KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT WETENSCHAPPEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
Celestijnenlaan 200A – B-3001 Heverlee – België

HIERARCHICAL AND STOCHASTIC ALGORITHMS FOR RADIOSITY

Promotor:
Prof. dr. ir. Yves D. Willems

Proefschrift voorgedragen tot
het verkrijgen van de graad
van doctor in de informatica

door

Philippe BEKAERT

14 december 1999

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTY OF SCIENCE
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE
Celestijnenlaan 200A – B-3001 Heverlee – Belgium

HIERARCHICAL AND STOCHASTIC ALGORITHMS FOR RADIOSITY

Committee:
Prof. René K. Boel, chairman
Prof. Yves D. Willems, advisor
Prof. Ronald Cools
Prof. Kadi Bouatouch (Université de Rennes 1, IRISA)
Prof. Werner Purgathofer (Technische Universität Wien)
Prof. Mateu Sbert (Universitat de Girona)

Thesis submitted in fulfillment of the requirements for the degree of "doctor in de informatica"

by

Philippe BEKAERT

14 december 1999

The research that led to this dissertation has been supported by a grant from the Flemish Institute for the Promotion of Scientific-Technological Research in Industry (I.W.T. grant nr. 941120) and by the Fund for Scientific Research – Flanders (F.W.O.-Vl. grant nr. G.0105.96).

Hierarchical and Stochastic Algorithms for Radiosity

Philippe Bekaert

Department of Computer Science,
Katholieke Universiteit Leuven

ABSTRACT

The radiosity method is a physically based method to compute the illumination in a virtual environment with diffuse (matte) surfaces. It allows to generate very realistic images of such environments by computer, and it is suitable for quantitative predictions of the illumination.

In the radiosity method, a number of simplifying assumptions are made that can however lead to certain image artifacts. In this dissertation, the numerical error introduced by these assumptions is analysed. The analysis allows to propose new algorithms in which this error, the discretisation error, is efficiently controlled during the computations by means of hierarchical refinement.

The radiosity method also requires the solution of very large non-sparse systems of linear equations (about 100,000 equations is common). Moreover, the coefficients of these systems are non-trivial four-dimensional integrals. The main part of this dissertation is devoted to an in-depth study of how the Monte Carlo method can be applied in this context.

The Monte Carlo method is suitable for reliable computation of the coefficients of the systems of equations. It also leads to algorithms that do not require explicit computation and storage of these coefficients. A systematic overview of such algorithms is presented. Previously proposed algorithms of this type are compared and some new algorithms are developed. Next, the application of several variance-reduction techniques is described, and the use of low-discrepancy sampling in this context is discussed. Finally, new ways to incorporate higher-order radiosity approximations and hierarchical refinement are proposed.

The resulting Monte Carlo radiosity algorithms do not only appear to be more reliable, but also often lead more rapidly to usable images than their deterministic counterparts. They require significantly less computer storage, and they are more user friendly. It is expected that these algorithms will stimulate the use of the radiosity method in a wide spectrum of applications.

I am indebted to anyone who contributed to this dissertation, through discussions, comments, material support, encouragement or otherwise.

In particular, I would like to thank my advisor Prof. Y. Willems (who has always been present when I really *needed advice) and the members of the reading committee: Prof. R. Boel, Prof. R. Cools, Prof. K. Bouatouch, Prof. W. Purgathofer and Prof. M. Sbert.*

I will always remember the (sometimes quite animated) discussions of papers and ideas with my current and former colleagues Frank Suykens, Phil Dutré and Eric Lafortune. The construction of our Cornell box replica (which does not *fit through the door anymore) is definitely a memorable highlight of the past few years!*

The members of the "Alma gang", and the other colleagues at the department of computer science, contributed a lot in making the preparation of this thesis (and the Alma-food) enjoyable.

Scientific events, such as the rendering workshops and graphics conferences in all seasons, have always been (sometimes much*) more than instructive events thanks to many co-attendees.*

I would like to express thanks to the computer system maintainance staff and the secretaries as well: everything went smooth on the technical and administrative side of this thesis (including last-minute printing of the pages with colour images from our terrible graphics machines).

Many of the results presented in this dissertation have been obtained in close collaboration with László and Attila Neumann, Mateu Sbert and Jan Přikryl. This collaboration has been considerably facilitated thanks to Prof. Purgathofer, who has been so kind to host me twice at the "Institut für Computergrafik" in Vienna. Numerous colleagues in Vienna have made my stays there extremely pleasant.

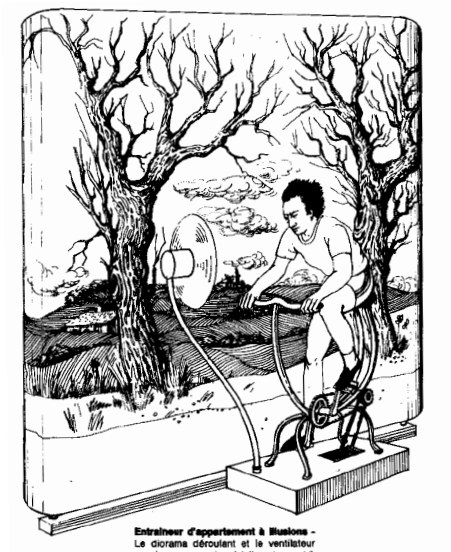
László Neumann was the one who convinced me to start doing research on Monte Carlo radiosity in 1995. Mateu Sbert provided almost daily critical advice and has been a considerable source of inspiration for much of the recent work on the subject. This dissertation would have been quite different without their constructive remarks and discussions.

Finally, the contribution of friends and family may not be directly visible in the contents of this dissertation, but their lasting support has been indispensable. Thanks!

Heverlee – December 1999.

Aan Annelies.

Contents



Entraineur d'appartement à Illusions - Le diorama déroulant et le ventilateur synchrones avec le pédalier, donnent à l'utilisateur de ce "home-trainer" particulier l'illusion d'une randonnée en plein air.

Virtual Reality: The ventilating fan and revolving diorama are driven by the home trainer pedals in order to give the impression of a bicycle trip in open air.
J. Carelman, *Catalogue d'objets introuvables – tome 2*, Brodard et Taupin, Paris, France, 1978.

1 Introduction

This dissertation addresses a problem in computer graphics. More specifically, it deals with physically based global illumination (§1.1) with the radiosity method (§1.2). The objectives of this dissertation are stated in §1.3. An overview will be presented in §1.4.

1.1 Physically-based global illumination

The goal of physically based global illumination is to compute the illumination in a, not necessarily existing, environment in a physically accurate way by computer. It allows to generate photo-realistic computer images in which illumination effects such as soft shadows, glossy reflections and colour bleeding effects are reproduced with high fidelity. Such effects are called *global* illumination effects because they are due to interactions of light in which multiple surfaces in a virtual environment are involved. This is in contrast with so called *local* illumination effects, which are due to interaction between a light source, a single surface and a viewing position only.

Global illumination effects in image synthesis can be reproduced by other means than by physically based illumination computations as well. For many global illumination effects, such as soft shadows, ad-hoc algorithms that exploit the capabilities of 3D graphics hardware, have been proposed (see for instance [11]). Computing the illumination in a physically correct manner however, will not only lead to ultimate realism, but the result can also be used for quantitative prediction of the illumination in a virtual scene.

Physically based global illumination finds applications in areas such as architectural design visualisation — in addition to the use of scale models for instance — civil engineering, lighting design and lighting optimisation, fine arts, virtual reality and computer entertainment. These are also the main areas that can benefit from the work presented in this thesis.

First, the problem and requirements of physically based global illumination are briefly formulated and an overview of previously proposed approaches is given. A more complete and elaborate introduction to physically based global illumination, as well as the radiosity method (§1.2), can be found in [56, 29, 150, 41].

1.1.1 Input

In general, the input of a physically based global illumination system consists of:

- A description of the *geometry* of the scene to be rendered. In practice, the surfaces in the scene are approximated by a set of simple surface primitives such as triangles and planar convex quadrilaterals, spheres, cylinders, tori, NURBS surfaces . . .

- A description of the *light scattering properties* of the surfaces in the scene. The light scattering properties of a surface are modelled by the *bidirectional reflectance and transmittance distribution function* (BRDF/BTDF). The BRDF/BTDF

basically expresses what fraction of light power coming in from a first direction will be scattered into a second direction. In general, the BRDF/BTDF are furthermore functions of location, wavelength of light, and time. In practice, simple mathematical models, such as described in [99] or [179], are used in order to specify the light scattering properties;

- A description of the *light sources* on the scene: a number of surfaces in the scene will not only scatter incident light, but also spontaneously emit light. The intensity of spontaneously emitted light, as a function of location, direction, wavelength and time, is expressed by a function called the *emittance distribution function* (EDF);

- For image synthesis, also a description of the *virtual camera* is needed: the virtual observer position in 3D space, the viewing direction, a direction that will appear as vertically up in the image, the image resolution and the horizontal and vertical field of view angles.

1.1.2 Output

The output consists of some kind of an approximate representation of the illumination in a virtual environment.

The illumination on the surfaces of a scene is commonly quantified by a quantity called *radiance*. Radiance expresses the intensity of the illumination as a function of location, direction, wavelength of light, and time. The relation between radiance and the scene geometry and optical surface characteristics is expressed by a mathematical equation called the *rendering equation* [83]. The rendering equation is a second-kind Fredholm integral equation of dimension 7 in the general case: 3 dimensions for position, 2 for direction and 1 for wavelength and time each. It results after making several simplifications to the general theory of light transport in physics [56, 174]. It is an instance of the general Boltzmann equation, which also describes other linear transport problems, such as neutron and radiative heat transport. In physically-based rendering, an approximate representation of the illumination in a virtual environment is obtained by numerically solving the rendering equation in some way.

Most often, physically-based illumination computations are used in order to generate a photo-realistic image of the scene. The ultimate goal is for this image to invoke the same visual experience as if the virtual scene were realised and viewed under corresponding conditions in reality. The computed radiances then need to be converted into RGB or CYMK colour values for display. This conversion needs to take into account the characteristics of the output device, e.g. by doing gamma correction, as well as of the human vision system. This conversion is called tone mapping [173, 105].

1.1.3 Requirements

Physically based global illumination algorithms ideally fulfil the following requirements:

- *Speed*: the result should be computed as quickly as possible; Speed is crucial in design applications where a designer is waiting for the rendered output in order to decide whether or not further refinements of the design are necessary.

Rendering speed also largely determines the production cost of computer animation movies. Ideally, rendering speed should be such that the illumination of a scene can be computed rapidly enough to follow the computer display refresh rate (typically 50 to 100 Hz);

- *Accuracy*: the radiance can only be approximated to finite accuracy. For lighting design applications, a relative accuracy of a few percent is sufficient. For image synthesis, the accuracy should be high enough so that displayed colour values are perceived as correct [126];

- *Reliability*: the algorithm should yield a result of controllable accuracy for a class of input models as wide as possible. It should not fail unexpectedly. In particular, it should deal well with *geometrical and optical complexity*. Although better complexity measures exist ([50] for instance), scene complexity is often expressed by the number of surface primitives, e.g. polygons, in the model. Many scene models being used at this time consist of hundreds of thousands or even millions of polygons;

- *User-friendliness*: in order to be useful for non-experts, it is mandatory that the algorithms need as few non-intuitive parameters as possible. Good default values should be available for these parameters. In particular, no tedious trial-and-error cycles should be required in order to appropriately condition the input data and determine parameter values.

1.1.4 Approaches

In order to create a photo-realistic image, the average radiance perceived through each pixel of the image needs to be computed. This can be done either by direct computation of pixel intensities in a *pixel-driven approach*, or by projection of a precomputed *object-space* radiance solution:

Object-space approaches Object space approaches first compute a representation of the radiance function on the surfaces of the objects in a virtual environment. In order to create an image from a given viewpoint, the visible surfaces through each image pixel are determined using ray-casting, a scan-line visibility algorithm or the Z-buffer algorithm. The average radiance in each pixel is computed from the average radiance radiated towards the viewing position from the surfaces that are visible in the pixel. Some examples:

- In the classical radiosity method (§1.2), particle tracing [120] and density estimation [145] for instance, the average radiance on each polygon in a diffuse polygonal environment is computed.

- Some non-diffuse radiosity-like algorithms [4, 161] compute the average radiance emitted by polygons in the scene towards other polygons in the scene. In [77, 148, 25], an angular instead of spatial parametrisation of the directional dependence of radiance is used.

The main advantage of an object-space radiance representation is that the computed result can be used easily for the synthesis of multiple images, e.g. in a virtual building walk-through. In the classical radiosity method, 3D graphics hardware is used in order

to carry out the projection step, yielding image generation times which are fractions of a second once a world-space radiance representation has been computed.

A second advantage of object-space algorithms is that significant re-use of the computed results is often possible after a change to the geometry of light emission or scattering properties of the surfaces in the virtual scene [52, 22, 39]. Object-space algorithms therefore may be very attractive for lighting design and optimisation applications [86, 138].

The main limitation of object-space algorithms is the enormous storage required for representing highly direction-dependent radiance, such as on a mirror. Even the storage of diffuse illumination can be prohibitive in scenes consisting of millions of surfaces. Current object-space algorithms such as radiosity also often have high intermediate storage requirements for the so called form factors (see §1.2).

Pixel-driven approaches Pixel-driven algorithms directly compute the average radiance in each pixel without first computing an object-space representation of the radiance on the surfaces in the scene. Examples of pixel-driven image synthesis algorithms are ray-tracing [182, 30, 83], path tracing with direct computation of pixel intensities [43] and bidirectional path tracing [96, 175].

The main advantage of pixel-driven algorithms is that storage of any data other than the scene geometry and materials can be avoided. They are suited for more complex models than feasible with object-space radiance algorithms. They will eventually yield correct results for a wide class of light emission and scattering models, no matter how directionally-dependent the resulting radiance is. Moreover, pixel-driven algorithms are often very user-friendly, one of the key reasons for the popularity of the ray-tracing algorithm.

The main disadvantage of pixel-driven approaches is that all computations need to be done over from scratch when the viewing position is changed. Object space algorithms may also yield results of fair quality significantly faster than pixel-driven algorithms.

Multi-pass approaches A promising approach is to compute radiance in object-space *as much as possible*. A pixel-driven algorithm is then used to render only the radiance contributions of which the computation and storage in object-space is not feasible. Such approaches are called *multi-pass* approaches [177, 149, 23, 79, 164].

1.2 The radiosity method

The radiosity method, first introduced in [58, 116, 27], is an object-space physically based image synthesis method for environments that consist exclusively of surfaces that are perfectly diffuse (matte, or Lambertian) emittors and reflectors of light: the EDF and BRDF of the surfaces in the scene do not depend on direction. (The BTDF, describing refraction of light through surfaces, is not taken into account in the basic radiosity method.)

In static diffuse scenes, the EDF and BRDF are fully determined by spatial location and the wavelength of light only. The advantage of such a simplification is that the radiance will not depend on direction either: a fixed point on a surface in the scene

will be perceived with the same intensity and colour regardless of the viewing position. This reduction of the dimension of the radiance function makes it feasible to store an accurate object-space representation even for fairly large scenes.

In diffuse environments, it is more appropriate to quantify the illumination at a given location and wavelength using the quantity *radiosity* rather than radiance.

1.2.1 Outline

The radiosity method basically consists of four steps. These steps are enumerated here for the classical radiosity method [58, 116, 27], but are the same in more advanced algorithms. A derivation of the method will be presented in chapter 2.

1. Discretisation of a virtual environment into planar convex polygons, called *patches*. For each patch i, the intensity of self-emitted illumination, expressed by the self-emitted radiosity E_i (unit $[W/m^2]$), and diffuse reflectance ρ_i is determined. The diffuse reflectance is a dimensionless number between 0 and 1 expressing what fraction of incident illumination is reflected. Both the diffuse reflectance and emittance are assumed constant over each patch;

2. Calculation of *form factors* F_{ij} for each pair of patches i and j. The form factor F_{ij} is a dimensionless number that expresses what fraction of the incident illumination on patch i is due to patch j;

3. Solution of the system of linear equations

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j . \tag{1.1}$$

   The unknowns B_i are the average total radiosity on patch i (unit $[W/m^2]$) and express the intensity of the total diffuse illumination on i. There is one unknown and one equation per patch in the scene. Due to the size of this system of equations, iterative solution methods such as Jacobi iterations or Southwell-relaxation [26] are used;

4. Visualisation of the solution as seen from one or more viewpoints. This step involves visible surface determination and tone mapping.

The equations (1.1) express that the illumination B_i on a patch i is the sum of the self-emitted illumination and the reflected incident illumination from other patches. The incident illumination is a weighted sum of the illumination on other patches j. The weights in this sum are the form factors F_{ij}. A fraction ρ_i of the incident illumination is reflected.

1.2.2 Problems

Unfortunately, the basic radiosity method suffers from several important problems. The main problems concern meshing and form factor computation and storage.

Meshing

The discretisation of the surfaces of the scene into patches needs to accurately capture illumination variations (figure 1.1 illustrates the kind of image artifacts that can appear due to improper meshing):

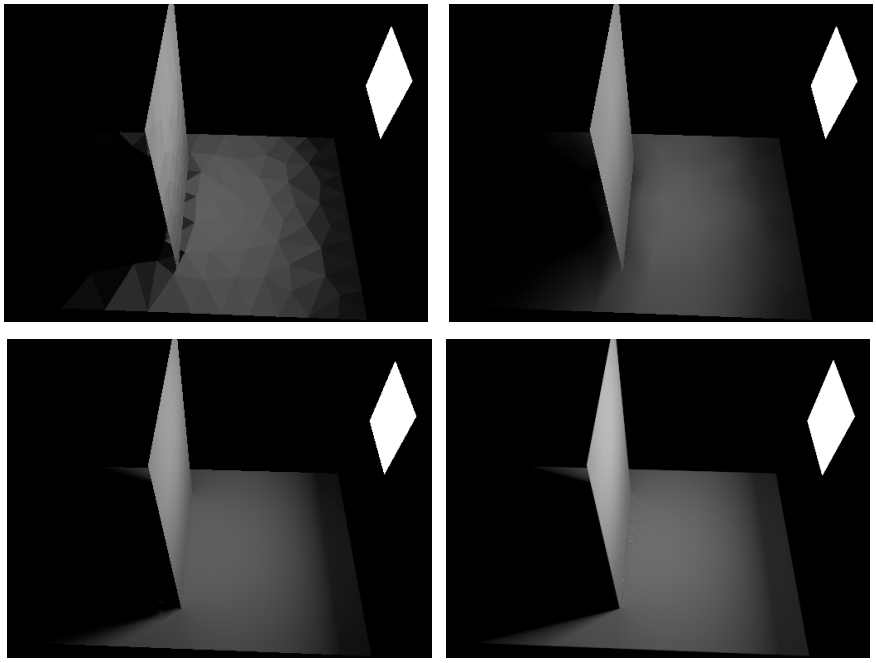


Figure 1.1: Image artifacts in the radiosity method due to improper meshing include light and shadow leaks, jagged shadow boundaries and shading discontinuities in regions where the illumination varies smoothly (upper-left image). Shading discontinuities in smooth areas are washed out using Gouraud interpolation, but other artifacts persist (upper-right image). In the lower-left image, the patches have been partitioned into non-intersecting parts, and a cubic radiosity approximation has been computed. The result is accurate without smoothing, except near the shadow boundaries, where higher-order discontinuities have not been resolved. The lower-right image shows the correct result that would be obtained with full discontinuity meshing.

- Visibility changes w.r.t. a primary or bright secondary light source, will result in discontinuities of various order in the radiosity function [71, 103]. Most notable are discontinuities in value where surfaces touch each other. If not properly dealt with, these lead to light- and shadow leaks in an image. Discontinuities in the first and second derivative result at shadow boundaries and along various curves inside the penumbra zone cast upon a receiver surface by light source/occluder surface pairs. Ignoring these leads to image artifacts such as blurred or jagged shadow boundaries. In discontinuity meshing [71, 103, 38, 162, 163, 13], the scene is discretised along these discontinuity lines so that these lines appear as patch edges and no discontinuities occur on the interior of the patches;
- In absence of visibility changes, the radiosity function varies smoothly. The

classical radiosity method however only computes a single constant radiosity value per patch. Even with higher order polynomial radiosity approximations [70, 72, 185, 172] however, a sufficiently fine mesh is required in order to accurately represent illumination variations in such regions. On the other hand, too fine a mesh will result in unnecessary computations.

- The patches need to fulfil several topological requirements as well [8].

Mesh generation for radiosity has traditionally been a matter of trial and error, exercised by experts for each scene separately. A good meshing strategy will balance the requirements of accuracy and speed. Automatic mesh generation is a requirement in order to make the radiosity method user-friendly.

Form factor computation and storage

The accurate computation of the form factors F_{ij} requires the computation of a non-trivial integral and is by far the most time consuming step of the radiosity method. Moreover, a form factor needs to be computed and stored for each pair of patches. Storage requirements are huge even for scenes consisting of no more than a few thousands of patches.

Form factors are non-trivial to compute The form factor F_{ij} between two patches i and j is given by 4-dimensional integral. Analytical solutions exist only in case of full visibility and for constant radiosity approximations [9, 141]. In general, the form factor needs to be computed by numerical integration. The numerical integration is non-trivial due to the need to evaluate visibility between pairs of points, one on each patch, and due to potential discontinuities and singularities in the integrand:

- Visibility is calculated using either a Z-buffer [27], a ray-tracing [178] approach, or using analytical visibility algorithms [40, 123]. When ray-tracing is used, familiar ray-tracing acceleration techniques [55] can be exploited. Special purpose acceleration techniques, such as shaft culling [61] have been designed especially for form factor computation as well. Global visibility pre-processing [169] is another alternative to save visibility computation time during form factor computation;

- The form factor integrand will be discontinuous in value and in derivatives in case of partial visibility. Discontinuity meshing [71, 103, 38, 162, 163, 13], resolves part of these discontinuities. Full resolution of the discontinuities requires sophisticated geometric computations and data structures such as a back projection [38, 162] or the visibility skeleton [40];

- The form factor integrand contains a r^2 factor in the denominator, where r is the distance between a pair of points, one on each patch. For patches that share an edge for instance, the form factor integrand will exhibit a (weak) singularity along the shared edge. Special purpose numerical integration strategies have been proposed in order to deal with this singularity [139].

The number of form factors is huge The accurate computation of form factors is not only complicated, but also needs to be carried out $\mathcal{O}(n^2)$ times where n is the number of patches in the scene. A lot of research effort has been spent in reducing the number of form factors to be computed without sacrifying accuracy:

- *Sub-structuring* [28] reduces the number of form factors by subdividing patches into elements. Incoming illumination is always computed at the finer element level, based on the radiosity emitted by the patches, representing the average effect of the elements into which it has been subdivided. If k input patches are subdivided into n elements, only $\mathcal{O}(kn)$ form factors need to be computed, which is substantially less than $\mathcal{O}(n^2)$ if $n \gg k$;

- With *adaptive meshing* [28, 20, 70], an initially rough subdivision into elements is successively refined until a given quality criterion is met. In each refinement step, "inaccurate" elements are replaced by groups of new elements. The intermediately available form factors and radiosity solution are re-used as much as possible;

- In *hierarchical radiosity* [68] (§2.3) the idea of sub-structuring is extended to more than two levels. This results in automatic, adaptive meshing and a further reduction of the number of form factors to $\mathcal{O}(k^2 + n)$. If input polygons are also grouped together into a hierarchy of *clusters* [151, 147, 24], the number of form factors is reduced further to $\mathcal{O}(n)$;

- *Higher order radiosity approximations* [70, 72, 185, 172] (§2.2) allow an accurate representation of the radiosity function in absence of discontinuities with far fewer elements than with a constant approximation. A reduction of the number of elements n, implies also a reduction of the number of pairs of elements n^2. Per pair of elements however, a larger number of generalised form factors needs to be computed. The computation of generalised form factors for higher order approximations is also more difficult;

- In *view-importance driven* radiosity algorithms [152] (§9) the number of form factors to be computed and stored is reduced by taking the view point, from which an image is to be generated, into account during the radiosity computations. Not all form factors are equally important for the perceived illumination in a given image. In importance-driven radiosity, less relevant light transport is computed to lower accuracy than more relevant, e.g. directly perceived, illumination. Less relevant light transport includes illumination from surfaces that reaches the virtual observer only indirectly, after several (attenuating) reflections;

- An obvious way to avoid a large form factor storage cost is *not* to store form factors permanently in computer memory, but to re-compute them whenever they are needed. Clever *caching strategies* [159] have been proposed in order to avoid re-computation of form factors as much as possible. These allow a trade-off between computation time and storage requirements.

1.3 Objectives of this dissertation

This dissertation aims at the further improvement of the speed, accuracy, reliability and user-friendliness of the radiosity method through:

- an analysis of the discretisation error, introduced by representing the radiosity in an environment as a piecewise low-order polynomial approximation, and the development of an automatic, hierarchical meshing strategy that allows to compute the radiosity to given discretisation accuracy with minimal computation work and storage requirements;
- a thorough investigation of the use of the Monte Carlo method as an alternative for deterministic iterative solution methods for the linear system of radiosity equations. The Monte Carlo method (§4) is known for its reliability and versatility in the solution of a wide class of difficult problems. Monte Carlo techniques have been used for solving very large systems of linear equations. They do not require explicit knowledge or storage of the coefficients of the system. The need to explicitly compute and store form factors may thus be avoided.

The restriction to diffuse environments may appear a very limiting restriction at first sight. The radiosity method is however important for two reasons:

- First, a large part of the illumination in real environments is to good approximation diffuse. In interior-scenes, diffuse illumination is perceived as “soft” and “cozy” unlike mirror-like reflections. Diffuse illumination plays an important role in architecture;
- Second, the problem of computing diffuse illumination is a trimmed-down version of the general problem and exhibits similar characteristics. It differs mainly in dimension. It is our hope that successful approaches for the diffuse case, will carry over to the object-space computation of the illumination in glossy environments. It is clear that object-space computation of complete directional illumination will never be a preferred approach, but we expect that the development of more efficient, reliable and user-friendly object-space algorithms will allow to shift more computation work from the pixel-driven pass to the object-space radiance computation pass in multi-pass approaches. Eventually, this may result in more efficient synthesis of sequences of images of dynamic scenes.

1.4 Overview of this dissertation

This dissertation is organised as follows:

- Chapter 2 presents a short review of the radiosity method with higher order approximations and hierarchical refinement;
- In chapter 3, the discretisation error in higher-order Galerkin radiosity computations is analysed and some algorithms proposed in order to deal with it;
- Chapter 4 briefly reviews the basic principles and techniques of the Monte Carlo method. The Monte Carlo method will be used for more reliable form factor computation as well as for solving the radiosity system of equations. An overview is given at the end of this chapter;

- The chapters §5 to §8 present a systematic overview of basic Monte Carlo estimators that can be used in the context of radiosity with constant approximations:
  - Chapter 5 deals with Monte Carlo form factor computation. It also introduces basic sampling techniques that will be used in subsequent chapters;
  - Chapter 6 presents stochastic relaxation methods for the solution of the system of radiosity equations;
  - Chapter 7 presents the solution of the system of radiosity equations by random walk methods;
  - In chapter 8, some other Monte Carlo methods for linear systems, found in the general literature on this subject, are briefly mentioned.

  Special attention is paid to analysing the computational error of the algorithms and ways of controlling it;
- The chapters 9 to 11 deal with variance reduction techniques that can be applied in order to increase the efficiency of stochastic relaxation and random walk methods:
  - Chapter 9 presents applications of importance sampling. In particular, view-importance driven Monte Carlo radiosity algorithms are proposed;
  - Chapter 10 discusses the control variates variance reduction technique;
  - In chapter 11, various strategies for combining gathering and shooting radiosity estimates are proposed.
- Chapter 12 deals with the issue of low-discrepancy sampling in Monte Carlo radiosity. Various experiments are presented that also confirm theoretical results derived in previous chapters;
- In chapter 13, presents the extension of the Monte Carlo radiosity algorithms for constant radiosity approximations to higher-order approximations;
- Chapter 14 proposes per-ray refinement as a strategy to incorporate hierarchical refinement in Monte Carlo radiosity, paving the way for the development of radiosity algorithms in which both the discretisation and computational error is efficiently controlled;
- Chapter 15 concludes with a summary, a list of original contributions, and some directions for future research;
- The appendices contain additional information about our implementation of the algorithms described in this thesis.

In the context of this dissertation, an extensive software package, called RENDERPARK has been developed as well. RENDERPARK is a test-bed system for physically-based rendering algorithms. All empirical results described in this dissertation have been obtained using RENDERPARK[1].

[1] All experiments have been carried out on a Silicon Graphics Octane workstation with 195MHz R10000 processors and 256MB RAM memory.

2 The Radiosity Method

At the heart of every physically-based global illumination system lays the numerical solution of a mathematical equation that describes the light transport in a virtual environment. This dissertation focusses on physically based global illumination algorithms that compute an object-space representation of the illumination in a diffuse environment. In the context of this dissertation, every such algorithm is called a *radiosity* algorithm. Radiosity is the radiometric quantity that is best suited for quantifying the illumination in a diffuse scene. The first radiosity algorithms for image synthesis have been proposed by Goral et al. [58], Nishita et al. [116] and Cohen et al. [27].

In §2.1 and §2.2, the mathematical equations modelling light transport in a diffuse environment will be presented briefly. The remainder of this thesis basically deals with the efficient, reliable and accurate solution of these equations. This chapter is concluded in §2.3 with a description of hierarchical refinement, which is a key technique leading to automatic, adaptive meshing. Hierarchical refinement largely contributes to the user-friendliness and efficiency of the radiosity method.

2.1 The radiosity integral equation

Consider a virtual environment consisting of surfaces S [1]. In radiosity, a restriction is made to only the following two modes of light-matter interaction[2]:

1. Spontaneous diffuse emission of light. The intensity of self-emitted diffuse light as a function of location x on the surfaces, wavelength λ of the emitted light and time t is given by the *self-emitted radiosity* $E(x, \lambda, t)$ (unit $[W/m^2]$);

2. Diffuse reflection of light. The fraction of incident light power that is reflected as a function of location x, wavelength λ and time t is given by the diffuse *reflectance* $\rho(x, \lambda, t)$ (dimensionless). The fraction of incident light energy that is not reflected is absorbed, that is: transformed into heat or some other form of energy.

Directional formulation With only these two modes of light-matter interaction, also the total illumination on the surfaces of the scene will be diffuse and depends only on location, wavelength and time:

$$B(x, \lambda, t) = E(x, \lambda, t) + \frac{\rho(x, \lambda, t)}{\pi} \int_{\Omega_x} B(h(x, \Theta_x), \lambda, t) \cos\theta_x \, d\omega_{\Theta_x} \qquad (2.1)$$

where (see figure 2.1):

[1] This dissertation focusses on triangular or planar convex quadrilateral surfaces, but the results are easily generalised to any other surface types for which a bijective mapping to a triangle or square exists.

[2] See [56, 174] for an in-depth discussion of the simplifications made to the general physics theory of light transport.

- the *radiosity* $B(x, \lambda, t)$ (unit $[W/m^2]$) expresses the intensity of the total diffuse light of wavelength λ at location x and time t;

- Ω_x denotes the hemi-sphere of directions Θ_x pointing from x to the outside of the surface at x. θ_x denotes the angle between the direction Θ_x and the surface normal at x. $d\omega_{\Theta_x}$ denotes the "size" (unit $[sr]$) of a differential solid angle containing the direction Θ;

- $h(x, \Theta_x)$ denotes the nearest point on the surfaces S of the scene that is visible from x into the direction Θ_x. This point always exists in a closed environment.

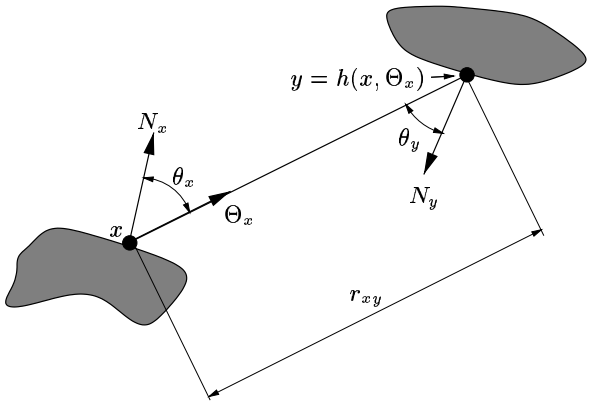


Figure 2.1: Diffuse light transport geometry

Because there is no cross-over between different wavelengths nor in time, we will drop the wavelength and time dependence in our notation:

$$B(x) = E(x) + \frac{\rho(x)}{\pi} \int_{\Omega_x} B(h(x, \Theta_x)) \cos\theta_x \, d\omega_{\Theta_x}. \tag{2.2}$$

There is one such equation per wavelength and per point in time. Equation (2.2) expresses that the radiosity $B(x)$ at $x \in S$ is the sum of self-emitted radiosity $E(x)$ plus the fraction of incident radiosity that is reflected. The incident radiosity is a weighted integral, over all directions Θ_x pointing from x outward the surface at x, of the radiosity $B(h(x, \Theta_x))$ of the first visible point $h(x, \Theta_x)$. The attenuation by $\cos\theta_x$ takes into account the apparent area at x in the direction Θ_x. A fraction $\rho(x)$ of the incident light is reflected. The division by π is required for energy conservation.

Spatial formulation It is often more convenient to transform the integral over the hemisphere Ω_x at x to an integral over the surfaces S of the scene. This leads to:

$$B(x) = E(x) + \rho(x) \int_S G(x, y) B(y) dA_y \tag{2.3}$$

with *geometric kernel*:

$$G(x,y) = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2}\text{vis}(x,y). \tag{2.4}$$

where

- dA_y denotes the area of a differential surface at the point $y \in S$;
- $\cos\theta_y$ is the cosine of the angle between the line connecting the points x and y and the surface normal at y;
- r_{xy} is the distance between the points x and y;
- $\text{vis}(x,y)$ is a predicate that takes the value 1 if x and y are mutually visible and 0 otherwise.

The geometric kernel $G(x,y)$ depends only on the geometry of the scene and not on the radiosity, self-emitted radiosity or reflectance.

Equations (2.2) and (2.3) are called *continuous radiosity equations*. In practice, $\rho(x) < 1$ for all surfaces in a scene. This is sufficient in order to guarantee that these equations have a unique solution. The kernel (2.4) is singular where surfaces touch each other. The singularity is a *weak singularity*. It is not present in the directional formulation (2.2) of the radiosity equation. The kernel (2.4) also has discontinuities in value due to abrupt changes in visibility $\text{vis}(x,y)$.

Equation (2.3) will sometimes be written in the more compact form

$$B(x) = E(x) + \int_S K(x,y)B(y)dA_y \tag{2.5}$$

with kernel

$$K(x,y) = \rho(x)G(x,y). \tag{2.6}$$

2.2 Galerkin radiosity

In numerical computing literature [35, 94, 125], basically two classes of methods are recommended in order to numerically solve integral equations like (2.3): the *Nystrom method*, also called *quadrature method*, and *projection methods* such as the *collocation* and the *Galerkin* method.

The Galerkin method for radiosity with higher order approximations has been proposed by Heckbert [72] and Zatz et al. [185]. In this section, a derivation of the method will be presented that makes clear how to apply the method with non-quadrilateral elements as well as with a function basis that is not the Cartesian product of 1-dimensional function bases.

Although an extensive comparison of the collocation and the Galerkin solution methods in the context of the radiosity problem could be an interesting topic of research, the Galerkin method is used in this dissertation as in most radiosity literature. The preference for the Galerkin method is based on tradition in finite element analysis: *"Galerkin methods provide consistently accurate, robust solutions to a wide variety of engineering problems"* [72].

2.2.1 Projection methods

The Galerkin method is a projection method. The basic idea of projection methods for solving linear integral equations such as (2.3) is to search for an approximate solution $\tilde{B}(x) \approx B(x)$ that is of a certain, known, "shape" which is inexpensive to evaluate for any point x. A convenient "shape" for $\tilde{B}(x)$ is defined as follows:

- The surfaces in the scene to be rendered are assumed to be a collection of *patches* i such as triangles or convex quadrilaterals or simple curved surfaces. Some numbering scheme on the patches is assumed so that each patch can be designated by an index, denoted i or j in this text;
- On each patch i, a number of independent "primitive shapes", called *basis functions*, $\psi_{i,\alpha}(x)$ are defined. The class of "shapes" considered on i contains all the possible linear combinations $f(x) = \sum_\alpha f_{i,\alpha}\psi_{i,\alpha}(x)$ of the basis functions. The basis functions are independent if none of the basis functions can be expressed as a linear combination of the other basis functions.

Example: using one basis function

$$\psi_i(x) = \begin{cases} 1 & x \in S_i \\ 0 & x \notin S_i \end{cases} \tag{2.7}$$

per patch, all per-patch piecewise constant functions can be represented. In §2.2.6, more details will be given concerning the higher-order basis functions that have been used in our implementation.

Projection methods thus try to find a "best" approximation

$$\tilde{B}(x) = \sum_i \sum_\alpha B_{i,\alpha}\psi_{i,\alpha}(x) \approx B(x) \tag{2.8}$$

for the true radiosity function $B(x)$, given a discretisation of the scene in patches i and a set of basis functions $\psi_{i,\alpha}(x)$ on each patch. With independent basis functions, $\tilde{B}(x)$ is uniquely and fully determined by its *basis coefficients* $B_{i,\alpha}$.

2.2.2 A bit of functional analysis

In order to understand how the basis coefficients $B_{i,\alpha}$ are determined in the Galerkin method, a couple of notions from functional analysis are required:

Scalar product of functions Consider two functions f and g defined on some domain D[3]. The integral

$$< f, g >= \int_D f(x)g(x)\,dx \tag{2.9}$$

is called the *scalar product* of the two functions. It can be seen as a generalisation of the well-known scalar product of 3D vectors $\vec{v} \cdot \vec{w} = v_1 \cdot w_1 + v_2 \cdot w_2 + v_3 \cdot w_3$. The set of "components" $f(x)$ of a function f is however continuous rather than discrete.

[3]This thesis deals only with real-valued functions for which the integral of the square of the function exists: $\int_D f(x)^2\,dx < \infty$.

Function norm With each scalar product corresponds a *norm*. The norm $\|f\|$, expresses the "size" of a function f. It is defined as the square root of the scalar product of f with itself:

$$\|f\|^2 =< f, f > .$$

A function is called *normalised* if it "has unit size":

$$\|f\| = 1.$$

Orthogonality of functions Just like 3D vectors, two functions f and g are called *orthogonal* if their scalar product is zero:

$$< f, g >= 0.$$

That will be the case if $f(x)$ is zero where $g(x)$ isn't and vice versa, but also when the functions "cancel" each other by appropriately changing sign and magnitude. A set of functions f_i is called an orthonormal set of functions if the functions are orthogonal to each other and they are normalised:

$$< f_i, f_j >= \delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

where δ_{ij} is *Kronecker's delta*. Note that scalar product, norm and orthogonality depend strongly on the domain D under consideration.

Expansion of a function w.r.t. a function basis Consider a set of functions ψ_i defined on a domain D. Then, every function f on D can be approximated by a linear combination

$$\sum_i f_i \psi_i(x) \approx f(x)$$

with real-valued coefficients f_i. The "closest" linear combination is called the *expansion* of the function f with respect to the set of functions ψ_i. Its coefficients can be computed by considering the scalar products of f with the ψ_i:

$$< f, \psi_j >\approx \sum_i f_i < \psi_i, \psi_j >.$$

There is one such equation per basis function. Together, these equations form a system of linear equations with the basis coefficients f_i as unknowns. In order to be able to solve this system of linear equations, the matrix $\Psi_{i,j} =< \psi_i, \psi_j >$ needs to have an inverse Ψ^{-1}. That will be the case if the functions ψ_i are independent.

Dual basis In practice, the basis coefficients f_i can be found quickly as scalar products:

$$\begin{aligned} f_i &= \sum_j < f, \psi_j > [\Psi^{-1}]_{j,i} \\ &= < f, \tilde{\psi}_i > \quad \text{with} \quad \tilde{\psi}_i(x) = \sum_j [\Psi^{-1}]_{j,i} \psi_j(x). \end{aligned} \tag{2.10}$$

The set of functions $\tilde{\psi}_i$ defined above is called the *dual basis* of the basis formed by the ψ_i. The original functions ψ_i are called the *primary basis*. The dual basis of a given basis is the unique set of linear combinations of the primary basis functions that satisfies the relations

$$< \tilde{\psi}_i, \psi_j >= \delta_{ij}. \tag{2.11}$$

for every primary basis function ψ_j.

A basis consisting of orthonormal functions is called an *orthonormal basis*. The dual basis of an orthonormal basis is the basis itself. The dual basis of a *orthogonal basis*, consisting of pairwise orthogonal but non-normalised functions ψ_i is

$$\tilde{\psi}_i(x) = \frac{1}{\|\psi_i\|^2}\psi_i(x). \tag{2.12}$$

Example with 2D vectors Consider the non-orthogonal basis $\{\vec{e}_1, \vec{e}_2\}$ with $\vec{e}_1 = (2, -1)$ and $\vec{e}_2 = (-1, 3)$ (see figure 2.2). What are the coefficients v_1 and v_2 of the vector $\vec{v} = (2, 1)$ w.r.t. this basis: $\vec{v} = v_1 \cdot \vec{e}_1 + v_1 \cdot \vec{e}_2$?

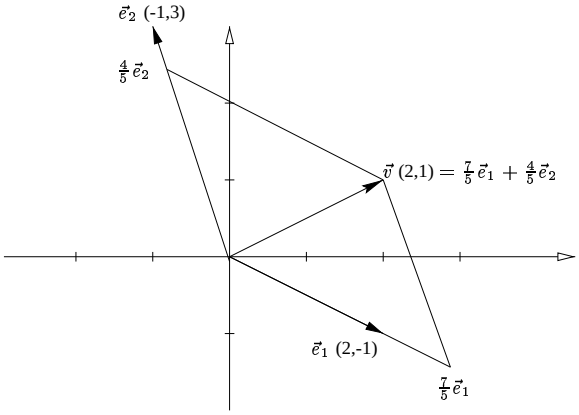


Figure 2.2: The basis coefficients of the vector $\vec{v}$ w.r.t. the basis $\{\vec{e}_1, \vec{e}_2\}$ are obtained most easily as a scalar product with dual basis functions.

First, the dual basis functions $\vec{d}_1 = (a_1, b_1)$ and $\vec{d}_2 = (a_2, b_2)$ need to be calculated using the relations (2.11):

$$\left\{ \begin{array}{rcl} < \vec{d}_1, \vec{e}_1 > & = & 1 \\ < \vec{d}_1, \vec{e}_2 > & = & 0 \end{array} \right. \iff \left\{ \begin{array}{rcl} 2a_1 - b_1 & = & 1 \\ -a_1 + 3b_1 & = & 0 \end{array} \right. \iff \vec{d}_1 = \left(\frac{3}{5}, \frac{1}{5}\right).$$

A similar calculation yields $\vec{d}_2 = (\frac{1}{5}, \frac{2}{5})$. The coefficients v_1 and v_2 are now easy to obtain by calculating scalar products of $\vec{v}$ with the dual basis functions $\vec{d}_1$ and $\vec{d}_2$:

$$\left\{ \begin{array}{rcccccc} v_1 & = & < \vec{v}, \vec{d}_1 > & = & \frac{3}{5} \cdot 2 + \frac{1}{5} \cdot 1 & = & \frac{7}{5} \\ v_2 & = & < \vec{v}, \vec{d}_2 > & = & \frac{1}{5} \cdot 2 + \frac{2}{5} \cdot 1 & = & \frac{4}{5} \end{array} \right.$$

Indeed: $\frac{7}{5}\vec{e}_1 + \frac{4}{5}\vec{e}_2 = \frac{7}{5}(2,-1) + \frac{4}{5}(-1,3) = (2,1) = \vec{v}$.

The basis coefficients f_i of some function f w.r.t. a function basis ψ_i can be determined in exactly the same way, using scalar products that are integrals (2.9) rather than the familiar vector scalar products in this example.

2.2.3 The Galerkin radiosity equations

In the Galerkin method, the radiosity coefficients $B_{i,\alpha}$ in $\tilde{B}(x) = \sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x) \approx B(x)$ are obtained by substituting $B(x)$ by $\tilde{B}(x)$ in the left and right hand side of (2.5):

$$\tilde{B}(x) \approx E(x) + \int_S K(x,y)\tilde{B}(y)dA_y.$$

The identity is only approximate because the integral in the right hand side will in general not be represented exactly by a linear combination of the chosen basis functions. The "best" approximation $\tilde{B}(x)$ is now obtained by expanding the left and right hand side w.r.t. the chosen basis $\psi_{i,\alpha}$ using (2.10) and (2.9), requiring that the basis coefficients be identical: for each i and α,

$$\begin{aligned}\int_S \tilde{\psi}_{i,\alpha}(x)\tilde{B}(x)dA_x &= B_{i,\alpha} \\ &= \int_S \tilde{\psi}_{i,\alpha}(x)E(x)dA_x + \int_S \tilde{\psi}_{i,\alpha}(x)\int_S K(x,y)\tilde{B}(y)dA_y dA_x. \quad (2.13)\end{aligned}$$

By defining

$$E_{i,\alpha} = \int_S \tilde{\psi}_{i,\alpha}(x)E(x)dA_x$$

and

$$K_{i,\alpha;j,\beta} = \int_{S_i} \tilde{\psi}_{i,\alpha}(x)\int_{S_j} K(x,y)\psi_{j,\beta}(y)\,dA_y\,dA_x. \quad (2.14)$$

one finally obtains the following system of linear equations in the radiosity coefficients $B_{i,\alpha}$:

$$B_{i,\alpha} = E_{i,\alpha} + \sum_{j,\beta} K_{i,\alpha;j,\beta}B_{j,\beta}. \quad (2.15)$$

The factors $K_{i,\alpha;j,\beta}$ are called *generalised patch-to-patch form factors*, for a reason that will become clear in §2.2.5.

2.2.4 Overview of the Galerkin radiosity method

The Galerkin method for radiosity consists of the following four steps:

1. Discretisation of the scene to be rendered into patches i. On each patch, a set of basis functions $\psi_{i,\alpha}(x)$ is fixed. For ease of implementation and robustness of the algorithm, a orthogonal set of basis functions that are nonzero only on a single patch is preferred;
2. Computation of the generalised patch-to-patch form factors $K_{i,\alpha;j,\beta}$ (2.14) for each pair of patches i and j. This involves the numerical computation of double integrals containing the visibility function between pairs of points $x \in S_i, y \in S_j$;
3. Solution of the Galerkin system of linear equations (2.15), yielding the radiosity coefficients $B_{i,\alpha}$;
4. Image generation by evaluating $\sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x)$ for one or more points $x \in S$ visible through each pixel of the image.

2.2.5 Constant radiosity approximations

Using the constant basis functions of (2.7), the following results are obtained:

$$\begin{aligned} \|\psi_i\|^2 &= \int_{S_i} 1\, dA_x &&= A_i \\ \tilde{\psi}_i(x) &= \frac{1}{\|\psi_i\|^2}\psi_i(x) &&= \frac{1}{A_i} \\ K_{i,j} &= \frac{1}{A_i}\int_{A_i}\int_{A_j} K(x,y)\, dA_y\, dA_x &&= \rho_i F_{ij} \end{aligned}$$

The reflectance $\rho(x) = \rho_i$ is assumed constant over each patch. F_{ij} is the classical *patch-to-patch form factor*:

$$F_{ij} = \frac{1}{A_i}\int_{A_i}\int_{A_j} G(x,y)\, dA_y\, dA_x. \tag{2.16}$$

With these assumptions, the Galerkin system of equations (2.15) becomes the classic radiosity system of equations (1.1), first proposed by Goral et al. [58]:

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j. \tag{2.17}$$

The four steps of the Galerkin radiosity method correspond with the four steps of the classical radiosity method, enumerated in §1.2.1.

2.2.6 Higher order approximations

Many choices are possible for the basis functions $\psi_{i,\alpha}$. In our implementation, we have used basis functions obtained by uniform mapping of "canonical" orthonormal basis functions $\psi_i(u,v)$ defined on a standard domain. The standard domain is the standard triangle $(0,0),(1,0),(0,1)$ for triangular patches and the unit square $[0,1]\times[0,1]$ for quadrilateral patches. This particular choice has proven to be adequate and convenient in implementation.

Canonical basis functions The canonical basis functions $\psi_i(u,v)$ are obtained by orthonormalisation of the elementary polynomials $1, u, v, u^2, uv, v^2, u^3, u^2v, uv^2, v^3$, ... on the standard domain using the algorithm of Gram-Schmidt from basic algebra. The resulting basis functions are plotted in figure 2.3. Details of their calculation are given in Appendix A. The computed radiosity solution $\tilde{B}(x)$ will be a linear combination of these basis functions after uniform mapping from a 3D triangle or convex planar quadrilateral to the standard triangle or unit square in 2D.

Uniform mapping The mapping from a 3D triangle or planar convex quadrilateral to the standard triangle or unit square was chosen so that equal areas in 3D correspond with equal areas in the 2D standard domain. The commonly used barycentric mapping for triangles is such a uniform mapping. For quadrilaterals however, a custom mapping was constructed. Details of the uniform quadrilateral mapping are given in Appendix B.

Generalised form factor computation Integrals over a 3D triangle or quadrilateral patch S_i are transformed to the 2D standard domain D corresponding with S_i:

$$\int_{S_i} f(x) dA_x = \int_D f(x(u,v)) J_i(u,v)\, du\, dv.$$

$J_i(u,v)$ is the Jacobian of the mapping from the 3D patch i to D:

$$J_i(u,v) = \frac{dA_x(u,v)}{du\, dv}.$$

The Jacobian expresses the differential area in 3D at the point $x(u,v) \in S_i$ that corresponds with the differential area $du\, dv$ in the standard triangle or unit square in 2D. The Jacobian of a uniform mapping is a constant: $J_i(u,v) = A_i$ for quadrilaterals and $J_i(u,v) = 2A_i$ for triangles (the surface area of the standard triangle $(0,0),(1,0),(0,1)$ is $1/2$).

Integrals in 3D, like (2.14), involving the basis functions $\psi_{i,\alpha}(x)$ are in this way transformed into integrals of the canonical basis functions $\psi_\alpha(u,v)$ on the standard triangle or unit square. The latter integrals can be computed numerically using deterministic numerical integration rules for the standard triangle or unit square. A compilation of such rules, based on [32, 31], is given in Appendix C. Various Monte Carlo integration algorithms will be presented in §5 and §13.2.

System solution The system of linear equations (2.15) can be solved using straightforward modifications to the Jacobi, Gauss-Seidel or Southwell relaxation algorithms. Algorithm 1 shows the modified Gauss-Seidel algorithm.

Visualisation of the solution The most straightforward way to generate images from the radiosity solution $\tilde{B}(x)$ is by tracing rays originating at the viewing position through each image pixel in 3D, and to evaluate $\tilde{B}(x) = \sum_\alpha B_{i,\alpha}\psi_{i,\alpha}(x)$ at the intersection point x with the first surface patch i of the scene. In order to evaluate $\psi_{i,\alpha}(x)$, the uniform parameters (u,v) of the point x in the standard triangle or unit

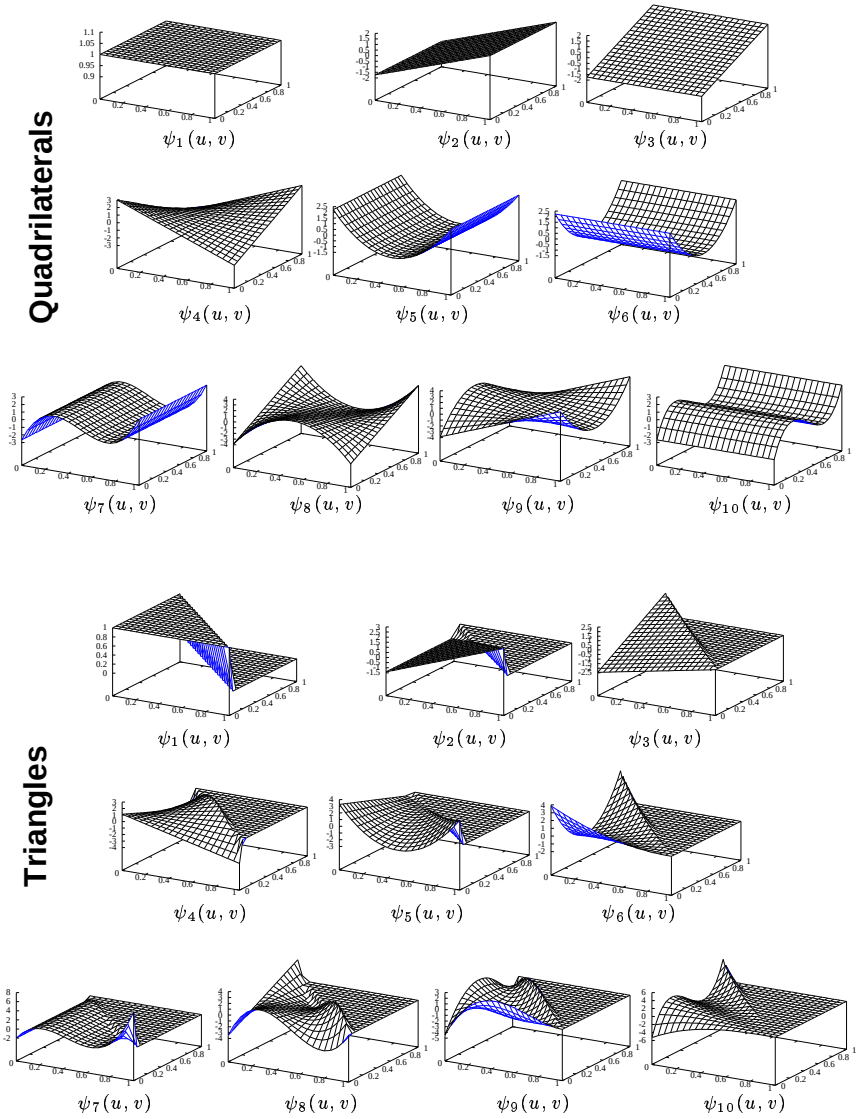


Figure 2.3: Basis functions used in our implementation. The bases are: constant = $\{\psi_1\}$, linear = constant $\cup$ $\{\psi_2, \psi_3\}$, quadratic = linear $\cup$ $\{\psi_4, \psi_5, \psi_6\}$, cubic = quadratic $\cup$ $\{\psi_7, \psi_8, \psi_9, \psi_{10}\}$. The computed radiosity solution $\tilde{B}(x)$ is a linear combination of these functions after mapping to the standard triangle or the unit square.

Algorithm 1: Gauss-Seidel solution of (2.15)

1. Initialise all $B_{i,\alpha} \leftarrow E_{i,\alpha}$;
2. Until convergence
   (a) for all patches i
      i. set $B_{i,\alpha} \leftarrow E_{i,\alpha}$ for all basis functions $\psi_{i,\alpha}$ on i;
      ii. for all patches j,
         A. if not yet computed before, compute the generalised form factors $K_{i,\alpha;j,\beta}$ for all basis functions $\psi_{i,\alpha}$ on i and $\psi_{j,\beta}$ on j;
         B. for all basis functions $\psi_{i,\alpha}$ on i, compute

$$B_{i,\alpha} \leftarrow B_{i,\alpha} + \sum_{\beta} K_{i,\alpha;j,\beta} B_{j,\beta}$$

square need to be determined using the uniform mapping: $\psi_{i,\alpha}(x) = \psi_{\alpha}(u,v)$. The average radiosity through the pixel is displayed after tone mapping.

It is possible to harness graphics hardware also for the visualisation step with higher order approximations:

- the nearest patch through each pixel can be found more quickly after ID-rendering: the patches in the scene are rendered with unique colours and flat shading. The frame buffer is read into main computer storage and the patch visible at the centre of each pixel determined by mapping the colour of the pixel;

- texture-mapping techniques can be used in order to determine the (u,v) parameters of the point visible at the centre of each pixel [69]. For quadrilaterals, the thus determined (u,v) parameters are bilinear parameters and still need to be transformed further into uniform parameters (see Appendix B).

2.3 Hierarchical refinement radiosity

The main problems of the radiosity method concern meshing and form factor computation. Meshing largely determines the solution accuracy that can be obtained. The mesh should be fine enough in order to accurately capture smooth illumination variations as well as illumination discontinuities such as at shadow boundaries. Too fine a mesh on the other side, will only lead to unnecessary computations. The number of form factors is quadratic in the number of mesh elements. For each form factor, a difficult integral needs to be computed.

Hierarchical refinement [68, 60] yields automatic adaptive meshing as well as a dramatic reduction of the number of form factors to be computed. This reduction in the number of form factors is due to the use of a multi-resolution representation of the radiosity function on a hierarchy of elements. Hierarchical refinement contributes immensely to the speed and user-friendliness of the radiosity method.

First, in §2.3.1 and §2.3.2, we illustrate how hierarchical refinement provides adaptive meshing and how a multi-resolution representation of the radiosity function

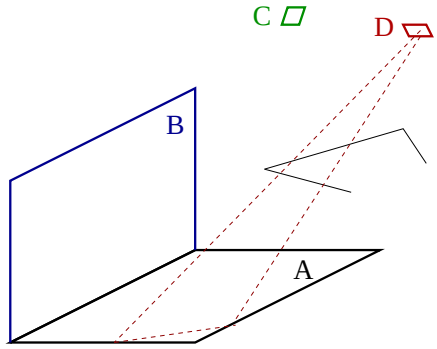


Figure 2.4: Simple example scene: patch A receives light from a nearby patch B, a small distant patch C and a partially occluded patch D.

leads to a reduction of the number of form factors that need to be computed. The algorithm for constructing the hierarchy of elements is outlined in §2.3.3. The modifications to the radiosity method needed in order to incorporate hierarchical refinement, including the so-called *push-pull sweep*, are briefly discussed in §2.3.4. Finally, a review of hierarchical refinement criteria and strategies for radiosity is given in §2.3.5.

2.3.1 Adaptive mesh generation

Consider the simple scene depicted in figure 2.4. The patch A receives light from three other patches: B, C and D. Suppose that a piecewise constant approximation of the radiosity on all elements is desired and assume that B, C and D emit a constant radiosity B_B, B_C and B_D respectively.

Consider now the radiosity at a point x on A due to B (cfr. §2.1):

$$B(x) = \int_{S_B} \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} \text{vis}(x,y) B_B \, dA_y = B_B \int_{\Omega_B(x)} \frac{\cos\theta_x}{\pi} \text{vis}(x, h(x,\Theta_x)) \, d\omega_{\Theta_x} . \tag{2.18}$$

S_B denotes the surface of patch B and $\Omega_B(x)$ the solid angle subtended by B on the hemisphere above $x \in S_A$. The radiosity at x due to the patches C and D is given by similar expressions.

A constant radiosity assumption on A is valid for the radiosity due to C, but not for the radiosity due to B or D:

- Patch B is a nearby patch. The solid angle $\Omega_B(x)$ under which B is "seen" will vary considerably for different points $x \in S_A$, and so will $\cos\theta_x$ and thus also the integrals in (2.18) and $B(x)$;
- Patch C is small and distant: both $\Omega_C(x)$ and $\cos\theta_x$ will be different only by a small amount for different points $x \in S_A$. The visibility factor $\text{vis}(x, h(x,\Theta_x)) = 1$ for all points x and directions Θ_x pointing to C;
- Patch D also is a small and distant patch, but an intervening surface blocks part of the light cast by D on A. The problem with patch D is caused by changing

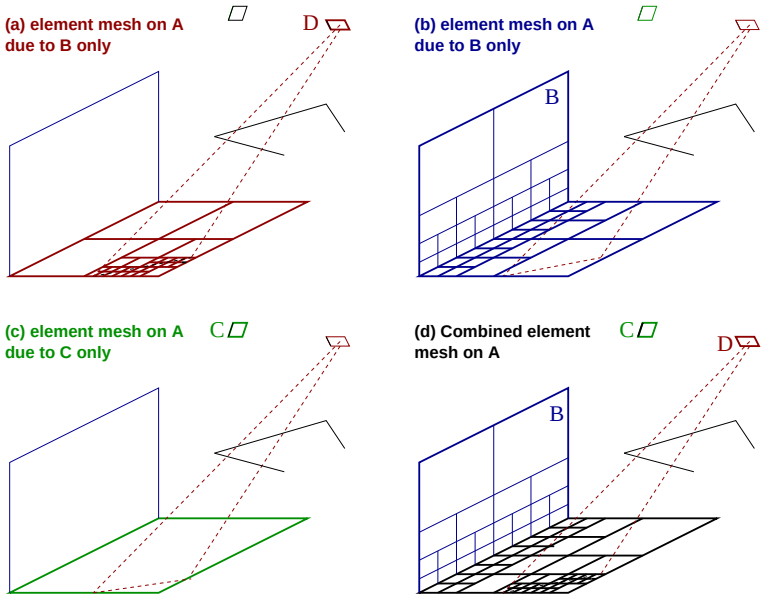


Figure 2.5: Adaptive meshing of A for (a) only source B, (b) only source C, (c) only source D and (d) combined element mesh.

visibility $\text{vis}(x, h(x, \Theta_x))$: for some points $x \in A$, D will be fully visible and — assuming that D is a strong light source — x receives a fair amount of radiosity from D. For other points $x \in A$ however, D is totally obscured by the intervening surface, and no light is received (directly) from D. Therefore, the constancy assumption will be violated for D.

Hierarchical refinement radiosity will refine A into a set of smaller elements. For each resulting element, the constancy assumption will be satisfied for any source. The figures 2.5a-c illustrate what subdivision into elements might be necessary for B, C and D separately. Figure 2.5d shows the combined set of elements. The resulting combined mesh will allow any light transport to A to be represented accurately, regardless of its source. Note that B will also need to be refined in order to accurately represent the illumination on B due to A.

2.3.2 Multi-resolution element mesh

A single combined adaptively-refined element mesh as shown in figure 2.5d will allow all illumination in the scene to be represented accurately. It will however lead to a lot of unnecessary computation. Consider e.g. the radiosity on patch A due to the small and distant patch C. No subdivision of A is necessary in order to represent the contribution of C with sufficient accuracy. The single combined element mesh however contains many small elements. For each element a form factor needs to be computed with C while one form factor with the whole of A would be sufficient.

In order to overcome this problem, hierarchical refinement radiosity uses a *multi-resolution* element mesh representing the radiosity on each patch at multiple levels of

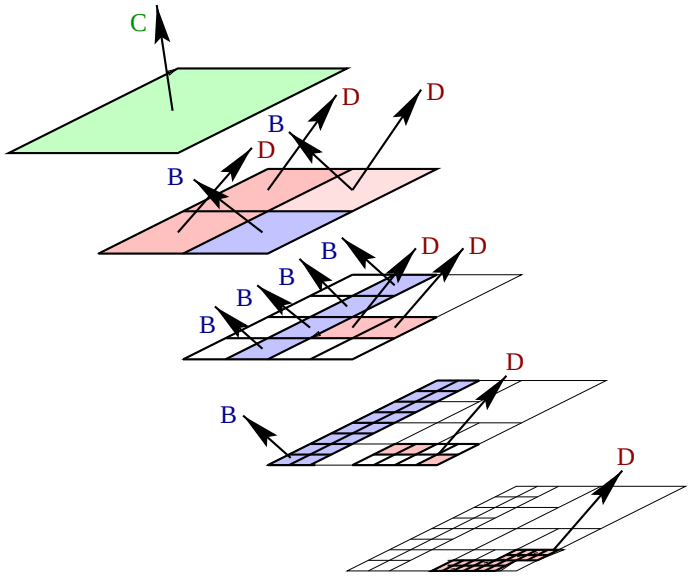


Figure 2.6: The multi-resolution element mesh for patch A. The arrows (shown for just a few elements in the lower two levels) indicate links to other patches or parts thereof.

detail. In the example, a regular quadtree subdivision is used. The element hierarchy for patch A is shown in figure 2.6. The top of the element hierarchy is a single element covering all A. One level below, the four elements correspond to the four quarters of A. In levels further below, an element of the parent level has been replaced by four quarter sub-elements unless the parent element already allowed the radiosity due to any source to be represented accurately.

Contrary to a single-level element mesh, an element hierarchy allows to compute the light transport at the *right level of detail* for each source instead of at one level of detail that suffices for *all* sources. The arrows in figure 2.6 indicate what contribution to the illumination on A is computed for each element. In order to compute the light transported from C to A, a single form factor is used between the top-level element of A and C. We say that a single *link* or *interaction* between A and C suffices. At the same time, light transport from D to A can be represented accurately on three of the four second-level quarter elements. On the fourth quarter element, a finer subdivision is needed, yielding links with sub-elements at lower levels. For each source, the same mesh elements on A are used as if there were no other sources (compare with figure 2.5a-c).

The Galerkin method with constant basis functions will compute the area-average of the radiosity on each element. The top-level element will represent the average radiosity on A as a whole. The second level elements represent the average radiosity on each quarter of A and so on. With the multi-resolution element mesh thus corresponds a multi-resolution representation of the radiosity. In this case, the representation is similar to a texture mip-map.

In the example, a constant approximation was used on every element. The basic idea and motivation is however exactly the same when higher order basis functions

are used. With the 'constantness assumption' then corresponds the assumption that the radiosity can be approximated well by a linear combination of the basis functions, whatever these basis functions are.

2.3.3 Hierarchical refinement algorithm

Algorithm 2 shows how the element hierarchies shown in figure 2.6 can be constructed automatically.

Consider a pair of patches, e.g. A and B in the example. The hierarchical refinement algorithm will first check whether an interaction between the top-level elements of A and B would already suffice in order to compute the light transport from B to A to desired accuracy (see figure 2.7). In order to check whether or not refinement is necessary, a *refinement criterion*, also called *oracle*, is needed. The oracle procedure takes information about the geometrical arrangement of the patches, visibility information as well as the source radiosity and returns whether or not an interaction between the two patches is admissible.

If an interaction between the two patches is not acceptable, a *refinement strategy* will prescribe what action needs to be undertaken in order to obtain pairs of interacting sub-elements of A and/or B that will allow light transport to be represented and computed to desired accuracy. Possible refinement actions include subdivision of one of the two elements (h-refinement), extending the basis on the receiver element in order to better approximate the received radiosity (p-refinement), relocation of mesh vertices (r-refinement) or combinations of these elementary strategies. In case of h-refinement, regular quadtree subdivision can be used as shown in the example. Sometimes, if information about the behaviour of the received radiosity function on the elements is known, a better subdivision choice can be made. This is the case if e.g. discontinuity curves have been determined using discontinuity meshing techniques [104, 10].

Refinement yields one or more new *candidate interactions*. The refinement procedure is called recursively for each of the candidate interactions. The candidate interactions will either be accepted or further refined until acceptance or until a recursion threshold has been reached. The result is a hierarchy of elements and interactions as illustrated in figure 2.6.

Algorithm 2: Hierarchical refinement: recursively refine a candidate interaction between a receiver ELEMENT rcv and a source ELEMENT src.

Refine(ELEMENT rcv, ELEMENT src)

1. if candidate interaction rcv←src is not admissible according to refinement criterion, then
   (a) subdivide one or both elements, or increase approximation order, . . .
   (b) recursively call Refine() for each resulting new candidate interaction element pair.
2. else
   (a) record accepted interaction rcv←src and compute form factors $K_{\text{rcv},\alpha;\text{src},\beta} \forall \alpha, \beta$.

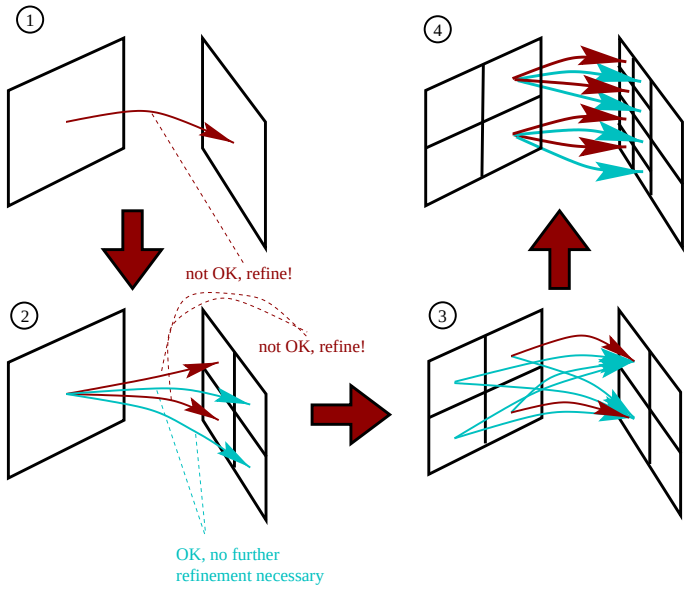


Figure 2.7: The hierarchical refinement algorithm first considers a candidate interaction between the top-level elements of the two patches. If the candidate interaction is deemed non-admissible by the refinement criterion, refinement is carried out, yielding a set of new candidate interactions to be considered in turn. Here, a candidate interaction is refined by regular subdivision of one of the interacting elements.

2.3.4 Radiosity with hierarchical refinement

What modifications are needed to an iterative radiosity system solver such as algorithm 1 in order to incorporate hierarchical refinement?

Without refinement, every iterative solver propagates not-yet-distributed (Southwell) or total (Jacobi, Gauss-Seidel) radiosity in some order from source patches to receiver patches in order to compute a better approximation of the radiosity on the receivers. All light propagated to a patch, is received by the whole patch.

With hierarchical refinement however, a tree-structure of elements corresponds to each patch. In order to compute the light transport from or to each patch, the iterative solver will traverse each element in the hierarchies. Each element in the hierarchy for a given patch will receive only a part of the radiosity propagated to the patch, determined by the interactions in which the element is involved as a receiver. In figure 2.6 for instance, the radiosity received from patch D will be scattered over 3 second level elements, 2 third-level elements, 3 fourth-level elements and 20 fifth-level elements in the element hierarchy of A. No radiosity from D is received at the top-level. When hierarchical refinement is carried out correctly, no contributions are forgotten and none are counted twice. All radiosity propagated to a patch will be available *somewhere* in the hierarchy, but it will in general be scattered among several elements.

Each element may however also be used as a source element in other interactions. The top-level element for patch A in the example may e.g. be used for computing the

light transport from A to C. The top-level element of A however receives no radiosity from D. Without taking measures, light transport from D via A to C would never be computed in this case. The iterative solver will therefore need to be extended in order to maintain a *consistent* multi-level representation of the radiosity on each patch. By consistent, we mean that each element in the hierarchy represents *all* radiosity on the corresponding surface of the patch to which it belongs, regardless of the level where the radiosity has been received. This is accomplished by first propagating received radiosity from elements at the top of the element hierarchy down the hierarchy. This sweep down the hierarchy results in a detailed representation of the full received radiosity at the leaf-elements of the hierarchy. Next, the hierarchy is traversed bottom-up, averaging the radiosity at sub-elements in order to obtain a rougher representation of the full radiosity at each parent element. This process, called a *push-pull sweep* is illustrated in figure 2.8. It relies on two elementary operations: the elementary push operation, in which radiosity is propagated from a parent element to its immediate children elements, and the elementary pull operation, in which the radiosity at the immediate children elements is filtered in order to obtain a best approximation at the parent element. The elementary push and pull operations are illustrated for constant approximations in figure 2.9.

Elementary push operation

The elementary push operation is used in order to propagate the radiosity of a parent element to its immediate children elements.

Consider an element S_i with sub-elements S_i^σ. Suppose a radiosity-like function $f(x) = \sum_\alpha f_{i,\alpha}\psi_{i,\alpha}(x)$ is given on the parent element S_i. $f(x)$ can be the radiosity $B(x)$ itself, or the unshot radiosity or yet another similar function, depending on the details of the radiosity system solver that is used. The elementary push operation computes the representation of $f(x)$ on the children elements S_i^σ.

The sub-elements cover disjunct parts of the parent element S_i. Consider therefore the restriction of $f(x)$ to each sub-element S_i^σ separately. The problem to be solved now is the determination of the coefficients $f_{i,\beta}^\sigma$ in

$$f(x) = \sum_\beta f_{i,\beta}^\sigma \psi_{i,\beta}^\sigma(x)$$

for points $x \in S_i^\sigma$. The coefficients can be determined by orthogonal projection (2.10):

$$\begin{aligned} f_{i,\beta}^\sigma &= < f, \tilde{\psi}_{i,\beta} > \\ &= \sum_\alpha f_{i,\alpha} h_{\alpha,\beta}^\sigma \quad \text{with} \quad h_{\alpha,\beta}^\sigma = < \psi_{i,\alpha}, \tilde{\psi}_{i,\beta}^\sigma > . \end{aligned}$$

The *filter coefficients* $h_{\alpha,\beta}^\sigma$ convert a low-resolution representation of a function on a parent element to a higher-resolution representation on the children elements. For orthogonal basis functions, cfr. (2.12):

$$h_{\alpha,\beta}^\sigma = \frac{1}{\|\psi_{i,\beta}^\sigma\|^2} \int_{S_i^\sigma} \psi_{i,\alpha}(x)\psi_{i,\beta}^\sigma(x)\, dA_x .$$

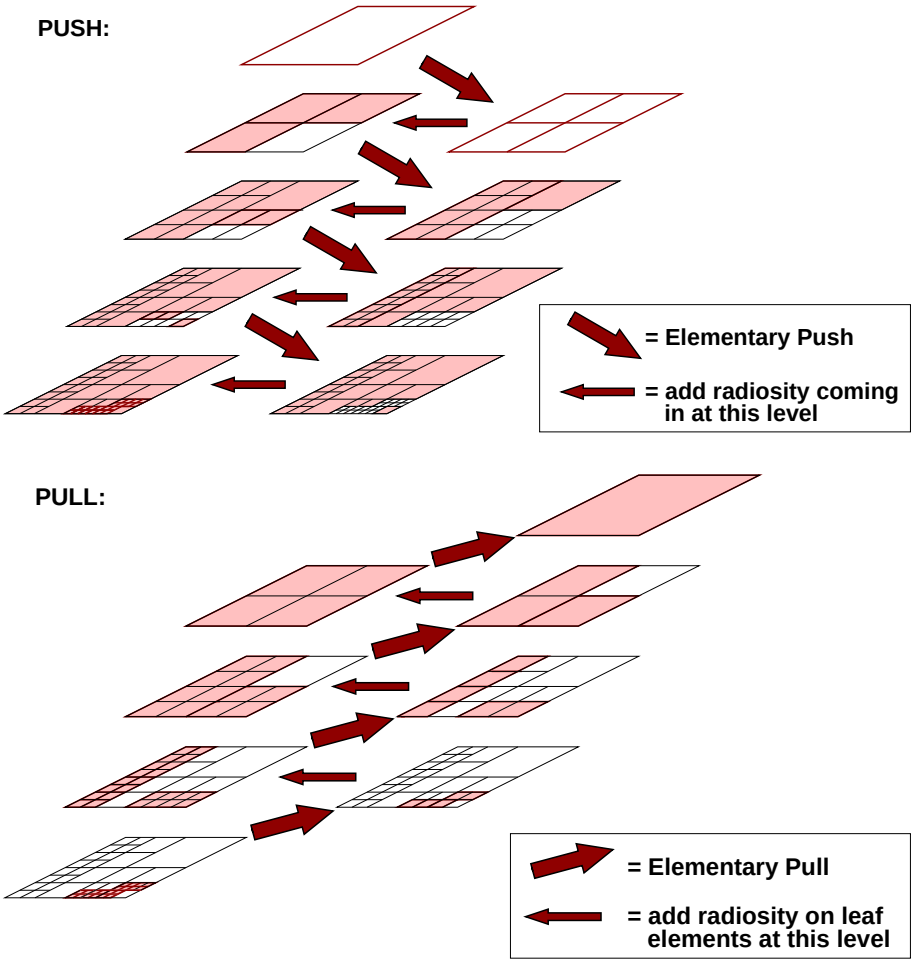


Figure 2.8: Push-pull sweep: first (top figure), the hierarchy of elements is traversed in a top-down direction, pushing down received radiosity at higher levels to lower levels. At each level, radiosity received at that level is added. At the end, a complete, detailed representation of received radiosity is obtained at the leaf elements of the hierarchy. Next (bottom figure), the hierarchy is traversed bottom-up, averaging the complete radiosity representation from lower levels. At the end, a representation of the complete received radiosity on all levels is obtained everywhere. The figure illustrates the push-pull sweep for the radiosity received on patch A from patch D in figure 2.6.

Example: For a constant approximation $\psi_i(x) = \psi_i^\sigma(x) = 1$ if $x \in S_i^\sigma$:

$$h_i^\sigma = \frac{1}{A_i^\sigma} \int_{S_i^\sigma} 1 \cdot 1 \, dA_x = 1.$$

Radiosity will be propagated unchanged from parent to children elements: $B_i^\sigma = B_i$ (see figure 2.9).

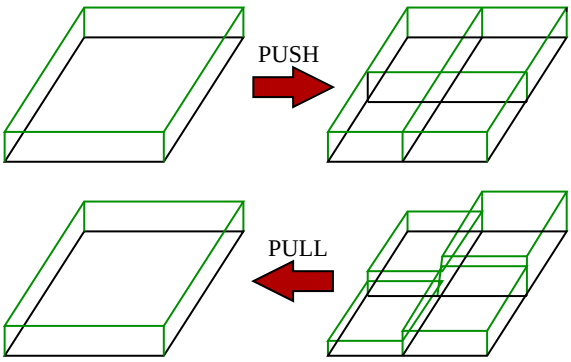


Figure 2.9: Elementary push and pull operation for constant approximations.

Elementary pull operation

The elementary pull operation is used in order to construct an approximation on a parent element for the radiosity given on an immediate sub-element.

Unlike above, the coefficients $f^{\sigma}_{i,\beta}$ in $f(x) = \sum_{\beta} f^{\sigma}_{i,\beta}\psi^{\sigma}_{i,\beta}(x)$ on each sub-element S^{σ}_{i} are given now. The elementary pull operation will compute the coefficients $f_{i,\alpha}$ for approximating f on the parent element S_i:

$$\sum_{\alpha} f_{i,\alpha}\psi_{i,\alpha}(x) \approx f(x).$$

Orthogonal projection (2.10) in this case yields

$$\begin{aligned} f_{i,\alpha} &= < f, \tilde{\psi}_{i,\alpha} > \\ &= \sum_{\sigma,\beta} f^{\sigma}_{i,\beta}\tilde{h}^{\sigma}_{\alpha,\beta} \quad \text{with} \quad \tilde{h}^{\sigma}_{\alpha,\beta} = < \tilde{\psi}_{i,\alpha}, \psi^{\sigma}_{i,\beta} > . \end{aligned}$$

The filter coefficients $\tilde{h}^{\sigma}_{\alpha,\beta}$ construct a lower-resolution representation on the parent element of the higher-resolution representation on the children elements. For orthogonal basis functions:

$$\tilde{h}^{\sigma}_{\alpha,\beta} = \frac{1}{\|\psi_{i,\alpha}\|^2} \int_{S^{\sigma}_{i}} \psi_{i,\alpha}(x)\psi^{\sigma}_{i,\beta}(x)\, dA_x .$$

Example: For constant approximations:

$$\tilde{h}^{\sigma} = \frac{1}{A_i} \int_{S^{\sigma}_{i}} 1 \cdot 1 \, dA_x = \frac{A^{\sigma}_{i}}{A_i}.$$

The radiosity at the children elements will be represented by their area-average on the parent element: $B_i = \sum_{\sigma} (A^{\sigma}_{i}/A_i) B^{\sigma}_{i}$ (see figure 2.9).

2.3.5 Refinement criteria and strategies

Crucial for the efficiency of the hierarchical refinement algorithm (cfr. §2.3.3) is the application of a good refinement criterion and strategy. In this section, we review which refinement criteria and strategies have been proposed in the past.

Refinement based on transported power

Hierarchical refinement radiosity was initially presented for constant radiosity approximations by Hanrahan et al. [68]. A cheap form factor estimate $\tilde{F}_{ij}$ ignoring visibility was used to measure the accuracy of a candidate interaction $i \leftarrow j$ from an element j to an element i:

$$\tilde{F}_{ij} = \frac{c_i}{\pi}\tilde{\Omega}_j \tag{2.19}$$

c_i is the cosine of the angle between the line connecting the midpoints of i and j and the normal on i. $\tilde{\Omega}_j$ is the solid angle subtended by the smallest sphere containing j at the midpoint of i. If $\max(\tilde{F}_{ij}, \tilde{F}_{ji})$ exceeds a given threshold F_ε, the candidate interaction $i \leftarrow j$ is deemed inaccurate. In the other case, the candidate link is considered admissible. If not admissible, the larger of the two elements i and j is subdivided using regular quadtree subdivision.

This subdivision strategy results in links with form factors approximately equal to the threshold F_ε. Since the sum of all form factors with a given element equals one, the number of links with each leaf element in the element hierarchies is bounded by approximately $1/F_\varepsilon$, independent of the total number of patches and elements in the scene. The number of form factors to be computed thus has been reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ with n the number of leaf elements resulting after refinement. In practice, the number of admissible links per leaf element is lower than $1/F_\varepsilon$ because links with higher level elements are effectively shared by all children elements.

The refinement procedure was however only used in [68] to refine the initial candidate interactions between each pair of input patches, yielding $\mathcal{O}(k^2+n)$ form factors to be computed, where k is the number of input patches. Unless the number of input patches is very small $k \ll n$, the k^2 term will be important. The computation of initial form factors between each pair of input patches is called *initial linking*. Initial linking, with its associated $\mathcal{O}(k^2)$ cost, is avoided by hierarchically grouping input patches in order to obtain a single hierarchy for the whole scene. This grouping of patches is called *clustering* [128, 90, 151, 147].

Hanrahan et al. [68] also observed that the number of form factors can be reduced considerably without affecting image quality by weighting the link error estimates (2.19) with the source element radiosity B_j and receiver element area A_i. Approximately the same amount of energy will be transported over the admissible links in that case. Weighting with receiver reflectance ρ_i as well further reduces the number of links without detoriating image quality.

An alternative way to reduce the number of form factors is by taking visibility information about candidate interactions into account: if a candidate interaction is fully occluded, refinement can be avoided because corresponding interaction at lower levels will be fully occluded as well [169]. A conservative procedure to determine full occlusion shall be used. Similarly, if a pair of candidate interacting elements is fully visible, a more relaxed refinement may suffice than for partially occluded interactions, where a higher variation of the received radiosity can be expected [54].

Form-factor and power-based refinement criteria use no information about what can be gained with non-constant approximations and as such have been applied to constant approximations only. Neither do they yield information about the variation of the received radiosity across the receiver element. This results for instance in sub-optimal shadow boundaries and too fine refinement in smooth areas. The main advantage of this criterion is its very low computational cost while yielding a fair image quality.

Refinement based on kernel smoothness

In earlier attempts to improve on power-based refinement, the variation of the radiosity kernel $G(x,y)$ between a pair of elements was taken into account. In [152], the maximum and minimum kernel value $G_{ij}^{\max} = \max_{x\in S_i, y\in S_j} G(x,y)$ and $G_{ij}^{\min} = \min_{x\in S_i, y\in S_j} G(x,y)$ are estimated by taking the maximum and minimum value computed between pairs of random points on both elements. A candidate interaction is considered accurate if

$$\Delta B_{ij} = \rho_i (G_{ij}^{\max} - G_{ij}^{\min}) A_j B_j < \varepsilon, \tag{2.20}$$

where ε denotes a predefined error threshold. Indeed, the received radiosity $B_j(x)$ due to j at any point $x \in S_i$ will lay between $\rho_i G_{ij}^{\min} A_j B_j$ and $\rho_i G_{ij}^{\max} A_j B_j$ since

$$G_{ij}^{\min} A_j \quad \leq \quad G_j(x) = \int_{S_j} G(x,y) dA_y \quad \leq \quad G_{ij}^{\max} A_j.$$

and $B_j(x) = \rho_i G_j(x) B_j$. In other words, by bounding radiosity kernel variation, the variation in radiosity over the receiver element is bounded as well.

A similar approach was used in [60] in order to drive hierarchical refinement with higher-order approximations. An interpolant

$$\tilde{G}_{ij}(x,y) = \sum_{\alpha}^{n_i} \tilde{\psi}_{i,\alpha}(x) \sum_{\beta}^{n_j} G_{\alpha,\beta} \psi_{j,\beta}(y) \tag{2.21}$$

is constructed for the radiosity kernel $G(x,y)$ using the n_i dual basis functions $\tilde{\psi}_{i,\alpha}$ on element i and the n_j primary basis functions $\psi_{j,\beta}$ on j. Next, the deviation from the interpolated kernel value $\tilde{G}_{ij}(x_k, y_k)$ is computed for every pair of points $(x_k \in S_i, y_k \in S_j)$ used for computing the generalised coupling coefficients (2.14) by numerical integration. The maximum deviation is used to decide whether further refinement is necessary or not. When applied to constant approximations, this approach would bound $\max(G_{ij}^{\max} - G_{ij}^{\text{av}}, G_{ij}^{\text{av}} - G_{ij}^{\min})$ with G_{ij}^{av} the average radiosity kernel value $G_{ij}^{\text{av}} = F_{ij}/A_j$.

Kernel variation is a sufficient condition for received radiosity variation, but not a necessary condition. Consider e.g. a very small element i touching a much larger element j as shown in figure 2.10. The received radiosity on i is proportional with the point-to-element-j form factors for points on i and will to good approximation be constant. The radiosity kernel $G(x,y)$ between i and j will however vary considerably. By further subdividing the small element i, the received radiosity can be made constant to arbitrary accuracy, but the radiosity kernel variation is not reduced at all.

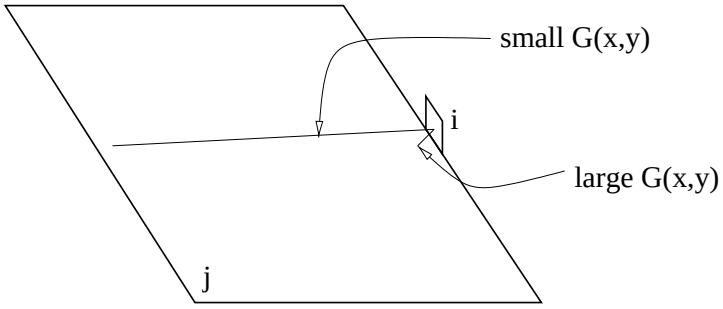


Figure 2.10: Bounding kernel variation is sufficient, but not necessary for bounding received radiosity variation. In this example, the variation in received radiosity can be made arbitrary small by taking a smaller element i. Kernel variation will however remain large.

Refinement based on smoothness of received radiosity

Because bounding kernel variation is not a necessary condition for bounding received radiosity variation, we can expect that hierarchical refinement based on kernel smoothness will yield hierarchical meshes with more elements and links than required. Optimal refinement can be expected by directly estimating how well the radiosity $B_j(x)$ received at $x \in S_i$ from S_j is approximated by a linear combination of the basis functions on S_i, i.e. by estimating the discretisation error directly.

This approach was first proposed by Lischinski et al. [104] for constant approximations. An interaction was deemed admissible if an estimate for

$$\delta_j(x) = \rho_i \max\left(\max_{x \in S_i} G_j(x) - F_{ij}, F_{ij} - \min_{x \in S_i} G_j(x)\right) B_j$$

was found to be smaller than a threshold ε. Indeed, the maximum and minimum received radiosity $B_j(x)$ at points $x \in S_i$ due to source j will differ no more than ε from the approximate average radiosity $B_{ij} = \rho_i F_{ij} B_j$ in that case. Pattanaik and Bouatouch [119] proposed a similar strategy for linear basis functions. In the next chapter, this approach will be generalised to arbitrary approximation order.

3 Discretisation Error Control

In a projection method, such as the Galerkin method (§2.2), an approximate solution $\tilde{B}$ is computed for the true solution B of the radiosity integral equation (2.5) or its alternatives. The approximate solution $\tilde{B}$ is a linear combination of basis functions $\psi_{i,\alpha}$, chosen as explained in §2.2. Even if the error in computing form factors and in system solution is absolutely negligible, Galerkin radiosity will not yield the true radiosity solution. The remaining error, which is due to the discretisation of the scene into elements, and the basis functions assumed on each element, is called the *discretisation error*[1]. Discretisation error causes artifacts in radiosity images, such as shadow and light leaks, jagged shadow boundaries, and shading discontinuities in areas where the radiosity varies smoothly (see figure 1.1).

The goal of this chapter is to study the effect of a given mesh and function basis on the discretisation error. First, a theoretical analysis of the discretisation error will be presented (§3.1). This analysis will be used for a-posteriori measurement of the discretisation error in a computed radiosity solution (§3.2). A discretisation error based refinement criterion and strategy for hierarchical refinement will be presented (§3.3). This refinement strategy generalises previous results for constant and linear approximations [104, 102, 119]. The combination of discretisation error measurement and error driven refinement allows to organise the computations in such a way that the radiosity in a scene is computed to given discretisation error with as few as possible elements and form factors (§3.4).

3.1 Analysis of the discretisation error

First, a theoretical analysis of the discretisation error is presented in this section. The following sections present applications of the framework developed here.

3.1.1 Galerkin discretisation error

Recall that the radiosity solution computed with the Galerkin method is of the form (2.8):

$$\tilde{B}(x) = \sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x). \tag{3.1}$$

The coefficients $B_{i,\alpha}$ are the solution of (2.15)

$$B_{i,\alpha} = E_{i,\alpha} + \sum_{j,\beta} K_{i,\alpha;j,\beta}B_{j,\beta} \tag{3.2}$$

with element-to-element form factors (2.14)

$$K_{i,\alpha;j,\beta} = \int_{S_i} \tilde{\psi}_{i,\alpha}(x) \int_{S_j} K(x,y)\psi_{j,\beta}(y)\, dA_y\, dA_x \tag{3.3}$$

[1] See [3] for a detailed classification of sources of error in global illumination computations.

and emittance coefficients

$$E_{i,\alpha} = \int_S \tilde{\psi}_{i,\alpha}(x) E(x) dA_x. \tag{3.4}$$

The element-to-element form factors (3.3) are integrals

$$K_{i,\alpha;j,\beta} = \int_{S_i} \tilde{\psi}_{i,\alpha}(x) K_{j,\beta}(x)\, dA_x$$

of *point-to-element form factors*

$$K_{j,\beta}(x) = \int_{S_j} K(x,y)\psi_{j,\beta}(y)\, dA_y. \tag{3.5}$$

The kernel is (2.4)

$$K(x,y) = \rho(x)G(x,y) = \rho(x)\frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2}\text{vis}(x,y). \tag{3.6}$$

The solution $\tilde{B}(x)$ is an approximation for the solution $B(x)$ of the radiosity integral equation (2.5):

$$B(x) = E(x) + \int_S K(x,y)B(y)\, dA_y. \tag{3.7}$$

The difference $\varepsilon(x) = B(x) - \tilde{B}(x)$ between $B(x)$ and $\tilde{B}(x)$ is called the *Galerkin discretisation error*. If the necessary integrals and sums are evaluated exactly, the error on the computed Galerkin solution $\tilde{B}(x)$ is fully determined by $\varepsilon(x)$. It will be discussed further (§3.5) how other sources of error can be incorporated in the framework.

3.1.2 The residual

The *residual* $\delta(x)$ of a given radiosity solution $\tilde{B}(x)$ is the function obtained by substituting $B(x)$ by $\tilde{B}(x)$ in the continuous radiosity equation (3.7) and making the difference between the right and left hand side:

$$\delta(x) = E(x) + \int_S K(x,y)\tilde{B}(y)\, dA_y - \tilde{B}(x) \tag{3.8}$$

In §3.1.3, the relation between the discretisation error $\varepsilon(x)$ and the residual $\delta(x)$ will be explained. First, the residual will be analysed a little more.

Theorem 3.1 *The residual $\delta(x)$ at a point $x \in S$ is a sum*

$$\delta(x) = \delta^e(x) + \sum_j \delta_j(x) \tag{3.9}$$

where

$$\delta^e(x) = E(x) - \tilde{E}(x) \quad \textit{with} \quad \tilde{E}(x) = \sum_{i,\alpha} E_{i,\alpha}\psi_{i,\alpha}(x) \tag{3.10}$$

the error made by approximating the self-emitted radiosity $E(x)$ by a linear combination $\tilde{E}(x)$ of the basis functions at x, and

$$\delta_j(x) = \sum_{\beta} B_{j,\beta}\left(K_{j,\beta}(x) - \tilde{K}_{j,\beta}(x)\right) \quad \textit{with} \quad \tilde{K}_{j,\beta}(x) = \sum_{i,\alpha} K_{i,\alpha;j,\beta}\psi_{i,\alpha}(x) \tag{3.11}$$

the error made by approximating the point-to-element form factors $K_{j,\beta}(x)$ by a linear combination $\tilde{K}_{j,\beta}(x)$ of the basis functions at x.

Proof:

$$\begin{aligned}
\delta(x) &= E(x) + \int_S K(x,y)\tilde{B}(y)\,dA_y - \tilde{B}(x) \\
&= E(x) + \int_S K(x,y)\tilde{B}(y)\,dA_y - \sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x) \\
&= E(x) + \int_S K(x,y)\left(\sum_{j,\beta} B_{j,\beta}\psi_{j,\beta}(y)\right) dA_y \\
&\quad - \sum_{i,\alpha}\left(E_{i,\alpha} + \sum_{j,\beta} K_{i,\alpha;j,\beta}B_{j,\beta}\right)\psi_{i,\alpha}(x) \\
&= \left(E(x) - \sum_{i,\alpha} E_{i,\alpha}\psi_{i,\alpha}(x)\right) \\
&\quad + \sum_{j,\beta} B_{j,\beta}\left(\int_{S_j} K(x,y)\psi_{j,\beta}(y)\,dA_y - \sum_{i,\alpha} K_{i,\alpha;j,\beta}\psi_{i,\alpha}(x)\right) \\
&= \left(E(x) - \tilde{E}(x)\right) + \sum_j \sum_\beta B_{j,\beta}\left(K_{j,\beta}(x) - \tilde{K}_{j,\beta}(x)\right) \\
&= \delta^e(x) + \sum_j \delta_j(x)
\end{aligned}$$

The residual has been used by Campbell [19] and Lischinsky [103] in order to drive non-hierarchical adaptive meshing for constant approximations. In the case of constant approximations, the terms $\delta_j(x)$ are nothing more than the difference between the point-to-element and the element-to-element form factor multiplied by the source radiosity and receiver reflectivity:

$$\delta_j(x) = \rho_i\left(F_{dA_x,j} - F_{ij}\right)B_j.$$

Also in the general case, the functions $E(x)$, $\tilde{E}(x)$, $K_{j,\beta}(x)$ and $\tilde{K}_{j,\beta}(x)$, needed in order to compute the residual for a given radiosity solution with coefficients $L_{i,\alpha}$, depend only on the given scene geometry, reflectance and emittance, and the set of basis functions being used. An efficient algorithm for evaluating them at selected points x will be presented below in §3.2.2.

3.1.3 Continuous error equation

The following theorem relates the discretisation error and the residual, and paves the way for the control of the discretisation error in radiosity algorithms:

Theorem 3.2 *The discretisation error $\varepsilon(x)$ is the solution a second-kind Fredholm equation with same kernel as the radiosity equation (2.3) but the residual as the source term:*

$$\varepsilon(x) = \delta(x) + \int_S K(x,y)\varepsilon(y)\,dA_y. \tag{3.12}$$

Proof:

$$\begin{aligned}
\varepsilon(x) &= B(x) - \tilde{B}(x) \\
&= E(x) + \int_S K(x,y)B(y)\,dA_y - \tilde{B}(x) \\
&= E(x) + \int_S K(x,y)\tilde{B}(y)\,dA_y - \tilde{B}(x) + \int_S K(x,y)\left(B(y) - \tilde{B}(y)\right)\,dA_y \\
&= \delta(x) + \int_S K(x,y)\varepsilon(y)\,dA_y
\end{aligned}$$

□

As a consequence of this theorem, in principle the same solution techniques can be used for computing the discretisation error in a given radiosity solution $\tilde{B}(x)$ as are used to solve the radiosity equation itself. Estimating the error for a given solution is called *a-posteriori error estimation*.

Note however that a projection method (§2.2.1) will require approximations for $\delta(x)$ and $\varepsilon(x)$ that are of higher polynomial order than the approximations used for $B(x)$, since $\delta(x)$ and $\varepsilon(x)$ are orthogonal to the function space in which the solution $\tilde{B}(x)$ was obtained. Projection methods for solving the error equation are therefore not recommended. Monte Carlo methods that avoid discretisation, similar to those used in e.g. stochastic ray tracing, are feasible. So is the Nystrom method [35, 125], with a numerical quadrature rule of sufficiently high precision. In practice however, we will be interested in a single number per element that characterises the discretisation error on each mesh element. This is outlined next.

3.1.4 Discrete error equations

A single number quantifying the discretisation error on an element i, can be obtained by using the p-norm:

$$\varepsilon_i^{(p)} = \left(\frac{1}{A_i}\int_{S_i} |\,\varepsilon(x)\,|^p\,dA_x\right)^{\frac{1}{p}}. \tag{3.13}$$

With $p = 1$, this expression yields the average discretisation error on the element, with $p = 2$ the RMS error. In the limit for $p \to \infty$, the maximum

$$\varepsilon_i^{(\infty)} = \max_{x \in S_i} |\varepsilon(x)| \tag{3.14}$$

is taken as error on the element. With these definitions, the dimension of the error $[W/m^2]$ is always the same as for radiosity.

Unless stated differently, the maximum-norm will be used in the remainder of this chapter. The superscript $^{(\infty)}$ will be dropped. Filling in (3.12) in (3.14) then yields

$$\varepsilon_i = \max_{x \in S_i} \left| \delta(x) + \sum_j \int_{S_j} K(x,y)\varepsilon(y) dA_y \right| \tag{3.15}$$

$$\leq \max_{x \in S_i} \left| \delta(x) + \sum_j \int_{S_j} K(x,y)\varepsilon_j dA_y \right| \tag{3.16}$$

An upper bound $\overline{\varepsilon}_i$ for ε_i can be obtained by solving the set of equations

$$\overline{\varepsilon}_i = \max_{x \in S_i} \left| \delta(x) + \sum_j \overline{\varepsilon}_j \int_{S_j} K(x,y) dA_y \right|. \tag{3.17}$$

Deriving similar equations for other norms is straightforward.

3.2 A-posteriori discretisation error estimation

In this section, it is described in detail how a-posteriori error estimation based on the theory in the previous section can be carried out in practice: A radiosity solution $\tilde{B}(x)$ has been computed and is considered given. The problem to be solved is: "How large is the discretisation error $\varepsilon(x)$ in this solution?".

In §3.2.1 we give an outline of the algorithm. Section §3.2.2 describes how the required coefficients can be computed efficiently. The approximations made and the advantage and cost of the algorithm are discussed in §3.2.3.

3.2.1 Outline of the algorithm

An upper bound $\overline{\varepsilon}_i$ for the discretisation error on each element i can be obtained by solving the set of equations (3.17), with (3.9):

$$\overline{\varepsilon}_i = \max_{x \in S_i} \left| \left(\delta^e(x) + \sum_j \delta_j(x) \right) + \sum_j \overline{\varepsilon}_j \int_{S_j} K(x,y) dA_y \right|. \tag{3.18}$$

The number of equations is equal to the number of elements in the scene.

The terms $\delta^e(x) + \sum_j \delta_j(x)$ in these equations can be considered source terms for error, comparable with the terms describing self-emitted radiosity in the set of equations to be solved in order to compute radiosity.

The integrals $\int_{S_j} K(x,y) dA_y$ are ρ_i times the classical point-x-to-element-j form factors $F_{dA_x,j}$ and result from the fact that the actual discretisation error $\varepsilon(y)$ on the elements j is replaced by a constant $\overline{\varepsilon}_j \geq \max_{y \in S_j} |\varepsilon(y)|$.

The set of equations (3.18) can be solved iteratively with Jacobi or Gauss-Seidel iterations. Such an iterative method will converge since, disregarding computational errors, for each point x:

$$\sum_j \int_{S_j} K(x,y) dA_y = \rho(x) < 1.$$

The main difference with previous algorithms for error estimation [102, 119] is in which way the error contributions from each interacting element j are added. In [102, 119], the absolute value of the sum in (3.17) is replaced by the sum of the absolute values of each term. The corresponding system of equations cannot be solved without sorting terms and dropping the smallest terms. No such sorting and dropping of terms is needed in order to solve (3.18). For the same reason, tighter error estimates can be expected from our approach.

In order to compute the maximum $\max_{x \in S_i}$ in (3.18), a numerical optimisation algorithm or an estimation method such as described in [75] should be used. In the implementation, we have however approximated $\max_{x \in S_i}$ by taking the maximum at only a small set of points x_k on each element i. Similar to [60], the points we used are the nodes of the numerical integration rule used to compute integrals on element i. We show in the next section §3.2.2 how $\delta_j(x_k)$ and $\int_{S_j} K(x_k, y) dA_y$ can be evaluated efficiently at these points x_k while computing the coupling coefficients $K_{i,\alpha;j,\beta}$. In §3.2.3, the effect of this approximation will be discussed.

Algorithm 3 shows the resulting algorithm when hierarchical refinement radiosity has been used to compute the approximate radiosity solution. In principle, expression (3.18) needs to be computed for all leaf elements i in the element hierarchies of each element p. The elements j to consider are then all elements interacting with the leaf element i and its parent elements. In our implementation, we have estimated the error at each level separately using (3.18). Next, the contributions from higher levels are added to the error estimate at lower levels. Finally, in order to obtain a consistent representation of the total error on each level, we have set the error at a parent level to the maximum error at the lower levels.

3.2.2 Efficient computation of the residual

In step 2a of EstimateErrorThisLevel() in algorithm 3, the residual $\delta_j(x_k)$ due to all elements j interacting with i, needs to be evaluated at a set of sample points $x_k \in S_i$. The residual needs to be evaluated in each iteration of algorithm 3. It is given by (3.11):

$$\delta_j(x) = \sum_\beta B_{j,\beta} \delta_{j,\beta}(x). \tag{3.19}$$

with coefficients

$$\delta_{j,\beta}(x) = K_{j,\beta}(x) - \tilde{K}_{j,\beta}(x) = K_{j,\beta}(x) - \sum_\alpha K_{i,\alpha;j,\beta} \psi_{i,\alpha}(x)$$

The sum in (3.19) is over all basis functions β on the source element j.

Algorithm 3: a-posteriori radiosity error estimation for hierarchical radiosity

EstimateError()

1. For each patch p, for each element i in the element hierarchy of p,
   (a) Set $\overline{\varepsilon}_i \leftarrow 0$
2. Until convergence,
   (a) For each patch p,
      i. For each element i in the element hierarchy of p, call EstimateErrorThisLevel(i);
      ii. Call PushPullError(p, 0).

EstimateErrorThisLevel(ELEMENT i)

1. Set a temporary variable $\epsilon_k \leftarrow 0$ for each sample point $x_k \in S_i$ (see §3.2.2);
2. For all elements j from which i receives radiosity, for each sample point $x_k \in S_i$,
   (a) Compute the residual $\delta_j(x_k)$ at x_k due to element j (see §3.2.2) and add to ϵ_k;
   (b) Add propagated error $\rho(x_k) F_{dA_{x_k},j} \overline{\varepsilon}_j$ to ϵ_k.
3. If i is a leaf element,
   (a) For each sample point x_k, add $\delta^e(x_k) = E(x_k) - \tilde{E}(x_k)$ to ϵ_k;
4. Set $\overline{\varepsilon}'_i \leftarrow \max_{x_k} |\epsilon_k|$.

PushPullError(ELEMENT i, FLOAT ϵ_{down})

1. Add $\overline{\varepsilon}'_i$ to ϵ_{down};
2. If i is a leaf element,
   (a) Set $\overline{\varepsilon}_i \leftarrow \epsilon_{\text{down}}$.
3. Else,
   (a) Set $\overline{\varepsilon}_i \leftarrow 0$;
   (b) For each child element c of i,
      i. Call PushPullError(c, ϵ_{down});
      ii. Set $\overline{\varepsilon}_i \leftarrow \max(\overline{\varepsilon}_i, \overline{\varepsilon}_c)$.

In the constant approximation case, the coefficients $\delta_{j,\beta}(x)$ are $\rho_i(F_{dA_x,j} - F_{ij})$, the difference between the point-x-to-patch-j form factor at x and the patch-i-to-patch-j form factor, which is the average point-to-patch form factor for points $x \in S_i$, multiplied by the reflectivity. In the general case, $\delta_{j,\beta}(x)$ expresses the error made by approximating the generalised point-x-to-element-j-with-basis-function-ψ_β form factor by a linear combination of the basis functions $\psi_{i,\alpha}$ on the receiving patch i. These coefficients are independent of the source radiosity coefficients $B_{j,\beta}$. They can be precomputed once at a set of sample points $x_k \in S_i$, for instance the first time they are needed. Once they have been calculated, the computation of the residuals at these sample points involves no more work than computing the sum of products shown in (3.19).

When the residuals are evaluated at the same sample points x_k that are used for numerical integration on the receiving element i, the coefficients $\delta_{j,\beta}(x_k)$ can be computed at a very low additional cost during the computation of the coupling coefficients $K_{i,\alpha;j,\beta}$. This is shown in algorithm 4.

Algorithm 4: Simultaneous computation of the coupling coefficients $K_{i,\alpha;j,\beta}$ and residual coefficients $\delta_{j,\beta}(x_k)$ between two elements i and j. Deterministic numerical integration is used. Integrals on the receiver element i are computed using a numerical integration rule with n nodes (u_k, v_k) and weights w_k, $k = 1, \ldots, n$. Similarly, integrals on the source element j are computed using a numerical integration rule with m nodes (u_l, v_l) with weights w'_l, $l = 1, \ldots, m$. On i, a basis with "a" basis functions $\psi_{i,\alpha}$ is defined. On j, a basis with "b" basis functions $\psi_{j,\beta}$ is used.

---

1. For all $k = 1, \ldots, n$, $l = 1, \ldots, m$, compute

$$K(x_k, y_l) = \rho(x_k) \frac{\cos\theta_{x_k} \cos\theta_{y_l}}{\pi \|x_k - y_l\|^2} \text{vis}(x_k, y_l).$$

2. For all $k = 1, \ldots, n$, $\beta = 1, \ldots, b$, compute

$$K_{j,\beta}(x_k) = \int_{S_j} K(x_k, y) \psi_{j,\beta}(y) dA_y \approx \sum_{l=1}^{m} w'_l K(x_k, y_l) \psi_\beta(u_l, v_l) J_j(u_l, v_l)$$

3. For all $\alpha = 1, \ldots, a$, for all $\beta = 1, \ldots, b$, compute

$$K_{i,\alpha;j,\beta} = \int_{S_i} \tilde{\psi}_{i,\alpha}(x) K_{j,\beta}(x) dA_x \approx \sum_{k=1}^{n} w_k \tilde{\psi}_\alpha(u_k, v_k) K_{j,\beta}(x_k) J_i(u_k, v_k)$$

4. For all $k = 1, \ldots, n$, $\beta = 1, \ldots, b$, compute

$$\delta_{j,\beta}(x_k) = K_{j,\beta}(x_k) - \sum_{\alpha} K_{i,\alpha;j,\beta} \psi_{i,\alpha}(x_k)$$

5. Return the $a \times b$ coupling coefficients $K_{i,\alpha;j,\beta}$, the $n \times b$ coefficients $\delta_{j,\beta}(x_k)$ and the n coefficients $K_{j,1}(x_k)$.

As explained in §2.2.6, the integrals are computed by transforming the elements to a standard 2D domain. The standard domain is the unit square for quadrilateral elements and the standard triangle for triangular elements. In algorithm 4, the transformed integrals on the the standard domain are computed using deterministic numerical integration (Appendix C). Different integration rules can be used on i and j:

$$\int_{S_i} f(x) dA_x \approx \sum_{k=1}^{n} w_k f(u_k, v_k) J_i(u_k, v_k) \tag{3.20}$$

with n nodes (u_k, v_k) and weights w_k on element i and

$$\int_{S_j} g(y) dA_y \approx \sum_{l=1}^{m} w'_l g(u_l, v_l) J_j(u_l, v_l)$$

with m nodes on element j. $J_i(u_k, v_k)$ is the Jacobian of the parameter mapping to i at a point (u_k, v_k) in the standard domain. With a uniform mapping (§2.2.6), the Jacobian is equal to the element area for quadrilateral elements and to twice the element area for triangular elements.

Different bases $\psi_{i,\alpha}$ and $\psi_{j,\beta}$ with a different number a and b of basis functions are allowed as well. In the implementation, the first basis function was chosen to be the constant basis function $\psi_1(u, v) = 1$. This way, the point-to-element form factors, required in step 2b of algorithm 3, result as the coefficients $K_{j,1}(x_k)$ in step 2 of algorithm 4.

The visibility factors $vis(x_k, y_l)$ in step 1 between cubature nodes x_k and y_l on element i and element j respectively are determined using ray casting, accelerated by using shadow caching and shaft culling [61]. In this simplified algorithm, no special care is taken of possible partial occlusion of the elements i and j. This will affect the quality of both the computed coupling coefficients and the error estimates. We will return to this topic in §3.5.

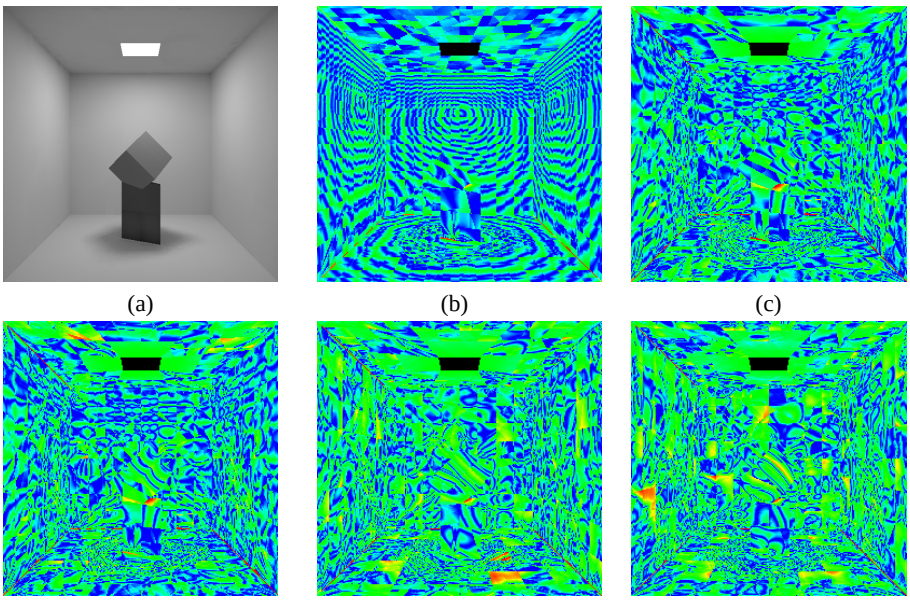
3.2.3 Empirical results and discussion

Two important simplifications were made in our algorithm for estimating the error on a computed radiosity solution:

- we have estimated $\max_{x \in S_i}$ in (3.17) by taking the maximum at only a small set of points x_k, which are the nodes of the numerical integration rule used on i;
- we have replaced the integral $\int_{S_j} K(x, y)\varepsilon(y)dA_y$ in (3.15) by $\overline{\varepsilon}_j \int_{S_j} K(x, y)dA_y$ with $\overline{\varepsilon}_j \geq \max_{y \in S_j} |\varepsilon(y)|$.

The first approximation was made in order to efficiently estimate the maximum over a element. We have found this approximation to be quite acceptable in most of the cases if an integration rule is used with slightly higher than the minimal required precision. The minimal precision is two times the order of the approximation on the element plus one [60]. This is illustrated in figure 3.1. In this figure, we have compared $|\sum_j \delta_j(x)|$ with the computed error estimate $\max_{x_k} |\sum_j \delta_j(x_k)|$ at each pixel of the image shown in (a) for various sets of basis functions. When using slightly more precise integration rules, the quality of both the computed coupling coefficients and the error estimates are visibly increased. The extra work implied thus cannot be seen as pure overhead.

The second approximation however leads to a serious overestimation of the error. We observed two reasons for this fact: First, the absolute value of the error $|\varepsilon(y)|$ is much smaller than $\varepsilon_j = \max_{y \in S_j} |\varepsilon(y)| \leq \overline{\varepsilon}_j$ on large parts of the source elements j. This can already be seen indirectly in figure 3.1. Second, $\varepsilon(y)$ will in general change sign on the source element. For error propagation (see (3.12)) the positive and negative parts tend to cancel each other. In (3.17) however, the absolute values are added together. For these reasons, we have found it more acceptable to ignore error propagation for the estimation of the error as in [102]: the maximum residual $\tilde{\varepsilon}_i = \max_{x_k \in S_i} |\delta^e(x_k) + \sum_j \delta_j(x_k)|$ alone yields a more realistic, although non-conservative, error estimate. A realistic error estimate is important in the context of the error control algorithm presented in §3.4.



(a) (b) (c)

(d) (e) (f)

Figure 3.1: False colour images indicating the ratio of the estimated maximum residual on each element and the actual residual per pixel for the scene shown in (a) and various sets of basis functions: (b) constant basis, (c) linear, (d) bilinear, (e) quadratic and (f) cubic. The actual residual was computed using per-pixel form factors. A green colour indicates that the measured approximation error is about equal to the estimated maximum for the element. A blue colour indicates that the measured error is smaller (by a factor of 10 for deep blue), a yellow or red colour that the error was underestimated (by a factor 3 and 10 respectively). We conclude that the discretisation error estimate is quite realistic, except on a few elements, where problems are due to computational errors, which have been ignored.

The main problem with our algorithm is the amount of storage required for the $n \times b$ coefficients $\delta_{j,\beta}(x_k)$ and the n coefficients $K_{j,1}(x_k)$ in addition to the $a \times b$ coupling coefficients $K_{i,\alpha;j,\beta}$. When e.g. using a quadratic basis, obtained by orthogonalising the 6 functions $1, u, v, u^2, uv, v^2$, a cubature rule of at least precision 5 is needed to compute the integrals. The minimal number of nodes for such a rule on quadrilaterals or triangles is 7 [32, 31]. This implies that there are $7 \times 6 + 7 = 49$ extra coefficients to be stored per pair of elements in addition to the $6 \times 6 = 36$ coupling coefficients.

The storage overhead for our algorithm can be significantly reduced by storing only one coefficient

$$\frac{\sum_\beta B_{j,\beta}\delta_{j,\beta}(x_k)}{B_{j,1}}$$

per cubature node x_k. When this coefficient is multiplied with $B_{j,1}$ the exact $\delta_j(x_k)$ is recovered. If error estimation is done while computing the radiosity solution, the radiosity coefficients $B_{j,\beta}$ will however change. This simplification relies on the fact that after a few iterations, the ratios $B_{j,\beta}/B_{j,1}$ will in general remain fairly constant.

Moreover, in a hierarchical radiosity algorithm, a significant change in the radiosity coefficients, will most often lead to refinement. While computing the coupling coefficients for the new interactions, a more accurate evaluation of the error is performed.

When ignoring error propagation, another n coefficients per pair of elements can be saved. With these simplifications, the storage overhead can be reduced to about 30% for a quadratic approximation and 15% for a cubic one, while the resulting error estimates still are realistic.

The quality of the error estimates was evaluated by comparing with the difference between a computed radiosity solution and a reference radiosity solution. In absence of inter-reflections and occlusion, the true radiosity in a scene can be computed analytically and be used as reference solution. In general however, the true radiosity cannot be computed exactly and a solution of higher accuracy must be used as reference solution. We found that our error estimates are quite trustworthy. This result will be illustrated indirectly when discussing the algorithm for error control in §3.4.2.

3.3 Error-based refinement indicator and strategy

3.3.1 Refinement indicator

In all previous hierarchical radiosity algorithms, the accuracy of a solution is controlled with a global link error threshold Δ. Before computing light transport from a source element j to a receiving element i, first the error Δ_{ij} on the transport is estimated. If this estimate Δ_{ij} for the error, that would be made by computing light transport from j to i, is larger then Δ, the candidate link between i and j needs to be refined. This results in new candidate interactions that are evaluated in turn until eventually, after sufficient refinement, admissible interactions result.

An expression for the error Δ_{ij} over a link $i \leftarrow j$ can directly be derived from (3.15): the contribution due to j to the error at i is:

$$\Delta_{ij} = \max_{x \in S_i} \left| \delta_j(x) + \int_{S_j} K(x,y)\varepsilon(y) dA_y \right|. \tag{3.21}$$

The candidate interaction $i \leftarrow j$ will need to be refined if $\Delta_{ij} > \Delta$ and will be accepted for computing light transport otherwise.

Expression (3.21) is a generalisation of the refinement indicator for constant and linear basis functions proposed in [102, 119]. For constant basis functions, (3.21) yields

$$\begin{aligned} \Delta_{ij} &\leq \max_{x \in S_i} |\rho_i B_j (F_{ij} - F_{dA_x,j}) + \rho_i \varepsilon_j F_{dA_x,j}| \\ &\leq \rho_i B_j \left(\max_{x \in S_i} F_{dA_x,j} - \min_{x \in S_i} F_{dA_x,j} \right) + \rho_i \varepsilon_j \max_{x \in S_i} F_{dA_x,j} \end{aligned}$$

which is very similar to the refinement indicator proposed in [102]. The indicator (3.21) is also very similar to the one for bilinear approximations presented in [119].

The same techniques as in §3.2 can be used in order to approximately compute expression (3.21). In particular, $\max_{x \in S_i}$ is approximated by the maximum at the numerical integration nodes x_k on i and $\varepsilon(y)$ is replaced by an approximation $\tilde{\varepsilon}_j$ for $\max_{y \in S_j} |\delta^e(y) + \sum_s \delta_s(y)|$, where the sum is over all elements s contributing light to j.

3.3.2 Refinement strategy

Deciding which element to refine

If the interaction needs to be refined, a first decision has to be made whether the source or receiver element shall be refined.

By refining the receiving element i, a better approximation of received radiosity as a linear combination of the basis functions defined on the sub-elements can be obtained. On the other side, if the source element j has been refined before, a more accurate representation of radiosity will be available on its sub-elements. By refining the source element, the propagated error from the source will be reduced. In order to decide which of two interacting elements i or j should be refined, $\max_{x \in S_i} |\delta_j(x)|$ shall be compared with $\max_{x \in S_i} |\int_{S_j} K(x,y)\varepsilon(y)dA_y|$. If the former — the residual due to j — is larger, the receiving element shall be subdivided. If the latter is larger, the propagated error from j dominates and the source j shall be refined. Often in this case, j will already have been refined before for other interactions, and a more accurate representation of radiosity on it is readily available.

Deciding how to refine

When either source or receiver has been selected for refinement, a second decision needs to be made *how* the element will be refined.

The most common refinement action undoubtly consists of subdividing an element into four sub-elements e.g. by connecting the midpoints of its edges. This is known as *regular quadtree subdivision*. On each sub-element, the same basis as on the parent is used. It is the best form of refinement if no knowledge about the behaviour of received radiosity is available. If such information is available, much better strategies are possible:

- If information about radiosity gradients is available, a binary subdivision along a line perpendicular to the gradient will tend to reduce the discretisation error most [19].
- If information about the location of discontinuity lines is available, it is advantageous to split elements across these discontinuity lines so that no significant discontinuities will appear on the interior of the children elements [104, 163, 13]. In [10], a novel approach has been proposed in which discontinuity-driven subdivision was restricted to significant discontinuity lines only. Also, multiple parallel element hierarchies were maintained on each patch. Each of these parallel hierarchies is optimally suited to compute light transport from certain elements, taking care that no contributions are omitted or counted double in the ensemble. Preliminary empirical results were encouraging, but this approach has not been pursued further due to the cost of implementing a proper discontinuity meshing algorithm.
- Subdivision is just one way to increase precision on a receiver element. It corresponds with h-refinement in the finite element method. Other refinement strategies, in particular switching to a higher order approximation (p-refinement), possibly in combination with element subdivision when a maximum approximation order has been reached on an element, may be well worth study in future research.

3.4 Discretisation error control

An error estimate computed with the algorithm proposed in §3.2 can be used in order to judge the acceptability of a given radiosity solution. If not acceptable, a more accurate solution is obtained in previously proposed hierarchical radiosity algorithms by decreasing the global link error threshold Δ. In the technique proposed here, a user controls the accuracy of a solution by specifying directly the maximum allowed absolute radiosity error ε instead. An outline of the method is given in §3.4.1. Empirical results and discussion follow in §3.4.2.

3.4.1 Outline of the algorithm

The basic idea is to *compute* a link error threshold Δ_i for each not further subdivided element i. This link error threshold Δ_i is such that if $\Delta_{ij} < \Delta_i$ for all interactions $i \leftarrow j$ of i and it's parent elements l, the total error $|\varepsilon(x)|$ will be smaller than the a-priori specified error threshold ε for all points $x \in S_i$. Since the real total error $|\varepsilon(x)|$ is not known, a realistic estimate $\tilde{\varepsilon}_i$, computed with the techniques of §3.2, is used instead.

In the algorithm we propose here, a link error threshold Δ_l is kept with *every* element l, not only the leaf elements. At the start of the radiosity computations, the link error thresholds Δ_l are initialised to a reasonable value (see below). During each iteration of the radiosity computations, refinement is carried out so that the link error Δ_{l,j_l} for every interaction $l \leftarrow j_l$ of each element l will be smaller than the link error threshold Δ_l for the element. The link error threshold Δ_l of a parent element will be the minimum link error threshold of its sub-elements. After gathering radiosity over each interaction, the total error $\varepsilon(x)$ is estimated on the leaf elements i of the element hierarchies. If the error estimate exceeds the maximum allowable error ε, the link error thresholds Δ_i at the leaf element i and Δ_l at its parent elements l are decreased by a fixed factor. Accumulating the error estimates at each level and possibly decreasing the link error thresholds are done with an extended push-pull sweep as shown in algorithm 5. A good, simple, starting value for Δ_i is $\Delta_i^{\text{initial}} = \varepsilon$ itself. With this starting value, too much refinement during the first iterations of the radiosity computation is avoided. The radiosity solution and error estimates become trustworthy only after a few iterations.

Decreasing the link error thresholds only by a small factor during each iteration will result in a larger number of iterations before reaching the requested accuracy ε. A large factor will eventually lead to over-refinement. In our implementation, we have used a factor 1.4. This factor should probably depend on the order of radiosity approximation used. Although we obtained satisfying results with this factor 1.4, a careful analysis in order to determine an optimal factor is an area for future research.

3.4.2 Empirical results and discussion

As will be explained in the next section §3.5, computational errors during the evaluation of the coupling and link error coefficients cannot be completely ignored and some simple preventive measures are taken in our current implementation:

- We imposed a minimal element area A_ϵ: subdivision of elements with area smaller than A_ϵ was not allowed. The link error threshold on such elements

Algorithm 5: Radiosity with error control

radiosity-with-error-control(ACCURACY ε)

1. For each patch p,
   (a) Initialise radiosity to self-emitted radiosity;
   (b) Initialise the link error threshold $\Delta_p \leftarrow \varepsilon$.
2. Until convergence, do for each patch p,
   (a) Refine all interactions $l \leftarrow j_l$ of elements l in the element hierarchy of p as described in §3.3 until all interaction errors Δ_{l,j_l} are smaller than the interaction error threshold Δ_l on l. j_l denotes an element from which l receives radiosity;
   (b) For all elements l in the element hierarchy of p,
      i. Gather radiosity over the interactions $l \leftarrow j_l$ of l;
      ii. Compute a realistic estimate $\overline{\varepsilon}'_i$ for the error in received radiosity at this level, using EstimateErrorThisLevel() in algorithm 3.
   (c) Call extended-push-pull(p, 0).

extended-push-pull(ELEMENT i, FLOAT ϵ^{down})

1. Add $\overline{\varepsilon}'_i$ to ϵ^{down};
2. If i is a leaf element,
   (a) Set the total radiosity on $i \leftarrow$ sum of self-emitted radiosity and received radiosity;
   (b) Set total error estimate on i, $\overline{\varepsilon}_i \leftarrow \epsilon^{\text{down}}$;
   (c) If $\overline{\varepsilon}_i$ is larger then the a-priori given error threshold ε, decrease the link error threshold Δ_i on i.
3. Otherwise,
   (a) Set $\overline{\varepsilon}_i \leftarrow 0$;
   (b) For each child element c of i,
      i. Push down the received radiosity on i and add to the received radiosity of the child element c;
      ii. Invoke recursively extended-push-pull(c, ϵ^{down}) for the child element;
      iii. Pull up the total radiosity on the child element c to i;
      iv. Set $\overline{\varepsilon}_i \leftarrow \max(\overline{\varepsilon}_i, \overline{\varepsilon}_c)$;
      v. Set $\Delta_i \leftarrow \min(\Delta_i, \Delta_c)$.

was not decreased either. A user will have to tolerate larger errors than ε on such elements;

- We changed the subdivision strategy of §3.3.2 in order to prevent a too large difference in area between elements in an interaction. Both the visibility estimation and the numerical integrations could be performed more accurately with this change. This change however results in more interactions than strictly necessary;

- In absence of occlusion, an analytical expression [9] was used for the point-x_k-to-element-j form factors $K_{j,1}(x_k)$.

The algorithm was verified by computing the illumination in a number of scenes at various accuracies and comparing the results. We found that the discrepancy between the computed results and a very accurate reference solution was always almost everywhere smaller than or about equal to the difference in requested accuracy. This also illustrates the reliability of the error estimates of §3.2. A typical result is shown in figure 3.2.

3.5 Other sources of error

So far in the discussion, other sources of error except due to discretisation have been ignored. In practice however, in particular errors in the computation of the form factors $K_{i,\alpha;j,\beta}$ can sometimes cause the computed radiosity solution and the error estimates to be unreliable if no special care if taken. The main sources of error are due to inexact visibility computation and due to imprecise numerical integration in the computation of the form factors and the error estimation coefficients.

A first strategy to deal with computational error is to prevent them right away (§3.5.1). Alternatively, we expect that more selective error control will be obtained by incorporating sources of computational error in the framework that is presented in this chapter (§3.5.2).

3.5.1 Prevention of other sources of error

Several techniques can be used in order to prevent visibility error and the error due to imprecise numerical integration:

- In order to deal correctly with visibility, the point-to-element form factors $K_{j,\beta}(x)$ (3.5), computed in step 2 of algorithm 4, can be restricted to the visible part of the source element j as seen from cubature node x_k on the receiving element i. As the visible part of the source element might be non-convex, the integral will possibly have to be split in several parts. Current algorithms and data structures to do so are however rather sophisticated or appear to require enormous amounts of storage [38, 162, 40];

- In the presence of occluders, the point-to-element form factors $K_{j,\beta}(x)$, viewed as a function of x, will also exhibit first-order discontinuities at for instance shadow boundaries. In order to correctly deal with these discontinuities, the receiving element i can be split along the discontinuity lines [71, 103, 104, 163, 13].

- When there is full visibility between two elements in an interaction, an analytical expression can be used for the inner integral $K_{j,1}(x)$ [9]. To our knowledge, no such analytical expressions exist for $K_{j,\beta}(x)$ with $\beta > 1$, for higher approximation on the source element j;

- An obvious way to obtain more precise numerical integration is simply by using better numerical integration rules. Such integration rules have more nodes and will therefore increase the computation cost;

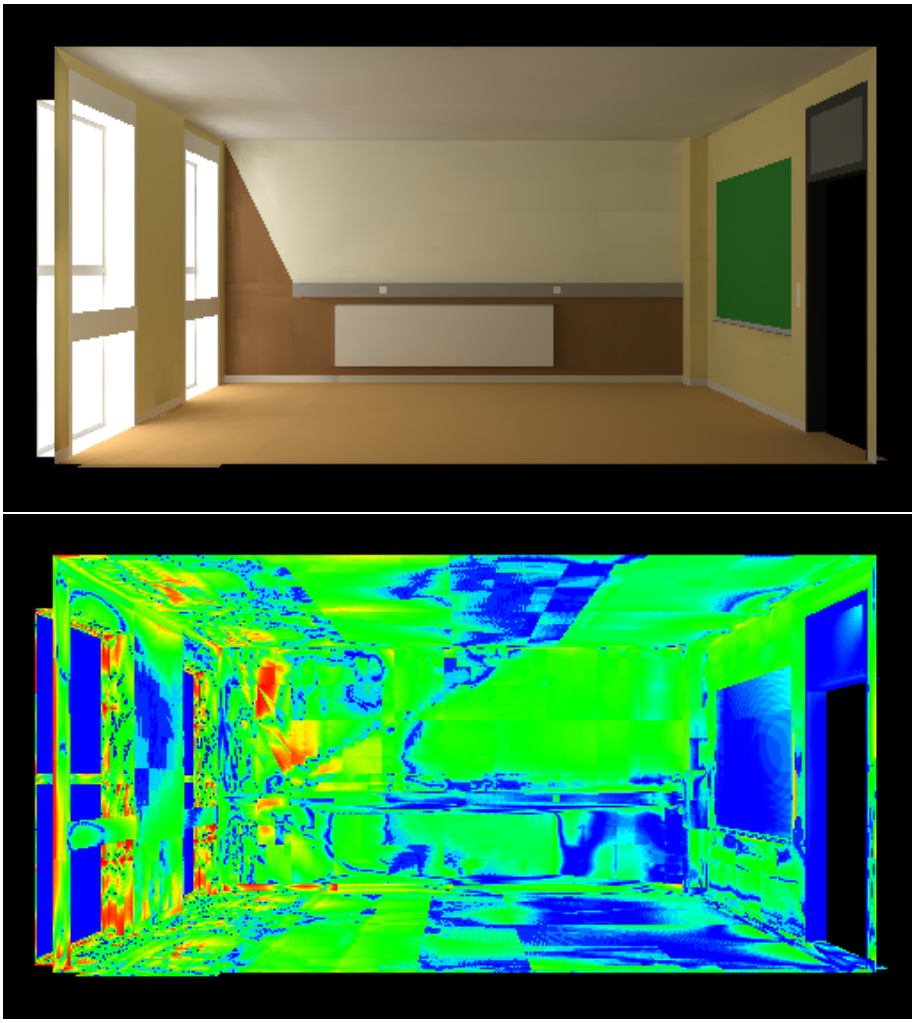


Figure 3.2: The top figure shows an image of an (empty) office, computed to 30 lux accuracy with a quadratic approximation on triangles and quadrilaterals. The average luminosity was about 600 lux. On the wall near the window, it was about 1300 lux. The bottom figure shows a false colour image indicating the ratio of the measured error and the requested accuracy (30 lux). The error was measured by computing the difference between the top image and a reference image, computed to much higher accuracy. A green colour indicates that the measured error is of the same magnitude as the requested accuracy. A blue colour indicates that the measured error was smaller and a yellow or red colour that the error was larger than 30 lux. The elements with larger error could not be subdivided further because their area was smaller than the minimum allowed element area.

- The point-x-to-element-j form factor integral (3.5) can be transformed to an integral over the solid angle $\Omega_j(x)$ under which j is seen from x. A useful transformation from the unit square to a spherical triangle can be found in [2];

- In §5, Monte Carlo integration techniques will be presented that allow to choose the number of samples adaptively in order to control numerical integration error.

3.5.2 Incorporation of other sources of error in the framework

A more selective approach to error control can result by adding extra terms describing these other sources of error to equation (3.12) and all other expressions derived from it. This requires a realistic estimate of the error on the computed radiosity due to potential inexact visibility calculations and due to the numerical integration rules that are used.

If such error estimates are available, they can be taken into account in the refinement criterion and strategy: a candidate link will be refined if the sum of all estimated error contributions exceeds the link-error threshold Δ. In case the criterion indicates that a candidate link is not admissible, the largest error source is determined, and appropriate actions are taken in order to reduce it.

If the largest error source appears to be due to inexact visibility on the receive element for instance, a selective discontinuity meshing algorithm will subdivide the receiver element along only those discontinuity lines that result in noticeable image artifacts. If the largest source of error however appears to be due to inexact numerical integration, one could switch to a numerical integration rule of higher precision (cfr. [53]). Embedded sequences of numerical integration rules, or adaptive integration, can be helpful in this context. More precise numerical integration may also be possible by using irradiance gradients [180, 3, 76].

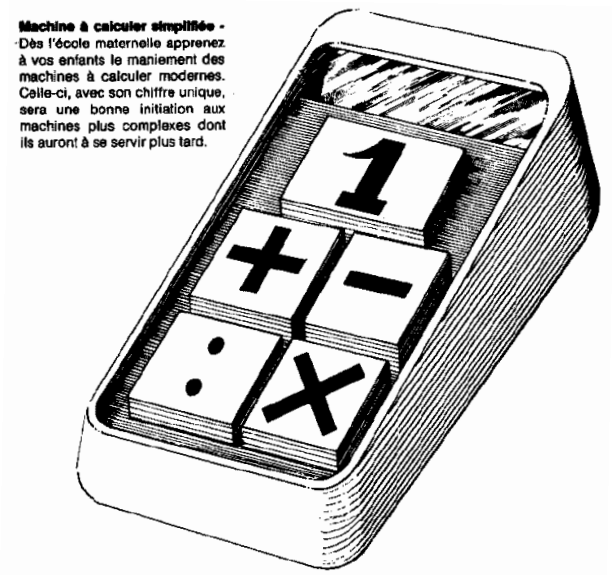
3.6 Conclusion

In this chapter, the discretisation error induced by a given mesh and set of basis functions has been analysed. Practical algorithms have been developed for a-posteriori evaluation of the discretisation error in a given radiosity solution, discretisation-error driven hierarchical refinement, and hierarchical radiosity computation to prescribed discretisation accuracy. Although only very simple measures have been taken in our implementation in order to deal with computational error, preliminary results are promising.

The main area for improvement is in the incorporation of sources of computational error in the presented framework, in particular due to inexact visibility calculations and imprecise numerical integration. This will require the development of realistic and efficient a-priori error estimates, and of selective discontinuity meshing algorithms.

In our experiments, only a very simple refinement strategy has been used. A second area for further research is the study of the more advanced discretisation-error based refinement techniques discussed in §3.3.2. Such techniques can lead to accurate radiosity solutions with fewer elements and less storage per element and link.

The following chapters of this dissertation focus on the Monte Carlo method as a method to control computational rather than discretisation error. In chapter 14, per-ray refinement will be presented as a strategy to combine Monte Carlo radiosity computation with hierarchical refinement, paving the way for algorithms in which both the discretisation and the computational error are dealt with adequately.



Simplified calculator: a good initiation for your children to the more complex machines they will need to use later in their life. J. Carelman, *Catalogue d'objets introuvables – tome 2*, Brodard et Taupin, Paris, France, 1978.

4 The Monte Carlo Method

The previous chapter focussed on the control of the discretisation error in hierarchical radiosity. The following chapters in this dissertation will present the Monte Carlo method as an alternative of full value for deterministic form factor computation and radiosity system solution. The following advantages of the Monte Carlo method will be demonstrated:

- flexible *computational error control*: heuristics will be developed that allow the amount of work to be tuned adaptively in order to achieve a prescribed computational accuracy;
- *reliability*: the Monte Carlo method automatically yields the correct result of a problem eventually;
- *low storage requirements* in radiosity system solution: the Monte Carlo method allows to solve very large systems of linear equations without explicit form factor computation and storage;
- *efficiency*: the amount of work that is required is rather loosely related to the number of polygons in the scene. In particular, fair-quality images showing complete diffuse illumination can often be obtained much more rapidly than with current deterministic methods;
- *compatibility*: it will be shown that the solution of the radiosity system of equations by Monte Carlo is compatible with the use of view-importance, higher order approximations and hierarchical refinement.

In this chapter, a brief overview of the Monte Carlo method in general is given. The reader is referred to numerous monographs and survey papers on the subject, such as [85, 66, 156, 127, 45, 17, 18, 64], for a more detailed description. In §4.1, a general description of the method and its benefits and current limitations is given. The main mathematical concepts behind the method are very briefly presented in §4.2. §4.3 and §4.4 deal with general techniques to make efficient use of the Monte Carlo method. This chapter is concluded with an overview of Monte Carlo methods in radiosity (§4.5), that will be described in detail in subsequent chapters.

4.1 Nature of the Monte Carlo method

The fundamental idea of the Monte Carlo method is to formulate the solution of a given mathematical problem as a parameter of a carefully chosen random variable. By sampling this random variable with the help of an electronic computer, the parameter yielding the solution of the problem is estimated. Most often, the mean value, also called the *expectation*, of the random variable is estimated. It is also possible to use the variance, or other parameters, but this text will deal only with Monte Carlo algorithms in which the mean value is estimated. The concepts of “random variable”, “expectation”, “variance” are briefly reviewed below in §4.2.

The main advantages of the Monte Carlo are its wide applicability and its conceptual simplicity. Its main limitation concerns its slow convergence.

4.1.1 Wide applicability

In principle, the Monte Carlo method can be applied to solve any problem for which a random variable can be designed such that the solution of the problem corresponds to the expectation (or another parameter) of the random variable. In order to be useful in practice, some, preferably efficient, algorithm for sampling the random variable by means of an electronic computer, shall be available.

Much Monte Carlo research has focussed on the design of efficient random variables for a wide variety of problems. This is fairly easy to imagine for problems that are stochastic in nature, such as for radiation transport problems in nuclear reactor design and in medicine, as well as for applications in operations research, e,g. involving queueing networks. Also for non-stochastic problems however, such as the solution of multiple integrals and certain types of integral equations and systems of linear equations, suitable and efficient random variables can be designed. An overview of estimators for systems of linear equations, such as in radiosity, is given in §4.5.

4.1.2 Simplicity

Once a suitable random variable and sampling algorithm have been designed, a Monte Carlo computation always consists of repeatedly drawing samples from the random variable and averaging the estimates obtained from each sample. If the variance of the random variable is finite (see below), the average of N samples converges to the required solution to arbitrary precision as more samples are taken. Although no deterministic error bounds are available in general, simple and non-application-specific algorithms exist to determine error bounds that are valid with arbitrary high probability if required.

Due to its conceptual simplicity, the Monte Carlo method is sometimes regarded as a method to avoid complicated mathematics when solving complicated mathematical problems. From an engineering point of view however, simplicity often implies reliability. My experience with Monte Carlo radiosity algorithms is that they are generally less error-prone and more reliable than their deterministic counterparts, while they take significantly less effort to implement.

4.1.3 Slow convergence

The convergence rate of Monte Carlo algorithms using independent samples is invariably $\mathcal{O}(1/\sqrt{N})$. This is discouragingly slow: in order to estimate the solution with 10 times higher accuracy (one more accurate decimal digit), 100 times more work is required.

Because of this, a lot of Monte Carlo research has focussed on the development of non-application specific techniques to obtain a smaller error for a given number of samples while keeping the $\mathcal{O}(1/\sqrt{N})$ convergence rate, or to increase the convergence rate:

1. *Variance reduction* techniques, such as importance sampling, stratified sampling (with fixed number of strata) and correlated sampling, aim at transforming a

given random variable into an equivalent one having lower *variance*. For a given number of independent samples, the error is reduced by a constant factor. The convergence rate remains $\mathcal{O}(1/\sqrt{N})$. A brief overview of the main variance reduction techniques will be given in §4.3;

2. A faster convergence rate, up to $\mathcal{O}(1/N)$, is obtained by using more uniform, but non-independent, sample number sequences in *quasi-Monte Carlo* methods [115]) or stratified sampling with increasing number of strata;

3. In *sequential Monte Carlo* methods a computation is split in stages such that the sampling strategy in a subsequent stage is adapted based on the result of previous stages. It has been shown that this may result in exponential convergence $\mathcal{O}(e^{-\lambda N})$ [63, 65, 154].

Although the basis for quasi-Monte Carlo and sequential Monte Carlo is quite old[1], both quasi-Monte Carlo and sequential Monte Carlo are topics of active ongoing research. Many unsolved theoretical and practical problems remain. We will therefore touch only briefly upon the application of quasi- and sequential Monte Carlo for radiosity.

4.1.4 Monte Carlo: a method of last resort

Because of its slow convergence, the Monte Carlo method is generally applied only as a method of last resort, when all other analytical or numerical methods fail. A notorious counter-example is the numerical integration of 1-dimensional functions. This application is often used to illustrate the principles of the Monte Carlo method as well as various variance reduction techniques. Most 1-dimensional integrals can however be solved much more efficiently using deterministic methods. The Monte Carlo method is however the only feasible method for solving high-dimensional integrals (dimension $d > 20$), or for non-smooth integrands on a complex domain: the amount of work required by deterministic methods generally is $\mathcal{O}(N^d)$, where d denotes the dimension of the integral. Deterministic methods also assume smoothness of the integrand and are designed for simple domains (e.g. simplices).

Something similar is true for the solution of linear systems: direct methods, such as Gaussian elimination, require $\mathcal{O}(n^3)$ work, where n denotes the size of the system, while iterative numerical methods, such as Jacobi-iterations, require $\mathcal{O}(n^2)$ operations in order to reach a fixed accuracy for linear systems such as in radiosity. Monte Carlo algorithms exist that require only $\mathcal{O}(n)$ operations. For small to moderately sized linear systems, direct or iterative numerical methods will be preferred, but for large systems of certain types, such as for solving the radiosity problem in an environment of several thousands of patches or more, Monte Carlo methods will be clear winners [34, 65].

4.2 Monte Carlo estimators

The random variable designed for estimating the solution of some problem, as explained above, is called an *estimator*. In the context of this work, only estimators

[1] For instance, Koksma's theorem [92] about the estimation of 1-dimensional integrals by uniform number sequences dates back to 1943, years before the earliest publications on the Monte Carlo method.

are considered for which the expectation is related to the solution of the problem under consideration. In this section, we briefly remind the reader of the main concepts needed in Monte Carlo and introduce our notation. A more complete and rigourous treatment can be found in aforementioned monographs and survey papers on the Monte Carlo method.

4.2.1 Random variables

In general, a random variable is a set of pairs (X, p_X) of "outcomes" X and associated probabilities p_X. The meaning of such a pair is that outcome X will be observed with probability p_X. The outcomes can be anything, e.g. the side appearing on top after tossing a coin. The probabilities p_X are positive real numbers that sum to 1. The random variables that will be needed in this work take two forms:

- (a_i, p_i), where a_i is a real number or vector and p_i denotes the probability associated with the term a_i;
- $(f(x), p(x))$, where $f(x)$ is the value of a real-valued function f defined on a, possibly multi-dimensional, domain D. $x \in D$ is a point in the domain that occurs with a probability density $p(x)$.

The former is a so called discrete random variable: the set of possible outcomes is countable. The latter is a continuous random variable.

4.2.2 The expectation of a random variable

In the context of this work, we need:

- for the discrete random variable (a_i, p_i) with n possible outcomes:

$$E_p[a] = \sum_{i=1}^{n} a_i p_i. \tag{4.1}$$

- for the continuous random variable $(f(x), p(x))$ on domain D:

$$E_p[f] = \int_D f(x)p(x)\,dx \tag{4.2}$$

The subscript p, denoting the probability distribution w.r.t. which the expectation is to be considered, will be dropped in future formulae if there is no confusion possible.

4.2.3 The variance of a random variable

The variance of a random variable is the expected square deviation from the expectation. It is a measure for the non-constantness of the outcome of a random variable and plays a major role in Monte Carlo error analysis. In the context of this work, we will need:

- for the discrete random variable (a_i, p_i) with n possible outcomes:

$$V_p[a] = \sum_{i=1}^{n} (a_i - E_p[a])^2 p_i = E_p[a^2] - (E_p[a])^2 = \sum_{i=1}^{n} a_i^2 p_i - \left(\sum_{i=1}^{n} a_i p_i \right)^2 \tag{4.3}$$

- for the continuous random variable $(f(x), p(x))$ on domain D:

$$V_p[f] = \int_D (f(x) - E_p[f])^2 p(x)\, dx = \int_D f^2(x) p(x)\, dx - \left(\int_D f(x) p(x)\, dx \right)^2 . \tag{4.4}$$

Note that the expectation and variance of a random variable may not be finite in some cases.

4.2.4 Simple Monte Carlo estimation of sums and integrals

The definition of expectation above immediately suggests the following simple Monte Carlo estimators:

- for a discrete sum $S = \sum_{i=1}^{n} a_i$: consider the random variable $\hat{S} = (na_i, \frac{1}{n})$ in which each term of the sum has equal probability $1/n$. It is easy to verify that $E[\hat{S}] = S$. The variance is given by $V[\hat{S}] = n \sum_{i=1}^{n} a_i^2 - S^2$.

  This estimator leads to the following simple algorithm to estimate a sum S: first choose randomly a term a_i from S whereby each term has the same chance $1/n$ of being selected. Return na_i, the value of the selected term times the number of terms, as an estimate for the sum.

  Summation of real numbers is however an operation that is implemented very efficiently in modern electronic computer hardware. The Monte Carlo estimation of a sum as illustrated here will not compete with straightforward summation by the machine unless the number of terms n is exceptionally large or the terms are not simple real numbers, but are the result of a lengthy computation. The latter is the case for the sums that appear in the Neumann expansion of the solution of the system of radiosity equations (see §4.5).

- for a (possibly multi-dimensional) integral $I = \int_D f(x)\, dx$: consider the random variable $\hat{I} = (V_D\, f(x), \frac{1}{V_D})$, where V_D denotes the volume of the domain D. It is equally easy to verify that $E[\hat{I}] = I$ and $V[\hat{I}] = V_D \int_D f^2(x)\, dx - I^2$.

  This estimator suggests to select a point $x \in D$, whereby each point has the same probability density $1/V_D$ of being selected. $V_D\, f(x)$ is returned as an estimate for the integral.

A simple method of obtaining better estimates of a quantity, is to draw multiple, independent, samples and to use their average as a new estimate. This practice will be justified below and the rate at which the estimation error is reduced as more samples are taken is analysed next.

4.2.5 Secondary estimators

Consider two estimators $\hat{S}_1$ and $\hat{S}_2$ yielding some quantities S_1 and S_2 as their expectation. Any linear combination $w_1\hat{S}_1 + w_2\hat{S}_2$ with constant weights w_1 and w_2 has expectation:

$$E[w_1\hat{S}_1 + w_2\hat{S}_2] = w_1E[\hat{S}_1] + w_2E[\hat{S}_2] = w_1S_1 + w_2S_2. \tag{4.5}$$

The variance is given by

$$V[w_1\hat{S}_1 + w_2\hat{S}_2] = w_1^2V[\hat{S}_1] + w_2^2V[\hat{S}_2] + 2w_1w_2\mathrm{Cov}[\hat{S}_1, \hat{S}_2] \tag{4.6}$$

with *covariance*

$$\mathrm{Cov}[\hat{S}_1, \hat{S}_2] = E[\hat{S}_1 \cdot \hat{S}_2] - E[\hat{S}_1] \cdot E[\hat{S}_2]. \tag{4.7}$$

If $\hat{S}_1$ and $\hat{S}_2$ are independent, the covariance is zero[2]. This result can easily be generalised to the linear combination of any number of estimators.

A special case is the linear combination of N times the same random variable $\hat{S}$: $\hat{S}_N = \sum_{i=1}^{N} \frac{1}{N}\hat{S}$. An outcome of $\hat{S}_N$ corresponds to the average of N independent samples from $\hat{S}$. The expectation is $E[\hat{S}_N] = E[\hat{S}] = S$, justifying the practice, mentioned above of using averages of independent samples. The variance is $V[\hat{S}_N] = \sum_{i=1}^{N} (\frac{1}{N})^2 V[\hat{S}] = V[\hat{S}]/N$.

4.2.6 Accuracy of the Monte Carlo method

There are two basic theorems that explain how the error on the average of N independent samples from a estimator $\hat{S}$ for the quantity S reduces as the number of samples N is increased. The error bounds provided by these theorems are however not deterministic, but probabilistic.

The first is *Chebyshev's inequality* which states that the probability that any sample from a random variable $\hat{S}$, with finite expectation $E[\hat{S}] = S$ and finite variance $V[\hat{S}]$, deviates more than $\sqrt{V[\hat{S}]/\delta}$ from S, with δ any positive number, is smaller than δ:

$$\mathrm{Prob}\left(|\hat{S} - S| \geq \sqrt{\frac{V[\hat{S}]}{\delta}}\right) < \delta. \tag{4.8}$$

Since $V[\hat{S}_N] = V[\hat{S}]/N$, for the average of N independent samples from $\hat{S}$, we obtain

$$\mathrm{Prob}\left(|\hat{S}_N - S| \geq \sqrt{\frac{V[\hat{S}]}{N\delta}}\right) < \delta.$$

[2] A zero covariance is a necessary, but not a sufficient condition for independence: there exist dependent estimators that also have zero covariance (see e.g. [85, p.13]).

With fixed probability δ, the average of N independent samples will be contained in intervals of size decreasing as $1/\sqrt{N}$ as the number of samples N is increased. The size of the intervals is also proportional to $\sqrt{V[\hat{S}]}$.

A stronger statement about the accuracy of a Monte Carlo computation is given by the *central limit theorem of probability*. This theorem states that the average of N independent samples of any random variable $\hat{S}$, with finite variance $V[\hat{S}]$ and expectation $E[\hat{S}]$, is asymptotically normal distributed as $N \to \infty$ and that, in the limit for large N:

$$\text{Prob}\left(a\sqrt{\frac{V[\hat{S}]}{N}} \leq \hat{S} - S \leq b\sqrt{\frac{V[\hat{S}]}{N}}\right) = \frac{1}{\sqrt{2\pi}}\int_a^b e^{-\frac{t^2}{2}}\,dt. \tag{4.9}$$

Tables exist for the integral on the right hand side. Asymptotically, the computed estimate using N samples is within one *standard error* $\sqrt{V[\hat{S}]/N}$ 68,3% of the time, within 2 standard errors 95.4% of the time and within 3 standard errors from the correct result 99.7% of the time.

4.2.7 Biased and consistent estimators

An estimator with expectation that differs from the quantity to be computed is called a *biased* estimator. The amount by which the estimator is "wrong" is called the *bias*: $B[\hat{S}] = E[\hat{S}] - S$. The total error on the resulting estimates is characterised by the sum of the bias and the square root of the variance. Biased estimators can be useful if a lower variance compensates for the bias. A full error analysis requires that both variance and bias are analysed and thus is more complicated than for unbiased estimators.

An estimator with finite variance and vanishing bias in the limit of large number of samples, is called a *consistent* estimator. This is for instance the case if the bias can be shown to be proportional to the variance, or if it is a decreasing fraction of the variance.

4.2.8 Estimating variance

In an actual computation, the variance $V[\hat{S}_N]$ can be estimated using the same N independent samples $s_i, i = 1, \ldots, N$ used to compute an estimate $\frac{1}{N}\sum_{i=1}^N s_i$ for S:

$$V[\hat{S}_N] \approx \frac{1}{N-1}\left[\frac{1}{N}\sum_{i=1}^N s_i^2 - \left(\frac{1}{N}\sum_{i=1}^N s_i\right)^2\right]. \tag{4.10}$$

Alternatively, if a closed form expression is available for $V[\hat{S}]$, and it is feasible to evaluate it efficiently on the computer, the variance can be approximated by filling in the actual estimate for the solution as well as any other required data in the closed form expression.

This text will frequently present variance formulae for the estimators that are proposed. Such formulae allow to better compare different Monte Carlo radiosity sampling strategies and to better understand the conditions under which each will be preferable. Closed form formulae for the variance can also be used for estimating error in actual computations as well as for combining different estimators.

4.3 Variance reduction techniques

The best understood way of increasing the efficiency of Monte Carlo algorithms is to transform one or more basic estimators, such as those for summation and integration presented in the previous section, to an equivalent estimator with lower variance. The lower the variance, the fewer samples will be needed to compute the quantity of interest to given accuracy and confidence.

In this section, a brief overview is presented of those variance reduction techniques that have been proven useful in the context of Monte Carlo radiosity. An excellent in-depth discussion of variance reduction techniques can be found in [81] as well as in other classic texts on Monte Carlo.

We shall see that in some ideal cases, an estimator with zero variance may result. Each sample of such an estimator yields the same estimate. If the estimator is also unbiased, these estimates will be equal to the quantity to be computed. An unbiased estimator with zero variance is called a *perfect estimator*. In practice, perfect estimators can only be approximated except in trivial cases, because they require that the solution of the problem under consideration be known in advance. Perfect estimators however form the basis of sequential Monte Carlo algorithms, allowing exponential convergence when carried out well.

Most of the variance reduction techniques described here also reduce the error when applied with low discrepancy sampling in quasi-Monte Carlo [158].

4.3.1 Importance sampling

The basic idea of importance sampling is to change the probability distribution of an estimator in such a way that each sample of the changed estimator will yield an approximately equal estimate.

In the case of Monte Carlo estimation of a sum $S = \sum_{i=1}^{n} a_i$, this means to sample terms from the sum so that each term a_i has a probability p_i, not necessarily $p_i = 1/n$, of being selected. In order to be unbiased, the resulting random variable is of the form $\hat{S}^I = (a_i/p_i, p_i)$. The probabilities p_i should be chosen so that the variance

$$V[\hat{S}^I] = \sum_{i=1}^{N} \frac{a_i^2}{p_i} - S^2 \tag{4.11}$$

is minimal. This is the case if p_i is taken proportional to a_i: $p_i = a_i/c$. In this case, each estimate yields $a_i/p_i = c$, which is obviously constant, so that the variance is zero. In order to be unbiased, c should equal S however, so that the solution of the problem would have to be known in advance.

Good variance reduction can be obtained already by taking p_i approximately proportional to a_i. In practice, the probabilities shall fulfil three conditions:

- They shall be be normalised: $\sum_{i=1}^{n} p_i = 1$;
- Selecting a term a_i with probability p_i shall not be too complicated: the computation cost of doing so shall not out-weigh the saving that fewer samples will suffice;
- The p_i shall never be zero if $a_i \neq 0$: every non-zero term of the sum, no matter how unimportant, shall have a nonzero probability of being sampled. Neither shall the p_i be too small compared to a_i^2 or an actual increase of variance could result [118].

4.3.2 Weighted sampling

The second requirement above can be relaxed: there are ways to obtain (nearly) the same result as in importance sampling without actually having to sample according to a, potentially very complicated, probability distribution p_i. Techniques to do so receive very little attention in most Monte Carlo texts. Although they are potentially very useful in global illumination, they appear not yet to have been studied. They will therefore be described here in a bit more detail.

Halton [63] noted that when a good probability density function (pdf) p is available, so that every estimate is about equal, good estimates of the quantity to be computed also result when actually sampling according to another, more convenient, pdf q and to do *as if* p was used. Such a technique is biased, but the bias $B[\hat{S}^B] = E[\hat{S}^B] - S$ is small when the target pdf p is good: $B[\hat{S}^B]$ even vanishes if p would lead to a perfect estimator. In that case, also $\hat{S}^B$ is perfect (and no longer biased).

The corresponding estimators for sums are of the form $\hat{S}^B = (a_i/p_i, q_i)$ with

$$E[\hat{S}^B] = \sum_{i=1}^{n} \frac{a_i}{p_i} q_i = S + \sum_{i=1}^{n} \frac{a_i}{p_i}(q_i - p_i). \tag{4.12}$$

If $p_i = a_i/S$, $E[\hat{S}^B] = S \sum_{i=1}^{n} q_i = S$ (because q_i is normalised). Such an estimator may have lower variance, but the bias is in general very hard to compute and does not vanish as more samples are taken.

A consistent class of estimators, based on this observation, has been proposed by Powell and Swann [124], following an idea of Handscomb [67]. The technique, called *weighted sampling*, was later generalised by Spanier [153, 158].

The basic idea is to consider p_i as a distribution of weights instead of as a probability distribution. Suppose the sum $T = \sum_{i=1}^{n} a_i p_i$ needs to be estimated. With importance sampling, N terms $a_{s_k}, k = 1, \ldots, N$ are selected with probability p_{s_k}. The resulting estimate for the sum is $\sum_{k=1}^{N} \frac{1}{N} a_{s_k}$: the primary estimates a_{s_k} are combined with equal weights $w_k = 1/N$. Selecting terms with probability p_{s_k} might however be complicated.

In a comparable situation, Handscomb proposed to take uniform samples, but to compensate for sampling the "wrong" pdf by weighting the primary estimates proportional to p_i: $w_{s_k} = p_{s_k} / \sum_k p_{s_k}$ instead of assigning them equal weights. For estimation of the sum $S = \sum_{i=1}^{n} a_i$ with *target* pdf p_i but using uniform sampling,

the following estimates are obtained:

$$S \approx \hat{S}_N^W = \sum_{k=1}^{N} \frac{p_{s_k}}{\sum_{k=1}^{N} p_{s_k}} \frac{a_{s_k}}{p_{s_k}} = \frac{\sum_{k=1}^{N} a_{s_k}}{\sum_{k=1}^{N} p_{s_k}}. \tag{4.13}$$

Powell and Swann showed that the variance $V[\hat{S}_N^W]$ of such an estimator is asymptotically (for large N) comparable with the variance $V[\hat{S}_N^I]$, that would be obtained by importance sampling according to the pdf p_i. The bias $B[\hat{S}_N^W]$ vanishes as $\sqrt{V[\hat{S}^I]}/N$, i.o.w. $\mathcal{O}(1/\sqrt{N})$ times faster than the standard error with N samples itself. Note that when $p_i = a_i/S$, the above estimates also always yield S without error.

Spanier generalised Powell and Swann's *weighted uniform sampling* estimators to the case where sampling is done according to a non-uniform *source* pdf q_i. The resulting estimates are of the form

$$S \approx \frac{\sum_{k=1}^{N} a_{s_k}/q_{s_k}}{\sum_{k=1}^{N} p_{s_k}/q_{s_k}}. \tag{4.14}$$

These techniques allow cheap samples to be combined in such a way that the effect of using more expensive samples is imitated, at the cost of some bias.

4.3.3 Control variates

A different way of reducing variance is suggested by formula (4.6) for the variance of the linear combination of estimators. While the variances are always positive, covariances can take both positive or negative values. When subtracting two estimators $\hat{S}$ and $\hat{T}$ with positive covariance $\mathrm{Cov}[\hat{S}, \hat{T}] > 0$, the variance $V[\hat{S} - \hat{T}] = V[\hat{S}] + V[\hat{T}] - 2 \cdot \mathrm{Cov}[\hat{S}, \hat{T}]$ of the combined estimator may even be lower than the variance of either of the combined estimators individually.

In the control variates technique, this effect is obtained by estimating the difference $S - T$ between a quantity S to be computed and a carefully chosen second quantity T, which is known so that it can be added to the difference $S - T$ in order to obtain S. Applied to the Monte Carlo estimation of sums, this would mean to find a second, known, sum $T = \sum_{i=1}^{n} b_i$ and writing

$$S = T + \sum_{i=1}^{n} (a_i - b_i).$$

The latter sum is estimated using Monte Carlo. This can be accomplished by estimating S and T simultaneously using the same random numbers. Using importance sampling with probability distribution p_i, the variance becomes

$$V[\hat{(S - T)}^I] = \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{p_i} - (S - T)^2. \tag{4.15}$$

A perfect estimator results if $a_i = b_i$, but this again implies that the solution of the problem under consideration would have to be known in advance. A second perfect

estimator results if p_i is chosen proportional to $a_i - b_i$, but also in that case it turns out that S would have to be known in advance. Note that $a_i - b_i$ may be negative so that care needs to be taken in order to sample a signed probability distribution function [155, 118].

4.3.4 Combining estimators

Suppose two estimators $\hat{S}_1$ and $\hat{S}_2$ for a given quantity S are available. Any linear combination $w_1\hat{S}_1 + w_2\hat{S}_2$ with constant weights $w_1 + w_2 = 1$ will then also be an estimator for S. The variance of the linear combination however depends on the weights. By minimising the variance (4.6), using the technique of Lagrange multipliers with constraint $w_1 + w_2 = 1$, it can be shown that the optimal weights fulfil

$$\frac{w_1}{w_2} = \frac{V[\hat{S}_2] - \mathrm{Cov}[\hat{S}_1, \hat{S}_2]}{V[\hat{S}_1] - \mathrm{Cov}[\hat{S}_1, \hat{S}_2]} \tag{4.16}$$

For independent estimators, the optimal weights are inversely proportional to the variance.

The optimal weights w_m for combining M estimators $\hat{S}_m, m = 1, \ldots, M$ for S, are obtained by solving the following system of $M+1$ linear equations with unknowns $w_1, \ldots, w_M, \lambda$:

$$\left\{\begin{array}{ccccccccc} V_{11}w_1 & + & V_{12}w_2 & + & \cdots & + & V_{1M}w_M & = & \lambda \\ V_{21}w_1 & + & V_{22}w_2 & + & \cdots & + & V_{2M}w_M & = & \lambda \\ & & & & \cdots & & & & \\ V_{M1}w_1 & + & V_{M2}w_2 & + & \cdots & + & V_{MM}w_M & = & \lambda \\ w_1 & + & w_2 & + & \cdots & + & w_M & = & 1 \end{array}\right. \tag{4.17}$$

where $V_{\alpha\beta} = \mathrm{Cov}[\hat{S}_\alpha, \hat{S}_\beta]$. Note that $V_{\alpha\beta} = V_{\beta\alpha}$ and that $V_{\alpha\alpha} = V[\hat{S}_\alpha]$. If N_m out of a total of N samples are taken from each estimator $\hat{S}_m$, yielding primary estimates $\tilde{S}_m^k, k = 1, \ldots, N_m$ for S, the combined estimate is:

$$S \approx \sum_{m=1}^{M} w_m \frac{1}{N_m} \sum_{k=1}^{N_m} \tilde{S}_m^k \tag{4.18}$$

If the covariances are estimated from the samples themselves as explained in §4.2.8, the result is slightly biased, but consistent.

4.3.5 Multiple importance sampling and mixture sampling

Veach [176] noted that a better combination may result by assigning potentially different weights w_m^k to each individual sample, even for samples from the same estimator:

$$S \approx \sum_{m=1}^{M} \frac{1}{N_m} \sum_{k=1}^{N_m} w_m^k \tilde{S}_m^k . \tag{4.19}$$

The corresponding combined estimator is unbiased as long as $\sum_{m=1}^{M} w_m^k = 1$ for every sample. Heuristics for determining the combination weights were derived for the important case that the estimators are different importance-sampling estimators. The basic idea behind these heuristics is to give a sample a weight that takes into account the probability that the sample would have resulted with the other estimators: if the sample would be generated only with low probability with the other estimators, it is given a large weight. Vice versa, when any other estimator would yield the same sample with high probability, the weight of the sample is reduced.

Applied to the Monte Carlo estimation of a sum $S = \sum_{i=1}^{n} a_i$, we are dealing with estimators of the form $\hat{S}_m = (a_i/p_i^{(m)}, p_i^{(m)})$. The primary estimates are of the form $a_{i_m^k}/p_{i_m^k}^{(m)}$ where i_m^k denotes the index of the k-th sample from the sum taken according to pdf $p^{(m)}$. A good heuristic for determining the weights is the so called *balance heuristic*:

$$w_i^{(m)} = \frac{N_m p_i^{(m)}}{\sum_{\alpha=1}^{M} N_\alpha p_i^{(\alpha)}} \tag{4.20}$$

leading to secondary estimates of the form

$$S \approx \frac{1}{N} \sum_{m=1}^{M} \sum_{k=1}^{N_m} \frac{a_{i_m^k}}{\sum_{\alpha=1}^{M} c_\alpha p_{i_m^k}^{(\alpha)}} \tag{4.21}$$

where c_α denotes the fraction of samples taken from $\hat{S}_\alpha$: $N_\alpha = c_\alpha \cdot N$.

Recently, Owen [118] has proposed a generalisation of this idea, which can be viewed as a unification of multiple importance sampling and the regression technique of the previous section.

4.3.6 Treating part of the problem by other methods than Monte Carlo

The above variance reduction techniques exploit knowledge of one or more "easy functions" p_i or b_i, in order to reduce sampling variance. Important efficiency gains may also result by treating parts of the problem under consideration by analytic or deterministic methods, if applicable.

Consider for instance the estimator $E_p[a] = \sum_i \sum_j a_{ij} p_{ij} = S$ where the probabilities p_{ij} can be decomposed in marginal and conditional probabilities: $p_{ij} = r_i q_{j|i}$. Then:

$$S = E_p[a] = \sum_i \sum_j a_{ij} p_{ij} = \sum_i r_i \left(\sum_j a_{ij} q_{j|i} \right) = \sum_i r_i E_q[a|i] = E_r \left[E_q[a|i] \right]. \tag{4.22}$$

$E_q[a|i]$ denotes the conditional expectation of a_{ij} w.r.t. the conditional probability distribution $q_{j|i}$ for given fixed i. The variance $V_p[a]$ can then be written as a sum

$$V_p[a] = E_r \left[V_q[a|i] \right] + V_r \left[E_q[a|i] \right] \tag{4.23}$$

Proof:

$$\begin{aligned}
V_p[a] &= \sum_i \sum_j a_{ij}^2 p_{ij} - \left(\sum_i \sum_j a_{ij} p_{ij} \right)^2 \\
&= \sum_i r_i \sum_j a_{ij}^2 q_{j|i} - \left(\sum_i r_i \sum_j a_{ij} q_{j|i} \right)^2 \\
&= \sum_i r_i \left[\sum_j a_{ij}^2 q_{j|i} - \left(\sum_j a_{ij} q_{j|i} \right)^2 \right] \\
&+ \left[\sum_i r_i \left(\sum_j a_{ij} q_{j|i} \right)^2 - \left(\sum_i r_i \sum_j a_{ij} q_{j|i} \right)^2 \right] \\
&= \sum_i r_i V_q[a|i] + V_r \left[\sum_j a_{ij} q_{j|i} \right] \\
&= E_r \left[V_q[a|i] \right] + V_r \left[E_q[a|i] \right] .
\end{aligned}$$

Formula (4.23) will be used several times in order to compute the variance of multi-dimensional estimators. It also indicates that if the inner sum $\sum_j a_{ij} q_{j|i}$ is computed by direct summation for instance, the variance $V_q[a|i] = 0$ and a reduction of the variance $V_p[a]$ results.

4.3.7 Other variance reduction techniques

The reader will find various other variance reduction techniques in Monte Carlo literature, such as antithetic variates, the use of orthogonal polynomials and conditional Monte Carlo. It is not clear to what extent these techniques can be applied efficiently in Monte Carlo radiosity, except possibly for the Monte Carlo computation of form factors. Remember however that it is our goal to avoid explicit computation of form factors, as well as their storage.

4.4 Sampling random variables

Sampling a random variable generally consists of two tasks: 1) selection of one of the possible outcomes ("events" or "states") of the random variable, and 2) computation of a value associated with the selected event or state. In the case of sums $S = \sum_{i=1}^{n} a_i$, one needs to select one or more indices i_k such that the probability of selecting each possible index corresponds to given probabilities p_i. In the case of integrals, a point x needs to be selected in a domain D, such that the probability of selecting a point from an infinitesimal region around x corresponds to a given probability density function $p(x)$. In this section, we briefly remind the reader of the most important techniques to carry out such sampling. Our focus is on sampling discrete probability functions, needed for estimating sums.

4.4.1 Inverting the cumulative distribution

The *cumulative probability distribution* (cdf) corresponding to $p_i, i = 1, \ldots, n$ is $P_i = \sum_{j=1}^{i} p_i$. Since p_i is normalised, $P_n = 1$. We take $P_0 = 0$. Random selection of an index i with probability p_i, can be accomplished by generating a uniform random number $\xi \in (0, 1)$ and finding the index i for which

$$P_{i-1} = \sum_{j=1}^{i-1} p_j \quad < \quad \xi \quad \leq \quad P_i = P_{i-1} + p_i \tag{4.24}$$

This can be done in $\mathcal{O}(\log_2 n)$ time per sample by binary search in a precomputed table with the P_i. Pre-computation of the P_i of course takes $\mathcal{O}(n)$ time.

4.4.2 Stratified sampling

The basic idea of stratified sampling is to place samples more uniformly by subdividing the domain to be sampled in a number of so called *strata*. An appropriate number of samples is drawn independently in each stratum.

Often, stratified sampling needs to be combined with importance sampling. A good practice is to take uniformly distributed samples in equal-sized strata and with equal number of samples in each stratum. These uniform samples are then transformed using the inverse of the cumulative importance-sampling probability distribution. It is advantageous to use the same random numbers for taking samples in each stratum as this leads to an increased convergence rate. For 1-dimensional integrals, this can be shown to be related to the trapezoid rule for integration, with $\mathcal{O}(1/N)$ convergence rate instead of $\mathcal{O}(1/\sqrt{N})$.

Algorithm 6 shows an efficient algorithm, based on these ideas, for taking N samples of a sum $S = \sum_{i=1}^{n} a_i$, using probabilities p_i [111]. A single random number ξ is generated. The i-th term of the sum is sampled $\lfloor P_i \cdot N + \xi \rfloor - \lfloor P_{i-1} \cdot N + \xi \rfloor$ times, where $\lfloor x \rfloor$ denotes the largest integer number smaller or equal to x. Roughly stated, the strata correspond to the terms a_i that would be sampled using a random number in each interval $(\frac{k-1}{N}, \frac{k}{N})$, $k = 1, \ldots, N$. Algorithm 6 has time complexity $\mathcal{O}(n)$.

Because the number of equal-sized strata increases exponentially with the dimension, straightforward stratified sampling is not feasible in higher dimensions. In higher dimensions, related techniques such as *Latin-hyper-cube sampling* shall be preferred.

4.4.3 Rejection sampling

The term rejection sampling indicates a class of sampling techniques in which *tentative* samples are proposed and tested for acceptability until an accepted sample results. If a tentative sample is rejected, a new tentative sample is generated and tested.

In this way, sampling techniques for pdf's can be developed that do not rely on the corresponding cdf. For sums for instance, it suffices to know a sum $T = \sum_{i=1}^{n} b_i$ with every $b_i \geq p_i$. One particular case is $b_i = \overline{p}$ with $\overline{p} \geq p_i, \forall i$. A term (index i_k for the k-th trial) is selected with probability b_{i_k}/T. In case all b_i are equal, this corresponds to uniform sampling. The tentative sample i_k is accepted if $p_{i_k} \leq b_{i_k} \cdot \xi_k$, where ξ_k is a random number in the range $(0, 1)$ as usual.

Algorithm 6: Stratified sampling of a sum $S = \sum_{i=1}^{n} a_i$ with N samples and pdf p_i.

1. Compute a random number $\xi \in (0, 1)$;
2. Initialise $N_{\text{prev}} \leftarrow 0$, $P \leftarrow 0$;
3. For all $i = 1, \ldots, n$,
   (a) $P \leftarrow P + p_i$;
   (b) $N_i \leftarrow \lfloor P \cdot N + \xi \rfloor - N_{\text{prev}}$;
   (c) Sample the i-th term of the sum N_i times;
   (d) $N_{\text{prev}} \leftarrow N_{\text{prev}} + N_i$.

The main drawback of such a technique is that the cost of generating samples can be quite high: the probability that a tentative sample is accepted equals S/T. Note that this technique does not require that the p_i are normalised in order to generate samples. The normalisation is however still needed for computing estimates, unless it can be made to cancel in some way.

4.4.4 Sampling a linear combination of pdf's

Sometimes, a pdf $q_i = \sum_{m=1}^{M} c_m p_i^{(m)}$, which is a linear combination of M primary pdf's $p^{(m)}, m = 1, \ldots, M$ needs to be sampled. In order to do so, a primary pdf $p^{(m)}$ is selected first with probability $c_m / \sum_{m=1}^{M} c_m$. Next, a sample is drawn using $p^{(m)}$. Although only a single primary pdf is sampled, the result is that a sample is generated according to the combined pdf: any pdf could have been chosen. The resulting estimates are of the same form as when the balance heuristic is used (equation (4.21)). The difference is that a primary pdf is chosen for each sample independently here, instead of allocating in advance the number of samples to be taken according to each pdf.

A special case occurs when estimating double sums $S = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}$ where terms are to be selected according to some probabilities p_{ij}. Although such a sum can always be written as a 1-dimensional sum, it is more convenient in some cases to first select a first index, for instance j with probability p_j, and next select the second index i with *conditional* probability $p_{i|j}$. The sampled probabilities fulfil $p_{ij} = p_{i|j} p_j$. Any of the sampling techniques described above can be used for sampling either j or i conditional upon j.

4.4.5 Other sampling techniques

Only some generally applicable sampling techniques have been surveyed here. Other general sampling techniques have received no attention yet in Monte Carlo radiosity or may be difficult to apply. Note that for many pdf's, custom sampling techniques can be developed [46]. An important example in the context of Monte Carlo radiosity is the selection of a patch j in a discretised scene, with probability equal to the form factors F_{ij} for a fixed patch i. This will be explained in §5.3 and §6.1.3.

4.5 Monte Carlo Radiosity

In the remaining chapters of this dissertation, a detailed overview will be given of the various ways in which the Monte Carlo method can be used in order to efficiently solve the systems of linear equations that arise in radiosity. Until chapter 13 we will deal with constant radiosity approximations. Chapter 13 explains how the methods for constant approximations can be generalised for higher order approximations.

There are various alternative forms of the system of equations (1.1), which models the radiosity problem with constant approximations. These alternative forms give rise to different Monte Carlo radiosity algorithms and will be referred to often in the following chapters. They are derived below. An overview of how the Monte Carlo method can be used in order to solve these systems of equations will be presented at the end of this section (§4.5.5).

4.5.1 The radiosity and power system: gathering radiosity and shooting power

In the first radiosity approaches [58, 27], the following systems of equations, derived in chapter 2, are solved:

$$B_i = E_i + \sum_j \rho_i F_{ij} B_j \tag{4.25}$$

with form factors

$$F_{ij} = \frac{1}{A_i} \int_{S_i} \int_{S_j} G(x,y)\, dA_y\, dA_x \tag{4.26}$$

$$= \frac{1}{A_i} \int_{S_i} \int_{\Omega_j(x)} \frac{\cos\theta_x}{\pi}\, d\omega_{\Theta_x}\, dA_x. \tag{4.27}$$

The system of equations (4.25) is traditionally solved using iterative numerical methods such as Jacobi and Gauss-Seidel [58, 57]. Because in each iteration step only a single radiosity value is updated, based on the radiosity values of all other patches, these methods have been called *gathering* methods. In each step, all form factors F_{ij} for a fixed patch i need to be computed. Traditionally, the *hemi-cube* method is used to do this [27].

Multiplying both sides of (4.25) by A_i, the surface area of patch i, and using the reciprocity relation $A_i F_{ij} = A_j F_{ji}$ (which follows immediately from (4.26)), leads to the following system of equations:

$$P_i = \Phi_i + \sum_j P_j F_{ji} \rho_i. \tag{4.28}$$

$P_i = A_i B_i$ is the total power emitted by i. $\Phi_i = A_i E_i$ is the self-emitted power.

In progressive refinement radiosity [26], this system of equations is solved using an iterative numerical solution method which is closely related to *Southwell relaxation* [60]. In progressive refinement radiosity, the radiosity of a large number of patches i is updated by *shooting* the unshot power from a single patch j. The formulation in terms of power instead of radiosity is convenient, because it is easier in practice to compute simultaneously all form factors F_{ji} for a given source j than for a fixed receiver i.

4.5.2 Adjoint systems of equations

A system of linear equations like (4.25) or (4.28) can be written in the form $\mathbf{C}\mathbf{x} = \mathbf{e}$ where $\mathbf{C}$ is a (given) $n \times n$ matrix of real numbers c_{ij}, $\mathbf{e}$ is a known vector of n real numbers e_i and the solution $\mathbf{x}$ is a vector of n real numbers x_i that need to be determined:

$$\sum_{j=1}^{n} c_{ij} x_j = e_i. \tag{4.29}$$

Many problems now can be expressed as scalar products $< \mathbf{w}, \mathbf{x} >= \sum_i w_i x_i$ of the solution $\mathbf{x}$ of such a system of equations and a *weight vector* $\mathbf{w}$. A special case is $\mathbf{w}^k$ with $w_i^k = \delta_{ik}$ where δ_{ik} denotes Kronecker's delta function: 1 if $i = k$ and 0 if $i \neq k$. With this choice, $< \mathbf{w}^k, \mathbf{x} >= x_k$ is the k-th component of the solution vector $\mathbf{x}$.

Such scalar products can also be obtained in the following way: consider the system of linear equations $\mathbf{C}^\top \mathbf{y} = \mathbf{w}$ with matrix $\mathbf{C}^\top$, which is the transpose of $\mathbf{C}$:

$$\sum_{j=1}^{n} c_{ji} y_j = w_i \tag{4.30}$$

The scalar product $< \mathbf{w}, \mathbf{x} >$ can then also be obtained as $< \mathbf{e}, \mathbf{y} >=< \mathbf{w}, \mathbf{x} >$:

$$< \mathbf{w}, \mathbf{x} >=< \mathbf{C}^\top \mathbf{y}, \mathbf{x} >=< \mathbf{y}, \mathbf{C}\mathbf{x} >=< \mathbf{y}, \mathbf{e} > .$$

The second equality follows directly from the definition of the scalar product of vectors and $c_{ij}^\top = c_{ji}$. In order to compute a component of the solution $\mathbf{x}$ of $\mathbf{C}\mathbf{x} = \mathbf{e}$, one can thus either solve $\mathbf{C}\mathbf{x} = \mathbf{e}$ directly, or solve the adjoint system $\mathbf{C}^\top \mathbf{y} = \mathbf{w}$ with suitable adjoint source $\mathbf{w}$.

4.5.3 Adjoint of the radiosity and power system: two kinds of importance

The adjoint system corresponding to (4.25) is:

$$Y_i = W_i + \sum_j Y_j \rho_j F_{ji}. \tag{4.31}$$

The order of the indices of the form factor is the same as in (4.28). Like (4.28), this system can thus be solved using a *shooting* solution method such as Southwell relaxation.

The adjoint system corresponding to (4.28) is:

$$I_i = V_i + \sum_j F_{ij} \rho_j I_j. \tag{4.32}$$

This system can be solved using the same *gathering* solution methods as for solving (4.25), such as Jacobi or Gauss-Seidel iterations.

Multiplying both sides of (4.32) by A_i yields (4.31) with $W_i = A_i V_i$ and $Y_i = A_i I_i$: W_i and Y_i are *power-like* quantities, proportional to the surface area A_i of patch

i, while V_i and I_i are *radiosity-like* quantities. These quantities are called *importance* or *potential* in literature (see e.g. [82, 84, 33, 156, 85]). The use of importance in radiosity was introduced by Smits et al. [152] and Pattanaik et al. [121, 122].

Importance expresses to what extent the self-emitted radiosity or power of each light source patch in the scene contributes to the total radiosity or power of a single, fixed, patch. The total radiosity B_i of a patch i for instance can be obtained as a weighted sum $B_i = \sum_{k=1}^{n} Y_k^i E_k$ of the self-emitted radiosity E_k of all light sources. The weights are the importances Y_k^i w.r.t. source, or direct, importance $W_k^i = \delta_{ik}$: only patch i is of direct importance.

4.5.4 Shooting and gathering importance

At first sight, it appears as if there are always four different ways to compute the radiosity B_i or power P_i of a patch i:

1. *gathering radiosity*: based on the scalar product $< B, W^i >$, with $W_k^i = \delta_{ik}$. B is computed by solving (4.25) using a gathering method;
2. *shooting (power-like) importance*: based on the scalar product $< E, Y^i >$. Y^i is computed by solving (4.31) with source term W^i defined above. A shooting solution method such as Southwell relaxation shall be used;
3. *shooting power*: based on the scalar product $< P, V^i >$, with $V_k^i = \delta_{ik}$. P is obtained by solving (4.28) using a shooting solution method;
4. *gathering (radiosity-like) importance*: based on the scalar product $< \Phi, I^i >$ solving I^i from (4.32) with source term V^i, defined above. Solving (4.32) requires a gathering solution method due to its similarity with (4.25).

In practice however, it turns out that the first and second option result in similar algorithms: shooting importance corresponds to gathering radiosity. The same is true for the third and fourth option above: shooting power corresponds to gathering importance.

4.5.5 Overview

The Monte Carlo method can be used in four different ways to solve linear systems such as (4.25), (4.28), 4.31) and (4.32):

- The coefficients of the system to be solved can be precomputed using the Monte Carlo method. A traditional, direct or iterative, solution method is then used in order to solve the approximate system of equations. In the case of radiosity, this corresponds to *form factor computation by Monte Carlo* (chapter 5).
- In *stochastic relaxation methods* (chapter 6), matrix-vector products $\mathbf{Cx}$ are estimated directly as a whole instead of estimating each term individually;
- *Random walk* methods (chapter 7) directly estimate the entire Neumann series expansion of the solution (if converging) by Monte Carlo;
- Various *other Monte Carlo methods* (chapter 8) estimate the solution by solving an equivalent problem by Monte Carlo.

The subsequent chapters 9 to 11, will deal with variance reduction techniques for the basic Monte Carlo estimators presented in chapter 5 to 8. Chapter 12 discusses low-discrepancy sampling in Monte Carlo radiosity. In chapter 13 and 14, the non-hierarchical Monte Carlo radiosity algorithms for constant approximations will be extended to higher order approximations and hierarchical refinement.

Computers have led to a novel revolution in mathematics. Whereas previously an investigation of a random process was regarded as being complete as soon as it was reduced to an analytic description, nowadays it is convenient in many cases to solve an analytic problem by reducing it to a corresponding random process and then simulating that process.

YU A. SHREIDER, "The Monte Carlo method – The Method of Statistical Trials", Pergamon Press, 1966

The only good Monte Carlos are dead Monte Carlos.

HALE F. TROTTER AND JOHN W. TUKEY, "Conditional Monte Carlo for normal samples", Proc. of the Symposium on Monte Carlo methods, Florida, March, 16 and 17, 1954.

5 Monte Carlo Form-Factor Computation

The form factors F_{ij}, which appear in the linear systems (4.25), (4.28), (4.31) and (4.32) describing the radiosity problem with constant approximations, are non-trivial four-dimensional integrals. The Monte Carlo method can be used in order to compute these form factors while solving the system using a deterministic solution method.

In this chapter, a brief survey is given of Monte Carlo form factor computation. The main goal of this chapter is to introduce the concept of *uniformly distributed lines*. Such lines can be used for form factor computation, but also play a central role in more direct Monte Carlo algorithms for the solution of the system of radiosity equations, to be discussed in subsequent chapters. Most of the techniques presented in this chapter have been proposed before by others. However, novel contributions in this chapter are:

- a new Monte Carlo form factor computation technique based on weighted sampling;
- an analysis of the variance of the form factor estimators, allowing to adaptively choose the number of samples for form factor computation in order to achieve a prescribed (computational) accuracy.

5.1 Uniform area sampling

The simplest way of computing the patch-to-patch form factor F_{ij} is by uniform random selection of pairs of points x on S_i and y on S_j in equation (4.26):

$$F_{ij} = \frac{1}{A_i} \int_{S_i} \int_{S_j} G(x,y)\, dA_y dA_x \tag{5.1}$$

with

$$G(x,y) = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} \text{vis}(x,y).$$

The corresponding Monte Carlo estimator $\hat{F}_{ij}^A$ has

- probability density function $p^A(x,y) = \frac{1}{A_i}\frac{1}{A_j}$;
- sample contributions $\hat{F}_{ij}^A(x,y) = A_j G(x,y)$.

It is easy the verify that

$$\begin{aligned} E[\hat{F}_{ij}^A] &= \int_{S_i}\int_{S_j} \hat{F}_{ij}^A(x,y) p^A(x,y) dA_y dA_x = \int_{S_i}\int_{S_j} A_j G(x,y) \frac{1}{A_i A_j} dA_y dA_x = F_{ij} \\ V[\hat{F}_{ij}^A] &= \int_{S_i}\int_{S_j} \left[A_j G(x,y)\right]^2 \frac{1}{A_i}\frac{1}{A_j} dA_y dA_x - F_{ij}^2. \end{aligned} \tag{5.2}$$

This technique is not satisfactory as it can yield very high, and even infinite, variance. The integrand of (5.2) has an r_{xy}^4 factor in the denominator. This factor causes a strong singularity for abutting patches. Unlike the weak singularity in (5.1), the singularity in (5.2) cannot be removed by directional integration. In such cases, increasing the number of samples N will not necessarily lead to an improved form factor estimate.

5.2 Uniform direction sampling

If equation (4.27) is used instead of (4.26), there is no singularity:

$$F_{ij} = \frac{1}{A_i} \int_{S_i} \int_{\Omega_j(x)} \frac{\cos\theta_x}{\pi} \, d\omega_{\Theta_x} dA_x. \tag{5.3}$$

The inner integral is an integral over the solid angle $\Omega_j(x)$ sub-tended by the visible part of patch j as seen from each point $x \in S_i$. $\Omega_j(x)$ can be determined by solving the visibility problem while projecting on the hemisphere above x. This would allow the inner integral to be computed analytically, using Lambert's formula [9], but it is an extremely expensive operation. It is therefore more appropriate to transform the inner integral domain to the unoccluded solid angle $\Omega_j^{\text{vis}}(x)$ sub-tended by j on the hemisphere above x (without taking visibility into account):

$$F_{ij} = \frac{1}{A_i} \int_{S_i} \int_{\Omega_j^{\text{vis}}(x)} \frac{\cos\theta_x}{\pi} \chi_j(h(x,\Theta_x)) d\omega_{\Theta_x}. \tag{5.4}$$

$h(x,\Theta_x)$ denotes the nearest point on the surfaces in the scene, seen from x in direction Θ_x. $\chi_j(y)$ is a predicate which takes value 1 if y is a point on patch j, and zero otherwise.

In order to uniformly sample a direction Θ_x from x to j, Arvo's algorithm for uniform sampling of spherical triangles [2] can be used. $\chi_j(h(x,\Theta_x))$ can be evaluated by tracing a ray from x into direction Θ_x and determining whether the first intersection with another surface in the scene occurs on patch j. The resulting estimator $\hat{F}_{ij}^D$ has

- pdf $p^D(x,\theta_x) = \frac{1}{A_i} \frac{1}{\Omega_j^{\text{vis}}(x)}$;
- sample contributions $\hat{F}_{ij}^D(x,\Theta_x) = \Omega_j^{\text{vis}}(x) \frac{\cos\theta_x}{\pi} \text{vis}_j(x,\Theta_x)$.

This estimator is unbiased. Its variance is given by

$$V[\hat{F}_{ij}^D] = \int_{A_i} \int_{\Omega_j^{\text{vis}}(x)} \left[\Omega_j^{\text{vis}}(x) \frac{\cos\theta_x}{\pi} \text{vis}_j(x,\Theta_x) \right]^2 \frac{1}{A_i} \frac{1}{\Omega_j^{\text{vis}}(x)} d\omega_{\Theta_x} dA_x - F_{ij}^2. \tag{5.5}$$

Unlike with uniform area sampling, the variance is always bounded:

$$V[\hat{F}_{ij}^D] \leq \left(\overline{\Omega}_{ij} \frac{\overline{c}_{ij}}{\pi} \right)^2$$

where

- $\overline{\Omega}_{ij}$ denotes the maximum unoccluded solid angle under which j is seen from a point x on i;
- $\overline{c}_{ij}$ denotes the maximum cosine w.r.t. the normal on i of a line connecting a point x on i and a point y on j.

In particular, $\overline{\Omega}_{ij} \leq 2\pi$ and $\overline{c}_{ij} \leq 1$, so that the variance is always smaller than 4.

5.3 Uniformly distributed lines

5.3.1 Cosine-distributed direction sampling

Estimator $\hat{F}_{ij}^D$ can be further improved by generating directions that are distributed according to $\cos\theta_x$ instead of being uniformly distributed. This yields a third patch-to-patch form factor estimator $\hat{F}_{ij}^C$ with

- pdf

$$p^C(x,\Theta_x) = \frac{1}{A_i}\frac{\cos\theta_x}{\int_{\Omega_j^{\text{vis}}(x)}\cos\theta_x d\omega_{\Theta_x}} = \frac{1}{A_i}\frac{\cos\theta_x}{\pi G_j^{\text{vis}}(x)}$$

- sample contributions $\hat{F}_{ij}^C(x,\Theta_x) = G_j^{\text{vis}}(x)\chi_j(h(x,\Theta_x))$.

$G_j^{\text{vis}}(x)$ denotes the unoccluded point-x-to-patch-j form factor.

The variance of this estimator is bounded by

$$\begin{aligned} V[\hat{F}_{ij}^C] &= \int_{S_i}\int_{\Omega_j^{\text{vis}}(x)} \left(G_j^{\text{vis}}(x)\chi_j(h(x,\Theta_x))\right)^2 \frac{1}{A_i}\frac{\cos\theta_x}{\pi G_j^{\text{vis}}(x)}\, d\omega_{\Theta_x} dA_x - F_{ij}^2 \\ &\leq \frac{1}{A_i}\max_{x\in S_i} G_j^{\text{vis}}(x)\int_{S_i}\int_{\Omega_j^{\text{vis}}(x)} \frac{\cos\theta_x}{\pi}\, d\omega_{\Theta_x} dA_x \\ &\leq \left(\max_{x\in S_i} G_j^{\text{vis}}(x)\right)^2. \end{aligned}$$

The maximum unoccluded point-to-patch form factor can be bounded in three ways:

1. based on directional integration:

$$G_j^{\text{vis}}(x) = \int_{\Omega_j^{\text{vis}}(x)} \frac{\cos\theta_x}{\pi}\, d\omega_{\Theta_x} \quad \leq \quad \overline{\Omega}_{ij}\frac{\overline{c}_{ij}}{\pi}; \tag{5.6}$$

2. based on surface area integration:

$$G_j^{\text{vis}}(x) = \int_{S_j} \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2}\, dA_y \quad \leq \quad A_j\frac{\overline{c}_{ij}\overline{c}_{ji}}{\pi \underline{r}_{ij}^2}; \tag{5.7}$$

   $\underline{r}_{ij}$ is a lower bound for the distance between points x on i and points y on j;

3. $G_j^{\text{vis}}(x) \leq 1$: point-to-patch form factors for constant approximations are always smaller than 1.

Conservative approximations for the quantities $\overline{\Omega}_{ij}$, $\overline{x}_{ij}$, $\overline{c}_{ji}$ and $\underline{r}_{ij}$ can be obtained efficiently and easily by considering bounding spheres for the patches. More advanced form factor bounding techniques [160] can be used as well in order to bound the variance. In any case, the minimum of the three bounds above can be used.

Unfortunately, direct use of estimator $\hat{F}_{ij}^C$ is not possible because there exist no efficient algorithms to generate cosine-distributed directions that are directed to a given patch j. In §5.4, a new form factor computation technique will be presented that yields almost the same low variance as $\hat{F}_{ij}^C$ using uniform area sampling.

5.3.2 Local uniformly distributed lines

It is however easy to generate cosine-distributed directions if the requirement that they must point towards a given patch is dropped. By generating cosine-distributed directions Θ_x in the whole hemisphere above points x on patch i, the following estimator $\hat{F}_{ij}^R$ is obtained:

- pdf

$$p^R(x, \Theta_x) = \frac{1}{A_i} \frac{\cos\theta_x}{\int_{\Omega(x)} \cos\theta_x d\omega_{\Theta_x}} = \frac{1}{A_i} \frac{\cos\theta_x}{\pi}$$

- primary estimates $\hat{F}_{ij}^R = \chi_j(h(x, \Theta_x))$.

$\Omega(x)$ denotes the full hemisphere above x.

This estimator is a so called *hit-or-miss* estimator: if a ray with uniformly chosen origin on i and cosine distributed direction has its nearest intersection point with the other surfaces in the scene on patch j, the estimate is 1. In the other case it is 0. The estimator estimates the probability that such a line hits patch j. We will call any line, constructed so that its intersection point with the patch i is uniformly distributed on the surface area of i, and so that its direction is cosine distributed w.r.t. the normal at the intersection point, a *uniformly distributed line*. If, as is the case here, it is constructed by explicitly sampling a uniform point on the surface of i, we will call the line a *local uniformly distributed line* w.r.t. i. In §5.3.3, we will discuss techniques to obtain uniformly distributed lines without explicit sampling an origin on a patch.

Because the estimator $\hat{F}_{ij}^R$ is unbiased — its expectation is equal to the form factor F_{ij} — we obtain the following theorem, which is of central importance in Monte Carlo radiosity:

Theorem 5.1 *The patch-to-patch form factor F_{ij} between two patches i and j in a discretised scene corresponds to the probability p_{ij} that a uniformly distributed line w.r.t. i has its nearest intersection point on patch j.*

Proof: A direct proof can be obtained by calculating the probability p_{ij} above:

$$p_{ij} = \int_{A_i} \int_{\Omega(x)} \chi_j(h(x, \Theta_x)) \frac{1}{A_i} \frac{\cos\theta_x}{\pi} d\omega_{\Theta_x} dA_x = \frac{1}{A_i} \int_{A_i} \int_{\Omega_j(x)} \frac{\cos\theta_x}{\pi} d\omega_{\Theta_x} dA_x = F_{ij}$$

The second equality holds because $\chi_j(h(x,\Theta_x)) = 0$ if Θ_x is not a direction pointing to the visible part of j as seen from x. □

Because $\hat{F}_{ij}^R$ is a hit-or-miss estimator with expectation F_{ij}, its variance is given by

$$V[\hat{F}_{ij}^R] = F_{ij}(1 - F_{ij}). \tag{5.8}$$

The efficiency of this estimator can be very poor however for small form factors F_{ij} since on the average only 1 line in $1/F_{ij}$ will hit j from i. On the other hand, the rejected lines can be used for estimating the other form factors F_{ik} while computing F_{ij}. The resulting algorithm for computing all form factors F_{ij} for fixed i is shown in algorithm 5.3.2 (see also figure 5.1).

Algorithm 7: Computes all form factors F_{ij} between a fixed patch i and another patch j using N_i local cosine-distributed lines.

---

1. Initialise $F_{ij} \leftarrow 0$ for all $j = 1, \ldots, n$;
2. For $k = 1, \ldots, N$, do
   (a) Choose a uniform random point x on S_i;
   (b) Choose a cosine distributed direction Θ_x w.r.t. the surface normal at x;
   (c) Determine the nearest intersection point $h(x,\Theta_x)$ of a ray with origin at x and direction Θ_x with a surface of the scene. Set $j \leftarrow$ the index of the patch containing this first intersection point $h(x,\Theta_x)$;
   (d) Set $F_{ij} \leftarrow F_{ij} + 1/N$.

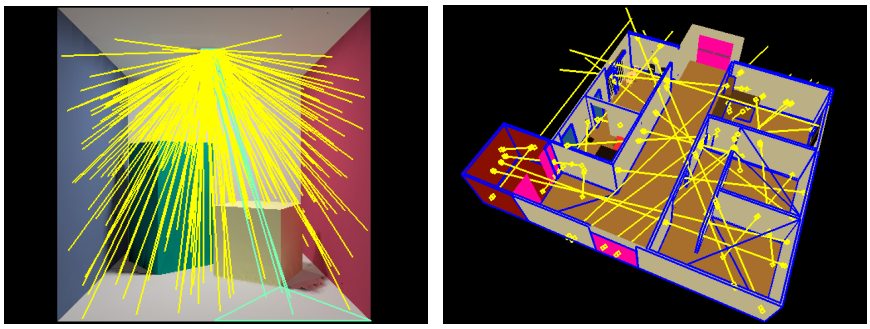


Figure 5.1: Local uniformly distributed lines (left) are constructed by explicitly sampling the origin on a patch in the scene. Global lines (right) are constructed without reference to any of the patches in the scene. Their intersection points with the surfaces in the scene are however also uniformly distributed. The angle between these lines and the normal on each intersected surface is cosine distributed, just like with local lines. The intersection points define spans on each line. Each global line span can be used bidirectionally for form factor computation between the connected patches.

This algorithm has been proposed as a ray-tracing based alternative for the hemi-cube method at the end of the 1980'ies [149, 142]. Note that it computes patch-to-patch form factors. This is unlike the hemi-cube method, which computes point-to-patch form factors.

5.3.3 Global uniformly distributed lines

There also exist techniques in order to construct uniformly distributed lines without explicitly sampling the origin on a patch in the scene. Uniformly distributed lines constructed without explicit sampling the origin on a patch, are called *global uniformly distributed lines*. The construction and properties of such lines have been studied extensively in *integral geometry* [129, 130, 131].

Construction of global uniformly distributed lines In the context of radiosity, the following methods have been used:

- **Two-points-on-a-sphere** method: two uniformly distributed points p and q are sampled on the surface of a sphere bounding the scene. The line connecting p and q can be shown to be a uniformly distributed line within the scene [130]. A field of N uniformly distributed lines is obtained by sampling N pairs of points p_k and q_k, $k = 1, \ldots, N$, on the surface of the bounding sphere;
- **Plane-intercept** method [123, 110, 131, 167]: a uniformly distributed point Θ is sampled on the unit sphere. As such, Θ is a uniform global direction. Consider the plane P through the origin and perpendicular to Θ: the plane equation is $\Theta \cdot x = 0$. Now consider the orthogonal projection of the scene onto this plane. Each uniformly sampled point x in the projection defines, together with Θ, a uniformly distributed line through the scene.

The resulting lines cross several surfaces in the scene. The intersection points with the intersected surfaces define *spans* of mutually visible patches along the line. (see figure 5.1). Each such a line span corresponds to *two* local cosine-distributed lines — one in both directions along the line — because the global uniformly distributed lines are uniformly distributed w.r.t. every patch in the scene. This is unlike local lines, which are uniformly distributed only w.r.t. the patch on which the origin was sampled.

It can be shown that the probability that a global uniform line, generated with the aforementioned algorithms, intersects a given patch i, is proportional to the surface area A_i [131]. If N global lines are generated, the number N_i of lines crossing a patch i will be

$$N_i \approx N \frac{A_i}{A_T}. \tag{5.9}$$

Form factor computation using global lines Theorem 5.1 implies that the form factor F_{ij} can be estimated as a fraction $F_{ij} \approx N_{ij}/N_i$ of a number N_{ij} of uniformly distributed lines w.r.t. i that have their next intersection on j and the total number N_i of uniformly distributed lines w.r.t. i. The basic idea of global line form factor computation is to generate a field of N uniformly distributed lines through the scene and to count the number of lines N_i and N_{ij} intersecting each pair of patches i and j. The ratios N_{ij}/N_i then yield estimates for the full form factor matrix [130]. This

corresponds to applying algorithm 5.3.2 with N_i *local* lines for each patch i separately, where $N_i \approx NA_i/A_T$ would be chosen proportional to the surface area A_i of i. The variance of the resulting estimates is,[1]

$$\frac{1}{N_i} V[\hat{F^R}_{ij}] = \frac{1}{N_i} F_{ij}(1 - F_{ij}). \tag{5.10}$$

In §11.1.2, it will be shown that also local lines, with origin chosen on a patch i with probability proportional to the surface area A_i of i, can be used bidirectionally, in the same way as global line spans.

5.3.4 Global versus local lines

The main advantage of global lines over local lines is that geometric scene coherence can be exploited in order to generate global lines more efficiently:

- In a naive ray-tracing implementation for instance, the two-points-on-a-sphere method would yield all k intersections of a line with the surfaces in the scene at the same cost of determining only the nearest intersection of a local line. In a properly constructed scene, the number of line spans on a global line is half the number k of intersection points. Since each span is used bidirectionally, this means that the global line yields the equivalent of k local lines at the same cost. Even when using ray-tracing acceleration techniques that allow to stop tracing a local line before all its potential intersections with the scene are determined, there still is a speed-up.

- The plane-intercept method allows bundles of parallel global lines to be generated using a Z-buffer like algorithm: first, a uniform random direction Θ is chosen. Next, a rectangular window is chosen in the plane, through the origin and perpendicular to Θ, that contains the orthogonal projection of the whole scene on the plane. A certain resolution for rendering is chosen in the window. Each pixel will correspond to a parallel global line. Finally, a suitable orthogonal projection matrix is set up and the scene projected onto the plane using a Z-buffer-like algorithm. Instead of keeping only the nearest Z-value in each pixel however, a full sorted list of all patches visible through each pixel is kept [110]. Alternatively, it is possible to use sweep-plane algorithms to solve the visibility problem analytically [123]. This corresponds to a bundle of parallel lines with infinite density [166].

The main limitation of global lines w.r.t. local lines is that their construction cannot easily be adapted in order to increase or decrease the line density on given patches. Combining (5.10) and (5.9), we see that the form factor variance is inverse proportional to the area of the source patch i:

$$\frac{1}{N_i} V[\hat{F}^R_{ij}] \approx \frac{1}{N} \frac{A_T}{A_i} F_{ij}(1 - F_{ij})$$

The variance will be high on small patches.

[1] These estimators also have a small, exponentially decreasing bias, which is negligible in practice [131]

5.4 Weighted area sampling

5.4.1 Motivation

Direction sampling for computing individual patch-to-patch form factors F_{ij} is quite expensive: the generation of uniformly distributed directions (§5.2) towards a given patch j, using the algorithm for uniform sampling of spherical triangles [2], is quite more involved than uniform area sampling. Uniform area sampling is cheap, but it has been shown in §5.1 that the variance of the resulting estimator $\hat{F}_{ij}^{A}$ can be very high, and even infinite for abutting patches. That means that there is no guarantee at all that by taking more samples, an improved estimate for the form factor between abutting patches is obtained.

5.4.2 Outline of the algorithm

We thus have a situation for which *weighted importance sampling* (§4.3.2) was designed. The target pdf $p(x, y)$ is such that for a given $x \in S_i$, the points $y \in S_j$ would correspond to the intersection points of cosine-distributed lines with origin at x:

$$p(x,y) = \frac{1}{A_i} p(y|x) \quad \text{with:} \quad p(y|x) = \frac{G^{\text{vis}}(x,y)}{\int_{S_j} G^{\text{vis}}(x,y) dA_y}$$

where $G^{\text{vis}}(x, y)$ is the unoccluded radiosity kernel function:

$$G^{\text{vis}}(x,y) = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} \quad ; \quad G(x,y) = G^{\text{vis}}(x,y)\text{vis}(x,y). \tag{5.11}$$

The integral in the denominator of $p(y|x)$ above is nothing else than the unoccluded point-x-to-patch-j form factor $G_j^{\text{vis}}(x)$, which can be computed analytically using Lambert's formula [9].

Application of formula (4.14) with source pdf $q(x, y) = \frac{1}{A_i}\frac{1}{A_j}$ (uniform area sampling) yields

$$\hat{F}_{ij}^{W1}(x_k, y_k; k = 1, \ldots, N) = \frac{\sum_{k=1}^{N} G^{\text{vis}}(x_k, y_k)\text{vis}(x_k, y_k)}{\sum_{k=1}^{N} G^{\text{vis}}(x_k, y_k)/G_j^{\text{vis}}(x_k)} \approx F_{ij}. \tag{5.12}$$

For nearby x_y and y_k, the large factor $G^{\text{vis}}(x_k, y_k)$ in the numerator will be cancelled by a large factor in the denominator.

The resulting estimator still is quite expensive however, because an analytical point-to-patch form factor needs to be computed for every sample. A cheaper estimator is obtained by using weighted importance sampling only in order to estimate the inner integral of (5.1). We propose to take N_i points x_k uniformly on S_i. For each x_k, N_j points $y_{kl}, l = 1, \ldots, N_j$ are sampled uniformly on S_j. The resulting estimates are:

$$\hat{F}_{ij}^{W2}(x_k, y_{kl}) = \frac{1}{N_i} \sum_{k=1}^{N_i} G_j^{\text{vis}}(x_k) \frac{\sum_{l=1}^{N_j} G^{\text{vis}}(x_k, y_{kl})\text{vis}(x_k, y_{kl})}{\sum_{l=1}^{N_j} G^{\text{vis}}(x_k, y_{kl})} \approx F_{ij}. \tag{5.13}$$

The number of analytical point-to-patch form factor evaluations is highly reduced. The corresponding algorithm is shown in algorithm 8.

Algorithm 8: Computes the patch-to-patch form factor F_{ij} between patches i and j using weighted importance sampling on the inner integral. Corresponds to estimator $\hat{F}_{ij}^{W2}$.

1. Initialise $F_{ij} \leftarrow 0$;
2. For $k = 1, \ldots, N_i$, do
   (a) Choose a uniform random point x on S_i;
   (b) Set $dF \leftarrow 0$, $dG \leftarrow 0$;
   (c) For $l = 1, \ldots, N_j$, do
      i. Sample a uniform random point y on S_j;
      ii. Trace a ray from x to y in order to evaluate $\text{vis}(x, y)$, compute $G^{\text{vis}}(x, y)$;
      iii. $dF \leftarrow dF + G^{\text{vis}}(x, y)\text{vis}(x, y)$;
      iv. $dG \leftarrow dG + G^{\text{vis}}(x, y)$.
   (d) Compute the unoccluded point-x-to-patch-j form factor $G_j^{\text{vis}}(x)$ using Lambert's formula [9];
   (e) $F_{ij} \leftarrow F_{ij} + \frac{1}{N_i}\frac{dF}{dG}G_j^{\text{vis}}(x)$.

5.4.3 Empirical results and discussion

Figure 5.2 shows some results obtained with algorithm 8 for a number of test configurations: two abutting or parallel patches with full visibility, and a more complex configuration with partial visibility. In all cases, weighted area sampling performs similarly to directional sampling (§5.2) and significantly better than uniform area sampling. How can we explain these good results, although uniform area sampling is used?

If both patches i and j are distant and small w.r.t. each other, the factors $G^{\text{vis}}(x, y)$ will be approximately constant and the estimates (5.13) will correspond to the average of N_i products of an analytically computed unoccluded point-to-patch form factor multiplied with the fraction of rays that do not hit occluders between i and j. Part of the problem has thus been solved analytically rather than by sampling.

If, on the other hand, i and j are abutting surfaces, $G^{\text{vis}}(x, y)$ will fluctuate enormously, but large factors in the numerator are compensated by equally large factors in the denominator. Sufficiently near a shared edge, visibility is almost always either fully occluded or fully unoccluded in practice. In the former case, the ratio of sums in (5.13) is 0. In the latter case, the ratio equals 1.

A more strict argument indicating that weighted sampling effectively removes the singularity is the following: because $\text{vis}(x, y) \leq 1$, the ratio of sums is bounded by 1, for every x_k. The variance of the estimator with N_i samples on i will be bounded by

$$V[\hat{F}_{ij}^{W2}] \leq \frac{1}{N_i} \int_{S_i} \left[G_j^{\text{vis}}(x)\right]^2 dA_x.$$

For a sufficiently high number of samples, the variance will be very close to the variance of the directional estimators presented above. The same variance bounds as presented in §5.3.1 can be used in order to determine the number of samples needed to compute the form factor with prescribed accuracy.

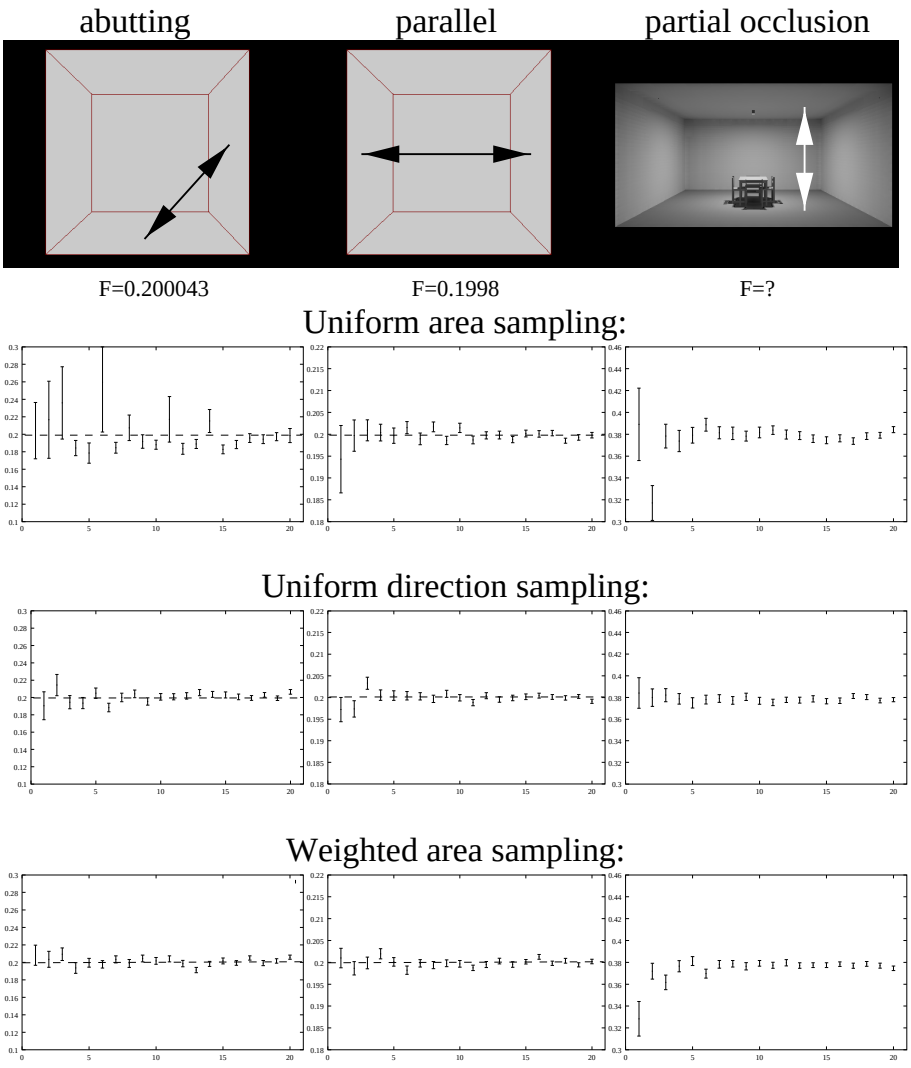


Figure 5.2: Results obtained with weighted area sampling for computing patch-to-patch form factors (algorithm 8). The graphs show the average and the standard deviation of the results of 100 runs with $k \times k$ samples each, as a function of k. k sample points were chosen on the receiver patch. For each point on the received patch k sample points on the source patch have been chosen randomly. In all cases, weighted area sampling performs similarly to uniform direction sampling and significantly better than uniform area sampling. In the partial occlusion example, a slight bias is visible when the number of samples is low.

5.5 Choosing an appropriate number of samples

In the previous section, closed form expressions for the variance of the form factor estimators allowed to derive practical upper bounds for the variance. Conservative approximations for the quantities $\overline{\Omega}_{ij}$, $\overline{c}_{ij}$, $\underline{r}_{ij}$ that appear in these bounds can be computed efficiently and easily using for instance bounding spheres for the patches (figure 5.3). More advanced techniques that have been designed to bound the form factor itself [160] can be used as well.

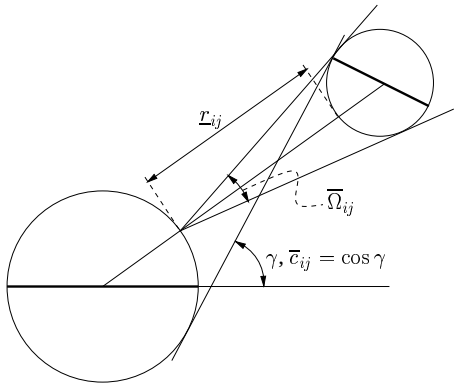


Figure 5.3: The quantities $\overline{\Omega}_{ij}$, $\overline{c}_{ij}$ and $\underline{r}_{ij}$ that appear in the form factor variance bounds can be computed efficiently using for instance bounding spheres for the patches (shown here in 2D).

The variance upper bounds can be used instead of the exact (unknown) variance itself in order to determine in advance how many samples will be needed in order to compute a form factor to given accuracy Δ and confidence. For a 99.7% confidence level for instance, the number of samples N should be chosen high enough so that (see §4.2.6)

$$3\sqrt{\frac{V[\hat{F}_{ij}]}{N}} \approx \Delta \tag{5.14}$$

The ability to choose the number of samples appropriately in advance so easily, is a major advantage over deterministic form factor computation methods: for other form factor computation methods, the error is much harder to control.

The final goal however is to compute the radiosity in a scene to prescribed accuracy, say ε. We hope to do so by controlling the computational error on each form factor. In principle, it is possible to determine in advance what error Δ can be tolerated on each form factor in order to compute the radiosity solution to prescribed accuracy ε. Arvo [2, §6.3] sketches how a relation between Δ and ε can be established. Many approximations have to be made however in order to obtain a manageable expression. Without going into the details, the resulting estimates for the required number of samples per form factor are far too conservative, leading to much more precise (and costly) form factor computation than necessary.

Most often, the accuracy of a radiosity solution is controlled directly through per-link error thresholds Δ, rather than by a global error threshold ε. In that case, it is

possible to determine an appropriate number of samples on the fly during the radiosity system solution as follows. Each form factor F_{ij} always appears in products like $\rho_i F_{ij} B_j$, where B_j is a currently available intermediate radiosity value during system solution. Equation (5.14) can be used in order to determine the required number of samples with $\Delta / \rho_i B_j$ instead of Δ in the right hand side.

Sometimes, when the form factor F_{ij} is needed again during a later iteration, it will appear that a higher accuracy is needed than before, for instance because B_j has increased. In such cases, it is not required to re-compute the form factor from scratch: it suffices to take a number of additional samples. The number of additional samples can be determined easily by keeping a count how many samples have been used previously. Formula (5.14) indicates how many samples are needed in total. The number of additional samples required is the difference between the total number of samples required and the number of samples that was already drawn before.

5.6 Conclusion

This chapter presented an overview of Monte Carlo techniques to compute form factors. Monte Carlo form factor integration is simple, and allows form factors to be computed to any prescribed accuracy Δ with given confidence: Monte Carlo integration leads to reliable form factor computation.

A new algorithm to compute patch-to-patch form factors, based on weighted importance sampling, has been developed. Although surface-area sampling is used, it does not suffer from infinite variance due to the integrand singularity for abutting patches. Its variance is almost as small as when cosine-distributed directional sampling is used. The price to be paid is a slight bias, which decreases $\mathcal{O}(\sqrt{1/N})$ more rapidly than the standard deviation for N samples. This algorithm can be incorporated easily in existing hierarchical refinement radiosity systems. It will provide more reliable form factor computation in the context of the algorithms of chapter 3. A generalisation to higher order approximations will be presented further in §13.2.

The main limitation of explicit form factor computation by Monte Carlo is that it is hard to determine in advance what error Δ can be tolerated on each form factor in order to obtain a prescribed accuracy ε on the total radiosity. Current heuristics to relate Δ with ε require many approximations and lead to unrealistic, conservative choices for the number of samples required for each form factor.

In the next chapters, it will be shown that more direct radiosity system solution by Monte Carlo allows computational error control on the total radiosity much more easily. In the techniques that will be presented, explicit form factor integration and storage can even be completely avoided.

6 Stochastic Relaxation Radiosity

In this and the next chapters, the Monte Carlo method will be used for more direct solution of the system of radiosity equations. In this chapter, stochastic relaxation methods are discussed. Stochastic relaxation methods have received no attention yet in general Monte Carlo literature. They are however applied with good success in the context of the radiosity problem.

Stochastic relaxation algorithms for radiosity have been proposed first by P. Shirley and L. and A. Neumann et al. [142, 111]. This chapter will present a systematic overview in which stochastic relaxation algorithms are situated and theoretically compared. Some new stochastic relaxation algorithms are proposed and their computational cost is analysed[1].

In relaxation methods, the coefficients c_{ij} of a linear system $\mathbf{Cx} = \mathbf{e}$ appear only in sums of the form $\sum_j c_{ij}x_j$. The basic idea of stochastic relaxation methods is to estimate these sums as a whole by Monte Carlo rather than to compute each coefficient c_{ij} individually. In the context of radiosity, the coefficients c_{ij} contain the form factors F_{ij}. Direct estimation of the sums $\sum_j c_{ij}x_j$ for radiosity yields the following benefits:

1. The form factors F_{ij} for fixed i form a probability distribution that can be sampled efficiently using uniformly distributed lines, introduced in chapter 5. By doing so, it turns out that an *accurate numerical value of the form factors is never required*: the form factors do not need to be computed or stored;

2. The Monte Carlo method allows sums to be estimated by evaluating only a fraction of the terms. The effect of missing terms is taken into account implicitly due to the fact that they *could* have been sampled. In the case of radiosity, the *time complexity is reduced from* $\mathcal{O}(n^2)$ *to* $\mathcal{O}(n \log n)$, where n is the number of patches in the scene;

3. Using *progressive variance reduction* techniques, the need for complicated a-priori estimates of the required number of samples is largely avoided. Good default values for the number of samples in each step of the computations are available.

First, a general outline of stochastic relaxation methods and their application to radiosity is presented in §6.1. Several examples are worked out in detail in the following sections: Gauss-Seidel (§6.2), Southwell (§6.3), Jacobi (§6.4) and others (§6.5). Progressive variance reduction techniques in stochastic relaxation methods are discussed in §6.6.

[1] The investigation and development of stochastic relaxation algorithms in the context of this dissertation, was largely performed in collaboration with L. and A. Neumann.

6.1 General outline

6.1.1 Relaxation methods

The basic idea of relaxation methods, such as Jacobi and Gauss-Seidel iterations, Southwell relaxation (related to progressive refinement radiosity) and the conjugate gradients method [57, 59], is to construct a sequence of approximate solutions $\mathbf{x}^{(k)}, k = 0, 1, \ldots$ that converges to the true solution $\mathbf{x}$ of a linear system $\mathbf{C}\mathbf{x} = \mathbf{e}$. A next approximation $\mathbf{x}^{(k+1)}$ is constructed by adding a correction $\Delta\mathbf{x}^{(k)}$ to a current approximation $\mathbf{x}^{(k)}$: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}$.

As a measure for the error $\mathbf{x} - \mathbf{x}^{(k)}$ of an approximation $\mathbf{x}^{(k)}$, the residu

$$\mathbf{r}^{(k)} = \mathbf{e} - \mathbf{C}\mathbf{x}^{(k)} \tag{6.1}$$

is used. Usually, $\mathbf{x}^{(0)} = 0$ is taken as a first approximate solution. In that case,

$$\mathbf{r}^{(0)} = \mathbf{e} - \mathbf{C}\mathbf{x}^{(0)} = \mathbf{e}.$$

Relaxation methods differ in the choice of the correction vectors $\Delta\mathbf{x}^{(k)}$. We will discuss several possibilities below. In Jacobi- and Gauss-Seidel iterations, a special choice of the correction vectors $\Delta\mathbf{x}^{(k)}$ allows to avoid explicit computation and storage of the residu vectors. In Southwell-relaxation and in the conjugate gradients method however, the residu is used in order to select a correction vector $\Delta\mathbf{x}^{(k)}$ that can lead to convergence more rapidly. After determination of the correction vector $\Delta\mathbf{x}^{(k)}$, the new residu vector $\mathbf{r}^{(k+1)}$ corresponding to $\mathbf{x}^{(k+1)}$ is then obtained as follows:

$$\begin{aligned} \mathbf{r}^{(k+1)} &= e - \mathbf{C}\mathbf{x}^{(k+1)} = \mathbf{e} - \mathbf{C}(\mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}) = (\mathbf{e} - \mathbf{C}\mathbf{x}^{(k)}) - \mathbf{C}\cdot\Delta\mathbf{x}^{(k)} \\ &= \mathbf{r}^{(k)} - \mathbf{C}\cdot\Delta\mathbf{x}^{(k)}. \end{aligned} \tag{6.2}$$

The basic relaxation algorithm is shown in algorithm 9

Algorithm 9: Basic relaxation algorithm.

1. Choose initial guess $\mathbf{x}^{(0)}$;
2. $\mathbf{r}^{(0)} \leftarrow \mathbf{e} - \mathbf{C}\mathbf{x}^{(0)}$;
3. For $k = 0, 1, \ldots$ until convergence, do
   (a) Compute correction $\Delta\mathbf{x}^{(k)}$ based on $\mathbf{x}^{(k)}$ and/or $\mathbf{r}^{(k)}$ or other information;
   (b) $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}$;
   (c) $\mathbf{r}^{(k+1)} \leftarrow \mathbf{r}^{(k)} - \mathbf{C}\cdot\Delta\mathbf{x}^{(k)}$.

Most often, $\Delta\mathbf{x}^{(k)}$ is chosen to have only a single nonzero component so that the matrix-vector product $\mathbf{C}\cdot\Delta\mathbf{x}^{(k)}$ requires only $\mathcal{O}(n)$ multiplications and additions. In general, the matrix-vector product requires $\mathcal{O}(n^2)$ work. It will be shown below that the Monte Carlo method allows to estimate the full matrix-vector product with only $\mathcal{O}(n \log n)$ work in the context of the radiosity problem.

In radiosity, the residu $\mathbf{r}^{(k)}$ has the meaning of *unshot* radiosity, power or importance, depending on which of the four systems of equations (4.25), (4.28), (4.31),

(4.32) is being solved. The approximate solution $\mathbf{x}^{(k)}$ corresponds to the *shot* radiosity, power or importance. If the residu is explicitly computed, it is customary in radiosity to display the sum $\mathbf{x}^{(k)} + \mathbf{r}^{(k)}$ after each iteration step [59]. In the progressive refinement radiosity method for instance, the *total* and unshot power are kept rather than the shot and unshot power.

6.1.2 Monte Carlo estimation of the matrix-vector products for radiosity

Consider the classical radiosity system (4.25). The coefficients of (4.25) are $c_{ij} = \delta_{ij} - \rho_i F_{ij}$. The matrix-vector product $\mathbf{C} \cdot \Delta\mathbf{x}$ in (6.2) takes the form

$$(\mathbf{C} \cdot \Delta\mathbf{x})_i = \Delta x_i - \sum_{j \neq i} \rho_i F_{ij} \Delta x_j. \tag{6.3}$$

A sum like above can be estimated using Monte Carlo as follows: N times, a term with index $j_k, k = 1, \ldots, N$ is selected randomly with probability p_{j_k}. As an estimate,

$$\Delta x_i - \frac{1}{N} \sum_{k=1}^{N} \frac{\rho_i F_{ij_k} \Delta x_j}{p_{j_k}} \approx (\mathbf{C} \cdot \Delta\mathbf{x})_i$$

is used. When the probabilities p_j are chosen well, so that the *variance* of the estimator is low, good estimates for the sum can be obtained by sampling only a small fraction of the terms. In other words: the sums can be estimated accurately *without computing all form factors.* This is a very fundamental characteristic of the Monte Carlo method that potentially makes it more efficient than direct summation: *the effect of the terms that are not sampled is taken into account by the fact that these missing terms* **could** *have been sampled.*

Since $F_{ij} \geq 0, \forall i, j$ and $\sum_{j=1}^{n} F_{ij} \leq 1, \forall i$, the form factors F_{ij} for fixed i can be used as a probability distribution p_j for estimating the sum above. In this case, the estimates become

$$\Delta x_i + \frac{1}{N} \sum_{k=1}^{N} \frac{\rho_i F_{ij_k} \Delta x_j}{F_{ij_k}} = \Delta x_i + \frac{\rho_i}{N} \sum_{k=1}^{N} \Delta x_j \approx (\mathbf{C} \cdot \Delta\mathbf{x})_i.$$

Because the form factor F_{ij_k} cancels in numerator and denominator, its numerical value does not need to be known.

The amount of computation work is clearly proportional to the number of samples N. How many samples are required in order to estimate the sum to given precision depends on the variance $\rho_i \sum_j |\Delta x_j| \, x_i' - [x_i']^2$, which is only loosely related to the number of patches n. In the remainder of this chapter, several concrete examples will be worked out. It will be shown that the *time complexity of stochastic relaxation methods can be $\mathcal{O}(n \log n)$* rather than $\mathcal{O}(n^2)$ for radiosity.

6.1.3 Sampling the form factor probability distribution

The question now is how a term j can be selected with probability F_{ij}. The answer is given by the same theorem 5.1 that was used to estimate the numerical value of the

form factor F_{ij} in the previous chapter: since the probability that a uniformly distributed line through i will have its nearest intersection point in the scene on j equals the form factor F_{ij}, it suffices to *sample such a uniformly distributed line through i and determine the patch j which is hit next*. Patch j will be chosen with probability F_{ij}. Using ray-tracing to do so, accelerated with a hierarchy of bounding volumes, the cost of taking a sample is $\mathcal{O}(\log n)$.

All three construction techniques for uniformly distributed lines can be used: local cosine-distributed lines, global lines using the two-points-on-a-sphere construction, and global lines using the plane-intercept construction. With local lines, the patch i containing the origin is chosen explicitly. With global lines, the patch containing the origin cannot be chosen explicitly: the origin of a global line span will be contained in i with probability A_i/A_T. Often however, a sum like (6.3) needs to be computed for every patch i. In this case, each global line will yield a contribution to every patch it intersects in the scene. It was discussed before that global line spans can be generated at a lower cost than local lines, but that small patches are hit rarely. In literature, only local lines or global line bundles using the plane-intercept technique with a Z-buffer like algorithm [110, 167] have been used in the context of stochastic relaxation methods. The use of global lines constructed with the two-points-on-a-sphere technique has not yet been described in literature, but is equally well possible.

6.2 Stochastic Gauss-Seidel iterative method

6.2.1 The Gauss-Seidel iterative method

The Gauss-Seidel iterative method is usually formulated as an iterative method in which each component of the solution is updated in turn as follows:

$$x_s \leftarrow \left(e_s - \sum_{j \neq s} c_{sj} x_j \right) / c_{ss} \tag{6.4}$$

According to the definition (6.1), the s-th component of the residu vector $\mathbf{r}$ equals

$$r_s = (\mathbf{e} - \mathbf{C}\mathbf{x})_s = e_s - \sum_j c_{sj} x_j$$

so that the Gauss-Seidel update step (6.4) can also be written as:

$$x_s \leftarrow (r_s + c_{ss} x_s) / c_{ss} = x_s + r_s / c_{ss}.$$

The Gauss-Seidel iterative method thus is a relaxation method in which the correction vector $\Delta\mathbf{x}$ is chosen so that

$$\Delta x_i = \delta_{is} r_s / c_{ss}.$$

It is interesting to see what happens with the residu components r_i after the update:

$$r_i \leftarrow r_i - (C \cdot \Delta x)_i = r_i - c_{is} r_s / c_{ss} \tag{6.5}$$

The special choice of correction vector $\Delta\mathbf{x}$ makes the residu component r_s zero. It is true that the other residu components change value as well, and will possibly become larger, but as a whole — measured using some vector norm $\|\mathbf{r}\|$ — an improvement is made if the system of equations fulfils certain requirements [59].

6.2.2 Regular stochastic Gauss-Seidel iterative method

Usually, the residu vector is not used explicitly in Gauss-Seidel iterations. Rather, (6.4) is evaluated directly. With this choice, the Gauss-Seidel method is suited in order to solve the classical radiosity system (4.25) or possibly the adjoint power system (4.32). Indeed, the coefficients of (4.25) are $c_{sj} = -\rho_s F_{sj}$ if $s \neq i$ and $c_{ss} = 1$, yielding

$$B_s \leftarrow B'_s = E_s + \rho_s \sum_{j \neq i} F_{sj} B_j. \tag{6.6}$$

The indices of the form factor appear in the right order so that this expression can be estimated using Monte Carlo as explained in the previous section §6.1.3. This leads to an estimator $\hat{b}_s$ with

- probability distribution $p(j) = F_{sj}$. This pdf can be sampled with uniform lines w.r.t. s as explained above;
- sample contributions $\hat{b}_s(j) = \rho_s B_j$.

The expectation is

$$E[\hat{b}_s] = \sum_j \hat{B}_s(j) p(j) = \sum_j \rho_s B_j F_{sj} = B'_s - E_s.$$

The resulting algorithm, with local line sampling, is shown in algorithm 10.

Algorithm 10: Regular stochastic Gauss-Seidel iterative method.

---

1. Initialise $B_i \leftarrow E_i$ for all patches i;
2. Cycle through the patches until convergence, for each selected patch s, do
   (a) Choose number of samples N (§6.2.3);
   (b) $B_s \leftarrow E_s$;
   (c) Do N times,
      i. Sample uniform random point x on patch s;
      ii. Sample cosine-distributed random direction Θ w.r.t. surface normal at x;
      iii. Determine patch i containing nearest hit of the ray with origin at x and direction Θ with the surfaces in the scene;
      iv. $B_s \leftarrow B_s + \frac{1}{N} \rho_s B_i$;

6.2.3 Time complexity

How many samples N need to be taken in each step in order to compute B'_s (or $B'_s - E_s$) in (6.6) to given accuracy ε with 99.7% certainty? In other words, how shall N be chosen so that

$$|\hat{b}_s - (B'_s - E_s)| \leq \varepsilon \ ?$$

The answer follows from the central limit theorem of probability (§4.2.6):

$$3\sqrt{\frac{V[\hat{b}_s]}{N}} \approx \varepsilon.$$

The variance is

$$V[\hat{b}_s] = \sum_j [\hat{b}_s(j)]^2 p(j) = \rho_s^2 \sum_j B_j^2 F_{sj}.$$

So, N shall be chosen

$$N \approx \frac{9}{\varepsilon^2} \cdot \rho_s^2 \sum_j B_j^2 F_{sj}.$$

The number of samples N depends on the radiosity distribution in the scene as seen from s and is asymptotically independent of n, the number of patches. Since tracing each ray takes $\mathcal{O}(\log n)$ time, we conclude that each step takes $\mathcal{O}(\log n)$ computation time. The number of steps before reaching an acceptable solution is $\mathcal{O}(n)$. The total time complexity of stochastic Gauss-Seidel iterations thus is $\mathcal{O}(n \log n)$.

At first sight, one would expect a dramatic improvement compared to deterministic Gauss-Seidel, which takes $\mathcal{O}(n^2)$ time. However, stochastic Gauss-Seidel, as described here, is not recommended in practice because:

- The variance will in general be very high unless the radiosity in the scene is nearly constant. In general, a high number of samples N will be required in every step to reduce the error to an acceptable level;
- In each iteration step, only a single component of the solution is updated.

By evaluating (6.5) instead of (6.4), a generally more efficient *incremental* stochastic Gauss-Seidel algorithm results. We call it an incremental stochastic relaxation method because in each step, the radiosities are incremented rather than replaced with a new radiosity value. The incremental Gauss-Seidel algorithm was first proposed by Shirley [144, 143] as a improvement over stochastic Southwell relaxation, which will be discussed next.

6.3 Stochastic Southwell relaxation

6.3.1 Southwell relaxation

In each Southwell relaxation step, the patch s with largest residu component r_s is relaxed instead of relaxing each component in turn. The correction vector $\Delta\mathbf{x}$ is identical as in Gauss-Seidel and also in this case, the updated residu component r_s will be zero after each step. In each iteration step, (6.5) is evaluated for every component i so that potentially all components of the solution will be updated. By relaxing the largest component in each step, faster convergence rates can be obtained, especially when there is a large variation in the size of the residu components. The iteration steps are however more expensive since a search for the largest residu component is required.

Southwell relaxation is particularly suited in order to solve the power system (4.28), or possibly the adjoint radiosity system (4.31): for the power system, $c_{is} = -\rho_i F_{si}$ if $i \neq s$ and $c_{ss} = 1$, so that $r_i \leftarrow r_i + \rho_i F_{si} r_s$ if $i \neq s$. Shirley [142] proposed to compute all form factors F_{si} for fixed s simultaneously using uniformly distributed local lines (estimator F_{is}^R, algorithm 5.3.2). The resulting algorithm (see below) is a ray-tracing based variant of progressive refinement radiosity: in each step, the patch s with highest unshot power $\Delta P_s = r_s$ is selected. The unshot power of the selected shooting patch is propagated into the environment by shooting rays with uniform random origin on the shooting patch, and cosine-distributed direction.

Algorithm 11: Stochastic Southwell relaxation [142].

1. Initialise $P_i \leftarrow \Phi_i$, $\Delta P_i \leftarrow \Phi_i$ for all patches i;
2. Until $\|\Delta P\| \leq \varepsilon$ or number of steps exceeds maximum, do
   (a) Search patch s with largest ΔP_s;
   (b) Choose number of samples N (§6.3.2);
   (c) Do N times,
      i. Sample uniform random point x on patch s;
      ii. Sample cosine-distributed random direction Θ w.r.t. surface normal at x;
      iii. Determine patch i containing nearest hit of the ray with origin at x and direction Θ with the surfaces in the scene;
      iv. $\delta P \leftarrow \frac{1}{N}\rho_i \Delta P_s$; $P_i \leftarrow P_i + \delta P$; $\Delta P_i \leftarrow \Delta P_i + \delta P$.
   (d) $\Delta P_s \leftarrow 0$;
   (e) Display image using P_i.

6.3.2 Time-complexity

How many samples N are needed in each iteration step in order to compute the radiosity increments $\delta B_i = \Delta P_s F_{si} \rho_i / A_i$ to the all other patches i with error less than ε with 99.7% certainty? In other words: how large needs N to be taken so that for all i:

$$|\,\delta\hat{B}_i - \delta B_i\,| = \frac{\rho_i \Delta P_s}{A_i}\,|\,\hat{F}_{si}^R - F_{si}\,| \quad < \quad \varepsilon\ ?$$

According to the central limit theorem of probability, N shall be chosen so that

$$\frac{\rho_i \Delta P_s}{A_i} \cdot 3\sqrt{\frac{V[\hat{F}_{si}^R]}{N}} \approx \varepsilon$$

or, since $V[\hat{F}_{si}^R] = F_{si}(1 - F_{si})$ (5.8),

$$N \approx \left(3\frac{\rho_i \Delta P_s}{A_i \varepsilon}\right)^2 F_{si}(1 - F_{si}).$$

Approximating $1 - F_{si} \approx 1$ and using $\Delta P_s F_{si} \rho_i / A_i = \delta B_i$,

$$N \approx \frac{9 \Delta P_s}{\varepsilon^2} \cdot \max_i \frac{\rho_i \, \delta B_i}{A_i}. \tag{6.7}$$

The relation between the number of patches n and the number of required samples N has been studied in [144] and in more detail in [132]. Consider a fixed scene. With a finer discretisation of the scene, which leaves s (and ΔP_s) unchanged, the radiosity increments δB_i and the reflectivities ρ_i will hardly change. Since the total surface area A_T is fixed, a finer discretisation of the scene will however affect A_i. As long as $\max_i(\rho_i / A_i)$ does not change however, the number of samples N does not need to be increased. If all patches $i \neq s$ would however be split in 2 new patches with equal area, N would have to be doubled. This would also be the case if only the patch with maximum ρ_i / A_i were split in two (a bad idea!). Although the truth thus is more complicated, often a linear relationship is assumed: on the average, the amount of samples N for one relaxation step will be about $\mathcal{O}(n)$.

The cost of taking one sample is $\mathcal{O}(\log n)$. The search for the patch s with highest unshot radiosity takes $\mathcal{O}(n)$ work. The amount of work for one relaxation step thus will be about $\mathcal{O}(n \log n)$. With the hemi-cube method to compute the form factors in each step, the time complexity of a single step is $\mathcal{O}(n)$. Since in total $\mathcal{O}(n)$ relaxation steps are required, the time complexity of stochastic Southwell relaxation is $\mathcal{O}(n^2 \log n)$ and is *not better than with a deterministic approach*!

6.3.3 Incremental stochastic Gauss-Seidel iterative method

In order to avoid the search for the patch with highest unshot power, Shirley [144] proposed to relax each patch in turn. The resulting algorithm is identical to algorithm 11 except that each patch s is selected in turn in step 2a. It corresponds to the Gauss-Seidel iterative method in which expression (6.5) is evaluated in each step instead of (6.4).

Unlike deterministic Gauss-Seidel, incremental stochastic Gauss-Seidel has a nearly identical convergence rate as the stochastic Jacobi iterative method, which will be presented first. The comparison follows next.

6.4 Stochastic Jacobi iterative method

6.4.1 The Jacobi iterative method

The Jacobi method is usually presented as a variation on the Gauss-Seidel method in which all components of the solution are relaxed simultaneously. In the k-th step, for all i simultaneously:

$$x_i^{(k+1)} \leftarrow \left(e_i - \sum_{j \neq i} c_{ij} x_j^{(k)} \right) / c_{ii} \tag{6.8}$$

Following the same reasoning as before, the correction vector $\Delta \mathbf{x}^{(k)}$ in the k-th step can be written as

$$\Delta x_i^{(k)} = r_i^{(k)} / c_{ii}.$$

The updated residu is

$$r_i^{(k+1)} = r_i^{(k)} - \sum_j c_{ij} r_j^{(k)} / c_{jj} = - \sum_{j \neq i} c_{ij} r_j^{(k)} / c_{jj}. \tag{6.9}$$

As with the Gauss-Seidel method, the choice to evaluate (6.8) or (6.9) in each step results in different algorithms. We will call the algorithm based on (6.9) the *incremental* and the algorithm based on (6.8) the *regular* Jacobi iterative method (§6.4.2 and §6.4.3 respectively). The former is called "incremental" because the radiosity result will be obtained as the sum of increments computed in each step.

6.4.2 Incremental stochastic Jacobi iterative method

When applied to the power system (4.28), expression (6.9) yields

$$\Delta P_i^{(k+1)} = \sum_{j \neq i} \Delta P_j^{(k)} F_{ji} \rho_i \tag{6.10}$$

or equivalently:

$$\Delta P_i^{(k+1)} = \sum_j \sum_{l \neq j} \Delta P_j^{(k)} F_{jl} \rho_l \ \delta_{li}.$$

These double sums (one for each patch i) can be estimated efficiently and simultaneously with the following estimators (one estimator for each patch i):

- sample a pair of patches (j,l) with probability $p(j,l) = p_j p_{l|j}$ with

$$\begin{aligned} p_j &= \Delta P_j^{(k)} / \Delta P_T^{(k)} \quad \text{with} \quad \Delta P_T^{(k)} = \sum_j \Delta_j^{(k)} \\ p_{l|j} &= F_{jl} \end{aligned}$$

  The former requires sampling a discrete probability distribution as explained in §4.4. The latter is sampled by tracing a local uniformly distributed line with origin on j. Global lines can be used as well, but enforce $p_j = A_j / A_T$;

- the sample contribution is $\Delta \hat{P}_i(j,l) = \rho_i \Delta P_T^{(k)} \delta_{li}$.

Sampling a pair of patches (j,l) is identical in all the estimators. The difference is only in the sample contributions $\Delta \hat{P}_i(j,l)$, so that all estimators can be sampled simultaneously. The expectation is:

$$E\left[\Delta \hat{P}_i\right] = \sum_{j,l} \Delta \hat{P}_i(j,l) p(j,l) = \sum_{j,l} \rho_i \delta_{li} \Delta P_T^{(k)} \frac{\Delta P_j^{(k)}}{\Delta P_T^{(k)}} F_{jl} = \sum_j \Delta P_j^{(k)} F_{ji} \rho_i = \Delta P_i^{(k+1)}.$$

When the discrete pdf p_j is sampled using the stratified sampling algorithm 6 [111], algorithm 12 results.

Algorithm 12: Incremental stochastic Jacobi iterative method.

1. Initialise total power $P_i \leftarrow \Phi_i$, unshot power $\Delta P_i \leftarrow \Phi_i$, received power $\delta P_i \leftarrow 0$ for all patches i and compute total unshot power $\Delta P_T = \sum_i \Delta P_i$;
2. Until $\|\Delta P_i\| \leq \varepsilon$ or number of steps exceeds maximum, do
   (a) Choose number of samples N (§6.4.5);
   (b) Generate a random number $\xi \in (0, 1)$;
   (c) Initialise $N_{prev} \leftarrow 0$; $q \leftarrow 0$;
   (d) Iterate over all patches i, for each i, do
      i. $q_i \leftarrow \Delta P_i / \Delta P_T$;
      ii. $q \leftarrow q + q_i$;
      iii. $N_i \leftarrow \lfloor Nq + \xi \rfloor - N_{prev}$;
      iv. Do N_i times,
         A. Sample random point x on S_i;
         B. Sample cosine-distributed direction Θ at x;
         C. Determine patch j containing the nearest intersection point of the ray originating at x and with direction Θ, with the surfaces of the scene;
         D. Increment $\delta P_j \leftarrow \delta P_j + \frac{1}{N} \rho_j \Delta P_T$.
      v. $N_{prev} \leftarrow N_{prev} + N_i$.
   (e) Iterate over all patches i, increment total power $P_i \leftarrow P_i + \delta P_i$, replace unshot power $\Delta P_i \leftarrow \delta P_i$ and clear received power $\delta P_i \leftarrow 0$. Compute new total unshot power ΔP_T on the fly.
   (f) Display image using P_i.

Schematically, algorithm 12 proceeds as follows:

$$
\begin{array}{lllll}
\text{initialisation:} & & \Delta P^{(0)} = \Phi & & P^{(0)} = \Phi \\
\text{iteration 1:} & \Delta P^{(0)} \longrightarrow & \Delta P^{(1)} = \mathcal{T}\Delta P^{(0)} & & P^{(1)} = \Delta P^{(0)} + \Delta P^{(1)} \\
\text{iteration 2:} & \Delta P^{(1)} \longrightarrow & \Delta P^{(2)} = \mathcal{T}\Delta P^{(1)} & & P^{(2)} = \Delta P^{(0)} + \Delta P^{(1)} + \Delta P^{(2)} \\
\text{iteration 3:} & \Delta P^{(2)} \longrightarrow & \Delta P^{(3)} = \mathcal{T}\Delta P^{(2)} & & P^{(3)} = \Delta P^{(0)} + \Delta P^{(1)} + \Delta P^{(2)} + \Delta P^{(3)} \\
& & \ldots & &
\end{array}
$$

In each iteration step, an increment $\Delta P^{(k+1)}$ is computed for given $\Delta P^{(k)}$, the result of the previous iteration step. Eventually, the solution is obtained as the sum of the increments: $P = \sum_k \Delta P^{(k)}$ with $\Delta P^{(0)} = \Phi$, the self-emitted power. Hence the name "incremental" Jacobi iterative method.

6.4.3 Regular stochastic Jacobi iterative method

Alternatively, expression (6.8) can be used. It is suited in order to solve both the gathering- or shooting-type equations. For the classical radiosity system (4.25), it

Algorithm 13: Regular stochastic Jacobi iterative method [111, 113, 112].

1. Initialise power $P_i \leftarrow \Phi_i$ and received power $\delta P_i \leftarrow 0$ for all patches i and compute total power $P_T = \sum_i P_i$;
2. $P_T^{(old)} \leftarrow 0$;
3. Until $P_T - P_T^{(old)} < \varepsilon$ or number of steps exceeds maximum, do
   (a) Choose number of samples N (§6.4.5);
   (b) Generate a random number $\xi \in (0, 1)$;
   (c) Initialise $N_{prev} \leftarrow 0$; $q \leftarrow 0$;
   (d) Iterate over all patches i, for each i, do
      i. $q_i \leftarrow f(P_i)/f(P_T)$;
      ii. $q \leftarrow q + q_i$;
      iii. $N_i \leftarrow \lfloor Nq + \xi \rfloor - N_{prev}$;
      iv. Do N_i times,
         A. Sample random point x on S_i;
         B. Sample cosine-distributed direction Θ at x;
         C. Determine patch j containing the nearest intersection point of the ray originating at x and with direction Θ, with the surfaces of the scene;
         D. Increment $\delta P_j \leftarrow \delta P_j + \frac{1}{N} \rho_j P_i / q_i$.
      v. $N_{prev} \leftarrow N_{prev} + N_i$.
   (e) $P_T^{(old)} \leftarrow P_T$;
   (f) Iterate over all patches i, *replace* power $P_i \leftarrow \Phi_i + \delta P_i$, and clear received power $\delta P_i \leftarrow 0$. Compute new total power P_T on the fly.
   (g) Display image using P_i.

yields

$$B_i^{(k+1)} \leftarrow E_i + \rho_i \sum_j F_{ij} B_j^{(k)}.$$

Simultaneous estimation of this expressions for all i by Monte Carlo is possible, but the variance of the resulting estimators turns out to be similar to the variance for the regular stochastic Gauss-Seidel iterative method (§6.2.2).

Applied to the power system (4.28), one obtains:

$$P_i^{(k+1)} \leftarrow \Phi_i + \sum_j P_j^{(k)} F_{ji} \rho_i.$$

The resulting estimators and the algorithm 13 [111, 113, 112] are very similar as in the incremental stochastic Jacobi iterative method, except that total power is propagated and that the result of each iteration step replaces the previous intermediate power solution rather than being added to it. No unshot power needs to be stored.

Schematically, algorithm 13 proceeds as follows:

$$\begin{array}{rl} \text{initialisation:} & P^{(0)} = \Phi \\ \text{iteration 1:} & P^{(0)} \longrightarrow P^{(1)} = \Phi + \mathcal{T}P^{(0)} \\ \text{iteration 2:} & P^{(1)} \longrightarrow P^{(2)} = \Phi + \mathcal{T}P^{(1)} \\ \text{iteration 3:} & P^{(2)} \longrightarrow P^{(3)} = \Phi + \mathcal{T}P^{(2)} \\ & \ldots \end{array}$$

The final solution is the result of the last iteration.

6.4.4 Time-complexity and discussion

The analysis of the time complexity is very similar for both regular as well as incremental Jacobi iterations.

Variance of the Jacobi estimators

Consider the incremental Jacobi estimators $\Delta\hat{P}^{(k+1)}$ for $\Delta P^{(k+1)}$ in $\Delta P^{(k+1)} = \sum_{j,l} \Delta P_j^{(k)} F_{jl}\rho_l\delta_{li}$ (§6.4.2). The variance $V[\Delta\hat{P}^{(k+1)}]$ of these estimators is:

$$\begin{aligned} V\left[\Delta\hat{P}_i^{(k+1)}\right] &= E\left[\left(\Delta\hat{P}_i^{(k+1)}\right)^2\right] - \left(E\left[\Delta\hat{P}_i^{(k+1)}\right]\right)^2 \\ &= \sum_{j,l}\left(\rho_i\delta_{li}\Delta P_T^{(k)}\right)^2 \frac{\Delta P_j^{(k)}}{\Delta P_T^{(k)}} F_{jl} - \left(\Delta P_i^{(k+1)}\right)^2 \\ &= \rho_i\Delta P_T^{(k)} \sum_{j,l} \Delta P_j^{(k)} F_{jl}\rho_i\delta_{li} - \left(\Delta P_i^{(k+1)}\right)^2 \\ &= \rho_i\Delta P_T^{(k)}\Delta P_i^{(k+1)} - \left(\Delta P_i^{(k+1)}\right)^2 \qquad (6.11) \\ &= \Delta P_i^{(k+1)} \cdot \rho_i \sum_j \Delta P_j^{(k)}(1 - F_{ji}) \\ &\approx \rho_i\Delta P_T^{(k)}\Delta P_i^{(k+1)}. \qquad (6.12) \end{aligned}$$

A similar reasoning for the regular Jacobi iterative method, applied to the power equations $P_i - \Phi_i = \sum_{j,l} P_j F_{jl}\rho_l\delta_{li}$, yields

$$\begin{aligned} V\left[\hat{P}_i\right] &= \rho_i P_T(P_i - \Phi_i) - (P_i - \Phi_i)^2 \qquad (6.13) \\ &\approx \rho_i P_T(P_i - \Phi_i). \qquad (6.14) \end{aligned}$$

Number of samples needed in a single iteration

In a single Jacobi iteration, n components are relaxed simultaneously. It was shown before (§6.3.2) that the computation cost for relaxing a single component is $\mathcal{O}(n \log n)$ with a stochastic and $\mathcal{O}(n)$ with a deterministic approach. At first sight, one would

expect that the cost of a full Jacobi iteration is n times the cost of a relaxation step in which a single component is relaxed. With a deterministic approach, (6.9) takes $\mathcal{O}(n^2)$ work indeed. *Using the stochastic approach that was outlined above however, the computational work still is only* $\mathcal{O}(n \log n)$.

Proof: Consider the $(k+1)$-th step of the incremental Jacobi method (§6.4.2). In order to compute all radiosity increments $\Delta B_i^{(k+1)} = \sum_j \Delta P_j^{(k)} F_{ji} \rho_i / A_i$ with error less than ε and with 99.7% confidence, the number of samples N shall be chosen so that

$$\frac{1}{A_i} \cdot 3\sqrt{\frac{V[\Delta \hat{P}_i^{(k+1)}]}{N}} \approx \varepsilon.$$

Filling in (6.12), we obtain:

$$N \approx \frac{9}{A_i^2 \varepsilon^2} \rho_i \Delta P_T^{(k)} \Delta P_i^{(k+1)}.$$

Using $\Delta P_i^{(k+1)} = A_i \Delta B_i^{(k+1)}$, we find

$$N \approx \frac{9 \Delta P_T^{(k)}}{\varepsilon^2} \cdot \max_i \frac{\rho_i \Delta B_i^{(k+1)}}{A_i}. \tag{6.15}$$

For a regular Jacobi iteration (6.4.3), we find, based on (6.14):

$$N \approx \frac{9 P_T}{\varepsilon^2} \cdot \max_i \frac{\rho_i (B_i - E_i)}{A_i}. \tag{6.16}$$

These relations are very similar to (6.7). The same discussion (§6.3.2) about the relation between N and n can be repeated here, suggesting $\mathcal{O}(n)$ number of samples for the n simultaneous relaxation steps. Generating each sample takes $\mathcal{O}(\log n)$ work. The resulting time complexity is $\mathcal{O}(n \log n)$. □

This result is not so surprising as might seem at first sight however: the number of samples N required to compute the radiosity increments $\Delta B_i^{(k+1)}$, resulting from all n subsequent relaxation steps, to specified accuracy ε, is simply the sum of the samples needed for each separate relaxation step according to (6.7) if the relaxation steps are estimated independently. This is the case here. A similar reasoning holds in the case of regular Jacobi iteration.

It is true that in the deterministic case, the cost of a full Jacobi iteration also is the sum of the costs of relaxating n components. The cost for relaxing a single component in the stochastic methods is however proportional to the power that is to be distributed. In the deterministic methods, the cost is always the same for each step, regardless how large or small the power to be distributed is.

Number of samples needed in total

In the incremental Jacobi method (§6.4.2), the solution P is obtained eventually as the sum of the increments computed in each iteration: $P_i = \sum_k \Delta P_i^{(k)}$, with $\Delta P_i^{(0)} =$

Φ_i, the self-emitted power. If N_k samples are used in the k-th iteration, then the variance in the k-th iteration is $V[\Delta\hat{P}_i^{(k)}]/N_k$. The total variance is

$$V[\hat{P}_i] = \sum_{k=1}^{K} \frac{1}{N_k} V[\Delta\hat{P}_i^{(k)}]$$

where K is the total number of iterations and $V[\Delta\hat{P}_i^{(k)}]$ is given by (6.11). A near-optimal allocation of samples over the individual iterations is obtained if $1/N_k$ is inverse proportional to $V[\Delta\hat{P}_i^{(k)}]$ (§4.3.4). Expression (6.12) indicates that the variance in the k-th step is approximately proportional to the total power $\Delta P_T^{(k-1)}$ to be propagated in the k-th iteration. The optimal allocation of samples over the iterations is thus obtained by choosing N_k proportional to $\Delta P_T^{(k-1)}$:

$$N_k \approx N \cdot \frac{\Delta P_T^{(k-1)}}{P_T}$$

where N is the total number of samples. The total variance of the iterative Jacobi method then becomes

$$\begin{aligned} V[\hat{P}_i] &= \frac{P_T}{N} \sum_{k=1}^{K} \frac{1}{\Delta P_T^{(k-1)}} \left(\rho_i \Delta P_T^{(k-1)} \Delta P_i^k - \left(\Delta P_i^k\right)^2 \right) \qquad (6.17) \\ &= \frac{1}{N} \left[\rho_i P_T (P_i - \Phi_i) - \sum_{k=1}^{K} \frac{P_T}{\Delta P_T^{(k-1)}} (\Delta P_i^{(k)})^2 \right] \approx \frac{1}{N} \rho_i P_T (P_i - \Phi_i). \end{aligned}$$

The total variance is identical to the variance (6.13) of a regular Jacobi iteration with N samples, except for the (negligible) terms $\sum_{k=1}^{K} \frac{P_T}{\Delta P_T^{(k-1)}} (\Delta P_i^{(k)})^2$ versus $(P_i - \Phi_i)^2$. We conclude that *the total number of samples required in order to compute the radiosity to given accuracy in the incremental Jacobi iterative method is the same as in a single regular Jacobi iteration with input radiosities sufficiently close to the solution. In both cases, formula (6.16) applies.*

Variance propagation

We have ignored that the input radiosities $\Delta B^{(k-1)}$ in the k-th iteration are the result of a stochastic computation and thus exhibit some error themselves too. (This is of course not the case for $k = 0$, with $\Delta B^{(0)} = E$.) The effect of a perturbation ε_i on the input radiosities $\Delta B_i^{(k-1)}$ is as follows:

$$E[\Delta\hat{P}_i^{(k)}] = \sum_j (\Delta P_j^{(k-1)} + A_j \varepsilon_j) F_{ji} \rho_i = \Delta P_j^{(k)} + A_i \rho_i \sum_j F_{ij} \varepsilon_j.$$

At first sight, the method thus appears to be biased. The apparent bias would be bounded by $\rho_i \overline{\varepsilon}$ where $\overline{\varepsilon}$ is an upper bound for $|\,\varepsilon_j\,|$ for each patch j. ε_j however is a random variable itself, with mean 0, so that the expected bias is 0 as well.

Perturbations ε_j on the input radiosities however do have an effect on the variances $V[\Delta\hat{P}_i^{(k)}]$. It is easy to derive bounds for the additional variance using a similar reasoning as above, and by using an upper bound for $|\varepsilon_j|$. In practice, these bounds are overly pessimistic, because the perturbations are independent from patch to patch and fluctuate around zero, rather than being coherent. If a sufficient number of samples is used, it is much more realistic to ignore variance propagation. In §6.4.5, a heuristic will be derived for determining the number of samples. This heuristic appears to suffice in practical application.

The "warming-up" problem of regular Jacobi iterations

The total required number of samples (6.16) in order to compute the radiosity to prescribed accuracy with incremental Jacobi iterations is the same as the number of samples required in the last of a complete sequence of regular Jacobi iterations. In regular Jacobi iterations, the result of each iteration replaces the previous results instead of being added to it. This implies that the required total number of samples in regular Jacobi iteration will be higher than in incremental Jacobi iterations.

In both cases, the number of samples required in the first iteration is proportional to Φ_T, the total self-emitted power in the scene. In the first iteration, about $\rho_{av}\Phi_T$ power is created in the scene. ρ_{av} denotes the area-average reflectivity. $\rho_{av}\phi_T$ is approximately the amount of power in direct illumination. In the second *incremental* Jacobi iteration, the amount of power to be propagated is $\rho_{av}\Phi_T$, while in the second *regular* Jacobi iteration, the total power $(1+\rho_{av})\Phi_T$ is to be propagated. If N_1 samples are used in the first iteration, $N_2 = \rho_{av}N_1$ samples are required in the second incremental iteration, while in the second regular iteration, $N_2 = (1+\rho_{av})N_1$ samples will be required. The problem is that the first N_1 of the N_2 samples in the second regular iteration are used in order to re-compute the direct illumination which was computed already in the first iteration. The same problem also occurs in later iterations ($k > 2$). It has been called the "warming-up" problem of regular Jacobi iterations. Intuitively, the problem is that the effect of higher order inter-reflections is incorporated initially too slowly in regular Jacobi.

In order to avoid re-estimation of the result of previous steps, Neumann et al. [112] proposed to store the count of shot rays N_i with all patches i. During each iteration, only the additionally needed number of rays $N_i - N_i^{(old)}$ is shot in step 3(d)iv of algorithm 13. Occasionally, due to statistical fluctuations, the new required number of rays can be lower than the number of rays that was shot before. In that case, the effect of the last rays is undone by shooting negative amounts of energy to the patches that were hit by these last rays. A deterministic sampling strategy makes it possible to find out efficiently where to *unshoot* power.

With this improvement, both variants of the stochastic Jacobi iterative method perform identically. Algorithm 12 has slightly higher storage needs since three instead of two power vectors need to be kept in storage. On the other hand, it does not need "tricks" such as deterministic sampling and unshooting previously shot power.

Alternatively, regular Jacobi iterations are very useful in order to improve an available complete radiosity solution computed with incremental Jacobi or even another method. This will be explained in §6.6.3.

Stochastic Jacobi versus incremental stochastic Gauss-Seidel

At first sight, one might expect that the incremental stochastic Gauss-Seidel iterative method (§6.3.3) is more efficient that the stochastic Jacobi iterative methods. This is however not so and could be proved by analysing the required number of samples in order to achieve a prescribed accuracy as done above for the stochastic Jacobi iterative method.

Intuitively, the difference is that in incremental Gauss-Seidel, the power contributions received from previous patches in a sweep is immediately propagated. In the Jacobi iterative method, these contributions are propagated during the next sweep. In a deterministic approach, the amount of work to be done does not depend on the magnitude of the power that is propagated and incremental Gauss-Seidel will converge in fewer sweeps. In the stochastic approach however, the number of samples is chosen proportional to the power to be propagated so that eventually exactly the same amount of samples (rays) will be needed in both cases.

It is even so that stochastic Jacobi iterations will be slightly more efficient in practice due to the stratified sampling [113]. This stratified sampling is not possible with stochastic incremental Gauss-Seidel because the unshot power of the patches is updated immediately during each step so that the unshot power of subsequent patches is not known in advance.

Stochastic Gauss-Seidel on the other hand, stores only two power vectors without tricks.

6.4.5 Implementation

Dealing with multiple colour channels

In practice, a multi-valued spectral representation of power needs to be computed. Often for instance, the power or radiosity needs to be computed at three different wavelengths, corresponding to the red, green and blue primary colours of a computer display. There will be a different estimator $\Delta\hat{P}_i^\lambda$ for each colour band λ. The probability p_j^λ of sampling a ray originating at patch j will differ for each colour band. There are basically two ways to deal with multiple colour values:

1. Do the computations for each colour band separately. For each colour band, its proper estimator is used. If Λ values are used in the colour representation, this leads to an increase of work by a factor of Λ. $\Lambda = 3$ for the aforementioned RGB colour model;

2. Use a combined estimator in order to propagate power in all colour bands simultaneously: each p_j^λ is replaced by a single combined probability $\tilde{p}_j$.

   In order to do so, a mapping $f(P)$ from a full spectral representation of power P to a single floating point value is required. In our experiments, we have found it most convenient to use the average power in each wavelength band: $f(P) = \sum_{\lambda=1}^{\Lambda} P^\lambda/\Lambda$. The combined probability is $\tilde{p}_j = f(\Delta P_j^{(k)})/f(\Delta P_T^{(k)})$.

   Since a "wrong" pdf $\tilde{p}_j$ is used for sampling in each colour band, the primary estimates need to be compensated accordingly by weighting with $p_j^\lambda/\tilde{p}_j$.

The latter strategy yields the following combined estimator:

- sample probability $\tilde{p}(j,l) = f(\Delta P_j^{(k)})/f(\Delta P_T^{(k)}) \cdot F_{jl}$;
- sample contributions:

$$\Delta\hat{P}_i(j,l) = \rho_i \delta_{li} \Delta P_T^{(k)} \cdot \frac{\Delta P_j^{(k)}/\Delta P_T^{(k)}}{\tilde{p}(j,l)} = \rho_i \delta_{li} \Delta P_j^{(k)}/\tilde{p}(j,l)$$

The only required changes to the algorithm 12 are:

- step 2(d)i: $q_i \leftarrow f(\Delta P_i)/f(\Delta P_T)$;
- step 2(d)ivD: $\delta P_j \leftarrow \delta P_j + \frac{1}{N}\rho_j \Delta P_i/q_i$.

The required changes to algorithm 13 are identical.

A heuristic for choosing the number of samples

In order to compute the radiosity to prescribed accuracy ε with 99.7% confidence, the total number of samples N in incremental Jacobi, or the last iteration of regular Jacobi, shall be chosen so that (6.16)

$$N \approx \frac{9P_T}{\varepsilon^2} \cdot \max_i \frac{\rho_i(B_i - E_i)}{A_i}.$$

A practical heuristic for choosing the number of samples is obtained by setting $B_i - E_i = B_{av}$ and $\varepsilon = B_{av}$, where B_{av} is the area-average radiosity $B_{av} = \sum_i A_i B_i/A_T = P_T/A_T$. Filling in these values in the expression above yields

$$N \approx 9A_T \max_i \frac{\rho_i}{A_i} < 9 \max_i \frac{A_T}{A_i}. \tag{6.18}$$

It is reasonable to skip the smallest patches in the scene. In our implementation, we have used this amount of samples already for the first iteration, in which only self-emitted power Φ_T is propagated. The amount of samples in subsequent iterations was chosen so that the power per ray, $P_{\text{ray}} = \Phi_T/N$ is kept constant.

(Overly) conservative choices for N can be made by setting $P_T = \Phi_T$ and using an a-priori upper bound for $B_i - E_i$. Such bounds have been derived for instance in [132].

6.5 Other relaxation methods

The computational cost for relaxing a single component in all deterministic relaxation algorithms is invariably $\mathcal{O}(n)$. The cost for relaxing m components simply is m times the cost for relaxing a single component. The only way how convergence can be reached faster, is by reducing the number steps. This can be done by

1. choosing better correction vectors $\Delta x^{(k)}$: over-relaxation [57, 48, 184, 59] and the conjugate gradient method [57, 72, 7] are examples of this approach;

2. appropriately weighing recent intermediate solutions as in the Chebyshev method [57, 7, 6].

Still, all the resulting algorithms require the computation of matrix-vector products $\mathbf{C} \cdot \Delta\mathbf{x}^{(k)}$. Estimating these with the Monte Carlo method yields stochastic variants of the algorithms.

In stochastic relaxation methods, the number of sweeps or steps is however not so important. It is rather the number of samples (rays in radiosity) that counts. The required number of samples is proportional to the magnitude of the corrections in each step. It is thus not clear a-priori whether the stochastic versions of more advanced relaxation methods will be beneficial, even when the deterministic versions are. In the following paragraphs, stochastic over-relaxation, the stochastic conjugate gradient method and the stochastic Chebyshev method are proposed and studied.

6.5.1 Stochastic over-relaxation

In over-relaxation, the correction $\Delta\mathbf{x}^{(k)}$ made in each step of an original relaxation method (e.g. Gauss-Seidel, Jacobi, Southwell) is exaggerated by multiplying it with a vector of carefully chosen over-relaxation factors $\omega^{(k)}$. The basic over-relaxation algorithm is shown in algorithm 14. The fundamental idea is to predict future corrections and apply them in advance. In the context of progressive refinement radiosity, this means that in each step, a shooting patch shoots already part of the power that it will receive only later on. If predicted well, the number of steps to convergence will be reduced. The choice $\omega_i^{(k)} = 1, \forall i$ yields no modification w.r.t. the original relaxation method. Over-relaxation in radiosity has been proposed by Feda and Purgathofer [48].

Algorithm 14: Basic over-relaxation algorithm.

1. Choose initial guess $\mathbf{x}^{(0)}$;
2. $\mathbf{r}^{(0)} \leftarrow \mathbf{e} - C \cdot \mathbf{x}^{(0)}$;
3. For $k = 0, 1, \ldots$ until convergence, do
   (a) Compute correction $\Delta\mathbf{x}^{(k)}$;
   (b) Compute over-relaxation factors $\omega^{(k)}$;
   (c) $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \omega^{(k)}\Delta\mathbf{x}^{(k)}$;
   (d) $\mathbf{r}^{(k+1)} \leftarrow \mathbf{r}^{(k)} - \mathbf{C} \cdot (\omega^{(k)}\Delta\mathbf{x}^{(k)})$.

Unfortunately, we cannot expect any benefit from stochastic over-relaxation. When $\mathbf{C} \cdot (\omega^{(k)}\Delta\mathbf{x}^{(k)})$ is estimated using Monte Carlo, the required number of samples for given accuracy will be higher, proportional to ω. In radiosity, the only effect is that rays that would be shot later anyways, are shot more early on. If ω is chosen too large, the power that was shot too much will need to be unshot afterwards. so that stochastic over-relaxation will even require extra work.

In short: while the number of sweeps may be reduced, the number of samples is not reduced and may even be higher if the over-relaxation factors are chosen too high.

6.5.2 Stochastic Chebyshev method

In the Chebyshev iterative method [57, 7], a reduction of the number of sweeps is obtained by combining recent approximations for the solution in a appropriate way. It however requires estimates of the smallest and largest eigenvalues λ_{min} and λ_{max} of the matrix C. Baranoski [6] has shown that for the classical radiosity system (4.25), $\nu = 1 + \rho_{avg}$ and $\mu = 1 - \rho_{avg}$ are usable estimates for λ_{max} and λ_{min} respectively. The Chebyshev algorithm (see e.g. [6]) contains radiosity-matrix-vector products that can be estimated by Monte Carlo using the same estimators as in the stochastic Jacobi method. The only difference is the magnitude of the correction vectors in each iteration.

It turns out that the analysis of the stochastic Chebyshev method is very similar to that presented above for the incremental stochastic Jacobi method. Also here, the final result is computed as the sum of the (positive) increments in each iteration. The magnitude of the individual increments is different, resulting in a different number of sweeps before convergence is achieved, but their sum is equal to the final solution, just like in the incremental Jacobi method. Since the same basic estimators are used in each step, the same formulae for the variance apply and the same total number of samples $N = P_T/P_{\text{ray}}$ will be required for computing the result to a given accuracy.

We conclude that also the computation cost of the Chebyshev method will be (almost) equal to that of the incremental Jacobi method. There can be at most minor difference, since the fixed cost per sweep is small, although not entirely negligible, and because stratified sampling may perform slightly differently.

6.5.3 Stochastic conjugate gradient method

The basic idea of the conjugate gradients method [57] is to consider the problem of solving the system of linear equations $\mathbf{Cx} = \mathbf{e}$ as a problem of finding the minimum of the functional $\phi(x)$:

$$\phi(x) = \frac{1}{2}\mathbf{x}^\top \mathbf{C}\mathbf{x} - \mathbf{x}^\top \mathbf{e}.$$

The matrix $\mathbf{C}$ is assumed to be symmetric and positive definite. The minimum value of ϕ is reached when $\mathbf{x} = \mathbf{C}^{-1}\mathbf{e}$. In each iteration step of the conjugate gradient method, a direction vector $\mathbf{p}^{(k)}$ and corresponding step-size $\alpha^{(k)}$ are carefully chosen so that if $\Delta\mathbf{x}^{(k)} = \alpha^{(k)}\mathbf{p}^{(k)}$, the residu $\mathbf{r}^{(k+1)}$ is minimised within some constraints. Each step requires that a matrix-vector product $\mathbf{Cp}^{(k)}$ is computed.

The conjugate gradient method however only works for symmetric and positive definite matrices $\mathbf{C}$. It does not work for the classical radiosity equations (4.25) or the power equations (4.28), unlike the other methods. The classical radiosity equations (4.25) can be transformed into a symmetric and positive definite system by multiplying each equation by A_i/ρ_i [109], yielding

$$\sum_j \left(\frac{A_i}{\rho_i}\delta_{ij} - A_i F_{ij} \right) B_j = \frac{A_i E_i}{\rho_i}.$$

Algorithm 15 shows the stochastic adaptation[2] of the conjugate gradients algorithm shown in [57, §10.2.7] applied to the transformed radiosity system shown above.

[2] Suggested in personal communication by László Neumann.

Algorithm 15: Stochastic conjugate gradients

1. Initialise $B_i \leftarrow 0$, $\Delta B_i \leftarrow A_i E_i / \rho_i$ and $D_i \leftarrow \Delta B_i$ for all patches i, compute $R_1 \leftarrow \sum_i [\Delta B_i]^2$ on the fly;
2. Until $\sqrt{R_1} < \varepsilon$ or number of steps has reached maximum, do
   (a) Compute $W_i = A_i D_i / \rho_i - \sum_j A_i F_{ij} D_j$ by estimating $\sum_j A_i F_{ij} D_j = \sum_j A_j D_j F_{ji}$ using Monte Carlo:
      i. initialise $W_i \leftarrow A_i D_i / \rho_i, \forall i$;
      ii. choose number of samples N;
      iii. take N samples as follows:
         A. Select patch j with probability $q_j = \mid A_j f(D_j) \mid \; / \sum_j \mid A_j f(D_j) \mid$;
         B. Select patch i by tracing a uniformly distributed local line from j;
         C. $W_i \leftarrow W_i - \frac{1}{N} A_j D_j / q_j$.
   (b) $\alpha \leftarrow R_1 / \sum_i W_i D_i$;
   (c) $R_2 \leftarrow R_1$;
   (d) Update $B_i \leftarrow B_i + \alpha D_i$ and $\Delta B_i \leftarrow \Delta B_i - \alpha W_i$ for all patches i, compute $R_1 \leftarrow \sum_i [\Delta B_i]^2$ on the fly;
   (e) $\beta \leftarrow R_1 / R_2$, update $D_i \leftarrow \Delta B_i + \beta D_i$ for all i.

The determination of the required number of samples N in step 2(a)ii will require a different error measure, similar to the one proposed by Neumann et al. in [109].

6.6 Progressive variance reduction

Unlike in their deterministic counterparts, the quality of intermediate results after each iteration step is not an issue in stochastic relaxation methods. Indeed: the computations can always be arranged in such a way that a complete run is done first with a small number of samples so that a first solution is obtained quickly. This first solution will be complete in the sense that the effect of all higher order inter-reflections will be present in it, but it will exhibit noisy artifacts. The quality of this first solution can be gracefully improved by increasing the number of samples. Two strategies to do so have been proposed in the context of Monte Carlo radiosity: *merging the solutions of different runs* [136] and a technique called *progressive ray refinement* [49].

6.6.1 Merging the results of different runs

The resulting power estimates $\hat{P}_i$, obtained with any stochastic relaxation method, are random variables with expectation P_i and a certain variance $V[\hat{P}_i]/N$, which is inverse proportional to the number of samples N. Suppose two different runs are performed with the same stochastic relaxation algorithm, one with N_1 samples and the second with N_2 samples. The resulting estimates are $\tilde{P}_i^1$ and $\tilde{P}_i^2$, with variance $V[\hat{P}_i]/N_1$ and $V[\hat{P}_i]/N_2$ respectively. According to (4.16), the optimal combination $w_1 \tilde{P}_i^1 + w_2 \tilde{P}_i^2$ of *independent* estimates is obtained if the weights are chosen inverse

proportional to the variances:

$$\frac{w_1}{w_2} = \frac{V[\hat{P}_i]/N_2}{V[\hat{P}_i]/N_1} = \frac{N_1}{N_2}$$

Since $w_1 + w_2 = 1$, we obtain

$$w_1 = \frac{N_1}{N_1 + N_2} \quad ; \quad w_2 = \frac{N_2}{N_1 + N_2}.$$

The variance on the combined result is then $V[\hat{P}_i]/(N_1 + N_2)$. Also if more than two independent results are to be combined, the combination weight for each run shall be chosen proportional to the number of samples in that run.

6.6.2 Parallel Monte Carlo radiosity

The runs can be done in parallel on different processors. If the number of samples in each run is chosen proportional to the speed of the processor executing the run, all computations will be finished simultaneously. It is extremely easy to perform parallel Monte Carlo radiosity computations on a heterogeneous network of processors with distributed memory if the scene database is duplicated for each processor. Without scene duplication, the parallelisation of Monte Carlo radiosity poses identical problems as in ray-tracing. On processors with shared memory, the scene geometry and material data can be shared by all processors. Of course, each processor still needs its own copy of the radiosity or power vectors as well as other variables in the stochastic relaxation algorithm.

6.6.3 Merging with the result of subsequent Jacobi iterations

Consider a complete solution $P^{(1)}$. A regular stochastic Jacobi iteration (algorithm 13) will transform $P^{(1)}$ into a new complete solution $P^{(2)}$ in a single sweep through the patches of the scene. Such a sweep can be executed more rapidly than a truely independent new run because of the stratified sampling and better data locality. If the same elementary ray power $P_{\text{ray}} = P_T/N$ (§6.4.5) is used in this sweep as was used for computing $P^{(1)}$, merging $P^{(1)}$ and $P^{(2)}$ with combination weights proportional to the number of samples will result in a better estimate [110, 113].

Also here, there will be variance propagation (§6.4.4). When a sufficient number of samples is used, for instance determined with the heuristic in §6.4.5, it is realistic to ignore variance propagation also in this case.

6.6.4 Progressive ray refinement

Alternatively, the result obtained using an incremental stochastic relaxation algorithm with given elementary ray power P_{ray} can be improved by restarting the computations with smaller $P'_{\text{ray}} = k \cdot P_{\text{ray}}$ with $k < 1$. If in the old run N samples were taken, the new run will take approximately $N/k > N$ samples. The basic observation of progressive ray refinement [49] is that the first N samples of the new run can be assumed to be identical as in the old run. Re-sampling these would only lead to a

waste of computation resources. Only the new $N/k - N = N(1-k)/k$ samples cause new computation work. The effect of the old samples in the new run can be incorporated by adapting the residual power $\Delta P \leftarrow k \cdot \Delta P + (1-k) \cdot \Phi$. After this adaptation, the old run is simply *continued* rather than restarted.

A general outline of progressive ray refinement is shown in algorithm 16. It was first proposed [49] in the context of Shirley's residu-based stochastic Gauss-Seidel iterations [143, 144], but can be applied to all other stochastic relaxation algorithms that explicitly use the residu as well.

Algorithm 16: Progressive ray refinement [49] (outline)

1. Choose number of samples N for first "run" according to formula (6.18), $P_{\text{ray}} \leftarrow P_T/N$
2. $n \leftarrow 1$;
3. until noisy artifacts are sufficiently reduced,
   (a) $k \leftarrow \frac{n-1}{n}$;
   (b) if $n > 1$, then $P_{\text{ray}} \leftarrow k \cdot P_{\text{ray}}$;
   (c) for all i, do $\Delta P_i \leftarrow k \cdot \Delta P_i + (1-k) \cdot \Phi_i$, $P_i \leftarrow k \cdot P_i + (1-k) \cdot \Phi_i$;
   (d) repeat stochastic relaxation steps with elementary ray power P_{ray} until $\|\Delta P\|$ has sufficiently decreased;
   (e) $n \leftarrow n + 1$;

Merging with stochastic Jacobi-iterations may be slightly more efficient in practice due to better data locality and the need for only a single sweep through the scene in order to obtain a new, (nearly) independent, solution.

6.7 Conclusion

The most important theoretical insight resulting from the study in this chapter concerns the computing cost of stochastic versus deterministic relaxation algorithms. The computing cost with a stochastic relaxation method is related to the number of samples (rays in radiosity) that needs to be shot, rather than to the number of patches in the scene. It appears that there is only a very loose relationship between the required number of samples and the number of patches. Unlike their deterministic counterparts, advanced relaxation methods, that result in fewer sweeps through the patches of the scene, do not yield a significant saving over simple Jacobi iterations with Monte Carlo because the number of required samples is not lower.

Because of this, the simple stochastic Jacobi methods are good candidates to use in practice. In particular, we recommend the use of a sequence of incremental stochastic Jacobi iterations until a first "complete" radiosity solution is obtained. The variance in this first solution is then gracefully reduced using regular stochastic Jacobi iterations taking the result of the previous iteration as their input. The algorithms for incremental and regular stochastic Jacobi iterations are very similar, which is convenient for the implementation.

Empirical results concerning stochastic relaxation methods have been presented in cited literature and will not be repeated here. The empirical verification of the variance

analysis will be presented in §7.4.4, when comparing with random walk methods, and in §12.3 in the context of low-discrepancy sampling. Figure 6.1 however illustrates our claim that stochastic relaxation can yield useful images much faster than deterministic relaxation algorithms.

Figure 6.1: Stochastic relaxation methods can yield useful images much faster than their deterministic counterparts. The shown environment consists of slightly more than 30,000 patches. The top image was obtained with incremental Jacobi iterations in less than 2 minutes using about 10^6 rays. Even if only 1 ray were used for each form factor, $9 \cdot 10^8$ rays would be required with a deterministic method. Noisy artifacts are still visible, but are progressively reduced. After about 30 minutes, they are not visible anymore.
The bottom images illustrate progressive variance reduction using regular Jacobi iterations after a first sequence of incremental iterations. The images are shown without Gouraud shading, so noisy artifacts are more visible. The images shown are obtained after 1, 4, 16, 64 and 252 (right-to-left, top-to-bottom) iterations of less than 2 minutes each.
The model shown is an edited part of the Soda Hall VRML model made available at the University of California at Berkeley, U.S.A.

The method therefore appears best suited to a human computer with a table of random digits and no calculating machine.

GEORGE E. FORSYTHE AND RICHARD A. LEIBLER,
"Matrix Inversion by a Monte Carlo Method", Math. Tabl. Aids. Comput., Vol. 4, 1950

7 Random Walk Radiosity

Contrary to stochastic relaxation methods, the solution of linear systems by random walks is a well-covered topic in Monte Carlo literature [82, 51, 181, 34, 66, 156, 64, 45, 127]. Its application to radiosity has been proposed by M. Sbert [131, 132].

In this chapter, the linear systems $\mathbf{C}\mathbf{x} = \mathbf{e}$ will be written in the form $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$. The radiosity systems in §4.5 are of this form. The solution $\mathbf{x}$ of such a system can, at least formally, be expressed as an infinite series, called the *Neumann series expansion* of $\mathbf{x}$:

$$\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{e} + \mathbf{A}^2\mathbf{e} + \dots . \tag{7.1}$$

For the i-th component,

$$x_i = e_i + \sum_{j_0=1}^{n} a_{ij_0} e_{j_0} + \sum_{j_0=1}^{n} \sum_{j_1=1}^{n} a_{ij_0} a_{j_0 j_1} e_{j_1} + \dots . \tag{7.2}$$

A sufficient condition for convergence is the existence of a positive number α so that $\|\mathbf{A}^\alpha\| < 1$.

If converging, an infinite sum like (7.2) can be sampled by constructing random walks over the set of components i of the system. In case of radiosity, these random walks can be imagined as the path of a light particle with special properties. If done appropriately, random-walk methods have the same advantages as stochastic relaxation methods.

The solution of linear systems by random walks is very similar to the Monte Carlo solution of second-kind Fredholm integral equations by random walks. In rendering algorithms such as stochastic ray tracing and particle tracing [30, 83, 120, 95, 41, 174], the rendering equation [83] or its adjoint integral equation [122] are solved by means of random walks.

This chapter is organised as follows: first, we explain the concept of an analog continuous or discrete random walk and its relation with Fredholm integral equations of the second kind and systems of linear equations $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$ (§7.1). Section §7.2 presents a very general recipe in order to construct unbiased random walk estimators for linear systems. The variance of these estimators is analysed in §7.3. Finally, the application of random walk estimators for linear systems in the context of radiosity is discussed in §7.4 and a comparison with stochastic Jacobi relaxation is made.

7.1 Random walks and particle transport simulation

7.1.1 Continuous random walks and integral equations

Light transport is an instance of a wider class of linear particle transport problems, that can be solved as follows [85, 33]:

1. Fix a description of the *state* X of a particle. In many applications, including illumination computation, particles are sufficiently characterised by their position x, direction Θ, energy E[1] and the time t;

2. Fix a description of the particle sources by means of a normalised *source (or birth) density function* $S(X)$ and a constant S_T expressing the total emission intensity. With $X = (x, \Theta, E, t)$, $S(X)$ expresses the relative intensity of emission of particles with energy E at time t from position x and into direction Θ;

3. Fix a description of how particles interact with the medium and surfaces in which they travel. Particles can be scattered or absorbed. If the scattering and absorption of the particles only depends on their present state, and not on their past history, particle scattering and absorption is fully determined by a *transition density function* $T(X \rightarrow X')$ from each state X to each other state X'. The transition density function need not to be normalised:

$$\rho(X) = \int_\Omega T(X \rightarrow X')dX'$$

describes the average number of particles resulting when a particle scatters at X. Here, Ω denotes the full state space of the particles. $\rho(X)$ can be larger than 1, e.g. in nuclear reactions, in which case the medium is called a multiplying or super-critical medium. If $\rho(X) \leq 1$, $\alpha(X) = 1 - \rho(X)$ expresses the intensity of absorption at X. In case $\alpha(X) > 0$, the medium is called absorbing or sub-critical;

4. Particle paths are simulated by sampling emission events according to the source density function $S(X)$ and subsequently sampling scattering events according to $T(X \rightarrow X')$ until the particle is either absorbed or disappears from the region of interest (assuming that the particle will not re-enter the region of interest). While simulating particle paths, events of interest are counted.

In general, this method is suited in order to compute weighted integrals of the particle density function $\chi(X)$:

$$G = \int_\Omega g(X)\chi(X)dX \tag{7.3}$$

The *response* or *detector* function $g(X)$ expresses our interest in particles with given location, direction, energy and time X. For instance by taking $g(X) = 1$ for particles located on a given surface and 0 on other surfaces, the particle flux on the surface will be computed.

The particle density $\chi(X)$ is the differential particle flux at given location, direction and energy at a fixed time. It is the sum of the source density and the density of particles from elsewhere that are scattered into X:

$$\chi(X) = S(X) + \int_\Omega \chi(X')T(X' \rightarrow X)dX'. \tag{7.4}$$

In short: *simulation of particle paths with given source density $S(X)$ and transition density $T(X' \rightarrow X)$ is a technique to sample points X in state space Ω with (non-normalised) density $\chi(X)$ that is the solution of the integral equation (7.4).*

[1] for photons: $E = \hbar c/\lambda$ with $\hbar$ Planck's constant, c the velocity of light and λ the wavelength of the photon.

7.1.2 Analog continuous light transport simulation

The continuous radiosity equation (2.3) is very similar to (7.4):

- Only the location of the particle is of interest: $X = x$;
- The source density $S(X)$ corresponds to the normalised self-emitted radiosity $E(x)/\Phi_T$, with Φ_T the total self-emitted power in the scene;
- The transition density $T(X' \rightarrow X)$ corresponds to $G(y,x)\rho(x) = G(x,y)\rho(x)$;
- The particle density $\chi(X)$ corresponds with the radiosity $B(x)/\Phi_T$.

Simulating particle paths with above specified source and transition density, results in particle hits that are distributed proportional to the radiosity $B(x)$ at each point x on the surfaces of the scene. In a very similar way, particles can be traced for rendering non-diffuse scenes as well as scenes containing participating media. Such simulation according to the emission and scattering densities dictated by the physics of the problem, is called *analog* simulation.

Analog light transport simulation for image synthesis has been proposed by several researchers:

- Pattanaik [120] proposed analog simulation of radiosity, choosing $g(x)$ in (7.3) equal to functions $\psi_i(x)$, which are 1 if $x \in S_i$ and 0 if $x \notin S_i$ for each patch i in the scene (cfr. §2.2.1). The flux P_i on each patch is computed this way. The resulting algorithm is shown in algorithm 17;
- Analog continuous random walk algorithms for higher order radiosity approximations have been proposed later by Bouatouch et al. [14] and Feda [47]. These algorithms basically use the dual basis functions $\tilde{\psi}_{i,\alpha}(x)$ (§2.2.3) as detector functions $g(x)$. Indeed, the coefficients $B_{i,\alpha}$ in $\tilde{B}(x) = \sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x)$ are given by scalar products

  $$B_{i,\alpha} = \int_S \tilde{\psi}_{i,\alpha}(x) B(x)\, dA_x\,.$$

  These scalar products are of the form (7.3). Recall that for orthogonal basis functions, $\tilde{\psi}_{i,\alpha}(x) = \psi_{i,\alpha}(x)/A_i$;
- Dutré [43, 41] proposed detector functions $g(x)$ that correspond to each pixel in an image in order to avoid any discretisation of the problem;
- Also density estimation [145] and the photon map approach [79], fall into this category of methods. The detector functions they use are however harder to characterise.

Algorithm 17: Continuous shooting random walk radiosity: computes global power solution P_i.

1. Initialise $P_i \leftarrow \Phi_i$ for all patches i;
2. Choose number of random walks N;
3. Do N times,
   (a) Select source patch p with probability $q_p = f(\Phi_p)/f(\Phi_T)$;
   (b) Select uniform random location x on patch p;
   (c) $\sigma \leftarrow \Phi_p/q_p$;
   (d) $\xi \leftarrow 1$;
   (e) While $\xi < f(\rho_p)$, do
      i. Sample cosine-distributed direction Θ w.r.t. the normal at x;
      ii. Trace a ray with origin x and direction Θ. Replace p by the patch that is hit first and x by the hit point;
      iii. $P_p \leftarrow P_p + \rho_p \cdot \sigma$;
      iv. $\sigma \leftarrow \sigma \cdot \rho_p/f(\rho_p)$;
      v. Sample new uniform random number $\xi \in (0, 1)$.

7.1.3 Discrete random walks and systems of linear equations

Just like the particle density resulting from a random walk with continuous source and transition density is the solution of a second-kind Fredholm integral equation, the (discrete) particle density χ_i resulting from a discrete random walk with source density π_i and transition density p_{ij} (for particles going from i to j), is the solution of a system of linear equations[2]:

$$\chi_i = \pi_i + \sum_j \chi_j p_{ji}. \tag{7.5}$$

This density can be used in order to estimate scalar products

$$G =< g, \chi >= \sum_k g_k \chi_k.$$

With the choice $g_i = \delta_{ir}$, the r-th component of the solution χ of the linear system (7.5) is computed.

7.1.4 Analog discrete light transport simulation

Using uniformly distributed lines, either local or global, we are able to simulate particle transitions from i to j according to the form factor F_{ij}. Due to the order in which the indices of the form factors appear in the equations, random walk simulation is therefore suited to solve either the adjoint radiosity equation (4.31) or the power equations (4.28):

[2]Note the switch of indices p_{ji} instead of p_{ij}!

Analog discrete gathering random walk The radiosity B_i of a patch i can be obtained by computing the solution Y^i of

$$Y_k^i = \delta_{ik} + \sum_j Y_j^i \rho_j F_{jk}$$

on the light sources in the scene and evaluating $< E, Y^i >= \sum_k E_k Y_k^i$.

This can be done by tracing random walks originating at i ($\pi_k = \delta_{ik}$) and with transition probabilities $p_{jk} = \rho_j F_{jk}$. A transition can be simulated by first deciding whether the particle will be scattered (with probability ρ_j) or absorbed (with probability $\alpha_j = 1 - \rho_j$). If not absorbed, a uniformly distributed line through j is traced. The patch k hit next by this line is selected for a next event.

It is not necessary to store the full vector Y_k^i. In practice, each time a light source k is hit, a contribution $\frac{1}{N_i} E_k$ is recorded to B_i. N_i is the number of random walks that are traced, originating at i.

It is also convenient to suppress the absorption at the source i itself. This absorption suppression needs to be compensated by multiplying each contribution to B_i by ρ_i. Since also only non-self-emitted illumination is estimated, self-emitted radiosity E_i needs to be added explicitly. The resulting algorithm is shown in algorithm 18.

Analog discrete shooting random walk The power P_i can also be computed by solving

$$P_i = \Phi_i + \sum_j P_j F_{ji} \rho_i.$$

First, the source term Φ_i needs to be normalised: $\pi_i = \Phi_i / \Phi_T$. Random walks are traced with source probability π_i (proportional to the self-emitted flux) and transition probabilities $p_{ji} = F_{ji} \rho_i$. In order to simulate transitions, first a uniformly distributed random line through j is traced in order to determine i, the next patch to be visited. Next, a survival test is done with ρ_i being the probability of survival. If the particle survived, a contribution Φ_T / N is recorded on the patch i on which the particle survived. N is the total number of random walks being traced.

In practice, it is convenient to record a contribution also if the particle is absorbed, in other words: for each collision. In order to do so, the contributions need to be compensated by a factor ρ_i, where i is the hit patch. Also in this case, only non-self-emitted power is estimated. Self-emitted power needs to be added explicitly. The resulting algorithm is shown in 19.

Dealing with multiple colour bands As in stochastic Jacobi relaxation, the sampling probabilities are different for each colour band. The algorithms 17, 18 and 19 show the analog gathering and shooting random walk algorithms with combined probabilities. As before, f is a function that maps a spectrum to a single-valued real number, such as the average colour value.

7.1.5 Discrete versus continuous analog shooting

The difference between the continuous shooting algorithm 17 and the discrete shooting algorithm 19 is small in practice: in continuous shooting, a particle is scattered

Algorithm 18: Discrete gathering random walk radiosity: computes a single radiosity component B_i.

1. Initialise $B_i \leftarrow E_i$;
2. Choose number of random walks N_i;
3. Do N_i times,
   (a) Set current patch $p \leftarrow i$;
   (b) $\sigma \leftarrow \rho_i$;
   (c) $\xi \leftarrow 1$;
   (d) While $\xi < f(\rho_p)$, do
      i. Sample uniform random location x on patch p;
      ii. Sample cosine-distributed direction Θ w.r.t. the normal at x;
      iii. Trace a ray with origin x and direction Θ. Replace p by the patch that is hit first;
      iv. $B_i \leftarrow B_i + \frac{1}{N_i}\sigma E_p$;
      v. $\sigma \leftarrow \sigma \cdot \rho_p / f(\rho_p)$;
      vi. Sample new uniform random number $\xi \in (0, 1)$.

Algorithm 19: Discrete shooting random walk radiosity: computes global power solution P_i.

1. Initialise $P_i \leftarrow \Phi_i$ for all patches i;
2. Choose number of random walks N;
3. Do N times,
   (a) Select source patch p with probability $q_p = f(\Phi_p)/f(\Phi_T)$;
   (b) $\sigma \leftarrow \Phi_p / q_p$;
   (c) $\xi \leftarrow 1$;
   (d) While $\xi < f(\rho_p)$, do
      i. Sample uniform random location x on patch p;
      ii. Sample cosine-distributed direction Θ w.r.t. the normal at x;
      iii. Trace a ray with origin x and direction Θ. Replace p by the patch that is hit first;
      iv. $P_p \leftarrow P_p + \rho_p \cdot \sigma$;
      v. $\sigma \leftarrow \sigma \cdot \rho_p / f(\rho_p)$;
      vi. Sample new uniform random number $\xi \in (0, 1)$.

from the point of incidence on a patch. In discrete shooting, a particle is "warped" to a uniformly chosen new location on the patch.

These algorithms compute a different quantity: in continuous shooting, the com-

puted radiosity $\tilde{B}(x)$ at a point x on a patch i is:

$$\tilde{B}(x) = \frac{1}{A_i} \int_{S_i} B(x) dA_x, \quad x \in S_i$$

where $B(x)$ is the solution of the continuous radiosity equation

$$B(x) = E(x) + \rho(x) \sum_j \int_{S_j} G(x,y) B(y) dA_y$$

Assuming that the self-emitted radiosity $E(x)$ and the reflectivity $\rho(x)$ are constant on all patches, the discretisation error with continuous shooting is:

$$\varepsilon_1(x) = B(x) - \tilde{B}(x) = \rho_i \sum_j \int_{S_j} \left[G(x,y) - \frac{1}{A_i} \int_{S_i} G(x,y) dA_x \right] B(y)\ dA_y \tag{7.6}$$

With discrete shooting, the solution that is obtained is

$$\tilde{B}(x) = B_i$$

where B_i is the solution of the linear system

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

It was shown in §3.1 that the discretisation error in this case is given by

$$\begin{aligned} \varepsilon_2(x) = B(x) - \tilde{B}(x) &= \rho_i \sum_j \int_{S_j} \left[G(x,y) - \frac{1}{A_i} \int_{S_i} G(x,y) dA_x \right] \tilde{B}(y)\ dA_y \\ &+ \rho_i \int_S G(x,y) \left(B(y) - \tilde{B}(y) \right) dA_y \end{aligned} \tag{7.7}$$

The difference is

$$\varepsilon_2(x) - \varepsilon_1(x) = \rho_i \sum_j \frac{1}{A_i} \int_{S_i} \int_{S_j} G(x,y) \left(B(y) - \tilde{B}(y) \right) dA_y$$

In other words: the difference in discretisation error of discrete and continuous random walk radiosity equals the average propagated discretisation error in the discrete approach. There is no propagation of discretisation error in the continuous approach. Propagated discretisation error is responsible for e.g. *reflections of* light leaks in badly meshed scenes. In properly meshed scenes, propagated discretisation error is negligible. Indeed: in Galerkin methods, the average discretisation error $B(y) - \tilde{B}(y)$ is zero on every patch j. The average propagated discretisation error on a receiving patch i will therefore also most often be very near to zero.

While the discretisation error with a discrete random walk is at most slightly higher than with a continuous random walk, an empirical comparison in which quasi-Monte Carlo sampling was used, showed that the statistical error is reduced faster with the discrete random walk (see chapter 12). In total, the discrete random walk may be more efficient (see chapter 12).

7.2 Construction of random-walk estimators for linear systems

It is clear that only relatively few problems can be solved efficiently using a analog random walk approach. A larger arsenal of more widely applicable random walk estimators can be obtained by making abstraction from the physical model. The construction of random walks will be regarded merely as a tool for sampling certain probability distributions. This abstraction will also make it easier to analyse the efficiency of the random-walk solution of linear systems.

7.2.1 Path space

Consider the Neumann expansion of the solution $\mathbf{x}$ of the linear system $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$:

$$x_i = e_i + \sum_{j_1=1}^{n} a_{ij_1} e_{j_1} + \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} a_{ij_1} a_{j_1 j_2} e_{j_2} + \dots .$$

Each term of this infinite sum is uniquely determined by a sequence of component indices $j_0 = i, j_1, \dots, j_\tau$, where the length τ of the sequence can be any positive integer number. Such a sequence can also be thought of as a sequence of *states* visited by an imaginary particle undergoing a discrete random walk as described in §7.1.3. There thus is a one-to-one relation between discrete random walks on the set of components of a linear system, and terms of the solution of the system. In the case of radiosity with constant approximations, the states or components of the solution correspond to the patches in the scene. We will call the set of sequences $j_0, j_1, \dots, j_\tau$ *path space*. A full path will be denoted by the capital letter J.

The birth and transition probability distribution π_i and p_{ij} (§7.1.3) define a *probability measure* on path space: with each path $j_0, \dots, j_\tau$ is associated a probability

$$p(J) = \pi_{j_0} p_{j_0 j_1} p_{j_1 j_2} \dots p_{j_{\tau-1} j_\tau} \alpha_{j_\tau}.$$

α_i denotes the absorption, or termination, probability of paths at i:

$$\alpha_i = 1 - \sum_{j=1}^{n} p_{ij}.$$

The following assumptions are made:

- $\sum_i \pi_i = 1$ (normalised birth probabilities);
- $0 \leq p_{ij} \leq 1, \forall i, j$ (positive transition probabilities);
- The system is sub-critical: the survival probability $\sigma_i = \sum_j p_{ij} = 1 - \alpha_i \leq 1$ and $\sigma_i < 1$ for at least part of the states. We will also assume that the probability of reaching any state j from a given state i is nonzero. These assumptions suffice in order to guarantee that each random walk will have finite length with arbitrary high probability;

The basic idea of random walk solution of linear systems now is to transform each scalar product $< \mathbf{w}, \mathbf{x} >$, of a weight vector $\mathbf{w}$ with the solution $\mathbf{x}$ of $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$, to a corresponding scalar product in path space:

$$\sum_i w_i x_i = \sum_J s(J)p(J). \tag{7.8}$$

where $p(J)$ is the probability measure induced by a careful choice of birth and transition probability density distributions π_i and p_{ij}. By sampling paths J according to the probability measure, the scalar product can be obtained as

$$< \mathbf{w}, \mathbf{x} > \approx \frac{1}{N} \sum_{m=1}^{N} s(J_m)$$

J_m denotes the m-th traced random walk out of N. $s(J)$ denotes a contribution or *score* associated to path J. The problem now consists of designing good sets of birth and transitions density distributions with corresponding score functions, so that the scalar product (7.8) can be estimated with as little as possible work. We will see that there are plenty of possible solutions.

7.2.2 General form of the estimators

Finding the most general form of the unbiased random walk scores $s(J)$ is an intractable problem. Instead, we will look for a general condition of unbiasedness for random walk scores of the form

$$s(J) = \sum_{t=0}^{\tau} q_t(J)\omega_t^{\tau}(J)e_{j_t} \tag{7.9}$$

with

$$q_0(J) = \frac{w_{j_0}}{\pi_{j_0}} \quad \text{and} \quad q_t(J) = q_{t-1}(J) \cdot \frac{a_{j_{t-1}j_t}}{p_{j_{t-1}j_t}} \quad \text{if } t > 0. \tag{7.10}$$

The random walk estimators that are defined in this way, will yield a non-zero score only when they originate on a state for which $w_{j_0} \neq 0$ and when states are visited for which $e_{j_t} \neq 0$: they are *gathering* random walks. It is very important that $\pi_i > 0$ whenever $w_i \neq 0$ and also $p_{ij} > 0$ whenever $a_{ij} \neq 0$.

The coefficients ω_t^{τ} (to be fixed) determine the score contributed at the t-th state visited by a random walk of length τ. A different choice of these coefficients, leads to a different random walk estimator. The following theorem, adapted from [45], provides expressions that can be used in order to determine the coefficients $\omega_t^{\tau}(J)$ for unbiased random walk estimators of the form (7.9). A number of special choices will be discussed next.

Theorem 7.1 *Provided that all encountered infinite sums converge, if the coefficients* $\omega_t^\tau(J)$ *in (7.9) are chosen so that*

$$\sum_{t=0}^{\infty} \sum_{j_0,\ldots,j_t} w_{j_0} a_{j_0 j_1} \cdots a_{j_{t-1} j_t} e_{j_t} \times \left[1 - \omega_t^t - \sum_{\tau=t+1}^{\infty} \sum_{j_{t+1},\ldots,j_\tau} p_{j_t j_{t+1}} \cdots p_{j_{\tau-1} j_\tau} (\omega_t^\tau - \omega_t^{\tau-1}) \right] = 0, \quad (7.11)$$

the resulting random walk estimator is an unbiased estimator for the scalar product $< \mathbf{w}, \mathbf{x} >$ *with* $\mathbf{x}$ *the solution of* $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$.

Proof: Consider the scalar product of the weight vector $\mathbf{w}$ with the Neumann expansion (7.2) of $\mathbf{x}$, the solution of $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$:

$$< \mathbf{w}, \mathbf{x} >= \sum_{t=0}^{\infty} \sum_{j_0,\ldots,j_t} w_{j_0} a_{j_0 j_1} \cdots a_{j_{t-1} j_t} e_{j_t} \quad (7.12)$$

Consider the expected score of a random walk estimator with scores of the form (7.9):

$$E[s(J)] = \sum_{\tau=0}^{\infty} \left(\sum_{t=0}^{\tau} q_t(J) \omega_t^\tau(J) e_{j_t} \right) p(J) \quad (7.13)$$

The theorem is proved by converting (7.13) into a sum that can be subtracted term by term from (7.12) and requiring that the difference be zero. This takes the following steps and assumptions:

1. Substitute $p(J) = \pi_{j_0} p_{j_0 j_1} \ldots p_{j_{\tau-1} j_\tau} \alpha_{j_\tau}$ in (7.13);
2. Substitute $\alpha_{j_\tau} = 1 - \sum_{j_{\tau+1}} p_{j_\tau j_{\tau+1}}$;
3. Group terms that contain $\pi_{j_0} p_{j_0 j_1} \cdots p_{j_{\tau-1} j_\tau}$ for all τ;
4. Substitute $q_\tau(J) e_{j_\tau} \pi_{j_0} p_{j_0 j_1} \cdots p_{j_{\tau-1} j_\tau} = w_{j_0} a_{j_0 j_1} \cdots a_{j_{\tau-1} j_\tau} e_{j_\tau}$;
5. Change the summation order $\sum_{\tau=0}^{\infty} \sum_{t=0}^{\tau}$ into $\sum_{t=0}^{\infty} \sum_{\tau=t}^{\infty}$;
6. Assume that the coefficients $\omega_t^\tau(J)$ only depend on $j_t, j_{t+1}, \ldots, j_\tau$.

It is assumed that the summation order can be changed in step 5. □

In particular, an interesting class of random walk estimators results by requiring that the expression between brackets in (7.11) be zero for all t individually:

$$1 = \omega_t^t + \sum_{\tau=t+1}^{\infty} \sum_{j_{t+1},\ldots,j_\tau} p_{j_t j_{t+1}} \cdots p_{j_{\tau-1} j_\tau} (\omega_t^\tau - \omega_t^{\tau-1}) \quad (7.14)$$

A number of important examples are described next. They are illustrated in figure 7.1.

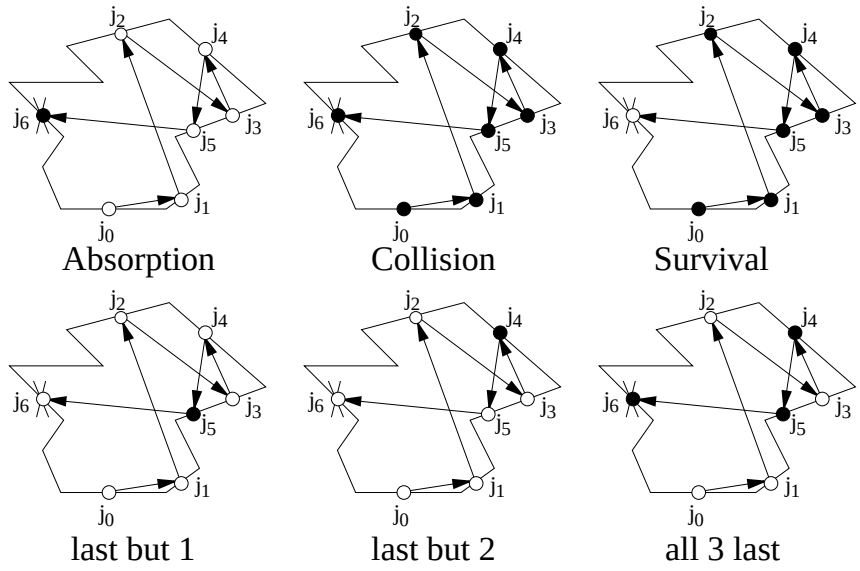


Figure 7.1: Random walk estimators for linear systems: the black dots indicate at what nodes the path yields a contribution. The contribution is nonzero only if e_{j_t} at the node is not zero.

7.2.3 The absorption estimator

The absorption estimator is the oldest random walk estimator for solving linear systems [51]. A contribution to the solution is only recorded at the final hit of the random walk, in other words: at absorption. In order to obtain such an estimator, the coefficients are taken $\omega_t^\tau(J) = 0$ for $\tau \neq t$. With this constraint, (7.14) yields:

$$1 = \omega_t^t(j_t) - \sum_{j_{t+1}} p_{j_t j_{t+1}} \omega_t^t(j_t) = (1 - \sigma_{j_t})\omega_t^t(j_t) \quad \Rightarrow \quad \omega_t^t(j_t) = \frac{1}{\alpha_{j_t}}$$

In full, the resulting *absorption estimator* score is

$$s^A(J) = \frac{q_\tau e_{j_\tau}}{\alpha_{j_\tau}} = \frac{w_{j_0} a_{j_0 j_1} \cdots a_{j_{\tau-1} j_\tau} e_{j_\tau}}{\pi_{j_0} p_{j_0 j_1} \cdots p_{j_{\tau-1} j_\tau} \alpha_{j_\tau}} \tag{7.15}$$

The unbiasedness of this estimator is guaranteed by construction, but can also be checked easily by straightforward calculation of $E[s^A(J)] = \sum_J s^A(J)p(J)$.

7.2.4 The collision estimator

With the collision estimator, first proposed by Wasow [181], a contribution is recorded at every visited state. In order to do so, the coefficients can all be taken $\omega_t^\tau = 1$. Also

with this choice, (7.14) is fulfilled. In full, the *collision estimator* scores are

$$\begin{aligned} s^C(J) &= \sum_{t=1}^{\tau} q_t e_{j_t} \\ &= \frac{w_{j_0}}{\pi_{j_0}} e_{j_0} + \frac{w_{j_0} a_{j_0 j_1}}{\pi_{j_0} p_{j_0 j_1}} e_{j_1} + \ldots + \frac{w_{j_0} a_{j_0 j_1} \cdots a_{j_{\tau-1} j_\tau}}{\pi_{j_0} p_{j_0 j_1} \cdots p_{j_{\tau-1} j_\tau}} e_{j_\tau}. \end{aligned} \quad (7.16)$$

The explicit calculation of $E[s^C(j)]$ is quite more elaborate than for the absorption estimator. The unbiasedness however follows by construction.

7.2.5 The survival estimator

Now we demand that a contribution is only recorded when the path is not terminated on a given state: $\omega_t^\tau \neq 0$ if $\tau \neq t$ and $\omega_t^t = 0$. (7.14) then yields:

$$1 = \sigma_{j_t} \omega_t^{t+1} + \sum_{\tau=t+2}^{\infty} \sum_{j_{t+1}, \ldots, j_\tau} p_{j_t j_{t+1}} \cdots p_{j_{\tau-1} j_\tau} (\omega_t^\tau - \omega_t^{\tau-1})$$

A possible solution is to choose all $\omega_t^\tau, \tau \neq t$ equal. The above expression then simplifies to

$$1 = \sigma_{j_t} \omega_t^{t+1} \quad \Rightarrow \quad \omega_t^\tau = \frac{1}{\sigma_{j_t}} \text{ (if } \tau \neq t\text{)}.$$

In full, the *survival estimator* scores are

$$\begin{aligned} s^S(J) &= \sum_{t=1}^{\tau-1} \frac{q_t e_{j_t}}{\sigma_{j_t}} \\ &= \frac{w_{j_0} e_{j_0}}{\pi_{j_0} \sigma_{j_0}} + \frac{w_{j_0} a_{j_0 j_1} e_{j_1}}{\pi_{j_0} p_{j_0 j_1} \sigma_{j_1}} + \ldots + \frac{w_{j_0} a_{j_0 j_1} \cdots a_{j_{\tau-2} j_{\tau-1}} e_{j_{\tau-1}}}{\pi_{j_0} p_{j_0 j_1} \cdots p_{j_{\tau-2} j_{\tau-1}} \sigma_{j_{\tau-1}}} \end{aligned} \quad (7.17)$$

Also this estimator is unbiased by construction.

7.2.6 Exotic estimators

Theorem 7.1 also leads to a lot more random walk estimators. Some examples, described by Ermakow [45] and Mikhailov [106] are presented here.

Estimators contributing at the last-but-l-th state only

Another family of random walk estimators, complementing the absorption estimator, results by demanding that a contribution is recorded only at the last but l-th visited state ($0 < l <= \tau$) [45]: $\omega_t^\tau \neq 0$ if $\tau = t + l$ and $\omega_t^\tau = 0$ if $\tau \neq t + l$. Filling in this constraint in (7.14) yields:

$$1 = \sum_{j_{t+1}, \ldots, j_{t+l}} p_{j_t j_{t+1}} \cdots p_{j_{t+l-1} j_{t+l}} \alpha_{j_{t+l}} \omega_t^{t+l}(j_t, \ldots, j_{t+l})$$

A simple solution of this equation is:

$$\omega_t^{t+l} = \frac{1}{\sigma_{j_t} \cdots \sigma_{j_{t+l-1}} \alpha_{j_{t+l}}}$$

This leads to the following score at state t for a path of length $\tau = t + l$:

$$s^{E(\tau-l)}(J) = \frac{q_{\tau-l}(J) e_{j_{\tau-l}}}{\sigma_{j_{\tau-l}} \cdots \sigma_{j_{\tau-1}} \alpha_{j_\tau}} \tag{7.18}$$

Estimators contributing at the last $l+1$ states only

Setting $\omega_t^\tau = 0$ for all $t < \tau - l$ and non-zero if $t \geq \tau - l$, leads to a family of estimators contributing a score only to the l last visited states [106, 45]. Filling in $\omega_t^\tau = 0$ if $\tau > t + l$ in (7.14) yields:

$$\begin{aligned} 1 = \omega_t^t \quad &+ \sum_{\tau=t+1}^{t+l-1} \sum_{j_{t+1},\ldots,j_\tau} p_{j_t j_{t+1}} \cdots p_{j_{\tau-1} j_\tau} (\omega_t^\tau - \omega_t^{\tau-1}) \\ &+ \sum_{j_{t+1},\ldots,j_{t+l}} p_{j_t j_{t+1}} \cdots p_{j_{t+l-1} j_{t+l}} (\alpha_{j_{t+l}} \omega_t^{t+l} - \omega_t^{t+l-1}) \end{aligned}$$

A simple solution of this equation is

$$\omega_t^\tau = \begin{cases} 0 & \text{if } t < \tau - l \\ 1/\alpha_{j_\tau} & \text{if } t = \tau - l \\ 1 & \text{if } \tau - l < t \leq \tau \end{cases}$$

The resulting estimator scores are

$$s^{M(\tau-l,\ldots,\tau)}(J) = \frac{q_{\tau-l}(J) e_{j_{\tau-l}}}{\alpha_{j_\tau}} + \sum_{t=\tau-l+1}^{\tau} q_t(J) e_{j_t} \tag{7.19}$$

All these random walk estimators are unbiased by construction.

7.3 Variance of the random walk estimators

We will now turn to the study of the variance $V[s]$ of the random walk estimators with scores $s(J)$ of the form (7.9).

7.3.1 General case

Consider first the score $s_i(J)$ of random walks $J = i, j_1, j_2, \ldots, j_\tau$, originating at i:

$$s_i(j_0, j_1, j_2, \ldots, j_\tau) = \delta_{i j_0} \sum_{t=0}^{\tau} q'(i, j_1, \ldots, j_t) \omega_t^\tau(j_t, j_{t+1}, \ldots, j_\tau) e_{j_t} \tag{7.20}$$

where

$$q'(j_k,\ldots,j_l) = \begin{cases} \frac{a_{j_k j_{k+1}}\ldots a_{j_{l-1}j_l}}{p_{j_k j_{k+1}}\ldots p_{j_{l-1}j_l}} & \text{if } k < l \\ 1 & \text{if } k = l \\ 0 & \text{if } k > l \end{cases}$$

The factors (7.10) correspond with $q_t(J) = \frac{w_{j_0}}{\pi_{j_0}} q'(j_0,\ldots,j_t)$. The expectation of $s_i(J)$ is $E[s_i] = x_i$ by construction.

The expectation $E[s]$ of the scores (7.9) can thus be written as

$$E[s] = \sum_i \pi_i \frac{w_i}{\pi_i} E[s_i].$$

This is a sum of the kind (4.22). According to (4.23), the variance $V[s]$ can be obtained as

$$V[s] = V[E[s_i]] + E[V[s_i]] = \sum_i \frac{w_i^2}{\pi_i}(V[s_i] + x_i^2) - <\mathbf{w},\mathbf{x}>^2 \tag{7.21}$$

The only unknowns in this expression are the variances $V[s_i]$. The following theorem, which is a generalisation of more specific theorems in [63, 45, 93, 135], shows how these can be computed:

Theorem 7.2 *Provided that the coefficients ω_i^τ are the same solution of (7.14) for all t, the variances $V[s_i]$ are the solution of the linear system*

$$V[s_i] = \nu_i + \sum_j \frac{a_{ij}^2}{p_{ij}} V[s_j] - x_i^2 \tag{7.22}$$

with

$$\nu_i = \sum_{j=1}^{n,\Delta} p_{ij} \left(E[\omega_{ij}]\, e_i + \frac{a_{ij}}{p_{ij}} x_j \right)^2 + \sum_{j=1}^{n,\Delta} e_i^2 p_{ij} V[\omega_{ij}] + 2 \sum_{j=1}^{n} e_i a_{ij} Cov[\omega_{ij}, s_j] \tag{7.23}$$

where

- Δ *is a imaginary* death state *with for all* life states $i = 1,\ldots,n$*:*

$$p_{i\Delta} = \alpha_i \quad p_{\Delta\Delta} = 1 \quad p_{\Delta i} = 0$$
$$a_{i\Delta} = a_{\Delta\Delta} = a_{\Delta i} = 0$$

- $\omega_{ij}(J)$ *is a random variable in path space, defined as follows:*

$$\begin{array}{lcll} \omega_{ij}(J) & = & \delta_{ij_0}\delta_{jj_1}\omega_0^\tau(i,j,j_2,\ldots,j_\tau) & \text{if} \quad J = j_0, j_1, \ldots, j_\tau, \tau \geq 2 \\ \omega_{ij}(J) & = & \delta_{ij_0}\delta_{jj_1}\omega_0^1(i,j) & J = j_0 j_1, \tau = 1 \\ \omega_{ij}(J) & = & \delta_{ij_0}\delta_{j\Delta}\omega_0^0(i) & J = j_0, \tau = 0 \end{array}$$

In particular, if for all t and τ, $\omega_t^\tau(j_t, j_{t+1}, \dots, j_\tau) = \omega_{j_t j_{t+1}}$ only depends on the current and next state j_t and j_{t+1} (for $t = \tau$, $j_{t+1} = \Delta$), then

$$\nu_i = \alpha_i(\omega_{i\Delta} e_i)^2 + \sum_{j=1}^{n} p_{ij} \left(\omega_{ij} e_i + \frac{a_{ij}}{p_{ij}} x_j \right)^2 . \tag{7.24}$$

Proof: The score $s_i(J)$ for each random walk $J = i, j, j_2, \dots, j_\tau$ can be written as

$$s_i(i, j_1, j_2, \dots, j_\tau) = \omega_0^\tau(i, j, j_2, \dots, j_\tau) e_i + \frac{a_{ij}}{p_{ij}} s'_{j_1}(j_1, j_2, \dots, j_\tau)$$

with

$$s'_{j_1}(j_1, j_2, \dots, j_\tau) = \sum_{t=1}^{\tau} q'(j_1, j_2, \dots, j_t) \omega_t^\tau(j_t, \dots, j_\tau) e_{j_t} .$$

s'_{j_1} is the score of a random walk $J' = (j_1, j_2, \dots, j_\tau)$ originating at j_1 with coefficients $\tilde{\omega}_{t-1}^{\tau-1}(j_{t+1}, \dots, j_\tau) = \omega_t^\tau(j_t, \dots, j_\tau)$. If ω_t^τ is the same solution of (7.14) for all t, then $\tilde{\omega}_{t-1}^{\tau-1} = \omega_{t-1}^{\tau-1}$, so that $s'_{j_1}(j_1, j_2, \dots, j_\tau) = s_{j_1}(j_1, j_2, \dots, j_\tau)$ and thus

$$s_i(i, j_1, j_2, \dots, j_\tau) = \omega_{ij_1}(i, j_1, j_2, \dots, j_\tau) e_i + \frac{a_{ij_1}}{p_{ij_1}} s_{j_1}(j_1, j_2, \dots, j_\tau).$$

Now consider the score $s_{ij}(i, j_1, j_2, \dots) = \delta_{jj_1} s_i(i, j, j_2, \dots)$ for random walks with fixed origin i and first state $j_1 = j$. Taking into account $p_{i\Delta} = \alpha_i$ and $a_{i\Delta} = 0$ for random walks $J = i$ of length $\tau = 0$, then

$$s_i(J) = \sum_{j=1}^{n,\Delta} p_{ij} s_{ij}(J) \quad ; \quad E[s_i] = \sum_{j=1}^{n,\Delta} p_{ij} E[s_{ij}] = E\left[E[s_{ij}]\right] = x_i$$

The variance $V[s_i]$ can then be obtained by application of (4.23):

$$V[s_i] = V\left[E[s_{ij}]\right] + E\left[V[s_{ij}]\right].$$

Filling in

$$\begin{aligned}
E[s_{ij}] &= E[\omega_{ij}]\, e_i + \frac{a_{ij}}{p_{ij}} E[s_j] = E[\omega_{ij}]\, e_i + \frac{a_{ij}}{p_{ij}} x_j \\
V\left[E[s_{ij}]\right] &= \sum_{j=1}^{n,\Delta} p_{ij} (E[s_{ij}])^2 - \left(\sum_{j=1}^{n,\Delta} p_{ij} E[s_{ij}] \right)^2 = \sum_{j=1}^{n,\Delta} p_{ij} \left(E[\omega_{ij}]\, e_i + \frac{a_{ij}}{p_{ij}} x_j \right)^2 - x_i^2 \\
V[s_{ij}] &= e_i^2 V[\omega_{ij}] + \left(\frac{a_{ij}}{p_{ij}} \right)^2 V[s_j] + 2 e_i \frac{a_{ij}}{p_{ij}} \mathrm{Cov}[\omega_{ij}, s_j] \\
E\left[V[s_{ij}]\right] &= \sum_{j=1}^{n,\Delta} p_{ij} V[s_{ij}]
\end{aligned}$$

yields (7.22) with (7.23). (7.24) follows from (7.23) if $\omega_{ij}(i, j, j_2, \dots)$ does not depend on $j_2, \dots$: in that case, $E[\omega_{ij}] = \omega_{ij}$ depends only on i and j and $V[\omega_{ij}] = \mathrm{Cov}[\omega_{ij}, s_j] = 0$. □

Corrolary 7.1 *The variance of the random walk estimators for linear systems will only be finite if there exists a positive integer number β so that $\|A^{*\beta}\| < 1$, where the matrix A^* has coefficients*

$$a^*_{ij} = \frac{a^2_{ij}}{p_{ij}}.$$

Proof: This condition is a necessary condition for the solvability of the set of linear systems (7.22). If all ν_i are finite, it is also a sufficient condition. □
Halton and Ermakow [64, 45] describe examples of linear systems for which the Neumann series expansion of the solution converges, but for which the condition described above cannot be fulfilled. Random walk estimators for such systems still are unbiased, but there is no guarantee that by sampling more random walks, a better estimate for the solution is ever obtained.

7.3.2 Variance of the absorption, collision and survival estimators

Corrolary 7.2 *Under the conditions of (7.24), the variances $V[s_i]$ can also be obtained as $V[s_i] = v_i - x_i^2$, where v_i is the solution of*

$$v_i = \mu_i + \sum_j \frac{a^2_{ij}}{p_{ij}} v_j \text{ with: } \mu_i = e_i^2 \left(\alpha_i \omega^2_{i\Delta} + \sum_{j=1}^{n} p_{ij} \omega^2_{ij} \right) + 2e_i \sum_{j=1}^{n} a_{ij} \omega_{ij} x_j. \tag{7.25}$$

Proof: By expansion of (7.22) with source term (7.24). □
This corrolary leads to the following results for the absorption, survival and collision estimators:

- Absorption estimator (7.15): $\omega_{i\Delta} = 1/\alpha_i$ and $\omega_{ij} = 0$ if $j \neq \Delta$:

$$\mu_i = \frac{e_i^2}{\alpha_i} \; ; \; v_i = \sum_s \frac{e_s}{\alpha_s} y_{is} \; ; \; y_{is} = \delta_{is} e_s + \frac{a^2_{is}}{p_{is}} e_s + \sum_{j_1} \frac{a^2_{ij_1}}{p_{ij_1}} \frac{a^2_{j_1 s}}{p_{j_1 s}} e_s + \cdots \tag{7.26}$$

- Collision estimator (7.16): $\omega_{ij} = 1$ for all $j = 1, \ldots, n, \Delta$:

$$\mu_i = e_i^2 + 2e_i \sum_{j=1}^{n} a_{ij} x_j = e_i(2x_i - e_i) \quad ; \quad v_i = \sum_s (2x_s - e_s) y_{is} \tag{7.27}$$

- Survival estimator (7.17): $\omega_{i\Delta} = 0$ and $\omega_{ij} = 1/\sigma_i$ if $j \neq \Delta$:

$$\mu_i = e_i \frac{2x_i - e_i}{\sigma_i} \quad ; \quad v_i = \sum_s \frac{2x_s - e_s}{\sigma_s} y_{is} \tag{7.28}$$

7.3.3 Source term estimation suppression

The random walk estimators above estimate scalar products $< \mathbf{w}, \mathbf{x} >$ with the full solution of $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$. Also the — known — zeroth order term $< \mathbf{w}, \mathbf{e} >$ is estimated. A slight variance reduction at no extra cost can be achieved by only estimating the — unknown — higher order terms $< \mathbf{w}, \mathbf{A}\mathbf{e} + \mathbf{A}^2\mathbf{e} + \cdots >= < \mathbf{w}, \mathbf{x} - \mathbf{e} >$. The known term $< \mathbf{w}, \mathbf{e} >$ is then added afterwards, without variance.

This effect can be obtained by modifying a "normal" random walk estimator, constructed as outlined above, as follows:

1. All traced paths are forced to have length $\tau \geq 1$ by suppressing absorption at the origin. The probability associated with each path becomes:

$$\tilde{p}(j_0, j_1, j_2, \ldots, j_\tau) = \pi_{j_0} \frac{p_{j_0 j_1}}{\sigma_{j_0}} p_{j_1 j_2} \ldots p_{j_{\tau-1} j_\tau} \alpha_{j_\tau} = \frac{1}{\sigma_{j_0}} p(j_0, \ldots, j_\tau);$$

2. The scores (7.9) are modified by omitting the contribution at the origin $t = 0$ and compensating for the modified probabilities by multiplying with σ_{j_0}:

$$\tilde{s}(J) = \sigma_{j_0} \sum_{t=1}^{\tau} q_t(J) \omega_t^\tau(J) e_{j_t}$$

If the coefficients ω_t^τ are constructed according to theorem 7.1, unbiased estimators for $< \mathbf{w}, \mathbf{x} - \mathbf{e} >$ result. Following a similar reasoning as in the proof of theorem 7.2, it is easy to show that the variances $V[\tilde{s}_i]$ for such random walks originating at i are related with the variances $V[s_j]$ of the unmodified random walk estimators like:

$$V[\tilde{s}_i] = \sigma_i \sum_{j=1}^{n} \frac{a_{ij}^2}{p_{ij}} \left(x_j^2 + V[s_j] \right) - (x_i - e_i)^2 \tag{7.29}$$

In case corrolary 7.2 is applicable, we obtain

$$V[\tilde{s}_i] = \sigma_i \sum_{j=1}^{n} \frac{a_{ij}^2}{p_{ij}} v_j - (x_i - e_i)^2 \; ; \; V[\tilde{s}] = \sum_{i=1}^{n} \frac{w_i^2 \sigma_i}{\pi_i} \sum_{j=1}^{n} \frac{a_{ij}^2}{p_{ij}} v_j - < \mathbf{w}, \mathbf{x} - \mathbf{e} >^2 \tag{7.30}$$

with v_i as defined in (7.25).

7.4 Random walk estimators for radiosity

Due to our assumption (7.10)[3] the theory of the §7.2 and §7.3 can be applied to either the classical radiosity system (4.25) or the adjoint power system (4.32).

[3] It is possible to derive shooting instead of gathering estimators directly, by assuming a different general form for the random walk scores in §7.2.2, but the derivation of the gathering estimators is more intuitive and elegant.

7.4.1 Gathering random walk radiosity estimators

When the general framework of the previous sections is applied to the classical radiosity system (4.25)

$$B_i = E_i + \sum_j \rho_i F_{ij} B_j \ ,$$

several *gathering* random walk estimators for radiosity are obtained. These estimators compute the radiosity B_k at a fixed patch k by tracing random walks originating at k. A contribution to B_k is recorded when the random walk hits one or more light sources ($E_i \neq 0$). The equations to be solved are $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$ with

- $e_i = E_i$, the self-emitted radiosity, and $x_i = B_i$, the total radiosity;
- $a_{ij} = \rho_i F_{ij}$;
- $p_{ij} = \rho_i F_{ij}$: when a particle hits a patch, first a decision is made to absorb it (probability $1 - \rho_i$) or to reflect it (probability ρ_i). Next, a uniformly distributed random line through i is traced and the nearest hit patch j determined;
- $w_i = \delta_{ik}$: we are computing the radiosity of a single patch k.
- $\pi_i = \delta_{ik}$: we will do so by tracing random walks all originating at k.

With these choices,

- $\sigma_i = \rho_i$, the reflectivity, and $\alpha_i = 1 - \rho_i$;
- $q_t(J) = 1$ for all t;
- $y_{is} = b_{is}$, the radiosity at patch i due to light source s.

$x_i - e_i$ corresponds with b_i, the total received radiosity at i. The resulting gathering random walk radiosity estimators, with source term suppression, are shown in table 7.1. The resulting algorithm for the collision gathering estimator was shown before (algorithm 18). The algorithms for the other estimators are very similar.

estimator	score $\tilde{s}(j_0 = k, \ldots, j_\tau) = \tilde{s}_k(J)$	variance $V[\tilde{s}_k]$
absorption	$\rho_k \frac{E_{j_\tau}}{1-\rho_{j_\tau}}$	$\rho_k \sum_s \frac{E_s}{1-\rho_s} b_{ks} - b_k^2$
collision	$\rho_k \sum_{t=1}^{\tau} E_{j_t}$	$\rho_k \sum_s (E_s + 2b_s) b_{ks} - b_k^2$
survival	$\rho_k \sum_{t=1}^{\tau-1} \frac{E_{j_t}}{\rho_{j_t}}$	$\rho_k \sum_s \frac{E_s + 2b_s}{\rho_s} b_{ks} - b_k^2$

Table 7.1: Gathering random walk estimators for radiosity. The expectation is $b_k = B_k - E_k$, the radiosity received on patch k.

7.4.2 Shooting random walk radiosity estimators

Random walks with transition probabilities $p_{ij} = \rho_i F_{ij}$, used in the gathering random walk above, can also be used in order to solve the following, slightly modified, adjoint power system (4.32):

$$(\rho_i I_i) = (\rho_i V_i) + \sum_j \rho_i F_{ij} (\rho_j I_j).$$

If the importance source term V_i is chosen $V_i = \delta_{ik}$, the radiosity B_k at a patch k can be obtained by the scalar product

$$B_k = \frac{1}{A_k} < \frac{\Phi}{\rho}, (\rho I) >= \sum_i \frac{\Phi_i}{A_k \rho_i} (\rho_i I_i).$$

This scalar product can be estimated with random walks that gather the importance I_i at the light sources from the source of importance, the patch k. In order to do so, we set:

- $e_i = \rho_i V_i = \rho_k \delta_{ik}$, the self-emitted importance, and $x_i = \rho_i I_i$;
- $a_{ij} = p_{ij} = \rho_i F_{ij}$; as for the gathering random walk;
- $w_i = \frac{\Phi_i}{A_k \rho_i}$;
- $\pi_i = \Phi_i / \Phi_T$: the probability of starting a path at a light source is chosen proportional to the self-emitted power of the light source.

These choices imply

- $\sigma_i = \rho_i$, the reflectivity, and $\alpha_i = 1 - \rho_i$;
- $q_t(J) = \frac{\Phi_T}{A_k \rho_i}$;
- $y_{is} = \delta_{sk} \rho_i I_i$;

The resulting shooting random walk estimators for radiosity, with source term suppression, are summarised in table 7.2. The collision shooting algorithm was shown before (algorithm 19). The algorithms for the other estimators are again very similar.

estimator	score $\tilde{s}(j_0, \ldots, j_\tau)$	variance $V[\tilde{s}]$
absorption	$\frac{\rho_k}{A_k} \frac{\Phi_T}{1-\rho_k} \delta_{j_\tau k}$	$\frac{\rho_k}{A_k} \frac{\Phi_T}{1-\rho_k} b_k - b_k^2$
collision	$\frac{\rho_k}{A_k} \Phi_T \sum_{t=1}^{\tau} \delta_{j_t k}$	$\frac{\rho_k}{A_k} \Phi_T (1 + 2\zeta_k) b_k - b_k^2$
survival	$\frac{1}{A_k} \Phi_T \sum_{t=1}^{\tau-1} \delta_{j_t k}$	$\frac{1}{A_k} \Phi_T (1 + 2\zeta_k) b_k - b_k^2$

Table 7.2: Shooting random walk estimators for radiosity. The expectation is $b_k = B_k - E_k$. If k would be the only source of radiosity, with unit strength, the total radiosity on k would be I_k. $\zeta_k = I_k - V_k = I_k - 1$ would be its received radiosity. ζ_k corresponds to $R_k \xi_k$ in [131, 132].

7.4.3 Comparison of random walk estimators for radiosity

The variance formulae in tables 7.1 and 7.2 allow to compare the efficiency of the random walk estimators [132, 131]:

- For gathering as well as shooting, the *survival estimator always has higher variance than the collision estimator*, due to the reflectivities in the denominator. Indeed: unlike the survival estimator, the collision estimator also yields a contribution when a particle is absorbed at k;
- Whether the collision estimator is better than the absorption estimator depends on the relative magnitude of
  - $2\zeta_k$ versus $\rho_k/(1-\rho_k)$ for shooting. In practice, the fraction of radiosity received back from it self is negligible, so that the collision estimator will be better;
  - $2b_s b_{ks}$ versus $b_{ks}E_s\rho_s/(1-\rho_s)$ on each light source s for gathering. Since the self-emitted radiosity E_s is generally much larger than b_s, the radiosity received at a light source, also for gathering, the collision estimator will generally be more efficient.

  We conclude that the collision estimator will generally be more efficient than the absorption estimator. This result should not surprise us: the collision estimator yields a contribution each time a particle hits a target patch, regardless of whether the particle is scattered or absorbed. The absorption estimator only yields a contribution when the particle is absorbed.

  The result that the collision estimator is more efficient than the absorption estimator is not valid in general: there are important cases in which the absorption estimator will be preferable. For instance, we shall see in §9 that the absorption estimator can be a perfect estimator for finite length paths, while the collision estimator can't.

We conclude that among the proposed analog radiosity estimators, the collision estimator will generally be the most efficient of all.

Whether the gathering estimators are better than the shooting estimators depends on the radiosity distribution in a scene and the surface area of the patch k on which the radiosity is computed. For sufficiently large patches, the shooting estimator will definitely be more efficient since the area appears in the denominator of the variance formulae of all shooting estimators (table 7.2). For sufficiently small patches, the gathering estimator will be better, since its variance does not depend on the surface area A_k (table 7.1). In practice, *the variance of the shooting estimator can be much lower, except on the smallest patches* of a scene.

The variance of the random walk estimators for radiosity, proposed above, is always finite: the matrix A^* mentioned in corrolary 7.1 has coefficients

$$a^*_{ij} = \frac{a^2_{ij}}{p_{ij}} = \rho_i F_{ij}.$$

Since $\sum_j \rho_i F_{ij} \leq \rho_i < 1$, the row-norm $\|A^*\| < 1$ so that the condition of corrolary 7.1 is fulfilled [135]. This may not be the case with other choices for the probabilities, for instance when taking survival probabilities $\sigma_i = \rho_{\text{av}}$ as in [88].

7.4.4 Collision shooting random walk versus stochastic Jacobi relaxation

(Almost) the same accuracy for the same amount of work

The variance of the collision shooting random walk estimator with N^{RW} random walks, is given by (table 7.2):

$$\frac{V^{RW}}{N^{RW}} = \frac{1}{N^{RW}} \left(\frac{\rho_k}{A_k} \Phi_T (1 + 2\zeta_k) b_k - b_k^2 \right). \tag{7.31}$$

The variance of the regular stochastic Jacobi relaxation estimator (expression (6.13), divided by A_k^2), with N^{SR} samples, is given by:

$$\frac{V^{SR}}{N^{SR}} = \frac{1}{N^{SR}} \left(\frac{\rho_k}{A_k} P_T b_k - b_k^2 \right). \tag{7.32}$$

In both cases, the subtracted term $-b_k^2$ is generally much smaller than the first term between brackets and can be ignored. Also the recurrent radiosity $\zeta_k \ll 1$ in practice. Moreover, it can be shown that the average number of rays to be traced for sampling N^{RW} random walks equals $N^{RW} P_T / \Phi_T$, so that *for same number of rays* $N^{SR} = N^{RW} P_T / \Phi_T$*, the variance of the collision shooting estimator and stochastic Jacobi iterations will be approximately equal* in practice:

$$\frac{V^{RW}}{N^{RW}} \approx \frac{1}{N^{RW}} \frac{\rho_k}{A_k} \Phi_T b_k = \frac{1}{N^{SR}} \frac{\rho_k}{A_k} P_T b_k \approx \frac{V^{SR}}{N^{SR}}. \tag{7.33}$$

This result has been confirmed in experiments and also appears to hold when low-discrepancy sampling is used instead of pseudo-random sampling. This will be illustrated in §12.3.2, when discussing the effectiveness of low-discrepancy sampling. It is also valid when comparing with a full sequence of incremental Jacobi iterations until convergence (variance expression (6.17) instead of (6.13)).

A first consequence is that the heuristic for determining the number of rays in a first stochastic Jacobi iteration (§6.4.5) can also be used in order to determine the number of random walks in collision shooting random walk radiosity. In our implementation, we have traced random walks in groups. After tracing a group of random walks, an intermediate image is shown. The result from different groups was merged as explained in §6.6.1 [136].

Another consequence is that the time-complexity of collision shooting radiosity random walks, and of stochastic relaxation radiosity is identical as well ($\mathcal{O}(n \log n)$ [131, 132]).

Intuitive explanation Both stochastic Jacobi iterative methods (§6.4) and discrete shooting collision random walk radiosity have an intuitive interpretation in the sense of particles being shot from patches. The particles have uniform starting position on the patches and they have cosine-distributed directions w.r.t. the normal on the patches. The number of particles shot from each patch is proportional to the power propagated from the patch. Since the three methods compute the same result, the same number of particles will be shot from each of the patches. If also the same random numbers are used to shoot particles from each patch, the particles themselves can also

be expected to be the same. This argument is not entirely correct. Besides a difference in the order in which the particles are shot (see figure 7.2), there is also a difference in the decisions whether a particle will be absorbed or will survive on the patch. The effect of multiple particles is averaged in a different way as well. The latter is probably the main cause for the (slight) difference in variance for the same number of rays.

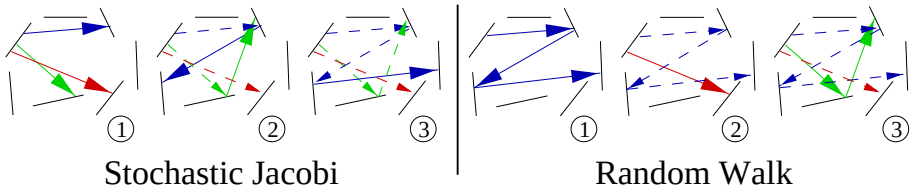


Figure 7.2: This figure illustrates the difference in order in which particles are shot in stochastic Jacobi iterations ("breadth-first" order) and in collision shooting random walk radiosity ("depth-first" order). Eventually, the shot particles are very similar. Survival decisions and in particular also the way how the results from multiple particles are averaged, are different in these algorithms.

Validation in an extreme case

The expressions (7.31) and (7.32) for same number of rays $N^{SR} = N^{RW} P_T/\Phi_T$, can be validated by measuring and calculating the mean square error in a simple environment. It should be possible to compute the variances in the environment fully analytically and the recurrent radiosity ζ_k should not be negligible. The environment we have used consists of an empty cube with unit sides (shown in figure 5.2 on page 80). The reflectivity ρ and emissivity E are chosen equal for all surfaces, so that $E + \rho = 1$. A variety of environments can be created by letting ρ take different values. The values $\rho = 0.1, 1/3, 0.5, 2/3, 0.9$, and 0.95 have been used in the comparison. The results of the comparison are shown in table 7.3.

ρ	1/10	1/3	1/2	2/3	9/10	19/20
ζ	0.00218	0.03125	0.0909	0.2353	1.3729	3.0336
RW (theory)	0.04915	0.54167	1.2727	2.588	10.076	20.218
RW (experiment)	0.0477	0.532	1.236	2.639	10.55	20.088
SR (theory)	0.05	0.5555	1.25	2.2222	4.05	4.5125
SR (experiment)	0.0500	0.554	1.198	2.180	3.887	4.323

Table 7.3: Observed mean square error (MSE) per ray for a selected patch in a homogeneous cube with unit sides and $\rho + E = 1$. There is a good correspondence between the empirical and the predicted values for both the (regular) stochastic Jacobi method and the collision shooting random walk. RW=Random Walk, SR=Stochastic Jacobi.

Empirical results The reported empirical results are the average mean square error per ray observed on a selected patch. They are the average MSE, after a sufficiently high number of runs (more than 20,000 in all cases) with $N^{SR} = 10000$ rays per run,

multiplied with N^{SR}. The result for the stochastic Jacobi iterative method have been obtained with regular iterations. The analytical solution $B = 1$ was used as input for these iterations. The experiment has also been repeated with the (inexact) result of a previous regular stochastic Jacobi iteration as the input for the next iteration. The observed mean square errors were not significantly different.

Theoretical results The theoretical values in table 7.3 are obtained as follows:

- $A_k = 1$, the total area $A_T = 6$;
- $\Phi_T = 6E = 6(1 - \rho)$;
- The total radiosity $B_k = 1$ for all patches. When $E_k + \rho_k = 1$ in a closed environment, the total radiosity B_k also is always equal to 1.
  Proof: Fill in $B_i = 1$ in the radiosity equations $B_i = E_i + \rho_i \sum_j F_{ij} B_j$. In a closed environment, $\sum_j F_{ij} = 1$. The solution is unique. □
- $b_k = \rho$;
- $P_T = \Phi_T/(1 - \rho)$, so that N^{RW} must be chosen $N^{SR}(1 - \rho)$.

With these choices, the expressions (7.31) and (7.32) become:

$$\begin{aligned} \frac{V^{RW}}{N^{RW}} &= \frac{1}{N^{SR}} \left[6\rho^2(1 + 2\zeta) - \frac{\rho^2}{1 - \rho} \right] \\ \frac{V^{SR}}{N^{SR}} &= \frac{1}{N^{SR}} 5\rho^2 \end{aligned}$$

$\zeta = I_k - V_k$ was determined by solving

$$I_i = V_i + \sum_j F_{ij} \rho_j I_j$$

analytically with $V_i = \delta_{ik}$, taking values 0.2 for the form factors between different patches. The true values of the form factors are 0.200043 for abutting patches and 0.1998 for parallel patches in the cube. This yields the following solution for ζ:

$$\zeta = \frac{0.2\rho^2}{1 - 0.2\rho(4 + \rho)}.$$

Filling in these values for ζ leads to the theoretical values shown in the table.

The correspondence between theory and experiment is quite good. The reader should realise however that the environment used for this validation experiment is a very untypical environment in practice. All experiments with "real" environments confirm the result mentioned in the beginning of this section: the observed statistical error for same number of rays is nearly identical regardless of whether regular or incremental stochastic Jacobi iterations, or collision shooting random walks are used (see §12.3.2).

7.5 Conclusion

In this chapter, random walk estimators for systems of linear equations have been described and their application to radiosity discussed. Unlike other texts on random walk solution of linear systems [66, 156, 64, 45, 127], a fully "constructive" approach has been proposed: it has been shown how unbiased random walk estimators can be designed and a new, very general theorem for their variance has been proved. The issue of source term estimation suppression, a rather unclear issue in many other texts on the random walk solution of linear systems, has been clarified.

The theoretical exposition allowed to re-derive some previously known results concerning the absorption, collision and survival estimator for radiosity [132, 131] in a very compact manner. The exposition here also emphasizes the duality between gathering and shooting estimators for radiosity. New, practical results in this chapter include:

- The comparison of discrete and continuous random walk methods in §7.1.5, showing that the discretisation error in both is nearly the same;
- The comparison of collision shooting random walk radiosity and the stochastic Jacobi iterative method, showing that in practice, the obtained accuracy for a given amount of work will be approximately equal (§7.4.4). Except on small patches, the collision shooting random walk estimator is generally the most efficient random walk estimator for radiosity (§7.4.3, [132, 131]);

The derived results on the variance of the absorption, collision and survival random walk estimators for radiosity will be used in further chapters, in order to analyse variance reduction techniques.

A possible area for future research is the analysis of the more exotic random walk estimators presented in §7.2.6. Preliminary experiments have shown that these estimators do not compete as such with other, better known, random walk estimators. They may however be useful in the context of advanced importance-sampling algorithms (chapter 9), because they yield a fixed number of contributions per path and so may result in perfect estimators.

8 Other Monte Carlo Methods for Linear Systems

In this chapter, a number of more exotic Monte Carlo techniques for the solution of linear systems are described. Their application for radiosity may be an interesting topic for further research.

8.1 Shreider's estimators

The random walk method described in the previous section can be used only for systems of equations $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$ for which the Neumann expansion of the solution $\mathbf{x}$ converges. This will be the case if $\|\mathbf{A}^k\| < 1$ for some positive integer k. It is — at least formally — possible to transform any linear system to a system for which this requirement is fulfilled. In order to have a finite variance however, the modified coefficient matrix $\mathbf{A}^* = \{a_{ij}^2/p_{ij}\}$ should fulfil $\|[\mathbf{A}^*]^m\| < 1$ for some positive integer m. Unfortunately, there exist linear systems satisfying the former requirement for which no transition probabilities p_{ij} can be found so that the latter requirement is fulfilled as well [64]. For such systems, random walk estimators won't work. Also stochastic relaxation methods cannot be applied for all possible linear systems: they have the same requirements as their deterministic counterparts.

Shreider [146, 17] proposed an entirely different Monte Carlo procedure in order to solve any linear system having a unique solution. The basic idea of Shreiders method is to consider to solution $\mathbf{x}$ of such a linear system $\mathbf{C}\mathbf{x} = \mathbf{e}$ with n equations as the centre of a n-dimensional ellipsoid:

$$\Psi(\mathbf{x}) = \Psi(x_1, \ldots, x_n) = \sum_{i=1}^{n} f_i \left(\sum_{j=1}^{n} a_{ij} x_j - e_i \right)^2 < R.$$

$f_1, \ldots, f_n$ are arbitrary positive numbers. R determines the size of the ellipsoid. The problem of finding the solution of the linear system is transformed to the problem of finding the centre of this ellipsoid. Two approaches have been proposed:

1. **Median method:** N uniform random n-dimensional points $\mathbf{x}^{(k)}$, inside the ellipsoid, are sampled. This can be done by rejection sampling: random uniform points are sampled in a (n-dimensional) bounding box for the ellipsoid. The points are rejected if $\Psi(\mathbf{x}^{(k)}) > R$. The median of the retained points is an estimate for the centre of the ellipsoid, and thus also for the solution of the linear system;

2. **Centre-of-Gaussian method:** The components x_k of the solution are also the expectation $x_k = E[y_k]$ of the coordinates y_k of n-dimensional points $\mathbf{y}$ with

Gaussian distribution $\exp(-\Psi(\mathbf{y}))$:

$$x_k = \int y_k e^{-\Psi(\mathbf{y})} dy_1 \cdots dy_n.$$

N uniform random points $\mathbf{x}^{(k)}$ are sampled in a n-dimensional bounding box for the ellipsoid. As an estimate for the solution,

$$x_i \approx \frac{\sum_{k=1}^{N} x_i^{(k)} e^{-\psi(\mathbf{x}^{(k)})}}{\sum_{k=1}^{N} e^{-\psi(\mathbf{x}^{(k)})}}$$

is taken. There exist also more complex distributions that have the same property, but may lead to smaller variance.

The main advantage of the second technique over the first, is that all n-dimensional samples are used. The number of rejected samples in the first technique can be very high if the bounding box for the ellipsoid is not properly chosen.

In both cases however, the quadratic form $\Psi(\mathbf{x}^{(k)})$ needs to be evaluated for each sample point $\mathbf{x}^{(k)}$. Computation of this quadratic form takes $\mathcal{O}(n^2)$ work. In order to be competitive with other methods for radiosity, cheap approximations for $\Psi(\mathbf{x})$ will need to be developed.

8.2 Dimov's acceleration technique

Recently, Dimov [37, 36] has presented a mapping technique that can increase the efficiency of random walk estimators. The basic idea is to transform the Neumann series expansion of the solution $\mathbf{x}$ of $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$ into an equivalent series that converges more rapidly, so that good approximations for the solution can be obtained by sampling only a small number of terms of the new series. The result is a reduction of the average length of the random walks.

A mapping was developed for matrices $\mathbf{A}$ that have negative, real, eigenvalues. Unfortunately, this is not the case for the radiosity equations. The construction of a new mapping, that will be suited for the radiosity problem, is another potential area for future research.

9 Importance-Driven Monte Carlo Radiosity

In chapter 6 and 7, an overview has been given of the basic estimators for solving the system of radiosity equations. In this chapter and the next chapters, the application of a number of variance reduction techniques to these basic estimators will be discussed. This chapter will deal with the application of importance sampling.

Importance sampling in random walk methods is a well-studied topic in Monte Carlo literature [82, 1, 33, 156, 85]. It has been applied to continuous random walk estimators for radiosity in [122, 44]. Its application to discrete random walk radiosity has been studied by Sbert et al. [135] and verified in the context of this dissertation. The main new result in this chapter concerns the application of view-importance sampling in stochastic relaxation radiosity, developed in the context of this dissertation in collaboration with L. and A. Neumann and J. Prikryl. The present discussion is a completely revised and significantly extended version of [108].

In chapter 7, the emphasis was on the derivation of good random walk estimator scores for given birth and transition probabilities π_i and p_{ij}. In the estimators shown in §7.4, transitions according to the form factor probabilities F_{ij} are used. Such an approach works well in case one has equal interest in all parts of a scene. By using importance sampling, it is possible to derive better birth and transitions probabilities in case the accurate computation of radiosity is more important in some region of a scene than in other regions.

9.1 Importance sampling in random walk methods

A random walk is characterised by its

- path creation, or *birth* probability distribution π_i, that expresses the probability of selecting the origin of a random walk in a given state (patch) i;
- transition probability distribution p_{ij}, that expresses the probability that a random walk, currently in state i, will make a transition to another state j.

It is convenient to consider also some derived probability distributions:

- survival probability distribution $\sigma_i = \sum_j p_{ij}$, expressing the probability that a random walk continues after a collision at state i;
- absorption, or *death* probability distribution $\alpha_i = 1 - \sigma_i = 1 - \sum_j p_{ij}$, expressing the probability that a random walk is terminated after a collision at state i.

These probability distributions fulfil certain requirements (see §7.2.1). They affect the variance $V[s]$ of a random walk estimator $s(J)$ of the form (7.9) for the scalar product

$< \mathbf{w}, \mathbf{x} >$ with $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$ in the following ways (see §7.3):

$$V[s] = \sum_i \frac{w_i^2}{\pi_i}(V[s_i] + x_i^2) - < \mathbf{w}, \mathbf{x} >^2 \tag{9.1}$$

$$V[s_i] = \nu_i + \sum_j \frac{a_{ij}^2}{p_{ij}} V[s_j] - x_i^2 \tag{9.2}$$

$$\nu_i = \alpha_i(\omega_{i\Delta} e_i)^2 + \sum_{j=1}^{n} p_{ij} \left(\omega_{ij} e_i + \frac{a_{ij}}{p_{ij}} x_j \right)^2 \tag{9.3}$$

The discussion here is limited to the (common) absorption, collision and survival estimators. When source term estimation is suppressed (§7.3.3), the variance is

$$V[\tilde{s}] = \sum_{i=1}^{n} \frac{w_i^2}{\pi_i} \sigma_i \sum_{j=1}^{n} \frac{a_{ij}^2}{p_{ij}} (V[s_j] + x_j^2) - < \mathbf{w}, \mathbf{x} - \mathbf{e} >^2$$

We now investigate what choice of the transition and birth probabilities will lead to minimal variance. We assume that all a_{ij} and x_i and e_i are positive or zero.

9.1.1 Optimal transition probabilities

The optimal transition probabilities can be found by minimising the variance source term $\nu_i - x_i^2$ in (9.2) for each estimator:

$$\nu_i - x_i^2 = \alpha_i(\omega_{i\Delta} e_i)^2 + \sum_{j=1}^{n} p_{ij} \left(\omega_{ij} e_i + \frac{a_{ij}}{p_{ij}} x_j \right)^2 - x_i^2. \tag{9.4}$$

The absorption estimator

For the absorption estimator (7.15), $\omega_{i\Delta} = 1/\alpha_i$ and $\omega_{ij} = 0$ if $j \neq \Delta$, so that (9.4) becomes:

$$\frac{1}{\alpha_i} e_i^2 + \sum_j \frac{a_{ij}^2 x_j^2}{p_{ij}} - x_i^2$$

Minimisation of this expression using the technique of Lagrange multipliers with constraint $\alpha_i + \sum_j p_{ij} = 1$, yields:

$$\begin{aligned} p_{ij} &= \frac{a_{ij} x_j}{e_i + \sum_j a_{ij} x_j} = \frac{a_{ij} x_j}{x_i} \\ \alpha_i &= \frac{e_i}{e_i + \sum_j a_{ij} x_j} = \frac{e_i}{x_i}. \end{aligned} \tag{9.5}$$

It is easy to verify that with this choice for all i, all $V[s_i] = 0$. Whether also $V[s] = 0$ depends on the birth probabilities, but it will be shown below that there exists a choice for these that makes $V[s] = 0$, resulting in a first perfect random walk estimator. As usual for a perfect estimator, knowledge of the solution is required in the probability distribution, so that a perfect estimator can only be approximated in practice. An interpretation for the perfect absorption estimator will be given below in §9.1.3.

The collision and survival estimator

In case of the collision estimator (7.16), all $\omega_{ij} = \omega_{i\Delta} = 1$, so that

$$\alpha_i e_i^2 + \sum_{j=1}^{n} p_{ij}\left(e_i + \frac{a_{ij}}{p_{ij}}x_j\right)^2 - x_i^2 = e_i^2\left(\alpha_i + \sum_j p_{ij}\right) + 2e_i\sum_j a_{ij}x_j + \sum_j \frac{a_{ij}^2 x_j^2}{p_{ij}} - x_i^2$$

needs to be minimised with constraint $\alpha_i + \sum_j p_{ij} = 1$. Unfortunately, no solution is found with $\alpha_i \neq 0$. If $\alpha_i = 0$, then we find

$$\begin{aligned} p_{ij} &= \frac{a_{ij}x_j}{\sum_j a_{ij}x_j} = \frac{a_{ij}x_j}{x_i - e_i} \\ \alpha_i &= 0 \end{aligned} \tag{9.6}$$

These choices for all i, again result in $V[s_i] = 0, \forall i$. The resulting perfect collision estimator however is an infinite random walk estimator, which definitely has a higher cost than the finite perfect absorption estimator. It is an example of a situation where the absorption estimator will be preferred over the collision estimator.

The perfect survival estimator is identical to the perfect collision estimator. This can be shown by deriving the optimal transition probabilities by minimising (9.4) with $\omega_{ij} = 1/\sigma_i$ and $\omega_{i\Delta} = 0$ as show above. This is not so surprising: a infinite collision estimator is a (infinite) survival estimator.

9.1.2 Optimal birth probabilities

The optimal birth probabilities π_i can be determined by minimising (9.1) (or alternatively (9.1) when source term suppression is used) with constraint $\sum_i \pi_i = 1$. For the general case $V[s_i] \neq 0$, we find:

$$\pi_i^2 \propto w_i^2(V[s_i] + x_i^2) \tag{9.7}$$

In particular, when $V[s_i] = 0$, we obtain as optimal birth probabilities:

$$\pi_i = \frac{w_i x_i}{\sum_k w_k x_k} \tag{9.8}$$

The resulting estimator $s(J) = \sum_i s_i(J)$ is then also perfect. Indeed, (9.1) then becomes:

$$\begin{aligned} V[s] &= \sum_i \frac{w_i^2}{\pi_i}(V[s_i] + x_i^2) - <\mathbf{w},\mathbf{x}>^2 \\ &= \sum_i \frac{w_i^2 x_i^2}{\frac{w_i x_i}{\sum_k w_k x_k}} - <\mathbf{w},\mathbf{x}>^2 \\ &= \left(\sum_i w_i x_i\right)^2 - <\mathbf{w},\mathbf{x}>^2 \quad = \quad 0. \end{aligned}$$

Also the optimal birth probabilities require a-priori knowledge of the solution.

9.1.3 Interpretation of the perfect random walk estimators

It is interesting to study the score $s(J)$ of a random walk with perfect birth probabilities (9.8) and transition/absorption probabilities (9.5):

$$\begin{aligned} s(j_0, j_1, \ldots, j_\tau) &= \frac{w_{j_0}}{\pi_{j_0}} \frac{a_{j_0 j_1}}{p_{j_0 j_1}} \cdots \frac{a_{j_{\tau-1} j_\tau}}{p_{j_{\tau-1} j_\tau}} \cdot \frac{1}{\alpha_{j_\tau}} \cdot e_{j_\tau} \\ &= \frac{w_{j_0} \sum_k w_k x_k}{w_{j_0} x_{j_0}} \frac{a_{j_0 j_1} x_{j_0}}{a_{j_0 j_1} x_{j_1}} \cdots \frac{a_{j_{\tau-1} j_\tau} x_{j_{\tau-1}}}{a_{j_{\tau-1} j_\tau} x_{j_\tau}} \cdot \frac{x_{j_\tau}}{e_{j_\tau}} \cdot e_{j_\tau} \\ &= \sum_k w_k x_k \quad = \quad < \mathbf{w}, \mathbf{x} > \end{aligned}$$

The score of any arbitrary random walk with given birth and transition probabilities equals the scalar product $< \mathbf{w}, \mathbf{x} >$, to be computed. Note that this random walk estimator has $\alpha_i = 0$ unless $e_i \neq 0$: a random walk will continue until a source is hit. When a source is hit, the random walk may be terminated and the score $< \mathbf{w}, \mathbf{x} >$ returned regardless of what source is hit. As usual in Monte Carlo, the effect of all other sources is brought into account by the fact that any source *could* have been hit.

It is harder to verify in this way that the optimal collision estimator is a perfect estimator. This is however so by construction. Also, a finite collision random walk estimator can not be perfect, because the number of contributions is not constant [85].

9.1.4 Approximation of perfect random walk estimators

In practice, the solution is never known in advance. It is however possible to obtain a significant variance reduction by using an approximate solution $\tilde{\mathbf{x}} \approx \mathbf{x}$ instead of the real solution $\mathbf{x}$ in the above formulae. This approximate solution needs to fulfil the following requirements. These requirements are re-phrasings of the general requirements good pdf's for importance sampling need to fulfil (see §4.3.1).

- Besides $\tilde{\mathbf{x}}$, also $\mathbf{A}\tilde{\mathbf{x}}$ needs to be known in order to guarantee proper normalisation of the transition probabilities;
- $\tilde{x}_j$ may never vanish if $x_j \neq 0$ and there exists a state i for which $a_{ij} \neq 0$. If it would vanish, then $\tilde{p}_{ij} = a_{ij}\tilde{x}_j / \sum_j a_{ij} x_j$ would be zero where $p_{ij} = a_{ij} x_j / x_i \neq 0$.

  Similarly, if approximate coefficients $\tilde{a}_{ij} \approx a_{ij}$ are used, $\tilde{a}_{ij}$ may never vanish if $a_{ij} x_j \neq 0$ for some j. If it would, important contributions to the solution may be missed.
- Sampling according to $a_{ij}\tilde{x}_j$ should not be too costly in order not to offset the gain of having to sample fewer paths.

A second approximation is to use the collision estimator with nearly-perfect transition probabilities for the absorption estimator. The resulting estimator is not perfect, but the generated paths are at least finite. Coveyou [33] has proven that for the thus modified collision estimator $\tilde{s}$,

$$1 \leq \frac{E[\tilde{s}^2]}{(E[\tilde{s}])^2} \leq 2$$

so that the variance $V[\tilde{s}] \leq (E[\tilde{s}])^2$: the statistical error $\sqrt{V[\tilde{s}]}$ with just 3 paths is already smaller than the result to be computed, with 99.7% certainty.

9.1.5 Collision density of the perfect absorption random walk

It is interesting to study the collision density χ_i (7.5) of the perfect absorption random walk estimator for the scalar product $< \mathbf{w}, \mathbf{x} >$ with $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$:

$$\chi_i = \pi_i + \sum_j \chi_j p_{ji} = \frac{w_i x_i}{\sum_s w_s x_s} + \sum_j \chi_j a_{ji} \frac{x_i}{x_j}$$

Compare this with the adjoint system

$$y_i = w_i + \sum_j y_j a_{ji} \quad \Rightarrow \quad (y_i x_i) = (w_i x_i) + \sum_j (y_j x_j) a_{ji} \frac{x_i}{x_j}$$

If this system has a unique solution (which we assume everywhere in this text), then

$$\chi_i = \frac{y_i x_i}{\sum_s w_s x_s} \tag{9.9}$$

In other words: the collision density χ_i of the perfect absorption random walk estimator is a weighted solution y_i of the adjoint system $\mathbf{y} = \mathbf{w} + \mathbf{A}^\top \mathbf{y}$. Since $\mathbf{x}$ and $\mathbf{w}$ are assumed to be known, the solution $\mathbf{y}$ of the adjoint system can be estimated by simply estimating the collision density and scaling

$$y_i = \sum_s w_s x_s \cdot \frac{\chi_i}{x_i}.$$

It will be explained in §9.2.3 what this means in the context of the shooting random walk estimator for radiosity.

9.2 Importance-driven random walk radiosity

In principle, the application of the theory above to the radiosity method is quite straightforward. There are however a number of practical problems.

9.2.1 Importance-driven gathering random walk radiosity

In gathering random walk radiosity (§7.4.1), the radiosity B_k for a given patch k is computed by estimating $\sum_i w_i B_i$ where $w_i = \delta_{ik}$ and

$$B_i = E_i + \sum_j \rho_i F_{ij} B_j.$$

The difference between the estimators for different patches k is only in the birth probabilities

$$\pi_i = \delta_{ik}.$$

The perfect transition probabilities are [135]:

- Absorption estimator:

$$p_{ij} = \frac{\rho_i F_{ij} B_j}{B_i} \quad ; \quad \alpha_i = \frac{E_i}{B_i} \tag{9.10}$$

- Collision (and survival) estimator:

$$p_{ij} = \frac{\rho_i F_{ij} B_j}{B_i - E_i} \quad ; \quad \alpha_i = 0 \tag{9.11}$$

In both cases, *transitions from i towards patches j from which i receives most radiosity $\rho_i F_{ij} B_j$ are favoured*. In the perfect absorption random walk, a path is allowed to terminate on a light source $E_i \neq 0$. The perfect collision estimator does not terminate.

It is possible to approximate these perfect estimators by using an approximate radiosity solution $\tilde{B} \approx B$ and by using approximate form factors $\tilde{F}_{ij} \approx F_{ij}$. Problems with this approach are:

- The approximate form factors can be accumulated during the computations, basically by counting the number of transitions between each pair of patches i and j. Care should be taken that $\tilde{F}_{ij} \neq 0$ if $F_{ij} \neq 0$ in order not to exclude potentially important radiosity contributions. For instance, if a form factor to a small, but bright, light source is accidently estimated to be zero — for instance because there has never been a transition to or from that light source before —, illumination due to this light source will be missing in the computed result;

- Besides $\tilde{B}$ and $\tilde{F}_{ij}$, also $\sum_j \rho_i \tilde{F}_{ij} \tilde{B}_j$ needs to be computed for proper normalisation of the transition probabilities. In general, the computation of $\sum_j \rho_i \tilde{F}_{ij} \tilde{B}_j$ for all i takes $\mathcal{O}(n^2)$ work.

The potential use in practice of a importance-driven gathering random walk, as outlined here, thus is rather questionable at first sight. There are however a number of possible solutions to these problems, that need to be investigated in further research:

- It is possible to ensure that all visible patches can be selected for a transition by harnessing multiple importance sampling [176, 118] with the probabilities $p_{ij}^{(2)} = \rho_i F_{ij}$, used in the non-importance driven gathering random walk, as auxiliary pdf;

- With hierarchical refinement, there will be only a fixed number of interactions (and form factors) for each element, so that computation of $\sum_j \rho_i \tilde{F}_{ij} \tilde{B}_j$ will take only $\mathcal{O}(n)$ work, which is acceptable.

A solution will need to be found for these problems first, before the issue of adaptive importance sampling can be attacked. In adaptive importance sampling, the computations are organised in stages. In each stage, a pdf is used which is based on the intermediate solution obtained in the previous stages. When done well, adaptive importance sampling may result in exponential convergence [63, 154, 100, 93].

9.2.2 Importance-driven shooting random walk radiosity

In shooting random walk radiosity (§7.4.2), the radiosity B_k of a given patch k is computed by estimating the scalar product $B_k = \frac{1}{A_k} < \frac{\Phi}{\rho}, (\rho I) >$ where

$$(\rho_i I_i) = (\rho_i V_i) + \sum_j \rho_i F_{ij} (\rho_j I_j) \tag{9.12}$$

with $V_i = \delta_{ik}$. The perfect birth probabilities (with source term estimation) are

$$\pi_i = \frac{\Phi_i I_i}{\sum_s \Phi_s I_s} = \frac{E_k \delta_{ki} + b_{ki}}{B_k} \tag{9.13}$$

The *probability of creating a path from a light source i shall be proportional to the radiosity at k due to that light source i.*

The perfect transition probabilities are [135]:

- Absorption estimator:

$$p_{ij} = \frac{\rho_i F_{ij} \rho_j I_j}{\rho_i I_i} = \frac{F_{ij} \rho_j I_j}{I_i} \quad ; \quad \alpha_i = \frac{\rho_i V_i}{\rho_i I_i} = \frac{\delta_{ik}}{I_i} \tag{9.14}$$

- Collision (and survival) estimator:

$$p_{ij} = \frac{F_{ij} \rho_j I_j}{I_i - V_i} \quad ; \quad \alpha_i = 0 \tag{9.15}$$

In both cases, *transitions towards regions with higher importance $\rho_j I_j$ are favoured*. With the absorption estimator, paths are allowed to terminate on the patch k of interest only: only for this patch, $\delta_{ik} \neq 0$, so that only $\alpha_k \neq 0$. The probability that a path survives a collision with patch k, equals $\sigma_k = \xi_k / I_k$. $\xi_k = I_k - V_k$ reflects the relative amount of radiosity (or power) that the patch of interest k receives back from itself.

The approximation of the birth and transition probabilities by using approximate form factors $\tilde{F}_{ij} \approx F_{ij}$ and an approximate importance solution $\tilde{I}_i \approx I_i$ poses the same problems as with the gathering random walk estimator for radiosity. The solutions are also the same.

9.2.3 View-importance driven random walk radiosity

There is however one additional problem with importance-driven shooting random walk radiosity: the importance I_i depends on the patch of interest k. In order to compute the radiosity of a different patch k', the solution process needs to be repeated for a different set of equations yielding a different importance solution. For this reason, perfect estimation of the radiosities by importance sampling would be feasible only with a gathering random walk.

Importance-driven shooting random walk radiosity can however still yield significant variance reduction by setting the weight vector $V_i = 1$ *for all patches that are visible in a particular view* of a scene. Setting the weight vector like this, marks all visible patches as being important (see figure 9.4 on page 156). Determination of the visible patches can be done easily with a Z-buffer like algorithm and ID-rendering,

like in previously proposed view-importance driven radiosity algorithms [152, 5] The importance I_i, resulting after solution of the importance equation (9.12), is called the *view-importance* of the patch i w.r.t. the view to be generated. With these, $\sum_i \Phi_i I_i$ yields the total power emitted by all visible patches in the view. I_i expresses to what extent the illumination emitted by patch i affects the illumination of visible patches in the view: $I_i P_i$ yields the flux emitted by all visible patches in a view, due to emission of power P_i by i.

The perfect random walk estimators, as introduced in the previous section, will be perfect estimators for the total flux emitted by the visible patches. They will not be perfect estimators for the radiosity emitted by individual patches. Consider however the collision density of the corresponding perfect absorption estimator (9.9):

$$\chi_i = \frac{P_i I_i}{\sum_s \Phi_s I_s}. \tag{9.16}$$

Although the random walk estimator is not perfect for computing radiosities of individual patches, paths are still biased towards to the region of interest ($\chi_i \propto I_i$). As a result, a lower variance will be obtained for patches in region of interest.

Formula (9.16) also suggests the following algorithm in order to compute a global radiosity (or better: power) solution for a scene, using view-importance biased random walks: whatever the choice for V_i — and thus also the resulting importances I_i — the power P_i emitted by any patch in the scene can be computed by *estimating the collision density* χ_i (by counting hits) of the biased paths, and using

$$P_i = \sum_s \Phi_s I_s \cdot \frac{\chi_i}{I_i}.$$

This procedure is unbiased for any choice of V_i so that $I_i \neq 0$ if $P_i \neq 0$. The choice of V_i however has an impact on the variance on the result. Choosing $V_i = 1$ for all visible patches in a view of a scene is a compromise that will reduce the variance on all visible patches. This choice will not be exceptional for any patch, unlike the strategy suggested in the previous section.

It is important to realise the difference between this algorithm and the algorithm of the previous section: in the previous section, importance was so chosen that $\sum_s \Phi_s I_s = P_k$ and P_k was obtained as the average score of the random walks that were generated. The algorithm in this section selects importance so that $\sum_s \Phi_s I_s$ yields the total power emitted by all visible patches in a view. The average score of the random walks would estimate this total power. The power emitted by individual patches is estimated through estimation of the collision density instead of averaging scores.

9.2.4 Computation of view-importance

So far, it has not been specified how the importance I_i in the importance-driven shooting random walk radiosity algorithms, outlined above, can be computed. Unlike in the analog shooting random walk method (§7.4.2), explicit computation of a global importance solution is required here: the computation of importance needs to be done as a first pass, before it can be used in order to bias random walks for shooting power towards important regions. Computation of a global importance solution can be done in three manners: by analog importance gathering random walks, by analog importance shooting random walks, and by radiosity-biased random walks:

Gathering importance with analog random walks

Equation (9.12) is immediately suggestive of a analog gathering random walk approach for computing importance: random walks are traced exactly as for gathering radiosity, with $\rho_i V_i$ taking the role of E_i and $\rho_i I_i$ taking the role of B_i. Random walks are traced from each patch k in turn ($w_i = \pi_i = \delta_{ik}$), towards the sources of importance ($V_i \neq 0$). The transition probabilities are $p_{ij} = \rho_i F_{ij}$ and can be simulated by means of uniformly distributed local or global lines as ever. The variance analysis can be taken over from gathering random walk radiosity almost literally.

Shooting importance with analog random walks

Consider the adjoint radiosity equations (4.31), related with (4.32) by multiplying both sides of (4.32) by A_i:

$$Y_i = W_i + \sum_j Y_j \rho_j F_{ji} \tag{9.17}$$

where $W_i = A_i V_i$ and $Y_i = A_i I_i$. Y_i can be obtained by estimating the collision density χ_i^Y of the random walk with birth probabilities $\pi_i = W_i / \sum_s W_s$ and transition probabilities $p_{ij} = \rho_i F_{ij}$. I_i is recovered as follows:

$$I_i = \sum_s A_s V_s \cdot \frac{\chi_i^Y}{A_i}.$$

The variance analysis can be copied almost without change from the variance analysis of the collision shooting random walk for radiosity. As with radiosity, analog shooting of importance will generally be preferred over analog gathering.

Importance computation by radiosity-biased random walks

Equation (9.17) above is the adjoint of the classical radiosity equations (4.25). The perfect absorption shooting random walk for (9.17) thus corresponds to the analog solution of

$$(Y_i B_i) = (W_i B_i) + \sum_j (Y_j B_j) \rho_j F_{ji} \frac{B_i}{B_j}$$

by estimation of the collision density χ_i^Z of the random walk with birth and transition probabilities:

$$\pi_i = \frac{W_i B_i}{\sum_s W_s B_s} = \frac{V_i P_i}{\sum_s V_s P_s} \quad ; \quad p_{ij} = \frac{\rho_i F_{ij} B_j}{B_i} \quad ; \quad \alpha_i = \frac{E_i}{B_i}.$$

The importance I_i is recovered by

$$I_i = \sum_s V_s P_s \cdot \frac{\chi_i^Z}{P_i}$$

and will be more accurate on patches with more intense illumination: in other words, on those patches where the importance is more important.

Radiosity-biased importance computation suggests the use of alternating phases in the computation: first, a first approximate global importance solution is sought using analog shooting random walks, next, a first importance-driven global power distribution is computed, next, a radiosity-driven second global importance solution is computed, etc . . . . The resulting estimates for radiosity and importance are combined as explained in §6.6.1.

9.2.5 View-importance driven random walk radiosity with analog transition probabilities

It was explained above that transition probability biasing is not without problems. Fortunately, also without transition probability biasing, a significant reduction of the variance in a view can be obtained once an approximate global view-importance solution has been computed. The basic idea is to modify only the birth and survival probabilities appropriately. The normalised transition probabilities $\pi_{ij} = p_{ij}/\sigma_i$ — expressing the probability that a path makes a transition from state i to state j after survival — are not modified. Local or global uniformly distributed lines can be used for sampling transitions as usual.

Optimal path creation probabilities with analog transitions

When the importance I_i on light sources is known, they can be used in order to bias the path creation probabilities in such a way that more paths are generated from important light sources and fewer from less important light sources.

Consider analog shooting random walk radiosity without source term estimation. The variance of this estimator is given by (7.30). In case of the analog collision estimator for estimating the radiosity on a fixed patch k:

$$V[\tilde{s}] = \frac{1+2\zeta_k}{A_k^2} \sum_i \frac{\Phi_i^2}{\pi_i \rho_i^2} \rho_i \sum_j \rho_i F_{ij} (\rho_j I_j) - b_i^2 = \frac{1+2\zeta_k}{A_k^2} \sum_i \frac{\Phi_i^2}{\pi_i} (I_i - V_i) - b_i^2$$

Using Lagrange multipliers again in order to minimise this expression with constraint $\sum_i \pi_i = 1$, yields

$$\pi_i \propto \Phi_i \sqrt{I_i - V_i}. \tag{9.18}$$

The optimal birth probabilities are proportional to the self-emitted power Φ_i of a light source and to the *square root* of the received importance $I_i - V_i$. This result [133] is also valid for the absorption and survival estimators.

Russian roulette and splitting

A global importance solution can also be used in order to bias the survival probabilities so that paths with low expected score are terminated prematurely. The survival probability in important regions, where paths will yield significant scores on the average, are artificially increased. Of course, when the survival probability of the analog

random walk is modified from σ_i to $\tilde{\sigma}_i$, the score of the random walk needs to be modified accordingly, by multiplying with $\sigma_i/\tilde{\sigma}_i$.

A related idea to to split particles in important regions in multiple parts, resulting in a branching random walk [81, 156, 107]. The scores associated with each branch are weighted so that the expected value of their sum is right.

Russian roulette and splitting have been studied in the context of stochastic ray tracing [12]. So far however, these techniques have not yet received attention in the context of view-importance driven Monte Carlo radiosity. Their study requires determination of the optimal number of off-spring particles on each patch, depending on the approximate importance solution I_i. A similar effect is however obtained easily with stochastic Jacobi iterations.

9.3 Importance-driven stochastic relaxation radiosity

The key to importance-driven stochastic relaxation radiosity [108] is the observation, made above, that the solution of a system $\mathbf{x} = \mathbf{e} + \mathbf{A} \cdot \mathbf{x}$ with a perfect absorption random walk estimator corresponds to the analog solution of a modified system $(y_i x_i) = (w_i x_i) + \sum_j (y_j x_j) a_{ji} x_i / x_j$ using a collision estimator. In case of shooting random walk radiosity, the modified system is

$$P_i I_i = \Phi_i I_i + \sum_j P_j I_j F_{ji} \rho_i \frac{I_i}{I_j} \tag{9.19}$$

where I_i is the importance corresponding with some importance source distribution V_i:

$$I_i = V_i + \sum_j F_{ij} \rho_j I_j. \tag{9.20}$$

Instead of solving (9.19) by collision density estimation, stochastic Jacobi iterations can be used as well. We will discuss here the case $V_i = 1$ for all visible patches in a view of a scene and zero for non-visible patches, yielding view-importance driven stochastic Jacobi iterations.

9.3.1 Importance-driven power propagation

When a global importance solution I_i is available for a view, it can be used in three manners in order to bias power propagation towards important regions. The computation of importance, using stochastic Jacobi steps as well, will be discussed subsequently.

Method A: analog transitions with total-importance weighting The problem consists in transforming a given power distribution P_i into a new power distribution $P_i' = \Phi_i + \sum_j P_j F_{ji} \rho_i$, so that computational effort is focussed in important regions, that is: on patches with high I_i.

A first way to do so is by weighting the probabilities of shooting a ray from a patch with the importance: if the importance is high, the probability of shooting a ray from the patch is increased. If low, the probability is attenuated. The contribution of the

ray at the hit patch is modified accordingly in order to retain an unbiased estimator. Non-view importance driven stochastic Jacobi iterations will appear as a special case, with $I_i = 1$ for all patches i.

In the same way as was done in §6.4, equation 9.19 can be reformulated as:

$$I_k(P'_k - \Phi_k) = \sum_i \sum_j (P_j I_j) F_{ji} \frac{\rho_i I_i}{I_j} \delta_{ik} \tag{9.21}$$

The sums on the right hand side can be estimated simultaneously for all patches k, using estimators like the following:

1. Select the ij-th term with probability $p^A_{ij} = p^A_j p^A_{i|j}$ as follows:
   (a) First select a patch j with probability $p_j = P_j I_j / Q^A_T$, where $Q^A_T = \sum_s P_s I_s$;
   (b) Next, select i conditional on j with probability $p^A_{i|j} = F_{ji}$. As usual, selection of i is carried out by finding the next intersection point of a uniformly distributed line through j.
2. The contribution associated with the ij-th term is

$$\mathcal{A}^k_{ij} = \frac{\delta_{ik}}{A_k I_k} Q^A_T \frac{\rho_i I_i}{I_j} \tag{9.22}$$

The expectation is b'_k. If N rays are used in a iteration, the contribution $\mathcal{A}^k_{ij}$ need to be divided by N for each ray.

The only changes that need to be made to algorithm 12 or 13 are:

- the computation of the probabilities q_i for selecting a ray origin on a patch;
- the score that is recorded on the patch hit by every ray.

The other parts of the algorithm remain unchanged, except that importance needs to be computed first. Computation of importance is explained further in this text.

The variance of the estimators $\mathcal{A}$ ($V[\mathcal{A}] = E[\mathcal{A}^2] - (E[\mathcal{A}])^2$), is determined by

$$E\left[(\mathcal{A}^k)^2\right] = \frac{\rho_k}{A_k} \left(\sum_s P_s I_s \right) \frac{1}{A_k} \sum_j \frac{P_j F_{jk} \rho_k}{I_j} \tag{9.23}$$

Method B: analog transitions with received-importance weighting A better estimator is obtained by selecting shooting patches j with probability that is proportional to the received importance $(I_i - V_i)$ instead of the total importance I_i.

Equation (9.21) is equivalent with

$$I_k(P'_k - \Phi_k) = \sum_i \sum_j P_j (I_j - V_j) F_{ji} \frac{\rho_i I_i}{I_j - V_j} \delta_{ik} \tag{9.24}$$

suggesting the following estimators:

1. Select a patch j with probability $p_j^B = P_j(I_j - V_j)/Q_T^B$ with $Q_T^B = \sum_s P_s(I_s - V_s)$.
2. Select patch i with conditional probability $p_{i|j}^B = F_{ji}$ using a uniformly distributed line from j as in method A;
3. Contribute

$$\mathcal{B}_{ij}^k = \frac{\delta_{ik}}{A_k I_k} Q_T^B \frac{\rho_i I_i}{I_j - V_j} \tag{9.25}$$

In order to compare the effectiveness (below), we need

$$E\left[(\mathcal{B}^k)^2\right] = \frac{\rho_k}{A_k}\left(\sum_s P_s(I_s - V_s)\right)\frac{1}{A_k}\sum_j \frac{P_j F_{jk}\rho_k}{I_j - V_j} \tag{9.26}$$

The main difference with method A is in the factor $\sum_s P_s(I_s - V_s)$, which will be significantly smaller than $\sum_s P_s I_s$ in (9.23): in practice, I_i is at most a few times higher than $V_i = 1$ for visible patches in a scene. $\sum_s P_s V_s$ will be a significant fraction of $\sum_s P_s I_s$. In particular, light sources that are directly visible in a view cause large contributions to $\sum_s P_s V_s$ (see figure 9.1).

Non-view importance driven stochastic Jacobi iterations results with the choice $I_i = 1/\rho_i$ and $V_i = 1/\rho_i - 1$. This choice for I_i and V_i satisfies (9.20).

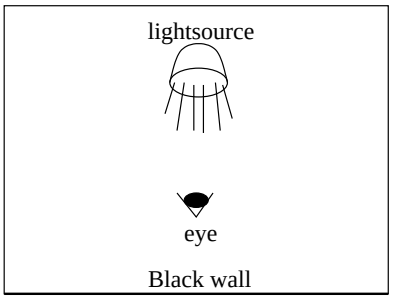


Figure 9.1: The light-source has a large product of power and total importance. All its light is however absorbed by the black wall behind the viewer, so it makes no sense to use a large number of rays to represent the power of this light-source. The solution is to use received importance instead of total importance.

Method C: non-analog transitions Methods A and B only modify the probability by which rays are shot so that more rays are shot from important patches and fewer from unimportant patches. The direction of the rays is cosine-distributed w.r.t. the normal of the patches. It is also possible to aim rays towards important regions, in addition to shooting more rays from important patches.

Since $I_j = V_j + \sum_i F_{ji}\rho_i I_i$, and all $I_i \geq 0$, the factors $F_{ji}\rho_i I_i/(I_j - V_j)$ form a probability density. This suggests a third kind of estimators for importance-driven power propagation:

1. Select a patch j with probability $p_j^C = p_j^B$;
2. Select a patch i conditional on j with probability $p_{i|j}^C = F_{ji}\rho_i I_i/(I_j - V_j)$;
3. Contribute

$$\mathcal{C}_{ij}^k = \frac{\delta_{ik}}{A_k I_k}\left(\sum_s P_s(I_s - V_s)\right) \tag{9.27}$$

We need

$$E\left[(\mathcal{C}^k)^2\right] = \frac{1}{A_k I_k}\left(\sum_s P_s(I_s - V_s)\right)\cdot\frac{1}{A_k}\sum_j P_j F_{jk}\rho_k. \tag{9.28}$$

Whether estimator C will be better than A and B for a given patch k depends on the ratio

$$\frac{E[\mathcal{B}^2]}{E[\mathcal{C}^2]} = \frac{\sum_j P_j F_{jk}\rho_k/(I_j - V_j)}{\sum_j P_j F_{jk}\rho_k/\rho_k I_k}.$$

If this ratio is larger than 1, C will have lower variance, and vice versa for B.

Some intuition in the relative benefit of method B and C can be gained as follows: consider a case where visible patches in a view do not receive much illumination directly from each other, for instance because they are not facing each other. If we assume as well that the received importance $I_i - V_i$ is approximately constant near the visible parts of the scene, I_k will be larger than $I_j - V_j$, the received importance of patches j from which k receives power, by about $V_k = 1$. If the average reflectivity near the visible parts of the scene is about 0.5, $I_j - V_j \approx 1$ at most, so that method C will yield a variance which is at most 2 times lower than with method B. Up to 4 times fewer rays would be needed in order to compute the radiosity in the visible parts of the scene to given accuracy.

Unfortunately, tracing rays that are aimed more towards important regions is significantly more complicated than tracing cosine-distributed rays. The problems are the same as in importance-driven gathering random walk radiosity (§9.2.1). Whether the lower variance offsets the higher cost per ray or not, depends on the actual scene and view, but the benefit of method C over method B is generally doubtful in practice.

Importance-driven versus analog power propagation

When will importance-driven power propagation be interesting?

A first important factor in answering this question is the variance of the involved estimators, which determines the number of samples that will needed in order to compute the radiosity for visible patches in a view $V_i \neq 0$ to given accuracy. For analog stochastic Jacobi iterations, we have

$$E[(\hat{b}'_k)^2] = \frac{\rho_k}{A_k}\sum_s P_s b'_k = \frac{\rho_k}{A_k}\left(\sum_s P_s\right)\cdot\frac{1}{A_k}\sum P_j F_{jk}\rho_k \tag{9.29}$$

It is easiest to compare with method C, expression (9.28). Whether C will be better than the analog method depends on the ratio

$$\frac{E[(\hat{b}'_k)^2]}{E[\mathcal{C}^2]} = \frac{\rho_k I_k \sum_s P_s}{\sum_s P_s(I_s - V_s)}$$

If this ratio is larger than 1, importance-driven iterations will have lower variance. This will be the case if relatively few patches of a scene are visible, so that $I_i \geq 1$ only in a small part of the scene. The ratio $\sum_s P_s / \sum_s P_s(I_s - V_s)$ will be large if the scene contains many, strong, but unimportant light sources. As a matter of fact, this ratio can be arbitrary large. In practice, we find that the variance of importance-driven stochastic Jacobi iterations will be lower than analog stochastic relaxation in most cases. The formula above also indicates that the variance reduction in the important parts of the scene is accompanied by a potentially very high increase of variance on unimportant patches. This conclusion is also valid for methods A and B.

The second factor determining the relative benefit of view-importance driven versus analog stochastic relaxation is the cost. The cost per ray in methods A and B is practically equal as in the analog case, but there is a significant extra cost due to the need to compute a global importance solution in addition to the radiosity solution. The amount of work spent in computing importance can be traded for the quality of the importance solution, but in practice, about a doubling of the cost can be expected.

Nonetheless, importance-driven stochastic relaxation will be highly beneficial in complex scenes, containing many light sources, of which only a small number are significant for a view. Typical examples include the visualisation of rooms in a large building.

9.3.2 Computation of importance

The computation of importance can be done in a very similar way as the computation of power. A given global importance solution I_i is transformed into a better one I'_i by using stochastic Jacobi iterations. As an initial importance distribution, the importance source $I_i = V_i$ (1 for visible patches in a view, and 0 for invisible patches) can be used. Due to the order in which the indices of the form factor appear, it is more convenient to use equation (4.31) for propagating importance instead of (4.32). This equation can be rewritten as

$$Y'_k - W_k = \sum_i \sum_j Y_j \rho_j F_{ji} \delta_{ik} \tag{9.30}$$

with $Y_i = A_i I_i$ and $W_i = A_i V_i$.

Method D: Analog importance propagation Analog importance propagation is directly based on (9.30):

1. Select a patch j with probability $p_j^D = Y_j \rho_j / \sum_s Y_s \rho_s$;

2. Select a patch i conditional on j with probability $p_{i|j}^D = F_{ji}$, using a uniformly local or global line originating at j as usual;

3. Contribute

$$\mathcal{D}_{ij}^k = \frac{\delta_{ik}}{A_k}\left(\sum_s Y_s \rho_s\right) \tag{9.31}$$

The expectation is $I_k' - V_k$. The variance is determined by

$$E\left[(\mathcal{D}^k)^2\right] = \frac{1}{A_k}\left(\sum_s Y_s \rho_s\right) \cdot (I_k' - V_k) \tag{9.32}$$

Method E: Analog transitions with total-radiosity weighting The duality between importance and radiosity can be exploited also in order to compute importance to higher accuracy where it is needed most. Accurate computation of importance is especially important on bright surfaces. A first way to take this into account is by biasing the probability p_j^D of shooting an importance-ray from a patch j proportional to the radiosity B_j of j. Multiplying both sides of (9.30) by B_i yields:

$$(Y_k' - W_k)B_k = \sum_i \sum_j Y_j B_j \rho_j F_{ji} \frac{B_i}{B_j} \delta_{ik} \tag{9.33}$$

1. Select patch j with probability $p_j^E = Y_j B_j \rho_j / Q_T^E$, with $Q_T^E = \sum_s Y_s B_s \rho_s$;
2. Select i conditional on j with probability $p_{i|j}^E = F_{ji}$;
3. Contribute

$$\mathcal{E}_{ij}^k = \frac{\delta_{ik}}{A_k B_k} Q_T^E \cdot \frac{B_i}{B_j} \tag{9.34}$$

The variance is determined by

$$E\left[(\mathcal{E}^k)^2\right] = \frac{1}{A_k}\left(\sum_s Y_s B_s \rho_s\right) \cdot \frac{1}{A_k} \sum_j \frac{Y_j \rho_j F_{jk}}{B_j} \tag{9.35}$$

Note that for power propagation, we assumed that the importance distribution was given and fixed. Here, for importance propagation, the radiosity distribution B_i is assumed to be given (and fixed). The analog case is retrieved by the choice $B_i = 1, \forall i$.

Method F: Analog transitions with received-radiosity weighting Again, we can do better by weighting only with received radiosity $B_j = B_j - E_j$:

$$(Y_k' - W_k)B_k = \sum_i \sum_j Y_j (B_j - E_j) \rho_j F_{ji} \frac{B_i}{B_j - E_j} \delta_{ik} \tag{9.36}$$

1. Select patch j with probability $p_j^F = Y_j (B_j - E_j) \rho_j / Q_T^F$, with $Q_T^F = \sum_s Y_s (B_s - E_s) \rho_s$;

2. Select i conditional on j with probability $p^F_{i|j} = F_{ji}$;

3. Contribute

$$\mathcal{F}^k_{ij} = \frac{\delta_{ik}}{A_k B_k} Q^E_T \cdot \frac{B_i}{B_j - E_j} \tag{9.37}$$

The variance is determined by

$$E\left[(\mathcal{F}^k)^2\right] = \frac{1}{A_k}\left(\sum_s Y_s(B_s - E_s)\rho_s\right) \cdot \frac{1}{A_k}\sum_j \frac{Y_j \rho_j F_{jk}}{B_j - E_j} \tag{9.38}$$

The variance will be lower than with total-radiosity weighting because of the factor $Q^F_T = \sum_s Y_s(B_s - E_s)\rho_s < Q^E_T = \sum_s Y_s B_s \rho_s$. The difference is due to the important light sources: $\sum_s Y_s E_s \rho_s$ can be quite large.

Method G: Non-analog transitions The important regions for importance propagation correspond to the bright surfaces in the scene. Also here, rays can be aimed towards bright surfaces in addition to shooting more rays from bright surfaces: because $B_j = E_j + \sum_i \rho_j F_{ji} B_i$, and $B_i, E_i \geq 0$, the factors $\rho_j F_{ji} B_i/(B_j - E_j)$ form a probability distribution. This suggests the following estimators:

1. Select a patch j with probability $p^G_j = Y_j(B_j - e_j)/Q^G_T$, with $Q^G_T = \sum_s Y_s(B_s - E_s)$;

2. Select patch i conditional on j with probability $p^G_{i|j} = \rho_j F_{ji} B_i/(B_j - E_j)$;

3. Contribute

$$\mathcal{G}^k_{ij} = \frac{\delta_{ik}}{A_k B_k} Q^G_T \tag{9.39}$$

The variance is determined by

$$E\left[(\mathcal{G}^k)^2\right] = \frac{1}{A_k B_k}\left(\sum_s Y_s(B_s - E_s)\right)(I'_k - V_k) \tag{9.40}$$

The same arguments about the relative benefit of methods C versus B for shooting power can be repeated for comparing this method G with method F above for shooting importance. The problems with aiming rays towards bright surfaces are again the same as those discussed for importance-driven gathering random walk radiosity. Unfortunately, in practice, scenes often have small and very bright light sources, so that aiming rays towards the light sources is even more difficult than aiming rays for shooting power towards the region of importance: the importance distribution is generally less "peaked" than the radiosity distribution of a scene. For this reason, this method G is not recommended in practice.

Also the arguments for comparing importance-driven power propagation versus analog power propagation can be repeated for importance propagation. The conclusions are the same, mutatis mutandis.

9.3.3 Outline of a complete algorithm

For practical use, we especially recommend

- method B for importance-driven power propagation;
- method F for radiosity-driven importance propagation.

It is however important that a sufficiently "stable" importance solution is available before it can be used for biasing power propagation. Vice versa, it is also important that the radiosity solution has sufficiently low variance before using it in order to bias importance propagation. Unfortunately, it is hard to derive precise conditions indicating when importance and radiosity are sufficiently stable for biasing the dual transport. We suggest the use of heuristics for choosing the number of samples as derived in §6.4.5. Such heuristics provide good default values.

In the implementation, we have taken the following strategy:

- First, compute a non-importance driven radiosity solution using incremental Jacobi iterations as explained in §6.4. The initial number of samples was chosen according to (6.18):

$$N \approx 9 \max_i \frac{A_T}{A_i}$$

  ignoring the smallest patches in the scene;

- Next, visible patches in a view of a scene are marked by setting $V_i = 1$ for visible patches. The importance source of non-visible patches should be initialised to 0. Visible patches can be determined using ID-rendering;
- Next, a first non-radiosity driven importance solution, using incremental Jacobi iterations, is computed with the same number (6.18) of samples;
- Finally, the solution is gradually improved by alternating regular importance-driven radiosity propagation steps and regular radiosity-driven importance propagation steps. A good heuristic for merging the result of different steps will be given below in §9.3.5.

It is not strictly necessary to compute a non-importance driven solution initially. Doing so however makes it easier to use the merging heuristic of §9.3.5.

9.3.4 Incremental view-importance computation

When the viewing parameters change, for instance in a walk-through application or animation, view-importance changes as well. A small change in viewing position or direction for instance, will most often however result in only a small change in view importance. For small viewing changes, incremental computation of view importance can yield the new view importance I_i^{new} faster than re-computation from scratch:

- First, the new direct importance V^{new} is determined by marking visible patches in the new view using ID-rendering. The difference $\Delta V = V^{\text{new}} - V^{\text{old}}$ is computed on the fly. ΔV_k can take three possible values: 1 for patches that were not visible previously and become visible in the new view, -1 for patches that were visible previously and move out of sight in the new view, and 0 for patches of which the visibility status does not change;

- Next, the importance increment $\Delta I = I^{\text{new}} - I^{\text{old}}$ is computed using incremental Jacobi iterations. The number of samples in the first iteration is chosen proportional to $\sum_k A_k \mid \Delta V_k \mid$ in such as way that each ray transports the same amount of importance as initially during the computations. If more rays would be needed after the viewing change than in the initial iterations, view-importance is recomputed from scratch rather than incrementally.

Note that ΔV may be negative on some patches, so that the probabilities p_j in the estimators described above become signed probability distributions. Such signed probability distributions can be sampled by independent sampling of the positive and negative parts, as explained in for instance [155]. In this case, the absolute value $\mid \Delta I \mid$ shall be used instead of I in the expressions for the probabilities p_j in §9.3.2. The contribution on the patch hit by each ray is modified accordingly by giving it the sign of ΔI_j.

9.3.5 Progressive variance reduction

Motivation Progressive variance reduction by merging the result of different runs, as explained in §6.6.1, assumes that the quality of the solution in the scene improves at the same rate during each iteration. In view-importance driven stochastic relaxation radiosity however, the quality of the solution improves much faster in important regions than in unimportant regions. After a viewing change, a new region of the scene becomes important, while the previously important region may become unimportant. New iterations after a viewing change will focus on the new important regions, while very little samples may contribute to the result in the previously important region. If the results from different steps were merged based solely on the total number of samples that is used (§6.6.1), one would find that a high quality result in the previously important region would degrade quickly, while too much weight is given to the low-quality results from previous steps in the new important region. In this section, a new merging heuristic is proposed that takes both the importance of each patch and the number of samples into account in order to adaptively merge the result from all iterations in such a way that the quality is improved maximally in a new important region without degradation of the quality in a previously important region. The same heuristic allows to switch the use of importance on and off during the computations.

The new heuristic The new heuristic for merging importance-driven radiosity solutions is based on expression (9.28) for the variance of the "ideal" importance-driven radiosity estimator (method C, for clarity henceforth denoted as $\hat{B}_k$). Ignoring the $(E[\hat{B}_k])^2$ term, and with N samples:

$$V[\hat{B}_k] \propto \frac{1}{NI_k} \left(\sum_s P_s (I_s - V_s) \right).$$

Factors, such as the patch area A_k, that do not change between iterations are irrelevant for the merging heuristic. For the purposes of the heuristic, also the power (or radiosity) solution can be assumed constant if a non-importance driven radiosity solution is computed initially. Indeed: subsequent importance-driven radiosity propagation steps will be used only in order to *improve* the initial radiosity solution adaptively, without introducing large quantitative changes.

Non-importance driven iterations correspond with the choices $I_k = 1/\rho_k$ and $I_k - V_k = 1$, leading to (compare with §6.4.4):

$$V[\hat{B}_k] \propto \frac{\rho_k}{N}\left(\sum_s P_s\right).$$

Now consider two iterations, with N^1 and N^2 number of samples respectively and using importances I_k^1 and I_k^2. The source importance distributions are V_k^1 and V_k^2. Then:

$$\begin{aligned} V[\hat{B}_k^1] &\propto \frac{1}{N^1 I_k^1}\left(\sum_s P_s(I_s^1 - V_s^1)\right) = \frac{C^1}{I_k^1} \\ V[\hat{B}_k^2] &\propto \frac{1}{N^2 I_k^2}\left(\sum_s P_s(I_s^2 - V_s^2)\right) = \frac{C^2}{I_k^2} \end{aligned}$$

The constants C^1 and C^2 group all factors that are the same for all patches in the scene. These constants need to be computed only once during each iteration. The sums need to be computed anyways for proper normalisation of the probability distributions that are used for sampling ray origins.

According to (4.16), the optimal combination of the radiosity results on a patch k is obtained if they are combined with weights w_k^1 and w_k^2 that are inverse proportional to the variance:

$$w_k^1 \propto \frac{1}{V[\hat{B}_k^1]} = \frac{I_k^1}{C^1} \quad ; \quad w_k^2 \propto \frac{1}{V[\hat{B}_k^2]} = \frac{I_k^2}{C^2}$$

The variance of the combined result is

$$V[\hat{B}_k^{\text{comb}}] = w_k^1 V[\hat{B}_k^1] + w_k^2 V[\hat{B}_k^2] = \frac{V[\hat{B}_k^1]V[\hat{B}_k^2]}{V[\hat{B}_k^1] + V[\hat{B}_k^2]}.$$

In order to combine this combined result with the result of a third iteration, it shall receive a weight

$$w_k^{\text{comb}} \propto \frac{1}{V[\hat{B}_k^{\text{comb}}]} = \frac{V[\hat{B}_k^1] + V[\hat{B}_k^2]}{V[\hat{B}_k^1]V[\hat{B}_k^2]} = \frac{1}{V[\hat{B}_k^1]} + \frac{1}{V[\hat{B}_k^2]} = \frac{I_k^1}{C^1} + \frac{I_k^2}{C^2}.$$

A flexible merging strategy thus results by accumulating the factors I_k/C during each iteration on each patch k separately. When merging with the result of a new iteration, the combined result of all previous iterations shall receive a weight that is proportional to the accumulated factors. The new result on each patch k is weighted proportional with the factor I_k/C of the new iteration. The main difference with the merging strategy of §6.6.1 is that quality factors I_k/C are accumulated per patch individually here. These quality factors keep track of the importance of the patch during previous iterations as well as of the number of samples.

Discussion The heuristic has been derived for importance-driven radiosity propagation using method C. In practice, we recommend the use of method B instead. The heuristic however appears to work well with method B too. In §9.3.1, the variance of method B and C have been compared. It was shown that the variance of method B is higher in important regions and lower in unimportant regions. The heuristic, derived from the variance of method C, therefore will under-estimate the variance in important regions while over-estimating it in unimportant regions. The result is that a new solution will receive a slightly too high weight in important regions, while it will receive a slightly too low weight in currently unimportant regions. This is however not a problem in practice since the solution will be good anyways in important regions and bad in unimportant regions.

A similar heuristic can be derived for merging importance solutions as well. After a change of viewing parameters, the accumulated quality factors should be cleared to zero however because the assumption that the importance solution only fluctuates slightly around its correct value will not hold anymore.

9.3.6 Empirical results

Figure 9.2 shows a environment consisting of about 162,000 patches, rendered without using view-importance. Results are shown for about 1.1M rays (1 iteration, about 3 minutes) and for about 3,3M rays (3 iterations, about 9 minutes).

Figure 9.3 shows the same scene. This time, after the first non-importance driven iteration (same results as in the left column of figure 9.2), two importance-driven iterations were performed for view-point A. The solution for this view (middle-left image in figure 9.3) is considerably better than the corresponding solution after 3 non-importance driven iterations (middle-right image of figure 9.2), although approximately the same number of rays has been used for propagating radiosity. The solution for the unimportant view-point B has however not improved (bottom-left image in figure 9.3, compare with the bottom-left image in figure 9.2).

Next, the viewpoint was changed from viewpoint A to viewpoint B. The results after two more importance-driven iterations are shown in the right column of figure 9.3. The solution for viewpoint B (bottom-right image of figure 9.3) has drastically improved (compare with the bottom-left image of figure 9.3 or of figure 9.2). Although almost no samples are shot in the, now unimportant, region of viewpoint A, the new merging strategy of §9.3.5 takes care that the good solution for viewpoint A does not degrade (compare middle-right image of figure 9.3 with the middle-left image).

Figure 9.4 shows the importance solutions corresponding to the left and right column images of figure 9.3.

An importance-driven iteration however takes about twice as much computation time in our implementation, because the same number of rays is used for importance propagation as for radiosity propagation. Figure 9.5 finally compares images obtained with and without view-importance for about the same total amount of work (about 9 minutes). The view-importance driven solution still is clearly better.

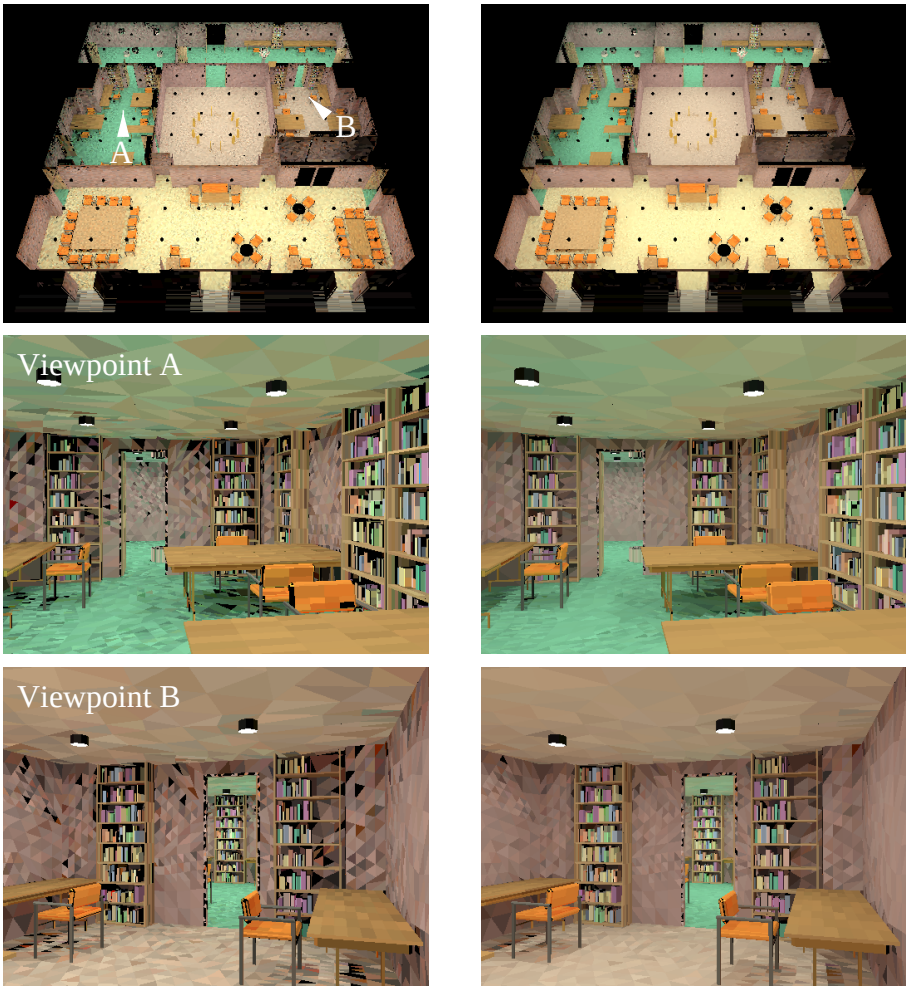


Figure 9.2: These images have been computed without using view-importance. The images in the left column took about 3 minutes of computation time. The images in the right column took about 9 minutes of computation time.

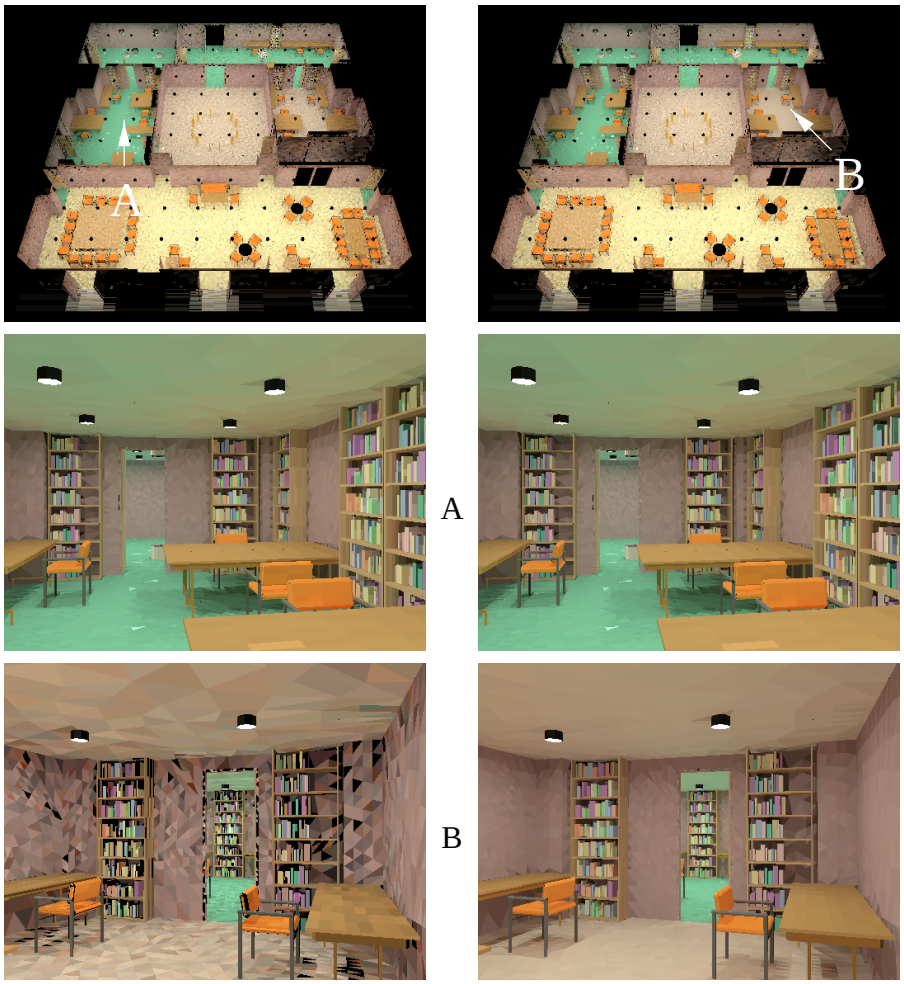


Figure 9.3: These images have been computed with view-importance, after an initial non-view importance driven iteration. The images in the left column are obtained after two importance-driven iterations for viewpoint A. The solution for this viewpoint (middle-left image) has been drastically improved. The solution in unimportant regions (viewpoint B, bottom-left image) has not improved. Next, the viewpoint was changed from viewpoint A to viewpoint B. The images in the right column show the results after two more importance-driven iterations for viewpoint B. The solution for viewpoint B has drastically improved (bottom-right). The new merging heuristic of §9.3.5 takes care that the good solution for the now unimportant region of viewpoint A does not degrade (middle-right).

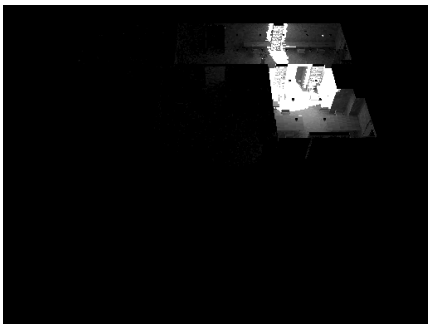



Figure 9.4: The importance distributions corresponding to the left and right column images of figure 9.3. The bright patches are directly visible in the view and have direct importance $V_k = 1$. The other patches only have indirect importance.


Figure 9.5: Two images obtained with approximately the same total amount of work (about 9 minutes). The left image has been obtained without view-importance. The right image has been computed with view-importance. With the help of view-importance, computation work is clearly concentrated in the area of interest.
The model shown in these figures is an edited part of the Soda Hall VRML model made available at the University of California at Berkeley, California, U.S.A.

9.4 Conclusion

A number of perfect random walk estimators, based on importance sampling, have been presented and their application to gathering and shooting random walks for radiosity studied. In order to optimise the computation of the radiosity of a single patch, the gathering random walk is best suited.

Of larger practical importance is the optimisation of the computations for all visible patches in a view. This is called *view*-importance driven random walk radiosity. A shooting random walk is best suited for this.

Several biasing strategies have been proposed and discussed. Unfortunately, sampling transitions according to other probability distributions than the form factors, poses some important practical problems, that need to be resolved in future research. View-importance driven random walk radiosity is also possible however while keeping analog transition probabilities. Two strategies are suggested: source biasing and Russian roulette and splitting. The latter has not yet been investigated in the context of random walk radiosity, but a similar effect is obtained easily in stochastic relaxation radiosity.

Various strategies for view-importance driven power propagation and radiosity-driven importance propagation using stochastic relaxation steps have been presented. In particular, the methods B (for power propagation) and F (for importance propagation) are recommended. Most often, a small change in viewing parameters results in only a small change in view-importance. Incremental computation will yield the new importances faster than re-computation from scratch in such cases. A new, very flexible, strategy has been proposed for merging the results from iterations with different number of samples or different importance distributions.

View-importance driven Monte Carlo radiosity will be very useful in large, complex, environments, of which only a small part is visible in a view. It requires that significant amount of extra work is done (the computation of view-importance), but by doing this extra work, a lot of irrelevant work can be avoided.

10 Control Variates in Monte Carlo Radiosity

This chapter investigates the use of control variates (a form of positive correlated sampling) in the context of Monte Carlo radiosity. Control variates have been applied before in stochastic ray-tracing in [97]. In the context of Monte Carlo radiosity, a similar idea, called "constant radiosity steps" [114], but based on different arguments, has been proposed. First, the application of control variates in the context of random walk methods will be explained in general (§10.1). Next, the application in gathering random walk radiosity algorithms will be discussed (§10.2). Finally, an improved constant radiosity step method for stochastic relaxation methods, based on variance minimisation, will be proposed (§10.3).

10.1 Control variates in random walk methods

10.1.1 Outline

Consider an approximation $\tilde{\mathbf{x}}$ for the solution $\mathbf{x}$ of $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$. The correction $\Delta\mathbf{x} = \mathbf{x} - \tilde{\mathbf{x}}$ then fulfils

$$\Delta\mathbf{x} = (\mathbf{e} + \mathbf{A}\tilde{\mathbf{x}} - \tilde{\mathbf{x}}) + \mathbf{A} \cdot \Delta\mathbf{x} \tag{10.1}$$

Proof:

$$\Delta\mathbf{x} = (\mathbf{I} - \mathbf{A}) \cdot \Delta\mathbf{x} + \mathbf{A} \cdot \Delta\mathbf{x}\ ;\ (\mathbf{I} - \mathbf{A}) \cdot \Delta\mathbf{x} = \mathbf{x} - \mathbf{A}\mathbf{x} + \mathbf{A}\tilde{\mathbf{x}} - \tilde{\mathbf{x}} = \mathbf{e} + \mathbf{A}\tilde{\mathbf{x}} - \tilde{\mathbf{x}}$$

□

This is true regardless of the error in the approximation $\tilde{\mathbf{x}}$. Now suppose $\Delta\mathbf{x}$ is computed using for instance a random walk method. The resulting estimate $\Delta\tilde{\mathbf{x}}$ for the correction $\Delta\mathbf{x}$ will not be exact, so that $\tilde{\tilde{\mathbf{x}}} = \tilde{\mathbf{x}} + \Delta\tilde{\mathbf{x}}$ will not be exactly equal to the solution $\mathbf{x}$ of the system to be solved either. However, regardless of the error on the first approximation $\tilde{\mathbf{x}}$, the error on the new approximation $\tilde{\tilde{\mathbf{x}}}$ is only determined by the error on the computed correction $\Delta\tilde{\mathbf{x}}$! Before we can show the potential benefit of equation (10.1), a theorem needs to be introduced:

Theorem 10.1 *The variances $V[s_i]$ (7.22) of the absorption, collision and survival random walk estimators for the system $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$ fulfils*

$$\|V[s_i]\| \leq \lambda \|\mathbf{e}\|^2. \tag{10.2}$$

where λ is some positive constant that depends on the problem.

Proof: Consider

$$V[s_i] = v_i - x_i^2 \quad \text{with} \quad v_i = \mu_i + \sum_j \frac{a_{ij}^2}{p_{ij}} v_j = \sum_s v_{is}\mu_s$$

where

$$v_{is} = \delta_{is} + \sum_j \frac{a_{ij}^2}{p_{ij}} v_{js}$$

is only dependent on the coefficients a_{ij} and the transitions probabilities p_{ij}. Denote the matrix v_{is} by $\mathbf{V}$, we then have that

$$\|V[s_i]\| = \|\mathbf{V} \cdot \mu - x_i^2\| \leq \|\mathbf{V}\|\|\mu\| + \|\mathbf{x}\|^2$$

Since

$$\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{e} + \mathbf{A}^2\mathbf{e} + \cdots \Rightarrow \|\mathbf{x}\| \leq (1 + \|\mathbf{A}\| + \|\mathbf{A}\|^2 + \cdots)\|\mathbf{e}\| = \frac{\|\mathbf{e}\|}{1 - \|\mathbf{A}\|}$$

The coefficients μ were given in (7.26), (7.27) and (7.28). For the absorption estimator, we have

$$\mu_s = \frac{e_s^2}{\alpha_s} \quad \Rightarrow \quad \|\mu\| \leq \max_s \frac{1}{\alpha_s} \cdot \|\mathbf{e}\|^2$$

Also for the other estimators, there exists a positive constant λ' so that $\|\mu\| \leq \lambda'\|\mathbf{e}\|^2$ and thus

$$\|V[s_i]\| \leq \left(\frac{1}{(1 - \|\mathbf{A}\|)^2} + \lambda'\|\mathbf{V}\| \right) \|\mathbf{e}\|^2$$

and the theorem follows. □

Applied to (10.1), this implies that if an approximation $\tilde{\mathbf{x}}$ for $\mathbf{x}$ is available so that $\|\mathbf{e} + \mathbf{A}\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\| < \|\mathbf{e}\|$, it will be more economical to compute the correction $\Delta\mathbf{x}$ — and thus $\tilde{\tilde{\mathbf{x}}} = \tilde{\mathbf{x}} + \Delta\mathbf{x}$ as well — to given accuracy than to compute $\mathbf{x}$ directly. In particular, if $\tilde{\mathbf{x}} = \mathbf{x}$ is the true solution of the problem $\mathbf{x} = \mathbf{e} + \mathbf{A}\mathbf{x}$ already, then $\|\mathbf{e} + \mathbf{A}\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\| = 0$, so that a perfect estimator results. Good variance reduction is already obtained by using approximations $\tilde{\mathbf{x}}$ that are sufficiently close to $\mathbf{x}$.

10.1.2 Sequential correlated sampling

The basic idea of sequential correlated sampling [63, 65, 154] is to construct a sequence of gradually improving approximations $\mathbf{x}^{(k)}$ to $\mathbf{x}$ by successive application of (10.1): (10.1) with an initial approximation $\mathbf{x}^{(0)}$ yields a correction $\Delta\mathbf{x}^{(0)}$, $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}^{(0)}$ is filled in in (10.1) again, yielding a new correction $\Delta\mathbf{x}^{(1)}$ and approximation $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \Delta\mathbf{x}^{(1)}$ etc . . . .

It turns out that if the number of samples N for computing each correction $\Delta\mathbf{x}^{(k)}$ is chosen sufficiently large, the error $\|\mathbf{x}^{(k)} - \mathbf{x}\|$ after the k-th stage decreases geometrically — like $\mathcal{O}(\gamma^k)$ (with $\gamma < 1$) rather than like $\mathcal{O}(\sqrt{1/k})$ — in a probabilistic sense:

Proof: A sketch of a possible proof goes as follows: consider $V^{(k)} = \|V[s_i]\|$ on the

result of the k-th stage. With N random walks in each stage, theorem 10.1 yields

$$\begin{aligned} V^{(k)} &\leq \frac{\lambda}{N}\|\mathbf{e}+\mathbf{A}\mathbf{x}^{(k)}-\mathbf{x}^{(k)}\|^2 \\ &\leq \frac{\lambda}{N}\|(\mathbf{I}-\mathbf{A})\cdot(\mathbf{x}-\mathbf{x}^{(k)})\|^2 \\ &\leq \frac{\lambda}{N}\|\mathbf{I}-\mathbf{A}\|^2\cdot\|\mathbf{x}-\mathbf{x}^{(k)}\|^2. \end{aligned}$$

$\mathbf{x}-\mathbf{x}^{(k)}$ is the error made in the $(k-1)$-th stage: $E[(\mathbf{x}-\mathbf{x}^{(k)})^2]=V^{(k-1)}$. Assuming that the central limit theorem can be applied, with more than $(0.997)^n$ confidence, $\|\mathbf{x}-\mathbf{x}^{(k)}\|^2 \leq 3V^{(k-1)}$. We obtain

$$V^{(k)} \leq \frac{9\lambda\|\mathbf{I}-\mathbf{A}\|^2}{N}V^{(k-1)}$$

The number of samples N in each stage can always be chosen so that the ratio $\gamma = 9\lambda\|\mathbf{I}-\mathbf{A}\|^2/N$ is smaller than 1, so that $V^{(k)} \leq \gamma V^{(k-1)}$ with probability larger than $(0.997)^n$, and thus $V^{(k)} \leq \gamma^k V^{(0)}$ with $\gamma < 1$, with probability that can be made as high as desired by choosing N sufficiently large. □

10.2 Control variates in random walk radiosity

The technique that was outlined above can be applied to gathering random walk radiosity (§7.4.1). Three possibilities are:

10.2.1 Self-emitted illumination as control variate: initial shooting pass

Using $\tilde{B} = E$ in (4.25) yields:

$$\begin{aligned} \Delta B_i &= \left(E_i + \sum_j \rho_i F_{ij} E_j - E_i\right) + \sum_j \rho_i F_{ij}\Delta B_j \\ &= \left(\sum_j \rho_i F_{ij} E_j\right) + \sum_j \rho_i F_{ij}\Delta B_j \end{aligned}$$

The *reduced source term* $\sum_j \rho_i F_{ij} E_j$ is nothing else than the direct illumination in the scene. ΔB corresponds to the non-self-emitted illumination. The original problem of finding the radiosity B_i due to self-emitted light E has been transformed into an easier problem of computing indirect illumination for given direct illumination. Direct illumination is computed in an *initial shooting pass* before using gathering random walks for indirect illumination computation only [131].

Using an initial shooting pass is obligatory in practice in order to make gathering random walks usable: a gathering random walk will only yield a non-zero score if

it happens to hit at least one light source. The probability of hitting a light source can be very low in practice, because light sources cover only a small fraction of the total surface area in most scenes. The surfaces receiving direct illumination cover a much larger fraction of the total area, so that by using direct illumination as the source term, gathering random walks have a much higher probability of yielding a non-zero contribution.

Of course, in practice, direct illumination is computed with some error as well. This error will be propagated into indirect illumination and will persist in the final solution. One possibility to remove the error in direct illumination is to apply a per-pixel final gather pass after the indirect illumination computations. The propagated error in indirect illumination generally is negligible.

A more elaborate initial shooting pass, that yields a more "smooth" reduced source, has been proposed in [21].

10.2.2 Constant control variate

The only choice for $\tilde{\mathbf{x}}$ that allows $\mathbf{A}\tilde{\mathbf{x}}$ to be calculated analytically in the case of radiosity, is the constant choice $\tilde{B}_i = \beta$. With this choice, we get

$$\begin{aligned} \Delta B_i &= \left(E_i + \sum_j \rho_i F_{ij} \beta - \beta \right) + \sum_j \rho_i F_{ij} \Delta B_j \\ &= (E_i - (1-\rho_i)\beta) + \sum_j \rho_i F_{ij} \Delta B_j \end{aligned}$$

The question now is how to determine an optimal value for β. Heuristics for choosing β can be derived by minimising the expected mean square error

$$E[MSE] = \sum_i A_i V[s_i] = \sum_i A_i \mu_i v_{is} - \sum_i A_i \Delta B_i^2.$$

Since v_{is} is as hard to compute as the radiosity itself, we propose to minimise only $\sum_i A_i \mu_i$. The optimal value for β will depend on the estimator that is used:

- absorption estimator (7.26): $\mu_i = e_i^2/\alpha$. Substitution of $e_i = E_i - (1-\rho_i)\beta$ and $\alpha_i = 1 - \rho_i$, yields the following expression to minimise:

  $$F^A(\beta) = \sum_i \frac{A_i}{1-\rho_i} (E_i - (1-\rho_i)\beta)^2$$

  By computing the derivative of $F^A(\beta)$ w.r.t. β and requiring that it vanishes, we find

  $$\beta^A = \frac{\Phi_T}{\sum_i A_i (1-\rho_i)}.$$

- collision estimator (7.27): $\mu_i = e_i(2x_i - e_i) = e_i(e_i + 2(x_i - e_i)) \approx e_i^2$:

  $$F^C(\beta) = \sum_i A_i (E_i - (1-\rho_i)\beta)^2$$

The heuristic optimal value for β is

$$\beta^C = \frac{\sum_i A_i(1-\rho_i)E_i}{\sum_i A_i(1-\rho_i)^2}.$$

- survival estimator (7.28): $\mu_i = e_i(2x_i - e_i)/\sigma_i \approx e_i^2/\sigma_i$:

$$F^S(\beta) = \sum_i \frac{A_i}{\rho_i}(E_i - (1-\rho_i)\beta)^2$$

The heuristic optimal value for β is

$$\beta^S = \frac{\sum_i A_i(1-\rho_i)E_i/\rho_i}{\sum_i A_i(1-\rho_i)^2/\rho_i}.$$

10.2.3 Empirical results and discussion

The constant control variate technique will yield very good results if the radiosity to be computed is to good approximation constant in the scene. In the extreme case that $E_i + \rho_i = 1$ for instance, when also $B_i = 1$ is perfectly constant, the heuristic optimal value of β will be 1 as well, for all three estimators. The source term $E_i - (1-\rho_i)\beta = E_i - (1-\rho_i) = 0$, so that nothing remains to be propagated.

Figure 10.1 shows a scene with sufficiently constant indirect radiosity, so that the constant control variate yields a visible improvement when applied after an initial shooting pass. In practice, the constant control variate technique for gathering random walk radiosity yields only a small variance reduction (because radiosity is not constant), but the additional cost is however very low as well.

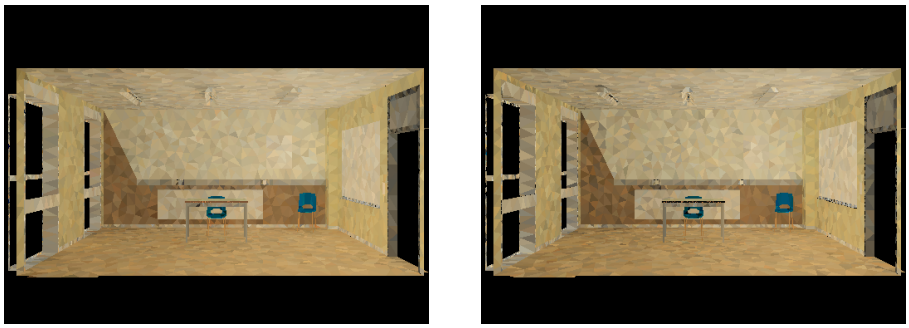


Figure 10.1: Indirect illumination in a simple office scene, computed with constant control variate (left) and without constant control variate (right) in collision gathering random walk radiosity. The constant control variate technique yields a slight, but visible, variance reduction. The RMS error is 27% smaller with constant control variate.

10.2.4 Sequential correlated sampling

There are two problems with the application of sequential correlated sampling to gathering random walk radiosity:

- Sequential Monte Carlo will definitively be preferred for very accurate results, since a Monte Carlo approach with independent samples has much slower convergence rate. The accuracy that is needed for image synthesis is typically of the order of magnitude of 1% relative error. It is questionable whether the number of samples needed in a single stage isn't too high in order to make sequential Monte Carlo competitive with other approaches, such as the shooting random walk, for such low accuracy;

- The source term for the k-th stage contains the matrix-vector product $\mathbf{A}\mathbf{x}^{(k)}$, which cannot be calculated analytically for radiosity since this would require that all form factors are known and stored in computer memory. It is questionable whether this matrix-vector product can be computed to sufficient accuracy at low enough cost to make sequential correlated sampling competitive with other Monte Carlo methods for radiosity.

10.3 Control variates in stochastic relaxation radiosity

10.3.1 Constant control variate

A constant control variate can also be beneficial in the context of stochastic Jacobi iterations, and leads to an improved version of the constant radiosity step proposed by Neumann et al. [114].

Consider the problem of computing a new power distribution P' for given power distribution P as follows:

$$P'_i - \Phi_i = \sum_i \sum_j P_j F_{ji} \rho_i \delta_{ik} \tag{10.3}$$

Now suppose $P_j = A_j B_j$ can be approximated by a constant radiosity β: $P_j = A_j\beta + A_j(B_j - \beta)$, then

$$P'_i - \Phi_i = A_i \rho_i \beta + \sum_i \sum_j A_j (B_j - \beta) F_{ji} \rho_i \delta_{ik}$$

The sums in the right hand side can be evaluated (simultaneously for all k) by Monte Carlo as follows:

1. Select a shooting patch j with probability $A_j \mid B_j - \beta \mid / \sum_s A_s \mid B_s - \beta \mid$;

2. Select a destination patch i using conditional probabilities $p_{i|j} = F_{ji}$, by tracing a uniformly distributed line originating at j;

3. Contribute

$$K^k_{ij} = \delta_{ik} \rho_i \sum_s A_s \mid B_s - \beta \mid \frac{A_j (B_j - \beta)}{A_j \mid B_j - \beta \mid}$$

   The ratio $A_j(B_j - \beta)/A_j \mid B_j - \beta \mid$ is $+1$ or -1 corresponding to whether $B_j > \beta$ or $B_j < \beta$ (the case $B_j = \beta$ will not occur).

The expectation is $E[K^k] = P_i' - \Phi_i - A_j \rho_i \beta$. The variance is determined by

$$E\left[(K^k)^2\right] = \rho_k \left(\sum_s A_s \mid B_s - \beta \mid \right) \cdot \sum_j A_j \mid B_j - \beta \mid F_{jk} \rho_k \tag{10.4}$$

10.3.2 Determination of the optimal constant radiosity

A very similar idea has been proposed by Neumann et al. [114], however from a different point-of-view. In [114], a heuristic optimal value for the constant radiosity β was obtained by projecting the current radiosity solution on a special radiosity hyperplane [109]. We found that the values of β, as derived there, do not always guarantee a reduction of the error.

A better value for β can be obtained by considering the variance of the Monte Carlo estimator described above. Expression (10.4) suggests to choose β so that

$$F(\beta) = \sum_s A_s \mid B_s - \beta \mid$$

is minimal. The function $F(\beta)$ is linear for values $\beta < B_{\min}$ or $\beta > B_{\max}$ in the scene: $\forall \beta < B_{\min}, \delta > 0$,

$$F(\beta - \delta) = \sum_s A_s (B_s - \beta + \delta) = \sum_s A_s (B_s - \beta) + A_T \delta = F(\beta) + A_T \delta$$

and $\forall \beta > B_{\max}, \delta > 0$,

$$F(\beta + \delta) = \sum_s A_s (\beta + \delta - B_s) = \sum_s A_s (\beta - B_s) + A_T \delta = F(\beta) + A_T \delta.$$

For values of β in between $B_{\min} < \beta < B_{\max}$, these two straight lines are connected by a smooth curve without local minima (see figure 10.2). In order to (probably) yield a reduction of the variance, the chosen value for β shall satisfy $F(\beta) < F(0) = \sum_s A_s B_s = P_T$. A simple numerical search algorithm guarantees that this requirement is fulfilled.

In order to minimise $F(\beta)$, a simple extension of the bisection method has been implemented. In the bisection method, an initial interval known to contain the minimum is successively refined by splitting it in two. The half interval containing the minimum is used as input for the subsequent refinement step. Because evaluation of $F(\beta)$ requires a full sweep through all the patches in the scene, we have found it more convenient to extend this method to 10 intervals. During each sweep, the value of $F(\beta)$ is determined simultaneously at values $\beta = B_{\min} + (B_{\max} - B_{\min}) \cdot k/10$, where $k = 0, \ldots, 10$. The interval, delimited by the smallest neighbouring values found for $F(\beta)$, was retained and refined in subsequent iterations until $\mid F(B_{\max}) - F(B_{\min}) \mid$ dropped below $10^{-4} F(0) = 10^{-4} P_T$.

The constant control variate technique can easily be applied in importance-driven stochastic Jacobi radiosity as well. It suffices to repeat the reasoning for expression (9.24) rather than (10.3). The expression to be minimised in order to determine the optimal value of β turns out to be

$$F^I(\beta) = \sum_s A_s \mid B_s - \beta \mid (I_s - V_s).$$

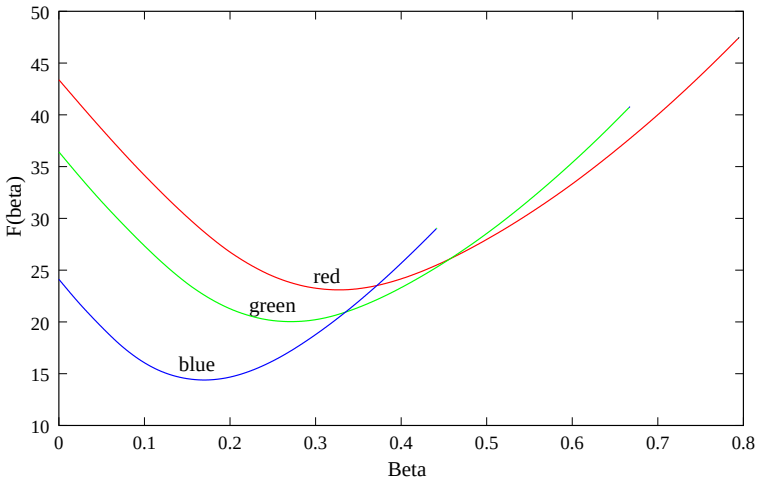


Figure 10.2: The curves $F(\beta)$ between $B_{\min}$ and $B_{\max}$ for the red, green and blue colour component in the test scene used in figure 10.3.

10.3.3 Empirical results and discussion

Figure 10.3 shows the results of the constant control variate technique in stochastic Jacobi iterations for the same scene as shown in figure 10.1. The reduction of the RMS error is somewhat larger in this case: 38% instead of 27%. The computation cost for determining a good value of β is however somewhat larger as well, but no more than a few percent of the total computation time.

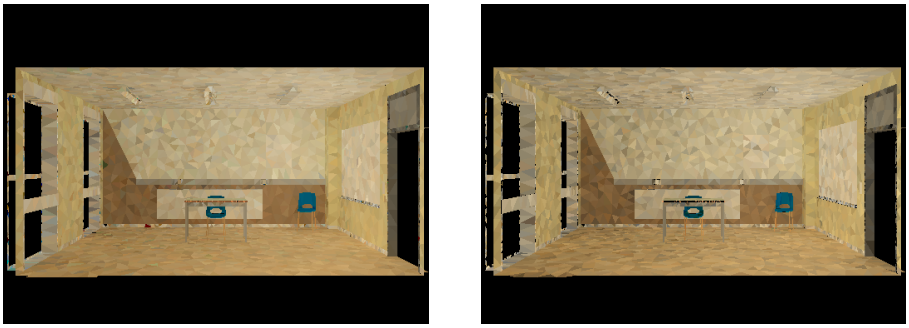


Figure 10.3: Indirect illumination in a simple office scene, computed with constant control variate (left) and without constant control variate (right) in stochastic Jacobi radiosity. The constant control variate technique yields a slight, but visible, variance reduction. The RMS error is 38% smaller with constant control variate.

10.4 Conclusion

In this chapter, the application of control variate variance reduction in Monte Carlo radiosity has been studied. The new results of this chapter are

- the constant control variate technique for gathering random walk radiosity (§10.2.2);
- an improved constant radiosity step (§10.3).

In practice, these techniques lead to a reduction of the variance by about 5-50%. The additional computation cost is however very low. A number of assumptions are made in the derivations however: the scene must be closed so that the sum of the form factors always equals 1 for all patches, and every surface in the scene must be illuminated. If these assumptions are not satisfied, the control variate techniques presented here may introduce a bias.

Sequential correlated sampling (§10.1.2) requires that matrix-vector products $\mathbf{A}\mathbf{x}^{(k)}$ are evaluated. Before sequential correlated sampling, promising geometric convergence rates, can be applied in the context of radiosity, an efficient solution to this problem needs to be found first. Even so however, it remains a question whether sequential correlated sampling will not be beneficial only for much higher accuracy than required for image synthesis.

11 Combining Estimators in Monte Carlo Radiosity

When more than one estimator exists for a given problem, the combination of estimates obtained by both independent or correlated sampling of each available estimator, can yield better estimates for a quantity to be computed. In this chapter, the combination of radiosity estimators based on gathering (equation (4.25)) and shooting (equation (4.28)), will be proposed. First, the combination of gathering and shooting in random walk radiosity will be discussed (§11.1). Section §11.2 deals with the combination of gathering and shooting in stochastic relaxation radiosity.

11.1 Combining gathering and shooting random walk radiosity

Gathering and shooting collision random walk radiosity, as explained in §7.4, are based on random walks with the same transition probabilities $p_{ij} = \rho_i F_{ij}$. We will discuss here how a single set of random walks with such transition probabilities can be used in order to obtain gathering as well as shooting estimates for the radiosity on all patches simultaneously. Three strategies will be proposed in order to combine these estimates into a single radiosity estimate at a negligible additional cost. The resulting estimate will be better than either the gathering or shooting estimates alone. The combination of gathering and shooting collision random walk radiosity has been developed in the context of this dissertation in collaboration with M. Sbert [134].

First, the relation between the collision gathering and shooting estimators will be investigated (§11.1.1). It will be explained in detail how the gathering random walk estimator can be used in order to obtain radiosity estimates on all patches in the scene simultaneously. Gathering and shooting over sub-paths of a random walk can be viewed as alternative importance sampling estimators for the same quantity. This leads to a first strategy to robustly combine gathering and shooting on certain patches (§11.1.2). Alternative strategies, based on variance approximations, are proposed in (§11.1.3). This section is concluded with some empirical results and a discussion (§11.1.4).

11.1.1 The relation between gathering and shooting

Gathering over sub-paths

Consider a random walk $J = j_0, j_1, \ldots, j_\tau$, generated with birth probabilities π_i, analog transitions probabilities $p_{ij} = \rho_i F_{ij}$ and corresponding termination probabilities $\alpha_i = 1 - \rho_i$. The score of the collision gathering random walk with source term

estimation suppressed, for estimating the radiosity on a patch k, is (see §7.4.1):

$$s^G(j_0,\ldots,j_\tau) = \frac{\delta_{kj_0}}{\pi_{j_0}} \sum_{t=1}^{\tau} \rho_{j_0} E_{j_t} = \sum_{t=1}^{\tau} \delta_{kj_0} \frac{\rho_{j_0} E_{j_t}}{\pi_{j_0}}.$$

A gathering random walk can be used in order to obtain gathering contributions at every visited patch $j_0,\ldots,j_{\tau-1}$ but the last, rather than only at the origin j_0 in the following way:

$$\tilde{s}^G(j_0,\ldots,j_\tau) = \sum_{r=0}^{\tau-1} \sum_{t=r+1}^{\tau} \mathcal{S}^{\leftarrow}(j_r,\ldots,j_t)\delta_{kj_r}$$

where $S^{\leftarrow}(j_r,\ldots,j_t)$ denotes the elementary gathering score contributed by the sub-path:

$$\mathcal{S}^{\leftarrow}(j_r,\ldots,j_t) = \frac{\rho_{j_r} E_{j_t}}{p_{j_r}}. \tag{11.1}$$

p_i is the sub-path birth density, satisfying

$$p_i = \pi_i + \sum_j p_j F_{ji} \rho_i. \tag{11.2}$$

$j_0,\ldots,j_\tau$ is called a *covering random walk* for the sub-paths $j_r,\ldots,j_t$. As presented here, a covering random walk $j_0,\ldots,j_\tau$ that visits a given patch i twice, gives rise to two separate sub-paths originating at i. These sub-paths contribute a different score. With each sub-path, a probability can be associated:

$$\mathcal{P}(j_r,\ldots,j_t) = p_{j_r} F_{j_r j_{r+1}} \rho_{j_{r+1}} F_{j_{r+1} j_{r+2}} \cdots \rho_{j_{t-1}} F_{j_{t-1} j_t} \tag{11.3}$$

Shooting over sub-paths

Now consider the collision shooting random walk estimator scores (7.4.2) for the radiosity on k:

$$s^S(j_0,\ldots,j_\tau) = \frac{\rho_k}{A_k} \frac{\Phi_{j_0}}{\pi_{j_0}} \sum_{t=1}^{\tau} \delta_{j_t k} = \sum_{t=1}^{\tau} \mathcal{S}^{\rightarrow}(j_0,\ldots,j_t)\delta_{kj_t}$$

where $\mathcal{S}^{\rightarrow}(j_r,\ldots,j_t)$ denotes the elementary shooting score contributed by a sub-path $j_r,\ldots,j_t$:

$$\mathcal{S}^{\rightarrow}(j_r,\ldots,j_t) = \frac{\Phi_{j_r}\rho_{j_t}}{p_{j_r} A_{j_t}}. \tag{11.4}$$

In theory, a covering shooting random walk can be used in order to compute the radiosity at all visited patches $j_t, t > 0$ due to every previously visited patch $j_r, r < t$ instead of only at the origin j_0:

$$\tilde{s}^S(j_0,\ldots,j_\tau) = \sum_{t=1}^{\tau} \sum_{r=0}^{t-1} \mathcal{S}^{\rightarrow}(j_r,\ldots,j_t)\delta_{kj_t} \tag{11.5}$$

In practice, shooting from other patches j_r than the origin j_0 of the covering path, is only possible if the sub-path birth densities p_{j_r}, needed in the scores (11.4), are known *in advance*. The sub-path birth densities, corresponding to $p_{ij} = \rho_i F_{ij}$ and source term estimation suppression, fulfil equation (11.2). The solution of this system of equations is as hard as the solution of the radiosity system of equations itself. With $p_{ij} = \rho_i F_{ij}$, shooting is thus only possible from the origin $j_r = j_0$ of the covering path.

Global uniform lines (§5.3.3) are however a notorious exception to this rule: with global uniform lines, used the global multi-path method [137], $\pi_i = A_i/A_T$ and $p_{ij} = F_{ij}$. In this case,

$$p_i = \frac{A_i}{A_T} + \sum_j p_j F_{ji}$$

It is easy to verify that $p_i = A_i/A_T = \pi_i$ is the solution of this equation, so that shooting from other patches than the origin j_0 of a random walk is possible in global multi-path methods. Global lines however allow less freedom in choosing the intersection densities with the patches in the scene.

The reason that gathering over sub-paths with origin $j_r \neq j_0$ is possible, is because it is sufficient to estimate the densities p_{j_r} *a posteriori*: gathering contributes a score to the *origin* of a sub-path, so that counting the number of contributions is sufficient. With shooting over sub-paths, a score is recorded at the *destination* j_t. For each destination, the sub-path birth probability on arbitrary other patches is needed.

Gathering is shooting over reversed sub-paths

Consider a sub-path $j_r, \ldots, j_t$. With the collision gathering random walk, a score $\mathcal{S}^{\leftarrow}(j_r, \ldots, j_t)$ is contributed to the radiosity at j_r with probability $\mathcal{P}(j_r, \ldots, j_t)$. This corresponds with the sampling of the following term in the Neumann expansion of the radiosity at j_r:

$$\begin{aligned}\mathcal{S}^{\leftarrow}(j_r, \ldots, j_t)\mathcal{P}(j_r, \ldots, j_t) &= \frac{\rho_{j_r} E_{j_t}}{p_{j_r}} p_{j_r} F_{j_r j_{r+1}} \rho_{j_{r+1}} F_{j_{r+1} j_{r+2}} \cdots \rho_{j_{t-1}} F_{j_{t-1} j_t} \\ &= \rho_{j_r} F_{j_r j_{r+1}} \rho_{j_{r+1}} F_{j_{r+1} j_{r+2}} \cdots \rho_{j_{t-1}} F_{j_{t-1} j_t} E_{j_t}. \quad (11.6)\end{aligned}$$

Now consider the probability associated with the reversed sub-path $j_t, \ldots, j_r$:

$$\begin{aligned}\mathcal{P}(j_t, \ldots, j_r) &= p_{j_t} F_{j_t j_{t-1}} \rho_{j_{t-1}} \cdots \rho_{j_{r+1}} F_{j_{r+1} j_r} \\ &= \frac{p_{j_t}}{A_{j_t}} A_{j_t} F_{j_t j_{t-1}} \rho_{j_{t-1}} \cdots \rho_{j_{r+1}} F_{j_{r+1} j_r} \quad (11.7) \\ &= \frac{p_{j_t}}{A_{j_t}} \frac{A_{j_r}}{p_{j_r}} p_{j_r} F_{j_r j_{r+1}} \rho_{j_{r+1}} F_{j_{r+1} j_{r+2}} \cdots \rho_{j_{t-1}} F_{j_{t-1} j_t} \\ &= \frac{p_{j_t}}{A_{j_t}} \frac{A_{j_r}}{p_{j_r}} \mathcal{P}(j_r, \ldots, j_t)\end{aligned}$$

Such a reversed sub-path yields a shooting score $S^{\rightarrow}(j_t, \ldots, j_r)$ at j_r with probability $\mathcal{P}(j_t, \ldots, j_r)$. This corresponds to sampling the same term in the Neumann expansion

of the radiosity at j_r as above:

$$\begin{aligned} S^{\rightarrow}(j_t,\ldots,j_r)\mathcal{P}(j_t,\ldots,j_r) &= \frac{\Phi_{j_t}\rho_{j_r}}{p_{j_t}A_{j_r}}\frac{p_{j_t}}{A_{j_t}}\frac{A_{j_r}}{p_{j_r}}\mathcal{P}(j_r,\ldots,j_t) \\ &= \frac{\rho_{j_r}E_{j_t}}{p_{j_r}}\mathcal{P}(j_r,\ldots,j_t) \quad = \quad (11.6). \end{aligned}$$

We conclude that shooting over reversed paths is an alternative importance sampling estimator for gathering over direct paths. The same is true for gathering over reversed paths versus shooting over direct paths. This observation allows to consider the combination of shooting and gathering over sub-paths as a case of multiple importance sampling [176].

11.1.2 Gathering and shooting as a case of multiple importance sampling

Each covering random walk $j_0,\ldots,j_\tau$ yields a set of sub-paths that can be used for gathering and shooting simultaneously. Gathering over a sub-path $j_r,\ldots,j_t$ results in a contribution $S^{\leftarrow}(j_r,\ldots,j_t)$ to the radiosity B_{j_r} at the sub-path origin j_r. Shooting results in a contribution $S^{\rightarrow}(j_r,\ldots,j_t)$ to the radiosity B_{j_t} at the destination patch j_t. There is however always a possibility that a sub-path is generated in reverse sense $j_t,\ldots,j_r$ (as part of some other covering path), in which case gathering yields a contribution $S^{\leftarrow}(j_t,\ldots,j_r)$ at j_t and shooting a contribution $S^{\rightarrow}(j_t,\ldots,j_r)$ at j_r.

The basic idea of multiple importance sampling [176], translated to this context, is to assign each sampled sub-path a weight that takes into account the probability $\mathcal{P}(j_t,\ldots,j_r)$ that it would have been generated in reverse sense: if there is a large probability that the sub-path would have been generated in reverse sense, it is given a small weight. If generating a sub-path in reverse sense is unlikely, its shooting and gathering score are given a large weight. As long as the sum of the weights for the direct and reversed sub-path equals 1, the result will be unbiased. One possible heuristic for determination of the weights is the so called balance heuristic. Applied to gathering and shooting over sub-paths, the scores of a sub-path $j_r,\ldots,j_t$ are assigned a weight (11.7)

$$\begin{aligned} w(j_r,\ldots,j_t) &= \frac{\mathcal{P}(j_r,\ldots,j_t)}{\mathcal{P}(j_r,\ldots,j_t)+\mathcal{P}(j_t,\ldots,j_r)} \\ &= \frac{p_{j_r}A_{j_t}}{p_{j_r}A_{j_t}+p_{j_t}A_{j_r}} \end{aligned}$$

The weighted scores of the sub-path so become:

$$\text{gathering (at } j_r\text{):} \qquad \frac{\rho_{j_r}\Phi_{j_t}}{p_{j_r}A_{j_t}+p_{j_t}A_{j_r}} \tag{11.8}$$

$$\text{shooting (at } j_t\text{):} \qquad \frac{\rho_{j_t}\Phi_{j_r}}{p_{j_r}A_{j_t}+p_{j_t}A_{j_r}}. \tag{11.9}$$

The radiosity at each patch is obtained by accumulating the weighted scores shown above, and dividing by the number N of covering paths.

Discussion Multiple importance sampling is a provably good and reliable way of combining estimators.

Unfortunately, it can only be applied for sub-paths that are suited for both gathering and shooting at the same time. In particular, with the analog random walk estimators (with $p_{ij} = \rho_i F_{ij}$), it can only be used on sub-paths $j_0, \ldots, j_t$ originating at the origin of the covering path $j_0, \ldots, j_\tau$: only in that case, the birth densities $p_i = \pi_i$ are known in advance. When the covering paths are analog shooting paths — the approach that makes most sense —, a combination with gathering is only possible at the light sources. With an initial shooting pass (§10.2.1), the illumination on patches with direct illumination can be improved, but not the illumination on patches that receive only indirect illumination: no suitable sub-paths for shooting originate at these.

Combined gathering and shooting in global multi-path radiosity Global multi-path radiosity is an exception to the rule above. In global multi-path radiosity, the sub-path birth densities $p_i = A_i/A_T$ are known in advance, so that the combination of gathering and shooting is possible over all sub-paths. The probability associated with each sub-path $j_r, \ldots, j_t$ is slightly different from above:

$$\mathcal{P}(j_r, \ldots, j_t) = p_{j_r} F_{j_r j_{r+1}} F_{j_{r+1} j_{r+2}} \cdots F_{j_{t-1} j_t}$$

The shooting and gathering weights can be computed in the same way as above however, and they turn out to be $w(j_r, \ldots, j_t) = 1/2$, always: shooting and gathering receive the same weight. The observation that global lines can be used bidirectionally with equal weights in both directions was made before by M. Sbert [130, 137, 131]. The balance heuristic confirms his observation.

11.1.3 Combination based on variance estimates

In order to be able to combine shooting and gathering estimates on more patches than only the sources, the more general combination strategy of §4.3.4 can be used. The basic idea is to compute gathering and shooting estimates for the radiosity on each patch separately first. Combination weights are chosen afterwards, inverse proportional to estimates for the variances. Unlike with multiple importance sampling, these combination weights are thus equal for all sub-paths contributing to a patch. On the other hand, also gathering contributions from sub-paths that are not suited for shooting, because the required sub-path birth density is not known in advance, can be taken into account.

First, the simultaneous computation of separate gathering and shooting radiosity estimates will be discussed. Next, variance-based combination strategies are proposed.

Simultaneous computation of gathering and shooting radiosity estimates

The random walks $J = j_0, \ldots, j_\tau$ are traced as in normal shooting random walk radiosity (algorithm 19): $\pi_i = \Phi_i/\Phi_T$, $p_{ij} = \rho_i F_{ij}$, with source term estimation suppression as explained in §7.3.3. A gathering and shooting estimate for the received radiosity b_i at each patch i,is obtained by averaging gathering and shooting scores from the traced random walks as follows (figure 11.1 illustrates the contributions of a single random walk):

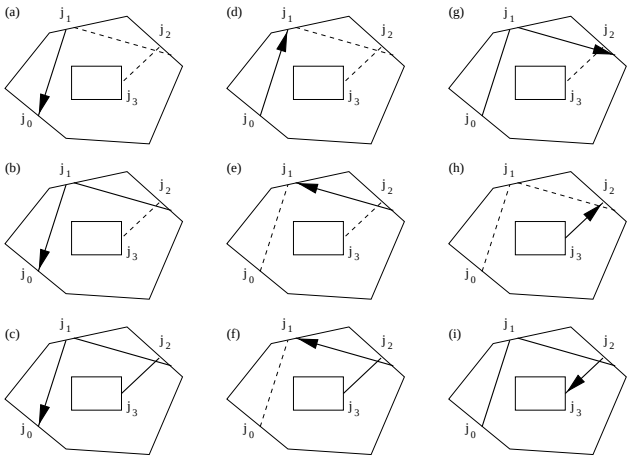


Figure 11.1: Contributions of a random walk j_0, j_1, j_2, j_3: (a,b,c) gathering at j_0; (d) shooting at j_1; (e,f) gathering at j_1; (g) shooting at j_2; (h) gathering at j_2; (i) shooting at j_3.

The shooting radiosity estimates are computed as in normal shooting random walk radiosity (algorithm 19): a shooting radiosity score accumulator $\tilde{B}_i^S$ is kept and initialised to zero for each patch i. Each time a random walk visits the patch i, a score

$$\frac{\Phi_{j_0}}{\pi_{j_0}} \frac{\rho_i}{A_i} = \frac{\rho_i \Phi_T}{A_i}$$

is added to $\tilde{B}_i^S$. Finally, $\tilde{B}_i^S$. is divided by N, the number of traced random walks, and self-emitted radiosity E_i is added.

The gathering estimates are computed as follows: a gathering score accumulator $\tilde{B}_i^G$ is initialised to 0 for each patch i as with shooting, but also a gathering score counter N_i^G, initialised to 0 as well, is kept. Each time a random walk visits a given patch i, a gathering score is added to $\tilde{B}_i^G$, and the gathering score count N_i^G is increased by one. Each individual visit counts: if a random walk visits a patch twice, two scores result. Suppose that the r-th patch j_r visited by a random walk $j_0, \ldots, j_\tau$ happens to be i. The gathering score at $i = j_r$ then is

$$\rho_i \sum_{t=r+1}^{\tau} E_{j_t}$$

Finally, $\tilde{B}_i^G$ is divided by N_i^G, the number of scores, and self-emitted radiosity E_i is added.

In this way, two independent radiosity estimates $\tilde{B}_i^G$ and $\tilde{B}_i^S$ are obtained. These estimates are independent because the gathering scores only depend on the "future" of a random walk while the shooting scores only depend on the "past" and the random walks are Markovian: the next patch that is visited depends on the current patch, but not on previously visited patches, so that the "future" of a path is independent of the past, before the current patch.

The gathering estimates are obtained simultaneously with the shooting estimates at a negligible additional cost: the extra work caused by computation and accumulation of the gathering scores is much smaller than the cost of tracing the random walks. There is an increase in storage requirements however: for each patch, not only a shooting score accumulator, but also a gathering score accumulator and gathering score count need to be kept in memory.

Optimal a-posteriori combination weights

The optimal a-posteriori combination $\alpha_i \tilde{B}_i^S + \beta_i \tilde{B}_i^G$ is obtained by choosing $\alpha_i + \beta_i = 1$ with each coefficient inverse proportional to the variance of the corresponding estimators (§4.3.4):

$$\frac{\beta_i}{\alpha_i} = \frac{V[\hat{B}_i^S]/N}{V[\hat{B}_i^G]/N_i^G}. \tag{11.10}$$

Closed formulae for the variances $V[\hat{B}_i^S]$ and $V[\hat{B}_i^G]$ were given in §7.4:

$$V[\hat{B}_i^G] = \rho_i \sum_s (E_s + 2b_s) b_{is} - b_i^2 \tag{11.11}$$

$$V[\hat{B}_i^G] = \frac{\rho_i}{A_i} \Phi_T (1 + 2\zeta_i) b_i - b_i^2 \tag{11.12}$$

Unfortunately, these formulae require very detailed knowledge of the radiosity solution, in particular the radiosity b_{is} received at each patch i due to each light source s. Such information is not available in practice. Near-optimal weights can be obtained however by using approximations for the variances in (11.10). We study two alternatives: heuristic approximations and the use of sample estimates of the variances.

Heuristical combination

A very simple but reasonably good heuristic for determining the weights [134] is obtained by introducing the following assumptions:

- ζ_i, the fraction of power at i due to own emission, is small in (11.12);
- $\sum_s (E_s + 2b_s) b_{is} \approx (\sum_s A_s (E_s + 2b_s)/A_T) \cdot \sum_s b_{is}$ in (11.11);
- Almost every patch in a scene receives direct illumination. With this assumption, almost every patch i can be considered a source patch after a initial shooting pass (§10.2.1) so that $\sum_s A_s b_s \approx \sum_i A_i b_i \approx \frac{\rho_{\text{av}} \Phi_T}{1 - \rho_{\text{av}}}$;

With these assumptions, the following approximation for (11.10) is obtained:

$$\frac{\beta_i}{\alpha_i} \approx \frac{N_i^G A_T}{N A_i} \cdot k \quad \text{with optimal } k = \frac{1 - \rho_{\text{av}}}{1 + \rho_{\text{av}}}. \tag{11.13}$$

The expected number of gathering contributions corresponds to the sub-path birth density p_i (11.2):

$$E[N_i^G] = N p_i \quad \text{with:} \quad p_i = \frac{\Phi_i}{\Phi_T} + \sum_j p_j F_{ji} \rho_i$$

so that $E[N_i^G] = NP_i/\Phi_T$. The following alternative expression for (11.13) is obtained:

$$\frac{\beta_i}{\alpha_i} \approx \frac{B_i}{E_{\text{av}}} \cdot k \quad \text{with optimal } k = \frac{1-\rho_{\text{av}}}{1+\rho_{\text{av}}}. \tag{11.14}$$

A gathering estimate will receive a larger weight on bright surfaces and a smaller weight on dim surfaces, because the number of gathering contributions will be higher on bright surfaces.

Combination based on sample estimates of the variance

The main disadvantage of a heuristical combination like above is that several rough approximations need to be made. An alternative is to use estimates for the variance $V[\hat{B}_i^G]$ and $V[\hat{B}_i^S]$ in (11.10), that are based on the observed shooting and gathering scores themselves. This can be done according to formula (4.10) and requires that not only the scores themselves are accumulated while tracing random walks, but also the square of the scores.

This approach yields optimal combination weights eventually, at the cost of slightly higher storage requirements for accumulating the sum of the square contributions besides the sum of the contributions themselves. It is however not necessary to keep the square of the scores for every colour band in a multi-channel spectral renderer: if s_λ denotes a score at wavelength band λ in a colour representation using Λ spectral values, it is sufficient to accumulate $\sum_{\lambda=1}^{\Lambda} s_\lambda^2$. Only two extra floating point numbers per patch are needed: one for the gathering and one for the shooting square scores.

A second disadvantage of this approach is that the variance estimates and the resulting combination weights may be unreliable for low number of samples. A worse estimator may receive a too large weight in the beginning of the computations. The weights however improve together with the variance estimates as the number of samples in increased.

11.1.4 Empirical results and discussion

Figure 11.2 shows the results obtained for the same office scene that was used also in chapter 10 (figures 10.1 and 10.3). This scene consists of about 25,000 patches. Combining gathering and shooting results in slight improvements with each of the presented combination strategies. The images, showing indirect illumination, have been obtained with about 50,000 random walks taking about 125,000 rays. The middle-right image, showing the combined result based on sample variance, exhibits some defects which are due to unreliable sample-based variance estimates for this low number of samples (about 5 rays per patch on the average).

In our experience, combining gathering and shooting yields improvements that are comparable with the use of the control variate technique of §10.2.2: a reduction of the variance by 5-50% is common. The additional computation cost is however negligible, but there is a slightly higher storage cost, due to the need to store separate gathering and shooting estimates, the gathering contribution counts, and the accumulated square scores.

Figure 11.3 shows the colour-coded combination weights obtained with the heuristic of expression (11.13) and based on sample estimates of the variance. A more reddish colour indicates that a higher weight is given to gathering. The heuristic (11.13)

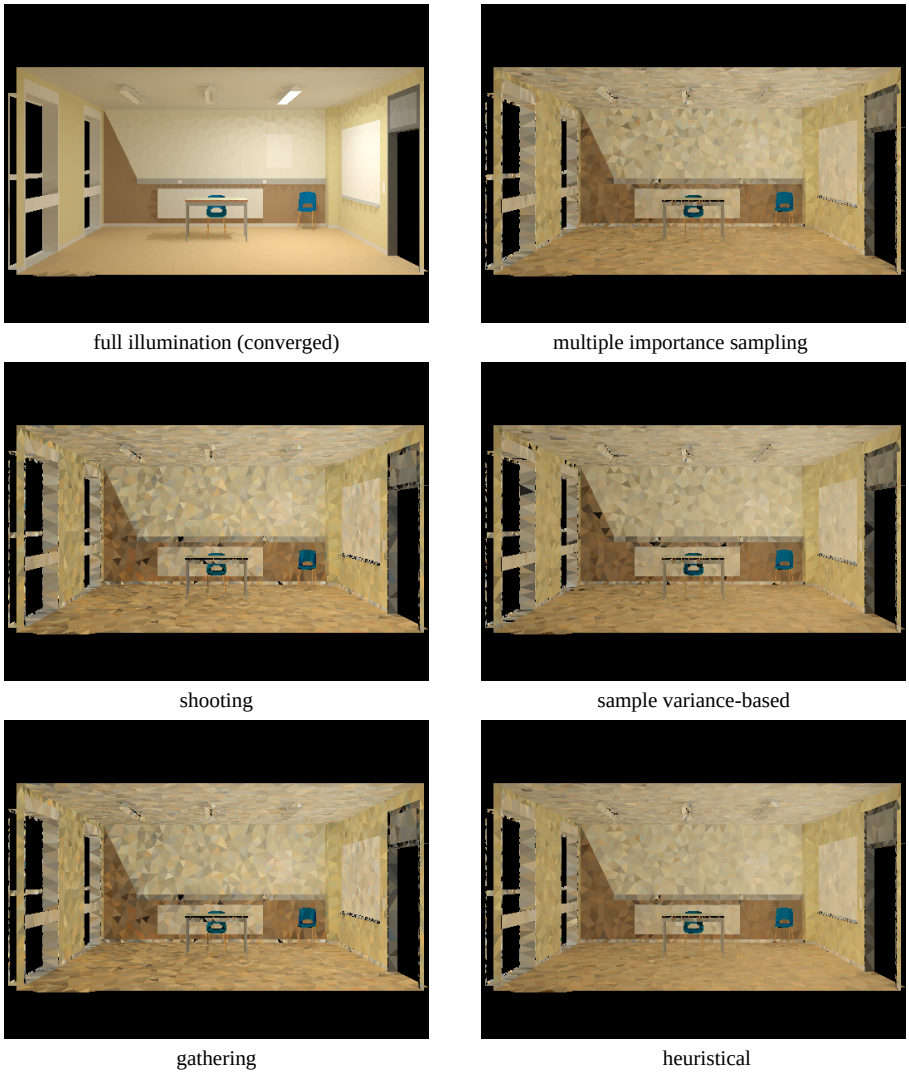
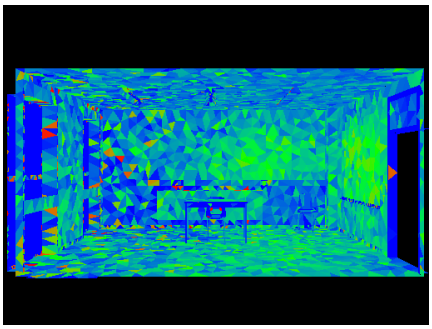
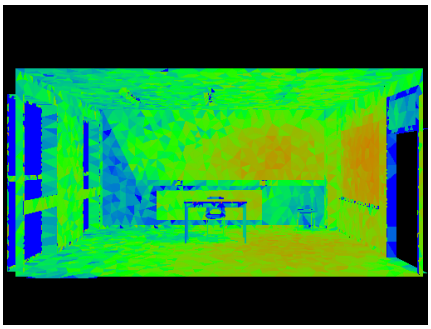


Figure 11.2: Results obtained by combining gathering and shooting random walk estimates for the indirect illumination in the scene shown (with also direct illumination) in the top-left image: shooting-only solution (middle-left, 0.046 normalised RMS error), gathering-only solution (bottom-left, 0.052 RMS error), combination based on multiple importance sampling (§11.1.2, top-right, 0.041 RMS error), using sample-based estimates of the variance (middle-right, 0.051 RMS error) and using the heuristic of expression (11.13) (bottom-left image, 0.042 RMS error). The relatively worse results with the sample-based variance estimates is due to the low quality of these estimates for the low number of samples (125,000 rays) that was used. With a higher number of samples, the combination weights will improve together with the variance estimates.



sample-variance based



heuristical

Figure 11.3: Colour coded weights used in the middle- and bottom-right images in figure 11.2: weights based on the sample-estimates of the variances (left) and heuristical weights (right). A more reddish colour indicates a larger weight for gathering. The heuristic of expression (11.13) assigns a higher weight to gathering than when sample-estimates of the variance are used. At this low number of samples, the weights based on sample-estimates of the variance are however not yet reliable and fluctuate largely from patch to patch.

assigns a higher weight to gathering than the combination strategy based on sample estimates for the variance. This is due to the approximations that are made in the derivation of the heuristic. The weights based on sample-estimates of the variance change abruptly from patch to patch. This is an indication of the unreliability of the variance estimates at the low number of samples used for these images. Eventually, for large number of samples, the optimal weights would be obtained. An interesting area for future research may be to design a combination strategy that uses heuristical weights at low sample size and the weights based on sample-estimates of the variances later on during the computations.

The variance-based combination strategies are asymptotically unbiased Since $E[\alpha_i \hat{B}_i^S + \beta_i \hat{B}_i^G] = E[\hat{B}_i^S + \beta_i(\hat{B}_i^G - \hat{B}_i^S)]$ and $E[\hat{B}_i^S] = B_i$, the bias is given by $E[\beta_i(\hat{B}_i^G - \hat{B}_i^S)]$. Since $\beta_i \leq 1$, the bias is bounded by $E[|\hat{B}_i^G - \hat{B}_i^S|] = \sqrt{\frac{2}{\pi}(V[\hat{B}_i^S]/N + V[\hat{B}_i^G]/N_i^G)}$ for sufficiently large N and N_i^G so that the central limit theorem applies.

Related work The combination of gathering and shooting algorithms has been studied before in bidirectional progressive refinement radiosity [42]: and bidirectional path tracing [96, 98, 175, 176]:

- Dutré et al. [42] describe a bidirectional progressive refinement algorithm in order to compute the radiosity of individual patches in turn by alternating steps in which radiosity or importance is propagated. In the bidirectional random walk radiosity algorithm that is proposed here, only power is propagated in order to compute the radiosity on all patches in the scene simultaneously. Random walks are used instead of deterministic Southwell relaxation;

- Lafortune and Veach [96, 98, 175, 176] have proposed bidirectional algorithms

for computing the average flux through each pixel in an image. Here, a world-space rather than image-space solution is computed. A second major difference is that here, we are interested in computing the illumination at both ends of a path, while in bidirectional path tracing, only the illumination at the viewer is of interest. A third difference is that in bidirectional path tracing, pairs of random walks, one originating at a light source, and one originating at the viewing position, are connected through next event estimators. The bidirectional random walk radiosity algorithm here, uses individual random walks bidirectionally, without next event estimators.

11.2 Combining gathering and shooting in stochastic Jacobi radiosity

Similar techniques for combining gathering and shooting in random walk radiosity, can be applied to stochastic Jacobi iterations (§6.4). The goal here is to obtain a score at both end-points of every line that is traced, instead of only at one end-point. In some cases, the result is the same as if the number of sampled lines is doubled. Unlike in random walk radiosity, the combination of gathering and shooting here can be done easily with multiple importance sampling.

11.2.1 Gathering and shooting in stochastic Jacobi iterations

Stochastic Jacobi iterations (§6.4), can be applied to solve both the system of classical radiosity equations (gathering):

$$B_k - E_k = \sum_i \sum_j \delta_{ik} \rho_i F_{ij} B_j$$

and, on the other hand, the power system (shooting):

$$P_k - \Phi_k = \sum_i \sum_j P_i F_{ij} \rho_j \delta_{jk}.$$

In both cases, the sum on the right hand side can be estimated by selecting terms, determined by the pair of indices $(i.j)$, according to some probability

$$\mathcal{P}_{ij} = p_i F_{ij}.$$

Selecting a patch j conditional on a patch i, according to the form factor probabilities F_{ij}, can be carried out by tracing a local or global uniformly distributed line originating at i, and determining the patch j containing its next intersection point with the surfaces in the scene. p_i denotes the probability (normalised) of selecting patch i for "shooting" such a line. In §6.4, where the application to the shooting equations was studied, p_i was chosen $p_i = P_i/P_T$,with $P_T = \sum_s P_s$, the total power in the scene. An arbitrary segment of a global line through the scene, constructed with the plane-intercept or two-points-on-a-bounding-sphere technique (§5.3.3), has a probability $p_i = A_i/A_T$ of intersecting a patch i.

The *radiosity* contributions associated by a line from i to j are:

$$\text{Shooting (at } j\text{):} \quad \mathcal{S}_{ij}^{\rightarrow} = \frac{P_i \rho_j}{p_i A_j} \tag{11.15}$$

$$\text{Gathering (at } i\text{):} \quad \mathcal{S}_{ij}^{\leftarrow} = \frac{\rho_i B_j}{p_i}. \tag{11.16}$$

Just like sub-paths in the random walk method, a line contributes only to either its origin i (in gathering), or to its destination j (in shooting). We would like to have a contribution at both end-points.

11.2.2 Gathering and shooting as a case of multiple importance sampling

It is very easy to see that shooting corresponds with reserve gathering in this case (and vice versa, gathering with reverse shooting):

$$\mathcal{S}_{ij}^{\rightarrow} \mathcal{P}_{ij} = \frac{P_i \rho_j}{p_i A_j} p_i F_{ij} = \rho_j \frac{A_i}{A_j} F_{ij} B_i = \rho_j F_{ji} B_i = \frac{\rho_j B_i}{p_j} p_j F_{ji} = \mathcal{S}_{ji}^{\leftarrow} \mathcal{P}_{ji}.$$

Again, shooting and reverse gathering can be viewed as alternative importance sampling estimators (for the shooting equation), so that the multiple importance sampling framework [176] can be applied. The balance heuristic yields the following weight for a line from i to j:

$$w_{ij} = \frac{\mathcal{P}_{ij}}{\mathcal{P}_{ij} + \mathcal{P}_{ji}} = \frac{p_i A_j}{p_i A_j + p_j A_i} \tag{11.17}$$

The weighted scores at both end-points of the line are:

$$w_{ij} S_{ij}^{\leftarrow} = \frac{\rho_i P_j}{p_i A_j + p_j A_i} \quad \text{on } i \tag{11.18}$$

$$w_{ij} S_{ij}^{\rightarrow} = \frac{\rho_j P_i}{p_i A_j + p_j A_i} \quad \text{on } j \tag{11.19}$$

Unlike in random walk radiosity, the probabilities p_i are always known in stochastic Jacobi iterations:

- Local lines:

$$p_i = \frac{P_i}{P_T} \quad \Rightarrow \quad w_{ij} = \frac{B_i}{B_i + B_j} \tag{11.20}$$

  A line shot from a bright patch i towards a dark patch j receives a high weight. Vice versa, if line from a dark to a bright patch receives a low weight, because there is a high probability that another line is shot in reverse sense. When connecting two patches with equal radiosity, the line has weight $w_{ij} = 1/2$;

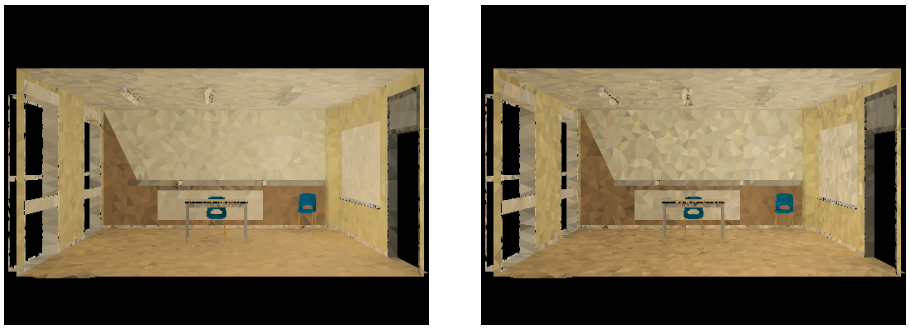
- Global line segments (or appropriately crafted local lines):

$$p_i = \frac{A_i}{A_T} \quad \Rightarrow \quad w_{ij} = \frac{1}{2}.$$

  This is the case with for instance global line bundles in the trans-illumination method [110, 165].

11.2.3 Empirical results and discussion

Figure 11.4 shows the result of combining gathering and shooting in stochastic Jacobi radiosity for the, now familiar, office scene. The reduced variance is very clearly visible. Although the gain is not so large in all scenes, the combined result will never be worse than the uncombined result either. Additional computation work is absolutely negligible and there is no extra storage cost. Considering that the implementation is extremely easy and that the technique combines very well with both the use of both view importance and the constant control radiosity variate technique of §10.3, there are no reasons not to use this technique.



bi-directional uni-directional

Figure 11.4: There are no reasons not to combine gathering and shooting in stochastic relaxation radiosity. Although the gain is not always as large as in this case, the combined result will never be worse either. The technique combines well with the other variance reduction techniques and is very easy to implement. Additional computation work is absolutely negligible and no extra storage is required.

11.3 Conclusion

In this chapter, the relation between gathering and shooting random walk radiosity has been analysed, showing how gathering random walks can be used in order to simultaneously estimate the radiosity on all patches in the scene. Gathering over a sub-path corresponds to shooting over a reversed sub-path so that a combination is possible based on multiple importance sampling. Unfortunately, this combination strategy only allows results to be combined on light sources, or on patches with direct illumination if only indirect illumination is being computed. Global multi-path radiosity [137] is an exception to this rule.

In order to be able to combine gathering and shooting estimates on other patches than only the light sources, combination strategies based on approximations of the variances have been proposed. In a first strategy, the closed form expressions for the variance of the collision shooting and gathering random walk estimators for radiosity have been approximated, leading to a simple heuristic for the combination weights. Alternatively, optimal combination weights will be obtained eventually when using sample-estimates for the variance. At low sampling densities however, these sample-variance based weights are not so reliable.

In all cases, a fair variance reduction can be obtained at very little additional cost (mainly storage). A possible topic for future research is the investigation of robust combinations of more elaborate heuristics, and the use of sample variances.

Contrary to random walk radiosity, combining gathering and shooting in stochastic Jacobi iterations is extremely simple with the balance heuristic. It works well in all cases and may yield up to a doubling of the efficiency at no extra cost. There is no reason not to use it.

12 Low-discrepancy Sampling in Monte Carlo Radiosity

In this section, we discuss the application of low-discrepancy number sequences instead of pseudo-random number sequences in the context of the radiosity problem. Low-discrepancy sequences are specially crafted number sequences that are more uniform that random (and pseudo-random) number sequences. They allow integrals to be computed with $\mathcal{O}(\log^d N/N)$ error instead of the typical $\mathcal{O}(\sqrt{1/N})$ of Monte Carlo integration (N denotes the number of samples, and d denotes the dimension of the integral). Low-discrepancy numbers are however not statistically independent, so that some care must be taken when using such sequences in algorithms that assume statistically independent samples.

The use of low-discrepancy sequences for radiosity has been proposed by A. Keller in [88, 87]. In [88], low-discrepancy sampling was used instead of pseudo-random sampling in the context of a continuous random walk radiosity algorithm that is very similar to algorithm 17 on page 110. In [87], low-discrepancy sampling was proposed for form factor computation using algorithm 5.3.2 on page 75. Neumann et al. proposed low-discrepancy sampling in the context of a stochastic Jacobi iterative method [112] (§6.4.4). The effectiveness of low-discrepancy sampling is similar for form factor computation and in stochastic Jacobi iterations. It turns out to be significantly more effective in discrete radiosity algorithms than in continuous radiosity algorithms like algorithm 17.

12.1 Quasi Monte Carlo integration

Quasi-Monte Carlo integration — integration by low-discrepancy sampling — is based on number theory rather than on probability. Consider an arbitrary set of N numbers $x_k, k = 1, \ldots, N$ in the half-open interval $[0, 1)$ and consider a function $f(x)$ defined on the closed interval $[0, 1]$. Koksma [92] showed that

$$\left| \frac{1}{N} \sum_{k=1}^{N} f(x_k) - \int_0^1 f(x)dx \right| < V_f \cdot D(\{x_k | k = 1, \ldots, N\}) \tag{12.1}$$

where

- V_f denotes the *variation* of the function f on the interval $[0, 1]$. The variation is the smallest upper bound (if finite) for

  $$\sum_{i=0}^{n} |\, f(t_{i+1}) - f(t_i) \,|$$

  for any possible subdivision of the interval $[0, 1]$ in sub-intervals determined by the numbers $0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = 1$. The variation is an indicator for the smoothness of the function f, and does not depend on the numbers x_k;

- $D(\{x_k|k = 1, \ldots, N\})$ denotes the *discrepancy* (modulo 1) of the set of N numbers $x_k, k = 1, \ldots, N$. The discrepancy D is the lowest upper bound for arbitrary intervals $(\alpha, \beta), 0 \leq \alpha < \beta < 1$ of $R_{\alpha,\beta}(\{x_k|k = 1, \ldots, N\})/N$, where

$$R_{\alpha,\beta}(\{x_k|k = 1, \ldots, N\}) = \#\{x_k \in (\alpha, \beta)|k = 1, \ldots, N\} - (\beta - \alpha)N$$

  is the difference between the number of points x_k that lay in the interval (α, β) and the size $(\beta - \alpha)$ of the interval times N. The discrepancy is an indicator for how uniformly the numbers x_k fill the interval $[0, 1)$. It does not depend on the function f.

Hlawka [74] later generalised (12.1) to functions and sequences defined on the unit hyper-cube I^d in arbitrary dimension *d*. Hlawka also proved a similar inequality for integrals over a wide class of sub-domains of I^d with non-zero volume, including convex domains of arbitrary shape.

The Koksma-Hlawka inequality (12.1) shows that the error in approximating an integral of f, by the average of N samples of f, will vanish in the limit for $N \to \infty$ if the discrepancy $D(\{x_k|k = 1, \ldots, N\}) \to 0$. Number sequences that have this property are called *uniform* number sequences. The rate at which the error vanishes is bounded by the rate at which the discrepancy vanishes. The potential of deterministic uniform number sequences for integration lays in the fact that sequences are known for which the discrepancy vanishes like $\mathcal{O}((\log N)^d/N)$. Such number sequences can lead to vastly superior convergence rates compared with the $\mathcal{O}(\sqrt{1/N})$ rate of classical Monte Carlo. Such sequences have been developed by Van der Corput, Halton, Sobol, Faure, Niederreiter and others [115, 158, 170, 18]. They are called *low-discrepancy* sequences. More information about the implementation and test of low-discrepancy sequences can be found in [16, 125].

Since a sum can always be expressed as an integral of a piecewise constant function, low-discrepancy numbers can equally well be applied for estimating sums. There are however a number of important facts to be taken into account concerning quasi-Monte Carlo integration versus classic Monte Carlo integration:

- Low-discrepancy numbers are *not random*: for instance, the difference between two subsequent numbers of the Van der Corput sequence (the 1-dimensional base-2 Halton sequence) [62], equals 0.5 half of the time. Efficient low-discrepancy number generators often even exploit correlation between successive numbers in order to gain efficiency;

- The Koksma-Hlawka inequality is valid for integrands with *finite variation*. It does not say anything about integrands with infinite variation, for instance integrands with non-axis aligned value discontinuities which occur frequently in the context of global illumination [125, 168]. Low-discrepancy numbers will often still result in lower error for same number of samples compared to pseudo-random numbers, but the convergence rate will be slower than $\mathcal{O}((\log N)^d/N)$;

- The Koksma-Hlawka inequality provides a *deterministic upper bound* for the integration error. Most often, it leads to gross over-estimates of the error. The problem is due to the variation V_f, which is most often not a good indicator for performance. The discrepancy D however is believed to be a good indicator for the convergence rate [18]. This is unlike the error analysis of classic

Monte Carlo integration, where probabilistic rather than deterministic, but at least *realistic* error estimates are relatively easy to obtain (see §4.2.6);

- The discrepancy is *asymptotically*, for large N, well approximated by $c(\log N)^d/N$ for some constant c, that depends on the sequence under consideration (recent low-discrepancy sequences have lower constant c). Nothing is stated about the discrepancy of relatively small sets of low-discrepancy numbers. Often in rendering, we deal with $N \approx 10$ or 100 samples for computing form factors or for the radiosity on each patch in a scene. For instance, it is possible that a first sequence with better asymptotical discrepancy has worse discrepancy for small N than some other sequence;
- The effect of *dimension*: for large dimension d, the error is dominated by the $(\log N)^d$ factor in the numerator, rather than the N in the denominator, unless $N > 2^d$ [18]. Quasi-Monte Carlo integration looses its effectiveness in higher dimensions. The convergence rate with low-discrepancy numbers is however never worse than $\mathcal{O}(\sqrt{1/N})$, as with pseudo-random numbers.

Through randomization, it is possible to find a solution to most problems of low-discrepancy sampling: realistic error estimates become available, a higher accuracy can be obtained in some cases, and the effectiveness can be preserved in high dimensions. These are still topics of ongoing research however [117].

12.2 Low-discrepancy sampling in radiosity

In practice, the application of low-discrepancy sequences often involves little more than replacing the pseudo-random number generator in an implementation. However, care must be taken in order to avoid potential problems due to correlations in algorithms that assume statistically independent samples.

12.2.1 Quasi-Monte Carlo form factor computation

The Monte Carlo estimators for form factors (§5) require either 2- or 4-dimensional random vectors. These are typically obtained by generating 2 or 4 subsequent pseudo-random numbers. Quasi-Monte Carlo form factor computation is obtained by employing 2- or 4-dimensional low-discrepancy vectors instead. In particular, we recommend:

- A 2- or 4-dimensional Halton sequence: the components of the i-th low-discrepancy vector are the radical inverse of i w.r.t. base 2, 3, 5 or 7 (the four first prime numbers). The base-b radical inverse $\iota_b(i)$ is the number obtained by expressing i is base b, and mirroring the digits of the representation [62]:

  $$i = a_0 + a_1 b + a_2 b^2 + \cdots \quad \Rightarrow \quad \iota_b(i) = a_0 b^{-1} + a_1 b^{-2} + a_2 b^{-3} + \cdots$$

  An implementation in the C programming language can also be found in [87];
- A 2- or 4-dimensional base-2 31-bits Niederreiter sequence. An efficient generator can be obtained for free [15].

In appendix D, a new mapping is proposed to transform a 2D low-discrepancy sequence on a square to a triangle.

12.2.2 Stochastic relaxation radiosity

In all stochastic relaxation radiosity algorithms, low discrepancy sampling can be used instead of pseudo-random sampling for selecting a patch j conditional on patch i with probability equal to the form factor F_{ij}. This is achieved by tracing a local or global uniformly distributed line through j:

- Local lines: 2 sample numbers are required for sampling the origin of a ray on i and 2 more sample numbers are required for sampling a cosine-distributed ray direction. A 4-dimensional low-discrepancy vector replaces 4 pseudo-random numbers. The same 4-dimensional sequences, mentioned above, can be used.

  Some care must be taken however in order to break correlations between ray directions of rays originating at neighbouring patches. The problem is illustrated in figure 12.1. Neumann et al. [112] proposed to start sampling the 4-dimensional sequence at different offsets for different patches. In the implementation, the offset for a patch with index i, was chosen $11 \cdot i$. The origin and direction of the k-th ray from patch i was constructed by using the 4-dimensional sample number with index $11 \cdot i + k$ in the sequence;

- Global lines with the two-points-on-a-sphere method: sampling both points on the sphere requires two times two sample numbers, so that here again, a 4-dimensional low-discrepancy sequence can be used [21]. Techniques to optimally reduce correlations are a topic of current research;

- Global lines with the plane-intercept method: a 2-dimensional construction sample vector is needed for choosing a plane normal. Within each plane, intercepts of parallel global lines can be selected by a separate 2-dimensional low-discrepancy sequence [165].

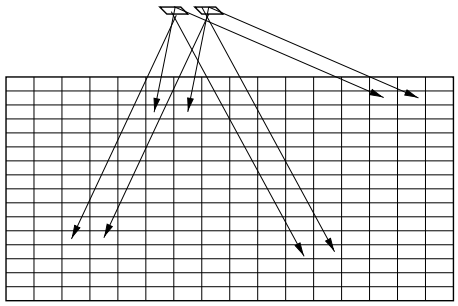


Figure 12.1: If sample vectors with same index are used on different patches, for choosing the origin and direction of the rays, distracting patterns may result in a computed image. This figure illustrates how such patterns may result due to clustered contributions from nearby sources.

12.2.3 Random walk radiosity

The of a random walk requires low-discrepancy vectors of higher dimension:

- 4 sample numbers are required in order to sample an origin of a particle on a light source, as well as an initial direction;

- In continuous shooting random walk, 2 extra random numbers are required in order to determine whether the particle is absorbed after some collision or in what direction it will continue its path. In discrete random walk, 4 rather than 2 sample numbers are needed per collision since also a uniform random new position needs to be chosen.

For sampling random walks of length l, $(4 + 2l)$-dimensional sample vectors are required in the continuous shooting random walk algorithm, and $(4 + 4l)$-dimensional sample vectors in the discrete shooting random walk algorithm: the 4 first components are used in order to sample the origin and initial direction of a particle on a light source. Successive groups of 2 or 4 components are used at each collision. A detailed description of an implementation can be found in [88]. Due to the high dimension, the effectiveness of low-discrepancy sampling is not so high when applied like this.

We found that the effectiveness of low-discrepancy sampling can be significantly increased by employing a similar sampling scheme as with local lines in stochastic relaxation radiosity (explained above): for each scattering decision a 2- or 4-dimensional sample vector is needed. For the k-th scattering event on a patch i, the sample vector with index $11 \cdot i + k$ from a single global 2 or 4-dimensional low-discrepancy sequence can be used. In the implementation, this implies that a count (k) of scattering events needs to be kept on every patch. The storage needed for one extra integer number per patch is well worth the increased accuracy for given amount of work.

12.3 Empirical results and discussion

12.3.1 Experiment description

In this section, three experiments are described. They concern:

- the error for given number of samples in regular and incremental Jacobi iterations, and in discrete collision shooting random walk radiosity (§12.3.2);

- the effectiveness of quasi-Monte Carlo sampling in continuous versus discrete collision shooting random walks (§12.3.3);

- the effectiveness of various low-discrepancy number generators in discrete collision shooting random walk radiosity (§12.3.4).

In each experiment, the measured RMS error in the result is studied as a function of the number of samples. In all cases, local line sampling is used.

The test scenes used are shown in figure 12.2. The labyrinth scene is a case in which the radiosity solution is analytically known to be constant and equal to one everywhere. This will always be the case regardless of geometry as long as for each individual patch $E_i + \rho_i = 1$. The degree of freedom in varying ρ_i per patch was used to create three labyrinth instances with different average reflectivity: 0.1, 0.5 and 0.9. Since the solution is known, the RMS error can be determined exactly. These tests exhibit better the theoretically expected behaviour of the algorithms.

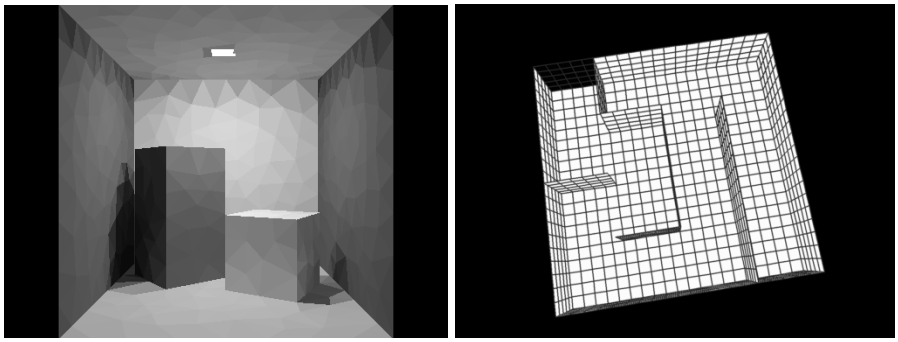


Figure 12.2: Test scenes used in the experiments: box (left) and labyrinth (right). The right image shows the reflectivities ρ_i of the patches in the high average reflectivity case $\rho_{av} \approx 0.9$. The self-emitted radiosities E_i were chosen such that $E_i + \rho_i = 1$. The radiosity solution B_i is constant and equal to one everywhere.

Scenes with constant illumination are however extremely rare in practice. The second test scene, a simple box containing two other boxes, will provide better indications of the behaviour of the algorithms in practice. Since the solution is not analytically known, the error has to be estimated by comparing with a reference solution. References images were computed using about 100 times more samples than indicated in the graphs. The RMS image difference with the reference image was plotted as an indication for the error. Although this includes also the effect of tone mapping and only the parts of the scene that are visible in the images are taken into account, it has no qualitative implications for our conclusions.

12.3.2 Experiment 1: local line sampling in discrete Monte Carlo radiosity algorithms

In the graphs in figure 12.3, the measured error of three local discrete Monte Carlo Radiosity (MCR) algorithms is plotted as a function of the number of samples with pseudo-random sampling and Halton low-discrepancy sampling. The three algorithms that are being compared are:

- The *stochastic ray radiosity* algorithm [113]: this algorithm uses regular Jacobi iterations (§6.4.3) merged as explained in §6.6.3. The higher error for low number of samples indicates the "warming-up" problem that was mentioned in §6.4.4;
- The *Well-Distributed Ray Set* radiosity algorithm [112]: in this stochastic Jacobi algorithm, the warming-up problem is solved by shooting only additionally required rays in regular Jacobi iterations (§6.4.4). Incremental Jacobi iterations (§6.4.2) with progressive ray refinement (§6.6.4) would perform very similarly;
- The *discrete collision shooting random walk* radiosity algorithm [132] (§7.4.2).

The graphs indicate that in the long-term, after warming-up, the obtained error for same number of samples is nearly identical for these three algorithms. For pseudo-

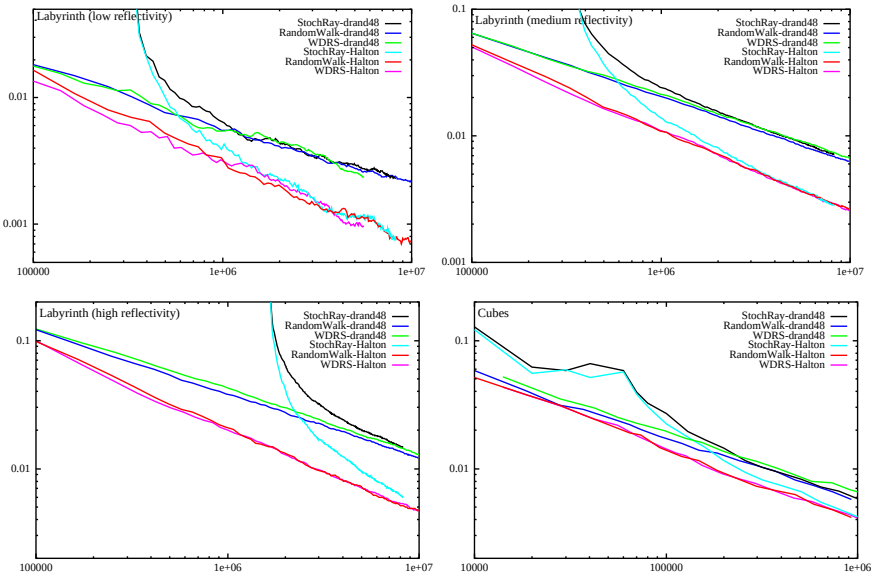


Figure 12.3: Empirical comparison of the error with three different discrete Monte Carlo radiosity algorithms in which local-line sampling is used. The long-term convergence rate of the three algorithms is identical, confirming the theoretical error analysis for pseudo-random sampling earlier in this dissertation. The experiment suggests that this result will be valid with low-discrepancy sampling as well. The graphs also indicate that with Halton low-discrepancy sample numbers, the long-term convergence rate can be significantly higher. Legend: StochRay = Stochastic Ray Radiosity [113], WDRS = Well-Distributed Ray Set Radiosity [112], RandomWalk = Discrete collision shooting random walk radiosity [132]. Measured RMS error is plotted against the number of samples, see §12.3.1.

random sampling, this confirms the theoretical comparisons of §6.4.4 and §7.4.4. The experiment suggests that this observation will also hold for low-discrepancy sampling. The graphs show that low-discrepancy sampling indeed leads to a significantly higher convergence rate.

The three algorithms have an intuitive interpretation in the sense of particles carrying light power that are shot throughout the scene. Intuitively, the difference between the algorithms is mainly in the order in which particles are shot and to a lesser extent also in survival decisions and in the way the results from individual particles are averaged (see §7.4.4).

12.3.3 Experiment 2: discrete versus continuous collision shooting random walk

In the graphs of figure 12.4, the convergence rate of the discrete versus continuous collision shooting random walk algorithms [120, 132] (§7.1) is compared when pseudo- or Halton quasi-random sample numbers are used. In an implementation, the difference between discrete and continuous collision shooting random walk radiosity is very small: a particle hitting a patch is warped to another, uniformly chosen, location on

the patch in the discrete random walk algorithm, while it is scattered from the point of incidence in the continuous random walk algorithm.

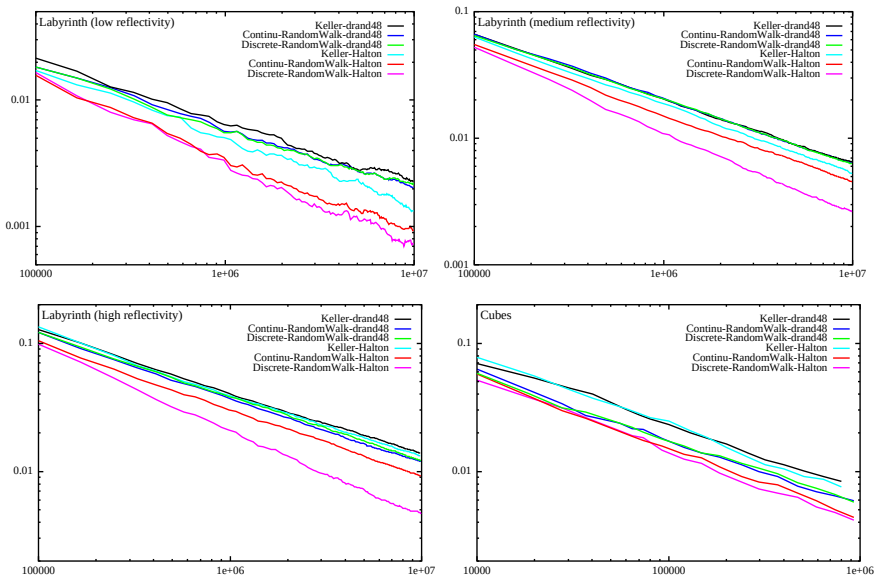


Figure 12.4: Empirical comparison of discrete versus continuous collision shooting random walk radiosity with pseudo- and Halton quasi-random sampling. Also Keller's quasi-Monte Carlo radiosity algorithm [88] is included in the comparison. Low discrepancy sampling appears to be more effective in discrete than in continuous random walk radiosity.

The convergence rate with random sampling is equal for the discrete and continuous random walk algorithms: $\mathcal{O}(1/\sqrt{N})$ with N the number of samples. With QMC sampling however, the convergence rate of the discrete random walk method can be significantly better than in the continuous random walk method: up to $N^{-0.7}$ versus $N^{-0.6}$. The experiment in §12.3.2 suggests that such a higher convergence rate will also be observed with stochastic relaxation radiosity. The difference appears to be larger for higher average reflectivity. This should not be surprising since direct illumination is computed identically in continuous and discrete particle tracing.

The graphs only show the computational error and not the discretisation error, which is the error caused by approximating the true radiosity solution $B(x)$ by a piecewise constant function (§3.1). In §7.1.5, it was shown that the difference in discretisation error will in general be very low however.

This experiment thus suggests that *with random sampling, the sum of computational and discretisation error for the same amount of work, will be very slightly higher with discrete Monte Carlo radiosity algorithms than with continuous algorithms. With QMC sampling however, lower computational error will compensate the slightly higher discretisation error with discrete algorithms, so that discrete algorithms may be preferred.*

The graphs in figure 12.4 also show the convergence rate obtained with Keller's algorithm [88] with both random and Halton sampling. The convergence rates obtained with this algorithm are inferior mainly because the particle paths in [88] are fully de-

termined by their sequence number without taking the local reflectance at each visited patch into account.

The result that low-discrepancy sampling is more effective in discrete than in continuous random walk radiosity, contradicts with theoretical predictions:

- Spanier [157] has proved a Koksma-Hlawka like theorem for random walk estimators by considering the function f which is being integrated implicitly in random walk methods. This function is a combination of the random walk score function and the mapping from canonical low-discrepancy vectors to random walks. The mapping from low-discrepancy numbers to a random walk is quite different for continuous and discrete random walks, so that the variation of this function can be expected to be different as well. A different variation does however not explain a different error reduction *rate* as the number of samples is increased. It only determines the absolute value of an upper bound for the error;

- The function being integrated contains discontinuities related to projected object boundaries in the scene [168]. The convergence rate can thus be expected to be lower than $\mathcal{O}(\log^d N/N)$. The convergence rate should be lower for higher dimensional sampling. This experiment however suggests a higher convergence rate for discrete random walks. Discrete random walks require a higher dimensional sampling, which contradicts this analysis;

- Also according to the discrepancy alone, one would expect a higher convergence rate with continuous random walk radiosity, because sample vectors of lower dimension are needed. Our experiments however indicate a faster reduction of the error as the number of samples in increased with discrete random walk radiosity.

That low-discrepancy appears to be more effective in discrete random walks, is probably due to the fact that the number of samples is quite small. Asymptotically, the convergence rate of continuous random walk may be higher, but probably only for a much higher number of random walks than required in image synthesis.

12.3.4 Experiment 3: the sample number generator

In the previous experiments, the Halton number sequence was used in order to characterise QMC sampling. The graphs in figure 12.5 compare the convergence rate of discrete collision shooting random walk radiosity with local line sampling, for other quasi-random number sequences. Besides pseudo-random and Halton sampling, also the results with the 4D scrambled Halton, Faure, generalised Faure and Sobol and Niederreiter sample number sequences [170] are shown. While a speed difference of up two a factor 2 may be observed with different QMC sequences, the graphs suggest that *no QMC sequence behaves systematically superior to the other sequences. The simple Halton sequence behaves well in all cases.*

12.4 Conclusion

- Low discrepancy sampling can increase the efficiency of Monte Carlo radiosity algorithms significantly at little extra cost. It appears to be more effective for

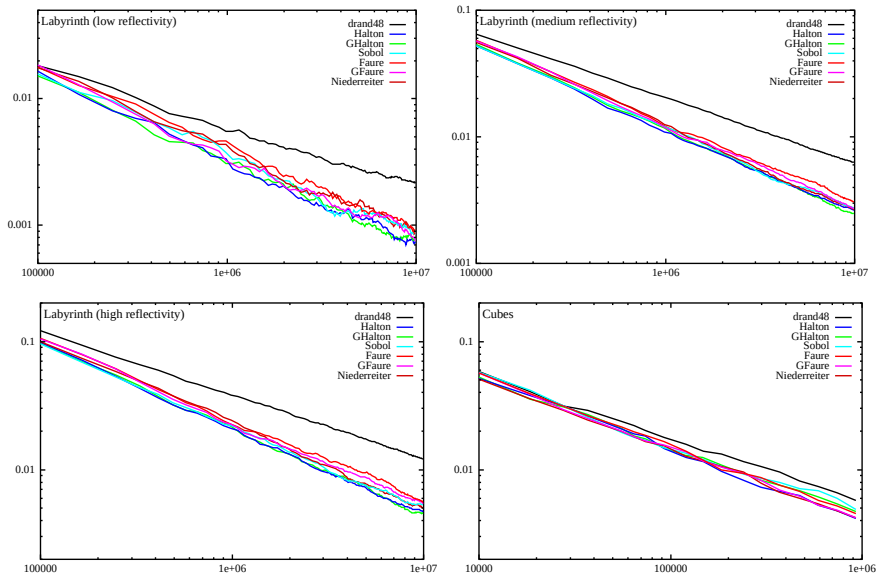


Figure 12.5: The influence of the QMC sample number sequence in discrete collision shooting random walk radiosity. No QMC sequence is systematically superior to the other sequences. The simple Halton sequence behaves well in all cases.

discrete than for continuous random walk radiosity. The experiments confirm that the performance of stochastic relaxation radiosity is the same in practice as of discrete collision shooting random walk radiosity for random sampling. They suggest that this is also true when quasi-random sampling is used;

- The reason why low-discrepancy sampling appears to be more effective in discrete rather than continuous random walk radiosity is not a resolved issue. Probably, the number of samples required in image synthesis is too low in order to exhibit the asymptotical converge rates predicted by the theory;
- No quasi-random sequence performs systematically better than the others. The simple Halton sequence performs well in all cases;
- Care must be taken in order to avoid correlations. With local lines, a good technique to break correlations is to start sampling a global 2- or 4-dimensional low-discrepancy sample sequence at different offsets for each patch;
- The same sampling technique also yields good results in random walk radiosity. The price to be paid is low: a scattering event counter needs to be stored with each patch.

Low-discrepancy sampling in random walk methods is however a topic of ongoing research. It is not yet fully understood, and several techniques are being designed in order to make it more effective in this context [18, 117].

13
Higher-Order Approximations

In the previous chapters, several applications of the Monte Carlo method to the solution of the radiosity problem with constant approximations have been presented. In this chapter, these methods will be extended to higher order radiosity approximations. In previous work, only the continuous shooting random walk method (§7.1.2) has been extended to higher order approximations [47, 14]. The extension of discrete Monte Carlo methods to higher order approximations is a new result developed in the context of this dissertation.

First, the equations to be solved and relevant previous work will be discussed (§13.1). Next, the extension to higher approximations will be proposed for Monte Carlo form factor computation (13.2), stochastic relaxation radiosity (§13.3) and discrete random walk radiosity (§13.4).

13.1 Problem formulation and previous work

The equations to be solved for higher order approximations have been derived in chapter 2. The problem consists in determining the coefficients $B_{i,\alpha}$ of an approximate radiosity solution $\tilde{B}(x) = \sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x) \approx B(x)$ where i denotes a patch in the scene, and $\psi_{i,\alpha}(x)$ are basis functions defined on patch i (§2.2.1). $B(x)$ is the true solution of the continuous radiosity equation (2.3) or its alternatives (§2.1):

$$B(x) = E(x) + \rho(x) \int_S G(x,y)B(y)dA_y. \tag{13.1}$$

Continuous higher order Monte Carlo radiosity Such coefficients $B_{i,\alpha}$ can be found directly, as scalar products of $B(x)$ with the dual basis functions $\tilde{\psi}_{i,\alpha}$ (§2.2.2):

$$B_{i,\alpha} = \int_S B(x)\tilde{\psi}_{i,\alpha}(x)\, dA_x. \tag{13.2}$$

Feda [47] and Bouatouch et al. [14] proposed to use an analog continuous shooting random walk in order to estimate these scalar products. The analog continuous shooting random walk can be viewed as a device to sample points $x \in S$ with density $\chi(x) = B(x)/\Phi_T$ (§7.1.2). The coefficients $B_{i,\alpha}$ are obtained by simultaneously accumulating averages

$$B_{i,\alpha} \approx \frac{\Phi_T}{N} \sum_{s=1}^{N} \sum_{t=0}^{\tau_s} \tilde{\psi}_{i,\alpha}(x_{s,t})$$

N denotes the number of random walks that is traced. $x_{s,t}$ is the t-th visited point of the s-th traced random walk. τ_s is the length of the s-th traced random walk.

For orthogonal basis functions obtained by uniform mapping of orthonormal basis functions on the unit square or standard triangle (§2.2.6), the dual basis functions are related with the primary basis functions like $\tilde{\psi}_{i,\alpha}(x) = \psi_{i,\alpha}(x)/A_i$. Constant approximations are obtained with one basis functions ψ_i per patch, with $\psi_i(x) = 1$ if $x \in S_i$ and 0 otherwise.

Feda observed that the required number N of random walks in order to compute a solution of fixed accuracy, using a K-th order product Legendre basis on the unit square, is $\mathcal{O}(K^2)$. An intuitive argument was given in order to explain this observation: *"Doubling the order K of the basis of a surface means doubling the maximum 'frequency' "* [47].

Discrete higher order Monte Carlo radiosity Alternatively, Galerkin discretisation can be used (§2.2.3), leading to a system of linear equations (2.15)

$$B_{i,\alpha} = E_{i,\alpha} + \sum_{j,\beta} K_{i,\alpha;j,\beta} B_{j,\beta} \tag{13.3}$$

with generalised patch-to-patch form factors (2.14):

$$K_{i,\alpha;j,\beta} = \int_{S_i} \tilde{\psi}_{i,\alpha}(x)\rho(x) \int_{S_j} G(x,y)\psi_{j,\beta}(y)\, dA_y dA_x. \tag{13.4}$$

In this chapter, we will generalise Monte Carlo form factor computation (chapter 5), stochastic relaxation (chapter 6) and random walk methods (chapter 7) to the solution of these equations.

13.2 Generalised form factor computation by Monte Carlo

13.2.1 Outline

In case $\rho(x) = \rho_i$ is constant on each patch i, $K_{i,\alpha;j,\beta} = \rho_i G_{i,\alpha;j,\beta}$ with

$$\begin{aligned} G_{i,\alpha;j,\beta} &= \int_{S_i} \tilde{\psi}_{i,\alpha}(x) \int_{S_j} G(x,y)\psi_{j,\beta}(y)\, dA_y dA_x. \\ &= \int_{S_i} \tilde{\psi}_{i,\alpha}(x) \int_{\Omega(x)} \frac{cos\theta_x}{\pi} \chi_j(h(x,\Theta_x))\psi_{j,\beta}(h(x,\Theta_x)) d\omega_{\Theta_x} dA_x. \end{aligned}$$

$\Omega(x)$ denotes the hemisphere above $x \in S_i$, $h(x,\Theta_x)$ denotes the nearest surface point visible from x in direction Θ_x, and $\chi_j(y)$ is 1 if $y \in S_j$ or 0 if $y \notin S_j$.

The algorithms for Monte Carlo form factor computation (chapter 5) can be generalised to higher-order radiosity approximations by merely multiplying the contribution of each ray between a point $x \in S_i$ and $y \in S_j$ with $A_i \tilde{\psi}_{i,\alpha}(x) \cdot \psi_{j,\beta}(y)$. Consider for instance form factor computation using local uniformly distributed lines:

- pdf:

$$p(x,\Theta) = \frac{\chi_i(x)}{A_i} \frac{\cos\theta}{\pi};$$

- sample contributions:

$$\hat{G}_{i,\alpha;j,\beta}(x,\Theta) = A_i \tilde{\psi}_{i,\alpha}(x)\chi_j(h(x,\Theta))\psi_{j,\beta}(h(x,\Theta))$$

All form factors $G_{i,\alpha;j,\beta}$ for a pair of patches i and j can be computed simultaneously using the same rays.

13.2.2 Variance

The variance $V[\hat{G}_{i,\alpha;j,\beta}]$ is bounded from above by

$$\begin{aligned} E[\hat{G}^2_{i,\alpha;j,\beta}] &= \frac{1}{A_i}\int_{S_i} A_i^2 \tilde{\psi}^2_{i,\alpha}(x) \int_{\Omega(x)} \frac{\cos\theta_x}{\pi} \chi^2(h(x,\Theta_x))\psi^2_{j,\beta}(h(x,\Theta_x))\, d\omega_{\Theta_x} dA_x. \\ &\leq \frac{\overline{c}_{ij}}{\pi} \cdot \frac{1}{A_i}\int_{S_i} A_i^2 \tilde{\psi}^2_{i,\alpha}(x) \int_{\Omega_j(x)} \psi^2_{j,\beta}(h(x,\Theta_x)) d\omega_{\Theta_x} \end{aligned} \tag{13.5}$$

with

$$\overline{c}_{ij} = \max_{x\in S_i, y\in S_j} \cos\theta_x.$$

$\overline{c}_{ij}$ is an upper bound for the maximum cosine of the angle w.r.t. the normal on i, under which points of j are seen from points on i. Consider the important case of orthogonal basis functions $\psi_{i,\alpha}(x)$ obtained by uniform mapping of orthonormal canonical basis functions $\psi_\alpha(u,v)$ on the standard triangle or the unit square ($dA_i(u,v) = A_i\, du\, dv$).

The inner integral (over $\Omega_j(x)$) of (13.5) can be bounded in two ways:

$$\begin{aligned} \int_{\Omega_j(x)} \psi^2_{j,\beta}(h(x,\Theta_x)) d\omega_{\Theta_x} &\leq \overline{\psi}^2_\beta \cdot \overline{\Omega}_{ij} \\ \text{or} \quad &\leq \int_{S_j} \frac{\cos\theta_y}{r^2_{xy}} \psi^2_{j,\beta}(y)\, dA_y \leq \frac{\overline{c}_{ji}}{\underline{r}^2_{ij}} \int_{S_j} \psi^2_{j,\beta}(y)\, dA_y = \frac{\overline{c}_{ji}}{\underline{r}^2_{ij}} A_j \end{aligned}$$

where

$$\begin{aligned} \overline{\psi}_\beta &= \max_{y\in S_j} |\,\psi_{j,\beta}(y)\,| = \max_{u,v} |\,\psi_\beta(u,v)\,| \\ \overline{\Omega}_{ij} &= \max_{x\in S_i} \Omega_j(x) \\ \overline{c}_{ji} &= \max_{y\in S_j, x\in S_i} \cos\theta_y \\ \underline{r}_{ij} &= \min_{x\in S_i, y\in S_j} r_{xy} \end{aligned}$$

The computation of $\overline{\Omega}_{ij}, \overline{c}_{ij}, \overline{c}_{ji}$ and $\underline{r}_{ij}$ has been discussed in §5.5. $\overline{\psi}_\beta$ only depends on the canonical basis functions on the unit square or standard triangle that are used and can be pre-computed once in advance.

After bounding the inner integral as shown above, the remaining outer integral (over S_i) of (13.5) becomes

$$\frac{1}{A_i}\int_{S_i} A_i^2 \tilde{\psi}^2_{i,\alpha}(x)\, dA_x = \frac{1}{A_i}\int_{S_i} \psi^2_{i,\alpha}(x)\, dA_x = \frac{1}{A_i} A_i \int \psi^2_\alpha(u,v)\, du\, dv = 1.$$

The following bounds for the variance result:

$$V[\hat{G}_{i,\alpha;j,\beta}] \leq \frac{\overline{c}_{ij}}{\pi}\overline{\Omega}_{ij}\overline{\psi}_{\beta}^{2} \tag{13.6}$$

$$\text{or} \quad \leq \frac{\overline{c}_{ij}\overline{c}_{ji}}{\pi \underline{r}_{ij}^{2}} A_j \tag{13.7}$$

The latter bound will not be satisfying for nearby or abutting patches i and j, when $r_{xy} \to 0$. The first bound on the other hand, may be too pessimistic for more distant patches with bad aspect ratio (for instance: long and thin). In all cases, the minimum of these two bounds can be used in order to determine the number of samples required in order to compute the form factors $G_{i,\alpha;j,\beta}$ to given accuracy.

13.2.3 Required work as a function of the approximation order

The upper bound (13.7) for the variance is independent of the used basis functions, and thus also the approximation order. When using deterministic integration rules (see for instance [32, 31]), a higher approximation order always requires a higher order integration rule — with more nodes.

That the number of samples seems not to depend on the approximation order with Monte Carlo, is a consequence of the use of orthonormal polynomials. Consider for instance a set of orthonormal polynomials $\psi(x)$ on the interval $[0,1]$: with uniform sampling, the variance is

$$V[\psi] = \int_{[0,1]} \psi^2(x)dx - \left(\int_{[0,1]} \psi(x)dx\right)^2 \leq \int_{[0,1]} \psi^2(x)dx = \|\psi\|^2 = 1$$

independent of the degree of the polynomial. Even stronger: $V[\psi] = 0$ if ψ is a constant function and $V[\psi] = 1$ for all non-constant functions ψ in the orthonormal set.

The previous statement does not imply that computing a higher order radiosity approximation with Monte Carlo would not require more work than the computation of a constant radiosity approximation. We analyse the required amount of samples for higher order approximations here.

Consider the solution of (13.3) by (deterministic) Jacobi iterations with generalised form factors $G_{i,\alpha;j,\beta}$ computed by Monte Carlo. Consider for instance the interaction between a patch j and a path i:

$$L_{i,\alpha} = \rho_i \sum_{\beta} G_{i,\alpha;j,\beta} L_{j,\beta}$$

Our goal is to choose the number of samples N for (simultaneous) computation of the generalised form factor $G_{i,\alpha;j,\beta}$ in such a way that the resulting error on

$$\text{Error}\left(\sum_{\alpha} L_{i,\alpha}\psi_{i,\alpha}(x)\right) = \text{Error}\left(\sum_{\alpha} L_{i,\alpha}\psi_{\alpha}(u,v)\right) \quad \approx \quad \varepsilon$$

for some threshold ε and confidence level. Application of the central limit theorem of probability, with 99.7% confidence level yields:

$$\begin{aligned} N &\approx \frac{9}{\varepsilon^2} V \left[\rho_i \sum_\alpha \psi_\alpha(u,v) \sum_\beta G_{i,\alpha;j,\beta} L_{j,\beta} \right] \\ &\approx \frac{9}{\varepsilon}^2 \rho_i^2 \left(\sum_\alpha \psi_\alpha^2(u,v) \sum_\beta L_{j,\beta}^2 V[\hat{G}_{i,\alpha;j,\beta}] \right. \\ &\quad \left. +2 \sum_{\alpha \neq \gamma; \beta \neq \delta} \psi_\alpha(u,v)\psi_\gamma(u,v) \mid L_{j,\beta} \mid \mid L_{j,\delta} \mid \mathrm{Cov}[\hat{G}_{i,\alpha;j,\beta}, \hat{G}_{i,\gamma;j,\delta}] \right) \end{aligned}$$

This expression can be simplified as follows:

- Following a similar reasoning as for bounding the variances $V[\hat{G}]$ above, it can be shown that the covariances are zero for orthogonal basis functions obtained by uniform mapping of orthonormal canonical basis functions;
- The variances $V[\hat{G}_{i,\alpha;j,\beta}] \approx V[\hat{G}]$ are approximately independent of the basis functions.

With these assumptions,

$$N \approx \frac{9}{\varepsilon^2} \rho_i^2 \left(\sum_\alpha \psi_\alpha^2(u,v) \right) \left(\sum_\beta L_{j,\beta}^2 \right) V[\hat{G}]. \tag{13.8}$$

In general, the higher order coefficients $L_{j,\beta>0} < L_{j,\beta=0}$ are small corrections to the constant approximation. On the average, $\psi_\alpha^2(u,v) \approx 1$, so that the following conclusion is reached: the *number of samples shall be taken approximately proportional to the number of basis functions on* i.

For a product basis of order K, the number of basis functions is K^2. The same dependence on the approximation order will be observed as in continuous shooting random walk algorithms for higher order approximations [47, 14]. The result derived here is however also valid for non-product basis functions, for instance those discussed in §2.2.6.

13.3 Higher-order stochastic relaxation radiosity

13.3.1 Outline

Unfortunately, the generalised form factors $G_{i,\alpha;j,\beta}$ do not form a probability distribution unlike the form factors $F_{ij} = G_{i,0;j,0}$. The generalised form factors can be negative, and their sum does not equal 1. In order to obtain stochastic Jacobi relaxation for higher-order approximations, one can proceed as follows.

The problem is to compute a new radiosity distribution $\tilde{B}'(x) = \sum_{i,\alpha} B'_{i,\alpha}\psi_{i,\alpha}(x)$ for a given radiosity distribution $\tilde{B}(x) = \sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x)$. Equation (13.3) can be rewritten as

$$\begin{aligned} B'_{i,\alpha} - E_{i,\alpha} &= \sum_{j,\beta} K_{i,\alpha;j,\beta} B_{j,\beta} \\ &= \sum_{j,\beta} \int_{S_i} \tilde{\psi}_{i,\alpha}(x)\rho(x) \int_{S_j} G(x,y)\psi_{j,\beta}(y)\, dA_y\, dA_x \cdot B_{j,\beta} \\ &= \int_S \int_S \tilde{\psi}_{i,\alpha}(x)\rho(x)G(x,y) \left(\sum_{j,\beta} B_{j,\beta}\psi_{j,\beta}(y) \right) dA_y\, dA_x \end{aligned} \tag{13.9}$$

This suffices in order to define a gathering stochastic Jacobi relaxation algorithm. However, consider:

$$B'_{i,\alpha} - E_{i,\alpha} = \int_S \left(\sum_{j,\beta} B_{j,\beta}\psi_{j,\beta}(y) \right) \int_S G(y,x)\rho(x)\tilde{\psi}_{i,\alpha}(x)\, dA_y\, dA_x \tag{13.10}$$

$$= \int_S \tilde{B}(y) \int_S G(y,x)\rho(x)\tilde{\psi}_{i,\alpha}(x)\, dA_x\, dA_y. \tag{13.11}$$

$$= \int_S \tilde{B}(y) \int_{\Omega(y)} \frac{\cos\theta}{\pi} \rho(h(y,\Theta))\tilde{\psi}_{i,\alpha}(h(y,\Theta))\, d\omega_\Theta\, dA_y. \tag{13.12}$$

The advantage of switching the integrals is that now, it is evident that the given radiosity $\tilde{B}(y)$ can be taken into account during sampling. Equation (13.11) leads to estimation procedures of the following kind:

1. Select a point y with normalised probability density $p(y)$. Two special choices for $p(y)$ will be discussed below;

2. Select a point x conditional on y with conditional probability $p(x|y) = G(y,x)$. This is the same as choosing a cosine-distributed direction Θ w.r.t. the normal on the surface at y, and determining the first intersection point $h(y,\Theta) = x$ with the surfaces in the scene. As usual, local or global uniformly distributed lines can be used. Global lines however enforce uniform area sampling for y: $p(y) = 1/A_T$;

3. Record a contribution

$$\hat{B}_{i,\alpha}(y,x) = \frac{\tilde{B}(y)}{p(y)}\rho(x)\tilde{\psi}_{i,\alpha}(x)\chi_i(x). \tag{13.13}$$

   As usual, $\chi_i(x)$ is zero if $x \notin S_i$ and 1 if $x \in S_i$: the contribution is non-zero only for the basis functions on the hit patch, containing x.

There is one such estimator for each patch i and basis function α. Because the difference is only in the scores, these estimators can be sampled simultaneously. The expectation $E[\hat{B}_{i,\alpha}] = B'_{i,\alpha} - E_{i,\alpha}$. We will now discuss two special cases:

Local lines

The ray origins y can be sampled more densely in bright regions:

$$p(y) = \frac{\tilde{B}(y)}{\int_S \tilde{B}(y) dA_y} = \frac{\tilde{B}(y)}{P_T}.$$

The scores (13.13) become

$$\hat{B}_{i,\alpha}(y, x) = P_T \rho(x) \tilde{\psi}_{i,\alpha}(x) \chi_i(x) \tag{13.14}$$

Sampling y proportional to $\tilde{B}(y)$ can be done as follows:

1. Choose a patch j that shall contain y with probability $p_j = P_j / P_T$;

2. Choose $y \in S_j$ by sampling according to a pdf

$$p_j(y) = \frac{\tilde{B}(y)}{P_j}$$

This can be accomplished efficiently using rejection sampling (§4.4.3), with the maximum radiosity value $\overline{B}_j = \max_{x \in S_j} \tilde{B}(x)$ as a reference value. The rejection test does not involve ray tracing and is very cheap. The number of rejections is generally low as well because the radiosity function $\tilde{B}(x)$ does in general not vary excessively on a single patch. There is a slight additional cost due to the need to determine the maximum radiosity value on the patch. In our implementation, we have simply taken the maximum on a 5×5 regular grid, as the maximum value of each individual basis function is reached on these points. The found value if only approximate, but appears to work well in practice.

It is instructive to see what (13.14) yields in the constant radiosity approximation case. In the constant radiosity case, $\tilde{B}(y) = B_j$, $\tilde{\psi}_i(x) = 1/A_i$ and $\rho(x) = \rho_i, x \in S_i$. One obtains as before in §6.4:

$$\hat{B}_i = \frac{1}{A_i} P_T \rho_i \chi_i(x).$$

Global lines

The intersection points have normalised density $p(y) = 1/A_T$ [131]. The corresponding scores (13.13) are

$$\hat{B}_{i,\alpha}(y, x) = A_T \tilde{B}(y) \rho(x) \tilde{\psi}_{i,\alpha}(x) \chi_i(x).$$

13.3.2 Variance and required work as a function of the approximation order

Consider the case with $p(y) = \tilde{B}(y)/P_T$, with scores (13.14). The variance $V[\hat{B}_{i,\alpha}]$ is bounded by

$$\begin{aligned} E[\hat{B}^2_{i,\alpha}] &= \int_S \int_S \left(P_T \rho_i \tilde{\psi}_{i,\alpha}(x) \chi_i(x) \right)^2 \frac{\tilde{B}(y)}{P_T} G(y,x)\, dA_x\, dA_y \\ &= \rho_i P_T \int_{S_i} \tilde{\psi}^2_{i,\alpha}(x) \cdot \rho_i \int_S G(x,y) \tilde{B}(y)\, dA_y\, dA_x \\ &\leq \rho_i P_T \overline{b'}_i \int_S \tilde{\psi}^2_{i,\alpha}(x)\, dA_x \end{aligned}$$

with

$$\overline{b'}_i = \max_{x \in S_i} \rho_i \int_S G(x,y) \tilde{B}(y)\, dA_y.$$

$\overline{b'}_i$ is the maximum received radiosity on i.

For orthogonal basis functions obtained by uniform mapping from orthonormal canonical basis functions, $\tilde{\psi}_{i,\alpha}(x) = \psi_{i,\alpha}(x)/A_i$ and $\int_S \psi^2_{i,\alpha}(x)\, dA_x = A_i$, so that

$$V[\hat{B}_{i,\alpha}] \leq \frac{\rho_i}{A_i} P_T \overline{b'}_i$$

Again, we find an upper bound for the variance that is independent of the approximation order. The computation of the variance with other choices of $p(y)$ can be done in a similar way and leads to the same conclusion.

Determination of the number of samples N, can happen in a similar way as discussed above for generalised form factor computation:

$$\begin{aligned} N &> \frac{9}{\varepsilon^2} V \left[\sum_\alpha \hat{B}_{i,\alpha} \psi_{i,\alpha}(x) \right] \\ &\approx \frac{9}{\varepsilon^2} \left(\sum_\alpha \psi^2_\alpha(u,v) \right) \frac{\rho_i}{A_i} P_T \overline{b'}_i \end{aligned}$$

We find the same dependency on the approximation order as for generalised form factor computation and in continuous shooting random walk methods for higher order approximations.

13.3.3 Variance reduction techniques

View-importance sampling View-importance can be used also here in order to increase or decrease the number of rays shot from each particular patch, based on its estimated importance for a particular view. It suffices to compute the average importance on each patch, so that the constant importance computation steps as discussed in §9.3 can be used without change. Importance is then used to bias $p(y)$. The scores are still given by (13.13).

Constant radiosity variate The constant radiosity variate technique of §10.3 can be generalised as follows:

$$\begin{aligned} B'_{i,\alpha} - E_{i,\alpha} &= \int_S (\tilde{B}(y) - \beta + \beta) \sum_S G(y,x)\rho_i \tilde{\psi}_{i,\alpha}(x)\, dA_x\, dA_y \\ &= \int_S (\tilde{B}(y) - \beta) \sum_S G(y,x)\rho_i \tilde{\psi}_{i,\alpha}(x)\, dA_x\, dA_y \\ &+ \beta\rho_i \cdot \int_S \tilde{\psi}_{i,\alpha}(x) \int_S G(x,y)\, dA_y\, dA_x \end{aligned}$$

The latter integral is

$$\int_S \tilde{\psi}_{i,\alpha}(x) \int_S G(x,y)\, dA_y\, dA_x = \int_S \tilde{\psi}_{i,\alpha}(x)\, dA_x .$$

For an orthogonal basis obtained by uniform mapping of a canonical orthonormal basis,

$$\int_S \tilde{\psi}_{i,\alpha}(x)\, dA_x . = \frac{1}{A_i} \int_S \psi_{i,\alpha}(x)\, dA_x = \delta_{\alpha,0} .$$

There is only a constant contribution $\rho_i\beta$.

The former integral can be estimated as outlined above, choosing $p(y) \propto | \tilde{B}(y) - \beta |$ if local lines are used. The determination of the optimal β can be done with the same technique as for the constant case (§10.3).

Bidirectional transports Also for higher order approximations, it is quite easy to achieve that each traced ray yields a contribution at both its end-points instead of only at one end-point: for a ray traced from y to x, the shooting contribution (at x):

$$\hat{B}_{i,\alpha}(y \to x) = \frac{\tilde{B}(y)G(y,x)\rho(x)\tilde{\psi}_{i,\alpha}(x)}{p(y)G(y,x)}$$

and gathering contribution (at y, derived from (13.9)):

$$\hat{B}_{j,\beta}(y \leftarrow x) = \frac{\tilde{\psi}_{j,\beta}(y)\rho(y)G(y,x)\tilde{B}(x)}{p(y)G(y,x)}$$

shall both be weighted by

$$w(y,x) = \frac{p(y)G(y,x)}{p(y)G(y,x) + p(x)G(x,y)} = \frac{p(y)}{p(y) + p(x)}.$$

There is a contribution to every basis function α at x and β at y. Note that in the constant approximation case, $p(y) = B_j/P_T$, so that the weights $w(y,x) = B_j/(B_j + B_i)$ correspond with (11.20). For global lines with $p(y) = 1/A_T$, the weights are $w(y,x) = 1/2$ as before as well.

13.4 Higher-order random walk radiosity

Although the generalised form factors $G_{i,\alpha;j,\beta}$ do not form a probability distribution, random walk algorithms for higher order radiosity approximations still are possible. Their derivation requires a more general, and at first sight quite awkward, type of random walk, but the resulting score functions turn out to be quite evident generalisations of the score functions for constant approximations.

Only a sketch of the derivation of the collision shooting and gathering random walk estimators for higher order approximations will be presented here. A more formal derivation, like in chapter 7, is possible by suitable definition of the state space. It turns out however that the variance of these estimators increases as $\mathcal{O}(K^{2\tau_{\text{av}}})$ instead of $\mathcal{O}(K^2)$ for a product basis of order K, where τ_{av} denotes the average length of the paths. Also the experiments indicate that the variance can be much higher than with stochastic relaxation, so that the discrete random walk method is not recommended in practice for higher order approximations.

13.4.1 Gathering random walk

Equation (13.3) can be written in the following form, similar to what was done above for stochastic relaxation:

$$\begin{aligned} B_{i,\alpha} &= E_{i,\alpha} + \sum_{j,\beta} \rho_i \int_S \tilde{\psi}_{i,\alpha}(x) \int_S G(x,y)\psi_{j,\beta}(y)\, dA_y\, dA_x \cdot B_{j,\beta} \\ &= E_{i,\alpha} + \sum_{j,\beta} B_{j,\beta} \int_S \int_S g_{i,\alpha;j,\beta}(x,y)\, dA_y\, dA_x \end{aligned} \tag{13.15}$$

where

$$g_{i,\alpha;j,\beta}(x,y) = \rho_i \tilde{\psi}_{i,\alpha}(x) G(x,y) \psi_{j,\beta}(y) \tag{13.16}$$

$B_{i,\alpha}$ can be estimated using random walks originating at i. Transitions from i,α to j,β can be sampled as follows:

1. Survival/absorption test with probability ρ_i of survival;
2. Choose a point x on patch i. At first sight, it seems best to sample x according to a pdf that is proportional to $|\tilde{\psi}_{i,\alpha}(x)|$. The benefit of doing so is however quite small, especially since one is generally interested in estimating $B_{i,\alpha}$ for all basis functions α. We therefore propose the uniform pdf $p(x|i) = \chi_i(x)/A_i$, which is independent of the approximation term α, so that the difference between the estimators for different α will only be in the scores, and not in the sampling;
3. Choose a point y conditional on x with pdf $p(y|x) = G(x,y)$. This can be done by tracing a uniformly distributed line originating at x. With global lines, x and y are sampled simultaneously according to the combined pdf $\chi_i(x)G(x,y)/A_i$;
4. Choose the patch j containing y as destination patch: $p(j|y) = \chi_j(y)$;
5. Choose a approximation term β on j with pdf

$$p(\beta|y) = \frac{|\psi_{j,\beta}(y)|}{\sum_\beta |\psi_{j,\beta}(y)|}.$$

The complete transition probability from i, α, to j, β using points $x \in S_i$ and $y \in S_j$ thus is

$$p_{i,\alpha;j,\beta}(x,y) = \rho_i \frac{\chi_i(x)}{A_i} G(x,y) \chi_j(y) \frac{|\psi_{j,\beta}(y)|}{\sum_\beta |\psi_{j,\beta}(y)|}. \tag{13.17}$$

Define for $x \in S_i$ and $y \in S_j$:

$$q_{i,\alpha;j,\beta}(x,y) = \frac{g_{i,\alpha;j,\beta}(x,y)}{p_{i,\alpha;j,\beta}(x,y)} = A_i \tilde{\psi}_{i,\alpha}(x) \left(\sum_\beta |\psi_{j,\beta}(y)| \right) \frac{\psi_{j,\beta}(y)}{|\psi_{j,\beta}(y)|} \tag{13.18}$$

The following scores for a path $i = j_0, \alpha = \beta_0 \to j_1, \beta_1 \to \cdots \to j_\tau \beta_\tau$ over points $x_0 \in S_{j_0}, y_1 \in S_{j_1}; x_1 \in S_{j_1}, y_2 \in S_{j_2}; \ldots; x_{\tau-1} \in S_{j_{\tau-1}}, y_\tau \in S_{j_\tau}$ then yield a first collision estimator for $B_{i,\alpha}$:

$$s_{i,\alpha;j_0,\beta_0,\ldots j_\tau,\beta_\tau}(x_0, y_1; \ldots; x_{\tau-1}, y_\tau) = \sum_{t=0}^{\tau} q_{j_0,\beta_0;j_1,\beta_1}(x_0, y_1) \cdots q_{j_{t-1},\beta_{t-1};j_t,\beta_t}(x_{t-1}, y_t) E_{j_t,\beta_t}.$$

A lengthy, but straightforward, calculation shows that $E[s_{i,\alpha}] = B_{i,\alpha}$ indeed. Source term estimation suppression yields:

$$s'_{i,\alpha;j_0,\beta_0;\ldots;j_\tau,\beta_\tau}(x_0, y_1; \ldots; x_{\tau-1}, y_\tau) = \rho_i \sum_{t=1}^{\tau} q_{j_0,\beta_0;j_1,\beta_1}(x_0, y_1) \cdots q_{j_{t-1},\beta_{t-1};j_t,\beta_t}(x_{t-1}, y_t) E_{j_t,\beta_t}.$$

The expectation $E[s'_{i,\alpha}] = B_{i,\alpha} - E_{i,\alpha}$.

The main difference w.r.t. the collision gathering random walk estimator for constant radiosity approximations is in the non-unity multiplicative factors (13.18) associated with each transition. These result because the generalised form factors $G_{i,\alpha;j,\beta}$ do not form a probability distribution and thus a non-analog sampling must be used.

The use of expected values The estimators above use only one approximation term β_t at each collision. Using a variance reduction technique called "the use of expected values", or "analytical summation" (§4.3.6), it is possible to derive improved estimators, that use all approximation terms at each collision.

Consider first the first-order term in the Neumann expansion of $B_{i,\alpha}$ according to (13.15). This term corresponds to direct illumination:

$$b^1_{i,\alpha} = \sum_{j,\beta} E_{j,\beta} \int_S \int_S g_{i,\alpha;j,\beta}(x,y)\, dA_y\, dA_x.$$

The expectation of the scores above for paths of length 1 that originate at i is

$$\begin{aligned} E[s^1_{i,\alpha}] &= \sum_{j,\beta} \int_S \int_S q_{i,\alpha;j,\beta}(x,y) E_{j,\beta} \cdot p_{i,\alpha;j,\beta}(x,y)\, dA_y\, dA_x \\ &= \sum_{j,\beta} E_{j,\beta} \int_S \int_S g_{i,\alpha;j,\beta}(x,y)\, dA_y\, dA_x \quad = \quad b^1_{i,\alpha} \end{aligned}$$

The summation over β can be done analytically as follows: consider the modified scores for length-1 paths from i,α to j,β over points $x \in S_i$ and $y \in S_j$:

$$\begin{aligned}\tilde{s}^1_{i,\alpha;j}(x,y) &= \sum_\beta p(\beta|y) q_{i,\alpha;j,\beta}(x,y) E_{j,\beta} \\ &= A_i \tilde{\psi}_{i,\alpha}(x) \sum_\beta \psi_{j,\beta}(y) E_{j,\beta}.\end{aligned}$$

Such scores shall be sampled will probability

$$p_{ij}(x,y) = \frac{p_{i,\alpha;j,\beta}(x,y)}{p(\beta|y)} = \chi_i(x)\frac{1}{A_i}G(x,y)\chi_j(y)$$

Then,

$$E[\tilde{s}^1_{i,\alpha}] = \sum_j \int_S \int_S \sum_\beta p(\beta|y) q_{i,\alpha;j,\beta}(x,y) E_{j,\beta} \cdot \frac{p_{i,\alpha;j,\beta}(x,y)}{p(\beta|y)}\, dA_y\, dA_x = b^1_{i,\alpha}$$

equals $b^1_{i,\alpha}$ as well.

Generalisation of this scheme to analytical summation over the approximation terms β_t at every collision, eventually yields the following scores:

$$\begin{aligned}&\tilde{s}_{i,\alpha;j_0,\ldots,j_\tau}(x_0,y_1;\ldots;x_{\tau-1},y_\tau) \\ &\qquad = \rho_i A_{j_0} \tilde{\psi}_{j_0}(x_0) \sum_{t=1}^{\tau} r_{j_1}(y_1,x_1)\cdots r_{j_{t-1}}(y_{t-1},x_{t-1})\tilde{E}(y_t) \qquad (13.19)\end{aligned}$$

with

$$\tilde{E}(y_t) = \sum_{\beta_t} E_{j_t,\beta_t}\psi_{j_t,\beta_t}(y_t) \text{ and } r_{j_k}(y_k,x_k) = \sum_{\beta_k} \psi_{j_k,\beta_k}(y_k) A_k \tilde{\psi}_{j_k,\beta_k}(x_k).$$

Source term estimation has been suppressed.

Sampling transitions with pdf $p_{ij}(x,y)$ can be done as described above, for sampling according to $p_{i,\alpha;j,\beta}(x,y)$, except that the last step of selecting a approximation term β is skipped. The remaining transition sampling procedure is *exactly* the same as for constant approximations: at each transition, first a survival/absorption decision is made, next, a uniform point x is chosen and a uniformly distributed line is traced to find y and the destination patch. Generalising the gathering collision random walk radiosity estimator for constant approximations (§7.4) to higher order approximations thus only requires that the scores be computed differently. The tracing of the paths itself remains unchanged.

The scores are best computed by accumulating the multiplicative factors $\rho_i A_i \tilde{\psi}_{i,\alpha}(x_0)$ at the origin and $r_{j_k}(y_k,x_k)$ at survived collisions, while tracing the path. That is a straightforward extension to algorithm 18 in §7.1.4.

Because part of the summations is done analytically instead of by sampling, the variance will be lower than with the first gathering random walk estimators for higher order approximations that were presented above.

13.4.2 Shooting random walk

The shooting random walk can be derived in a similar way by use of the adjoint equation. $B_{i,\alpha}$ can be obtained as a scalar product:

$$B_{i,\alpha} = \sum_{k,\gamma} B_{k,\gamma}\delta_{ik}\delta_{\alpha\gamma} = \sum_{k,\gamma} E_{k,\gamma} I_{k,\gamma} \tag{13.20}$$

where $I_{k,\gamma}$ is the solution of

$$I_{k,\gamma} = \delta_{ik}\delta_{\alpha\gamma} + \sum_{j,\beta} I_{j,\beta}\rho_j G_{j,\beta;k,\gamma}$$

This adjoint equation can be brought in a form that better corresponds with (13.15) so that a collision random walk estimator can be derived in exactly the same way:

$$(\rho_k I_{k,\gamma}) = (\rho_i \delta_{ik}\delta_{\alpha\gamma}) + \sum_{j,\beta} \rho_k \tilde{G}_{k,\gamma;j,\beta}(\rho_j I_{j,\beta})$$

with (by switching the integrals in the definition of $G_{j,\beta;k,\gamma}$):

$$\tilde{G}_{k,\gamma;j,\beta} = \int_S \psi_{k,\gamma}(y) \int_S G(y,x)\tilde{\psi}_{j,\beta}(x)\, dA_x\, dA_y = G_{j,\beta;k,\gamma}.$$

Instead of (13.20), the following scalar product is estimated:

$$\begin{aligned} B_{i,\alpha} &= \sum_{k,\gamma} \frac{E_{k,\gamma}}{\rho_k}(\rho_k I_{k,\gamma}) \\ &= E_{i,\alpha} + \int_S \tilde{E}(y_0) \int_S G(y_0,x_1) \sum_{j_1,\beta_1} \tilde{\psi}_{j_1,\beta_1}(x_1)(\rho_{j_1} I_{j_1\beta_1})\, dA_{x_1} dA_{y_0}. \end{aligned}$$

This formula suggests source term estimation suppression and path birth probabilities $\pi_{j_0} = \Phi_{j_0}/\Phi_T$, exactly as in the constant radiosity case. Also transitions are sampled exactly the same way as in the gathering random walk, and for a constant radiosity approximation. The score to $B_{i,\alpha}$ associated with a random walk $j_0 \to j_1 \to \cdots \to j_\tau$ over points $y_0 \in S_{j_0}, x_1 \in S_{j_1}; y_1 \in S_{j_1}, x_2 \in S_{j_2}; \ldots; y_{\tau-1} \in S_{j_{\tau-1}}, x_\tau \in S_{j_\tau}$ is:

$$\begin{aligned} &\tilde{s}_{i,\alpha;j_0,\ldots,j_\tau}(y_0,x_1;\ldots;y_{\tau-1},x_\tau) \\ &\quad = \Phi_T \frac{\tilde{E}(y_0)}{E_{j_0}} \sum_{t=1}^{\tau} \tilde{r}_{j_1}(x_1,y_1)\cdots\tilde{r}_{j_{t-1}}(x_{t-1},y_{t-1})\delta_{ij_t}\delta_{\alpha\beta_t}\rho_i\tilde{\psi}_{i,\alpha}(x_t) \end{aligned} \tag{13.21}$$

with E_{j_0} the average self-emitted radiosity on j_0 and

$$\begin{aligned} \tilde{E}(y_0) &= \sum_{\beta_0} E_{j_0,\beta_0}\psi_{j_0,\beta_0}(y_0) \\ \tilde{r}_{j_k}(x_k,y_k) &= \sum_{\beta_k} \tilde{\psi}_{j_k,\beta_k}(x_k) A_k \psi_{j_k,\beta_k}(y_k). \end{aligned}$$

Again, the required changes to the collision shooting random walk algorithm 19 for constant radiosity approximations are quite straightforward: the sampling itself of the random walks is exactly the same. The multiplicative factors $\Phi_T \tilde{E}(y_0)/E_{j_0}$ associated with the origin (on a light source) and $\tilde{r}_{j_k}(x_k, y_k)$ associated with survived collisions, are accumulated while tracing the path. Each time a patch i is hit, a score is recorded to every approximation term α. The score to $B_{i,\alpha}$ equals the accumulated weights times $\rho_i \tilde{\psi}_{i,\alpha}(x_t)$, where x_t is the point where i is hit. (The multiplicative factor $\tilde{r}_{j_t}$ is accumulated after scoring on j_t.)

13.4.3 Variance and required work as a function of the approximation order

A detailed variance analysis will not be given. It turns out that for both the gathering as well as the shooting estimator, the squared scores contains factors

$$\prod_{t=1}^{\tau-1} \sum_{\beta_t} \int_{S_{j_t}} \int_{S_{j_t}} A^2 \tilde{\psi}^2_{j_t,\beta_t}(x) \psi^2_{j_t,\beta_t}(y)\, dA_x\, dA_y$$

For an orthogonal basis obtained by uniform mapping of a orthonormal canonical basis,

$$\int_{S_{j_t}} \int_{S_{j_t}} A^2 \tilde{\psi}^2_{j_t,\beta_t}(x) \psi^2_{j_t,\beta_t}(y)\, dA_x\, dA_y = 1.$$

If there are M basis functions ψ_{j_t,β_t} on each patch j_t ($M = K^2$ for a K-th order product basis), the variance will contain factors

$$\prod_{t=1}^{\tau-1} M = M^{\tau-1}.$$

The variance for higher order approximations will depend very strongly on the path length.

Intuitively, this is clear: long paths make many transitions. With each transition is associated a multiplicative non-constant factor r (gathering) or $\tilde{r}$ (shooting). The product of many such factors can vary greatly and will vary more for longer products.

For a continuous random walk however, the difference between the scores for higher order approximations and constant approximations is not in the transitions, but only in a factor $\psi_{i,\alpha}(x)$ either at the origin of the random walk (for gathering) or at a hit point (shooting). There is no accumulation of factors at each transition. A detailed analysis of the variance, for instance by generalising the exposition of chapter 7 by replacing sums over the discrete set of states of a discrete random walk to integrals over the continuous set of states of a continuous random walk, will reveal the same factors $\sum_\alpha \psi^2_{i,\alpha}(u, v)$ in the variance as for Monte Carlo generalised form factor computation and stochastic relaxation radiosity.

13.5 Empirical results

The figures 13.1, 13.2 and 13.3 show the obtained results for the non-product constant, linear, quadratic and cubic bases described in §2.2.6. The number of basis functions

is 1, 3, 6 and 10 respectively. The number of samples was chosen proportional to the number of basis functions. Figure 13.1 shows results obtained with the continuous shooting random walk method [47, 14]. Figure 13.2 shows the results obtained with stochastic Jacobi relaxation using local lines, as proposed in this chapter. In both cases, the visual quality of the resulting images is similar, confirming the rule, derived in this chapter, that for an approximation with M basis functions, about M times more samples are required.

The visual quality of the images obtained with discrete collision shooting random walks, shown in figure 13.3, is worse for non-constant approximations. The difference is larger for higher order. Some patches in these figures show high "spike" noise. This noise is due to long paths, for which the product of the factors $\tilde{r}_{j_k}(x_k, y_k)$ can vary greatly.

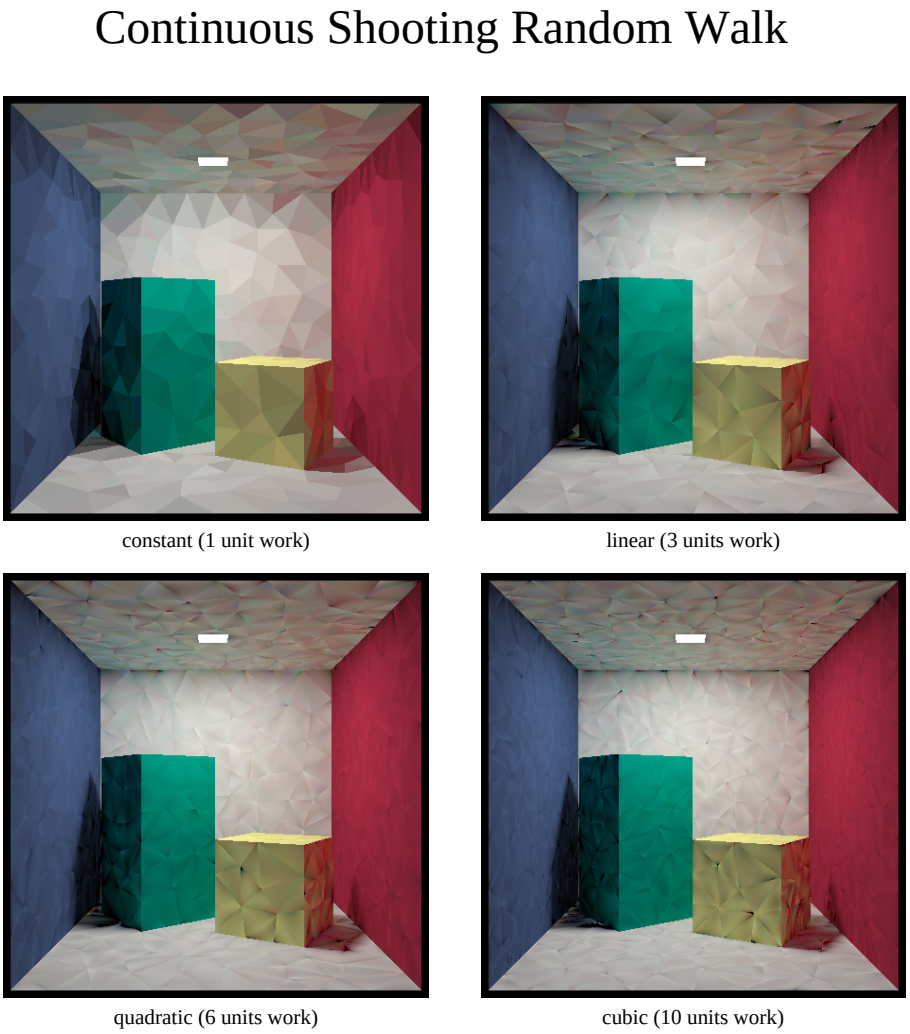


Figure 13.1: Higher-order radiosity approximations obtained with continuous shooting collision random walk radiosity [47, 14]. The number of samples was chosen proportional to the number of basis functions in each approximation. The variance in the results is similar.

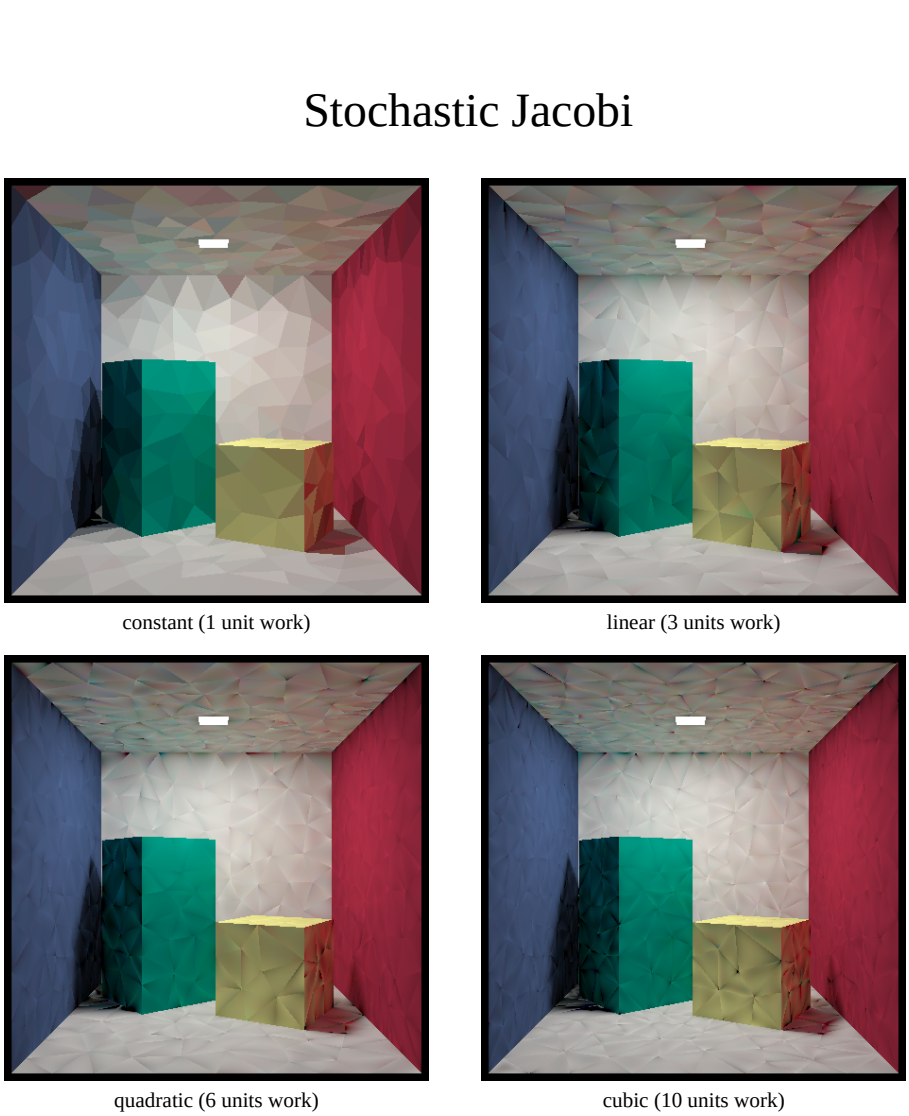


Figure 13.2: Higher-order radiosity approximations obtained with the stochastic relaxation radiosity method presented in this chapter. The number of samples was chosen proportional to the number of basis functions in each approximation. The variance in the results is similar. The amount of work is equal as in figure 13.1 for same approximation.

Figure 13.3: Higher-order radiosity approximations obtained with discrete shooting collision random walk radiosity, as explained in this chapter. The number of samples was chosen proportional to the number of basis functions in each approximation. For the same approximation, the amount of work was the same as for figures 13.1 and 13.2. The variance is however higher for non-constant approximations. The difference with the other methods is larger for higher approximation order. This is due to the multiplicative factors $\tilde{r}$ associated with each transition. Especially for long paths, high "spike" noise results.

13.6 Conclusion

In this chapter, the discrete Monte Carlo radiosity methods of the previous chapters have been extended to higher order approximations. Their cost has been analysed as a function of the approximation order. The conclusion is that

- Monte Carlo form factor computation for higher order approximations is very straightforward. The variance for individual form factors does not seem to depend greatly on approximation order for the basis functions proposed in §2.2.6. The computation cost for an approximation with M such basis functions is about M times higher than for a constant approximation. This is not because the Monte Carlo method has more problems computing the integrals, but because the variance for each basis function adds up;

- The derivation of stochastic relaxation methods for higher order approximations is quite different than for constant approximations, because the generalised form factors do not form a probability distribution. The continuous integral kernel $G(x, y)$ however can be used as a probability distribution. The use of view-importance, the constant control radiosity technique and bidirectional transports is equally well possible with the extended stochastic Jacobi method for higher order approximation. The amount of work is again proportional to the number of basis functions;

- The derivation of discrete random walk methods for higher order approximations is even more awkward, but leads to fairly straightforward extensions of the random walk radiosity estimators of chapter 7. Their variance is however considerable higher, so that the discrete random walk methods are not recommended in practice for higher order approximations. The reason is that with each transition, a non-constant multiplicative factor is associated. The product of these factors can vary greatly, especially for long paths and high approximation orders;

- With a continuous random walk method for higher order approximations [47, 14], there are no such multiplicative factors associated with each transition. The variance again is proportional to the number of basis functions, confirming an observation by Feda [47].

The possibility to compute higher order approximations with Monte Carlo radiosity allows very high quality to be obtained in regions of the scene where the radiosity varies smoothly. The (discrete) stochastic relaxation method is certainly not worse than the continuous random walk method, especially because the variance reduction techniques proposed in previous chapters appear to work well for higher order approximations too.

Figure 13.4 shows two converged images generated from a single radiosity solution of cubic order computed with the stochastic relaxation method for higher order approximations. Once the radiosity solution is available, the time needed in order to generate an image for a new viewpoint is only a fraction of a second, so that interactive "walk-throughs" are possible. The images illustrate that the computed higher order solution will be of very high quality on a sufficiently fine mesh, illustrating again that the Monte Carlo radiosity algorithms deal well with computational errors. The shown images exhibit some disturbing artifacts near shadow boundaries. These are due to

the discretisation of the scene and will not be reduced by the Monte Carlo method. Chapter 3 focussed on the control of exactly this kind of errors by means of hierarchical refinement. In the next chapter, we present a first step towards the synthesis of Monte Carlo solution of the radiosity equations, and the control of the discretisation error by means of hierarchical refinement according to the principles explained in chapter 3.

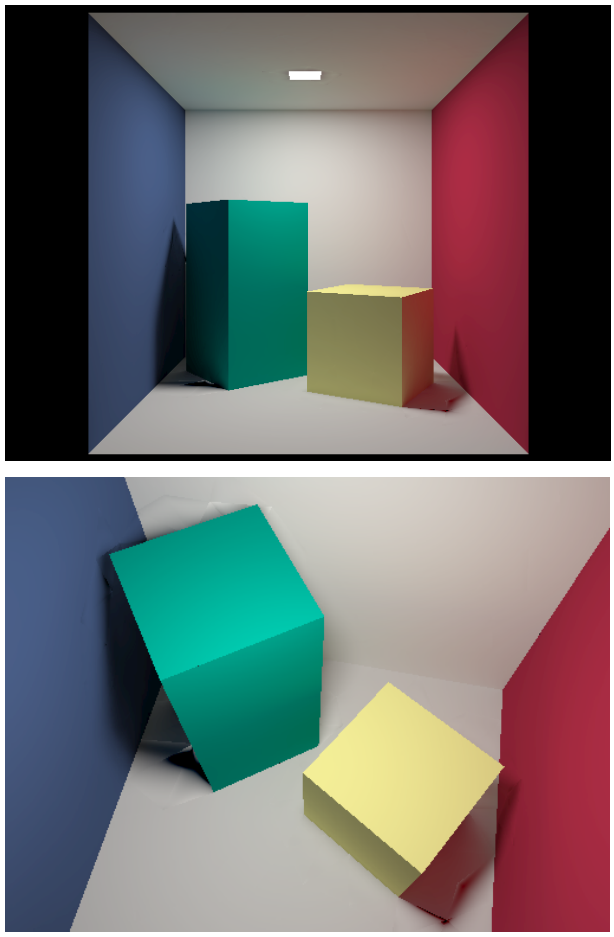


Figure 13.4: Two images generated from the same converged cubic approximation solution. Once the solution has been obtained, a new image for a new viewpoint can be generated in fractions of a second. These images illustrate also that the (discrete) higher order stochastic relaxation radiosity algorithm proposed in this chapter can yield very high image quality in regions where the illumination varies smoothly: computational error is dealt with effectively in Monte Carlo radiosity. In the neighbourhood of discontinuities however, disturbing image artifacts remain due to the discretisation error. The resulting image artifacts would be avoided if discontinuity meshing were used.

14 Hierarchical Monte Carlo Radiosity

In previous chapters, it has been shown that the Monte Carlo method is well suited in order to solve the systems of linear equations that arise in radiosity without hierarchical refinement. This chapter will deal with the incorporation of hierarchical refinement in Monte Carlo radiosity. Although the basic ideas presented here are more generally applicable, the text will focus on the stochastic Jacobi iterative method (§6.4).

The synthesis of hierarchical refinement and Monte Carlo radiosity offers advantages with respect to both deterministic hierarchical radiosity and non-hierarchical Monte Carlo radiosity. The relation with previous work will be discussed first in §14.1. In §14.2, per-ray refinement will be presented as a novel strategy in order to incorporate hierarchical refinement in Monte Carlo radiosity. Some implementation issues are covered in §14.3. Empirical results and the benefits and limitations of the new strategy will be discussed in §14.4.

14.1 Previous work

The incorporation of hierarchical refinement (HR) in Monte Carlo radiosity (MCR) offers advantages over both HR and MCR. Some previous approaches for incorporating HR in MCR are discussed at the end of this section.

Hierarchical refinement radiosity

Originally, hierarchical refinement has been proposed [28, 68] with two purposes:

- automatic adaptive meshing;
- a firm reduction of the number of form factors that needs to be computed and stored. With clustering [151, 147], the number of form factors is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ (cfr. §2.3.5).

Even so, deterministic hierarchical radiosity suffers from two important problems:

- Computational error in the form factors and the refinement indicator that drives hierarchical refinement, will result in incorrect or even missed energy transport. An example will be shown in §14.4. The issue of computational error has been discussed before in chapter 3;
- Although storage complexity is linear in the number of elements, storage requirements are still quite high [183]: usually, a few hundred bytes per element are needed for storing the interactions and form factors.

With Monte Carlo radiosity, form factor storage can be avoided. Monte Carlo radiosity is also more reliable than deterministic radiosity system solution.

Non-hierarchical Monte Carlo radiosity

The variance of shooting-type stochastic relaxation algorithms (chapter 6) and the shooting random walk estimators for radiosity (§7.4.2) is inverse proportional to the patch area. On small patches, the variance will be high. Besides providing automatic adaptive meshing, hierarchical refinement with clustering will lead to a reduction of the variance by letting small patches interact with other patches in group.

Previous work in combining hierarchical refinement in Monte Carlo radiosity

The first application of Monte Carlo in radiosity concerned the computation of all form factors from a given source patch with the whole environment as a ray-tracing based alternative for the hemi-cube method. Such form factor computation was needed in the context of progressive refinement radiosity (§5.3).

Several authors [101, 91, 89] have addressed the extension of such Monte Carlo form factor computation to hierarchical progressive refinement radiosity. The basic idea in these three proposals is identical: a large amount of rays is shot from the source patch. The surrounding scene is subdivided in receiver elements so that each receiver element (a surface or cluster) receives the same amount of rays. These approaches have the following disadvantages:

- they correspond to only a single, primitive refinement strategy. It is not clear how they can be adapted in order to incorporate more advanced refinement strategies;
- these approaches will only work well when a sufficiently large amount of rays is shot at once from the source patch. This is not the case in more advanced Monte Carlo radiosity algorithms.

More recently, Tobler et al. [171] have presented an adaptive meshing scheme for continuous shooting random walk radiosity (§7.1.2). By simultaneously keeping track of incident particles on successive hierarchical element levels, smoothness assumption violations can be detected. When an element is subdivided, part of the incident samples had to be forgotten, resulting in about 25% of the samples being "wasted". This technique also doesn't solve the under-sampling of small patches.

14.2 Incorporation of hierarchical refinement in Monte Carlo radiosity

14.2.1 Observations

The new strategy for incorporating hierarchical refinement in Monte Carlo radiosity is based on two observations:

Hierarchical refinement and wavelet pre-conditioning of systems of linear equations

Hierarchical refinement in radiosity is related with wavelet-based pre-conditioning techniques for linear systems [78]. Wavelet-based pre-conditioning transforms a given,

large, matrix into a multi-resolution representation by grouping entries, a bit similar to wavelet-based analysis and compression of images or 3D-meshes. The multi-resolution representation allows to solve the linear system faster.

Straightforward application of wavelet-based pre-conditioning in radiosity, would correspond to the computation of a very large and detailed form factor matrix between very fine elements first. Next, this form factor matrix would be simplified. It is clear that such an approach is not feasible. In radiosity, a top-down rather than bottom-up approach is taken: initial rough form factor entries are refined into more accurate entries after predicting in advance that the rough representation would not suffice [68, 60, 140].

The new strategy presented below can be viewed formally as the Monte Carlo solution of the preconditioned system of radiosity equations, using a top-down rather than bottom-up approach as in deterministic hierarchical radiosity.

Algorithmic difference between deterministic and stochastic iterative Jacobi iterations

Consider stochastic Jacobi iterations versus deterministic Jacobi iterations with Monte Carlo ray-traced form factors with uniformly distributed rays:

- Deterministic solution method: for each pair of patches, shoot a number of rays in order to compute the form factor between the patches. Multiply the form factor by the power of the source patch and the reflectivity of the receiver patch. Add the result to the received power of the receiver patch;
- Stochastic solution method: shoot a number of rays through the scene, for each ray, determine which patches are connected by the ray. Multiply the power of the source patch by the reflectivity of the receiver patch and divide by the expected number of rays between the two patches. Accumulate the result of each ray separately on the receiver patch.

The deterministic and stochastic method differ in the order in which rays are shot and in how their contribution to the form factor is translated into energy transfers: in the deterministic method, rays are shot in a “coherent” fashion, while in the stochastic method, subsequent rays rarely connect the same pair of elements. In the deterministic method, the contributions of all rays to the form factor between a fixed pair of patches is accumulated first and then translated into a power transfer from source to receiver patch. Because rays are shot “incoherently” in the stochastic method, the contribution of each individual ray is immediately translated into a (partial) energy transfer from source to receiver.

14.2.2 Per-ray refinement

Without hierarchical refinement, each shot ray in a Monte Carlo radiosity algorithm will be used in order to transfer power between the pair of patches that are connected by the ray. With hierarchical refinement, a hierarchy of elements is constructed on each patch (§2.3.2). We propose to incorporate hierarchical refinement in Monte Carlo radiosity as follows (see figure 14.1):

- Construct the element hierarchies lazily, per ray;

- For each ray, an appropriate pair of sub-elements containing the end-points of the ray is determined. Power is transferred between this pair of elements instead of between the patches connected by the ray.

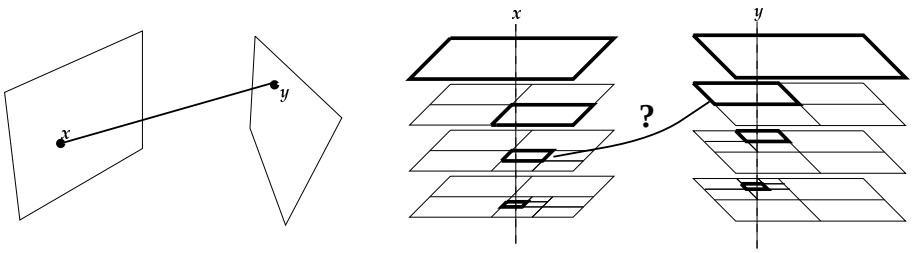


Figure 14.1: Per-ray hierarchical refinement in Monte Carlo radiosity: for each individual ray, connecting two points x and y, the algorithm will determine which level of the element hierarchies at x and y is appropriate for computing light transport from x to y. The element hierarchies are lazily constructed. In non-hierarchical Monte Carlo radiosity, light transport would be computed between the input patches containing the end-points x and y of the ray. In deterministic hierarchical radiosity, the interaction levels are determined once, before or during element-to-element form factor computation.

These effects are obtained by carrying out hierarchical refinement for each shot ray separately instead of in a single sweep as explained in §2.3.3. Except that refinement is done for each ray separately, the refinement procedure, shown in algorithm 20 is identical as in deterministic hierarchical radiosity (algorithm 2 on page 25, compare figure 14.2 with figure 2.7 on page 26).

The same hierarchical refinement indicators and strategies can be employed (at least in theory, some practical limitations will be discussed in the next sections). This is possible because hierarchical refinement indicators and strategies are based on scene geometry, reflectance, emittance and possibly the current intermediate radiosity solution. These data are also available in Monte Carlo radiosity.

Algorithm 20: Per-ray hierarchical refinement: recursively refine between a receiver ELEMENT rcv and a source ELEMENT src containing the end-points x and y of a ray shot in Monte Carlo radiosity. This algorithm shall be called with the top-level elements containing x and y. These top-level elements are the single top-level cluster containing the whole scene when clustering is used [151, 147], or the top-level surface elements corresponding to the patches containing x and y if no clustering is used.

Refine(ELEMENT rcv, ELEMENT src)

1. if candidate interaction rcv←src is not admissible according to refinement indicator, then
   (a) subdivide one or both elements, or increase approximation order, . . .
   (b) recursively call Refine() for the single new candidate interaction between sub-elements containing the end-points x and y of the ray.
2. else
   (a) transfer energy over the accepted interaction rcv←src;

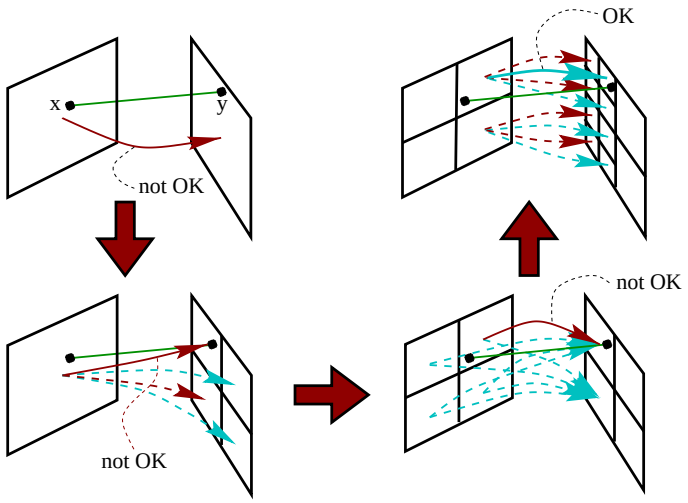


Figure 14.2: The per-ray hierarchical refinement algorithm first considers a candidate interaction between the top-level elements containing the end-points x and y of the ray. If the candidate interaction is deemed non-admissible by the refinement indicator, refinement is carried out, yielding a set of new candidate interactions. Of this set, only the single candidate sub-interaction between elements that contain the end-points x and y of the ray is further considered. Here, a candidate interaction is refined by regular subdivision of one of the interacting elements.

14.3 Implementation

14.3.1 Hierarchical stochastic Jacobi iterative method

In the non-hierarchical stochastic Jacobi iterative method, the double sum over pairs of patches i and j in the right hand side of the following equation is estimated by Monte Carlo:

$$P'_k = \Phi_k + \sum_i \sum_j P_i F_{ij} \rho_j \delta_{jk}.$$

The sum can be viewed as a sum over all non-hierarchical interactions in the scene. With hierarchical refinement, a similar sum needs to be estimated. This sum is a sum over all admissible interactions in the scene instead of a sum over pairs of patches.

The incorporation of hierarchical refinement in algorithm 13 requires the following changes:

- step 3d needs to iterate over all leaf elements in the scene, since at the leaf elements the most detailed representation of power is available;
- step 3(d)ivC is changed as follows: first, the nearest intersection point y of a ray, originating at x being shot in direction Θ, with the surfaces in the scene is determined. Next, the admissible pair of elements containing the end-points x and y is determined as explained in the previous section;
- step 3(d)ivD: accumulate $\frac{1}{N}\rho_r P_s / q_s$ on the admissible receiver element r containing y. s denotes the admissible source element, which contains the origin x of the ray;
- step 3f: received power δP is first pushed down to the leaf elements. At the leaf elements, self-emitted power Φ is added. Finally, it is pulled up and replaces P at every level. This push-update-pull sweep is identical as in deterministic hierarchical radiosity.

14.3.2 The number of rays in each iteration

With hierarchical refinement, radiosity will be received at different element levels instead of being received all on a single patch. The non-self-emitted radiosity $\tilde{b}(x)$ at a point x on a surface will be obtained as the sum of the partial radiosities b_l, received at each element l in the hierarchy that contains x (§2.3.4). It can be shown that the variance is given by

$$V[\hat{b}(x)] = \rho(x) P_T \sum_l \frac{b_l}{A_l} - (\tilde{b}(x))^2 \quad \leq \quad \rho(x) P_T \sum_l \frac{b_l}{A_l}. \tag{14.1}$$

The sum is over all elements in the hierarchy that contain x.

Without hierarchical refinement, the variance is given by (see §6.4.4):

$$V[\hat{b}(x)] = \rho(x) P_T \frac{b_i}{A_i} - (\tilde{b}(x))^2. \tag{14.2}$$

where i is the patch containing x. The radiosity b_i received on this patch would correspond with $\sum_l b_l$. The number of samples in an iteration was chosen so that the (computational) error on a patch with average radiosity B_{av} would be smaller than B_{av} with 99.7% confidence, using the following expression:

$$3\sqrt{\frac{V[\hat{b}]}{N}} \approx B_{av}.$$

Unfortunately, a similar derivation of a heuristic for choosing N in the case of hierarchical refinement fails, because the subdivision in elements l is not known in advance.

The resulting subdivision in elements depends on scene geometry and optical characteristics and also on the the refinement criterion and strategy that is used. Often, refinement is carried out so that each admissible interaction will transport approximately the same amount of power P_ε (see §2.3.5). It has been found reasonable to choose N as follows:

$$N \approx \frac{P_T}{P_\varepsilon} \tag{14.3}$$

This choice of N leads to

$$\frac{V[\hat{b}]}{N} \approx \rho(x) \sum_l \frac{P_\varepsilon}{A_l} b_l.$$

14.3.3 Quasi-Monte Carlo sampling

Quasi-Monte Carlo (QMC) sampling can yield a considerably improved convergence rate (see chapter 12). In non-hierarchical stochastic relaxation radiosity (§12.2.2), it was proposed to keep a sample index with each patch as a means of breaking correlations between low discrepancy samples. This sample index determines the next sample to be generated from the patch.

Hierarchical refinement requires sample sequences on sub-elements. As a sample sequence on a sub-element, we will take a subset of all samples generated on the patch containing the sub-element: those with origin within the sub-element (see figure 14.3). Taking the next sample on a sub-element involves generating subsequent samples on the patch until one is found with origin within the sub-element.

Whenever a leaf element is subdivided, a sample index need to be computed for each resulting new leaf element. The next-sample-index can be copied from the parent element.

When regular subdivision is used, finding a subsequent sample on a sub-element is considerably facilitated with a so called (t, k) low discrepancy sample sequence. In the implementation, a base-2 31-bits 4-dimensional Niederreiter sequence [15] was used. With this series, each group of four successive samples with index $4i, 4i+1, 4i+2, 4i+3$ on a parent element, will contain exactly one sample within each of the four regular sub-elements. The (t, k) property of this sequence also ensures that directions of rays with origin restricted to a sub-element still will be properly distributed.

In Appendix D, an appropriate sub-element numbering scheme and quadrilateral-to-triangle mapping is proposed that allows the same sample generation and counting routines to be used for triangular and quadrilateral elements.

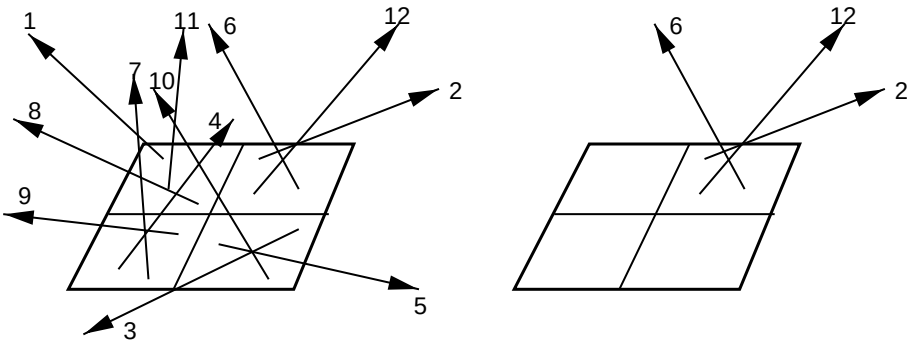


Figure 14.3: The left figure shows a number of samples on a parent element. A suitable subsequence of samples on a sub-element is obtained by skipping those samples that have their origin on other sub-elements than the one of interest. By choosing an appropriate number sequence [15], skipping irrelevant samples can be done efficiently and ray directions still will be well distributed.

14.4 Empirical results and discussion

Comparison with previous work

Deterministic hierarchical refinement radiosity Hierarchical Monte Carlo radiosity (HMC) can be viewed as a variant of HR in which form factors are computed (implicitly) as in MCR. It inherits the benefits of MCR: more reliable and efficient form factor computation while avoiding form factor storage without double work. Form factor storage was previously avoided in HR by recomputing form factors whenever needed (see e.g. [159]). In HMC algorithms, new form factor samples always *improve* the (implicit) form factor estimates.

As in MCR, every HMC sample connects mutually visible points. Because of this, never an attempt is made to refine interactions between totally occluded elements. In deterministic HR, global visibility pre-processing [169] have been proposed as a remedy. Global visibility pre-processing still can be used for faster ray tracing in MCR, but is not so required as in deterministic HR. For good element-to-element form factor computation in HR, the form factor integral should be restricted to the visible part of the source (§3.5). In order to obtain this effect, sophisticated algorithms and data structures, such as a back-projection algorithm or the visibility skeleton [162, 38, 40] have been proposed in deterministic HR. In HMC, the implicit form factor integration is automatically over the visible parts of the sources only.

Discontinuity meshing [71, 103] will however still be needed for high quality penumbra. Unfortunately, such algorithms are restricted to moderately complex scenes ($< 10^4$ input patches). HMC without discontinuity meshing will yield reasonably good image quality for *much more complex scenes*. We report on experiments indicating the reliability, efficiency and low storage requirements of HMC further in this section.

Non-hierarchical Monte Carlo radiosity The use of a multi-resolution representation of radiosity leads to a reduction of the variance. Imagine a hierarchical element mesh. With hierarchical refinement, the variance on the computed radiosity is given

by (14.1). Now image the "flat" mesh consisting of the leaf elements of the hierarchy only. Without hierarchical refinement, the variance is given by (14.2). The radiosity b_i on a "flat" element corresponds with the sum of the partial radiosities b_l gathered at the various levels of the hierarchical mesh. The difference is variance is

$$V^{\text{hierarchical}} - V^{\text{non-hierarchical}} = \sum_l \frac{b_l}{A_l} - \frac{\sum_l b_l}{A_i}.$$

The difference will always be negative because the area A_l of higher level elements is larger than that of the leaf element i, regardless of the refinement criterion and strategy that is used.

An intuitive explanation is as follows: by depositing radiosity on a higher level in a hierarchical mesh, it is distributed over several small elements instead of being concentrated on a single small element.

Previous approaches at combining hierarchical refinement in Monte Carlo radiosity Unlike [101, 91, 89], per ray refinement can be used for arbitrary refinement indicators and strategies. (There are some practical limitations that will be discussed below). Our strategy also still works for small amounts of rays shot at once from a source. For this reason, it is possible to generate first complete hierarchical radiosity solutions in a shorter time. More smooth progressive variance reduction is possible because the number of samples can be increased in smaller steps.

Compared to [171], the new strategy bases refinement on the geometric and optical properties of a scene, without "wasting" samples. With clustering in HMC, the undersampling problem is solved well.

Reliability

Figure 14.4 shows an example of a problem due to computational error on the refinement oracle in HR. The horizontal patch was erroneously classified as fully occluded from the light source early on during the refinement process in HR. The problem could be reduced by deferring visibility estimation until other criteria, such as small enough unoccluded form factor, are fulfilled [68]. This would however lead to many form factor samples being "wasted" in fully occluded candidate interactions. Visibility pre-processing [169] has been proposed as a solution. In HMC, all form factor samples always connect mutually visible points. The appropriate level of interaction is determined for each sample, after visibility detection. HMC is considerably less sensitive than HR to missing important interactions such as shown in this example.

Computation time

Figure 14.5 shows some results obtained with the hierarchical extension of the stochastic Jacobi iterative method as described in this paper. Although the models that are shown are quite complex, the images only required a few minutes of computation time[1]. While noisy effects are still visible, the most disturbing artifacts in the radiosity solutions are due to a bad initial mesh quality. These artifacts are however not unique to HMC and also show up with other radiosity algorithms.

[1] All experimental data were obtained on a SGI Octane workstation with 195MHz R10000 processors and 256MB RAM.

Comparing running times of HR and HMC for the cubicle office space, we found that a first image showing only direct illumination was obtained after 14 minutes with HR. Six Jacobi iterations, yielding a more complete illumination solution took 89 minutes. With HMC, a first solution, containing the effect of inter-reflections, was obtained already after 3.6 minutes with less than 2 visibility samples per interaction. We found that often no more than 5 to 10 samples per form factor suffice in order to obtain images in which noise is not noticeable. Many more samples are needed to ensure sufficiently reliable element-to-element integration.

Storage requirements

In HMC, interactions need not to be stored explicitly. Compared to deterministic HR with form factor storage, storage is reduced from (at least) 8 bytes per *interaction* with HR to 4 bytes per *element* in HMC with QMC sampling. Experiments within our test-bed rendering system RENDERPARK reveal that a reduction of storage requirements to about 20% is feasible.

Smoothness-based oracles

In the experiments above, the popular low-cost refinement oracle of [68] was used (see also §2.3.5). A smoothness-based oracle (see §2.3.5) will often however result in fewer elements and interactions, while shadow boundaries and other areas where the illumination varies quickly will be reproduced better. In low-gradient regions, the variance will be lower for same amount of samples, because larger elements will be used in order to represent slowly varying radiosity. In high-gradient regions however, a larger variance will be observed as the price for more accurate representation of the radiosity.

We have found that with the smoothness-based oracle of [102], the number of elements is typically reduced by a factor of about two. The analytical form factor and visibility computations in this oracle are however carried out several times for each sample, resulting in an increased computation cost by a factor of 10 to 20. In order to avoid repeated computation of the smoothness estimates, they can be stored for each link (the form factors themselves don't need to be stored with HMC). The observed increase in computation time (factor three) still wasn't compensated by the gain. The need for good low-cost refinement criteria is more urgent in HMC than in HR.

We achieved acceptable smoothness-based refinement similar to [102] by using cheap point-to-element form factor estimates at the vertices and midpoint of an element for estimating an upper form factor bound. The lower bound is always set to zero since it is not determined whether there is full or partial visibility in a link (full occlusion never has to be considered in HMC). Radiosity variation over an element for estimating propagated error is approximated by the difference between maximum and minimum radiosity on descendant leaf elements.

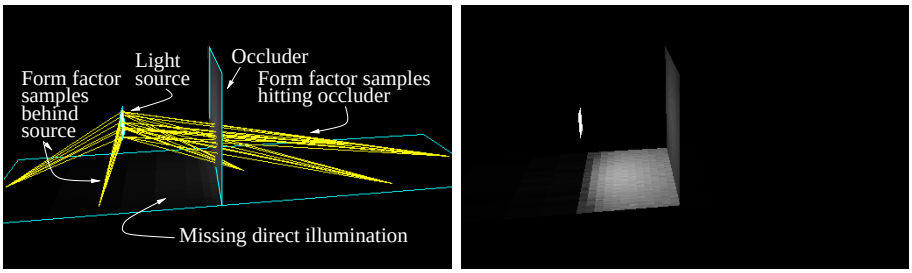


Figure 14.4: With HR (left), the horizontal patch was erroneously classified as fully occluded from the light source because all form factor sample lines either hit the receiver behind the source, or intersected the occluder surface. As a result, direct illumination is missing on the horizontal patch. HMC (right) is much less sensitive to such visibility mis-estimation problems.

Figure 14.5: Complex scenes rendered with hierarchical Monte Carlo radiosity: theatre (39,000 initial polygons, refined into 88,000 elements, 5 minutes), conference room (125,000 polygons, refinement yields 178,000 elements, 9 minutes) and cubicle office space (128,000 polygons, refined into 506,000 elements, 10 minutes).
Model credits: Candlestick Theatre: Design: Mark Mack Architects, 3D Model: Charles Ehrlich and Greg Ward (work conducted as a research project during the Architecture 239X course taught by Kevin Matthews formerly at UC Berkeley, College of Environmental Design). Conference room and cubicle space models by Anat Grynberg and Greg Ward (Lawrence Berkeley Laboratory, Berkeley, California).

14.5 Conclusion

In this chapter, the incorporation of hierarchical refinement in Monte Carlo radiosity has been studied. The synthesis of hierarchical refinement radiosity and Monte Carlo radiosity yields new algorithms that inherit the benefits of their ancestors, while important drawbacks are annihilated. They are well suited for reliable and fast rendering of complex diffuse models on low-cost platforms.

The incorporation of hierarchical refinement in the stochastic Jacobi iterative method (§6.4) has been worked out in detail. The hierarchical extension of other stochastic relaxation algorithms can be done in the same manner. The similarity between the stochastic Jacobi iterative method and the collision shooting random walk (§7.4.4) suggests that the same technique can also be used for collision shooting random walks, when tracing paths in parallel, in breadth-first order. When tracing paths in depth-first order, a separate, but fortunately restricted, push-pull operation will be necessary at each path node. One area of future research is the extension of the random walk theory in chapter 7 to the solution of wavelet-preconditioned linear systems. Such a framework will allow to incorporate hierarchical refinement also in the other random walk algorithms.

A second important area for future research in this context concerns the development of cheaper discretisation-error based refinement criteria and strategies for constant as well as higher order radiosity approximations. The main practical limitation of per-ray refinement is that the refinement procedure is carried out much more often than in deterministic hierarchical radiosity. The need for cheap refinement procedures is therefore more urgent with per-ray refinement than in deterministic hierarchical radiosity. Experiments indicate that a speed-up of one order of magnitude is feasible with a sufficiently cheap refinement procedure.

The Monte Carlo method deals effectively with computational error and allows to solve the system of radiosity equations with much less storage than deterministic approaches. It does not improve the discretisation error in the solution however. In order to deal well with discretisation error near discontinuities such as shadow boundaries, discontinuity meshing still is needed. Willmott et al. [183] pointed out that the use of a multi-resolution representation of radiosity also introduces some errors, which can be visible near abutting patches. These errors are due to accumulation of discretisation error at different levels in the hierarchy, and still will be present with hierarchical Monte Carlo radiosity as well.

Even without a solution of these problems however, hierarchical Monte Carlo radiosity promises to be a fast, view-independent rendering technique yielding image quality that is sufficient for many applications.

15 Conclusion

This dissertation has focussed on two topics: discretisation error analysis and control (§15.1.1) and Monte Carlo methods for radiosity (§15.1.2). A list of original contributions in this dissertation is given in §15.2. Some directions for future research follow in §15.3.

15.1 Summary

15.1.1 Discretisation error analysis and control

The discretisation error is the solution of an integral equation of the same kind as the radiosity integral equation itself, with same kernel, but with the residual as the source term instead of self-emitted radiosity. An efficient algorithm for approximately computing the residual during form factor computation has been proposed. It has been used in order to estimate the discretisation error in a given radiosity solution and for error-driven hierarchical refinement. The combination of discretisation error estimation and error-driven hierarchical refinement promises to lead to automatic algorithms in which the resulting discretisation error will be below a beforehand specified threshold with minimal refinement. Other sources of error, in particular computational error due to visibility mis-estimation and due to inexact numerical integration, can be incorporated in the presented framework. Although in the current implementation, only very simple measures have been taken to prevent problems due to computational error, the preliminary results are promising.

15.1.2 Monte Carlo methods for radiosity

A systematic overview has been presented of how the Monte Carlo method can be applied in order to solve the radiosity problem. The Monte Carlo method can be used in order to reliably compute form factors, but also leads to algorithms in which the radiosity problem is solved without explicit form factor computation and storage. Such algorithms have two important advantages:

- low storage requirements since no form factors ever need to be stored in computer memory;
- the difficult problem of computing an accurate numerical value for the form factors is avoided. This is possible because the form factors can be interpreted as probabilities that can be sampled efficiently. Importance sampling leads to expressions in which the form factors cancel in numerator and denominator. In this way, a numerical value for the form factors is no longer required.

It turns out that the resulting Monte Carlo algorithms are also efficient. The computation cost of these algorithms is determined by the number of samples that need to be drawn in order to compute the result to given accuracy, times the cost of drawing each sample. The samples typically correspond to rays traced through the virtual scene.

The number of rays that needs to be traced depends on the variance of the Monte Carlo estimators that are used. In practice, the variance appears to be low enough so that complete radiosity solutions of fair quality can be obtained by tracing rays between only a small fraction of the possible pairs of patches in the scene. As usual in Monte Carlo, the effect of contributions that have not been sampled explicitly is accounted for by the fact that these contributions *could* have been sampled.

Several variance reduction techniques, and the use of low-discrepancy sample sequences, further reduce the number of samples needed in order to achieve a prescribed accuracy. View-importance sampling allows to render parts of complex scenes much more efficiently by increasing the sample density in important regions and decreasing it in unimportant regions. There is a significant cost however, in the determination of what parts of the scene are important for a given view and what parts are not. Other techniques yield a more moderate variance reduction, but their additional cost is very small.

Furthermore, the Monte Carlo method leads to reliable algorithms. As more samples are drawn, eventually the correct solution of the problem under consideration — in this case the solution of the system of radiosity equations — is obtained. The Monte Carlo method deals well with computational error. By doing the computations in stages, and merging the results of each stage appropriately, it is possible to obtain complete radiosity solutions, providing a good qualitative idea of the final result, in a very short time. Once an initial image has been obtained, variance is progressively reduced without large quantitative changes in the result. Good default values for the number of samples in each stage can be chosen automatically. The resulting algorithms require less user input than deterministic radiosity algorithms and are easier to implement.

Finally, Monte Carlo radiosity algorithms to compute higher order radiosity approximations have been proposed. A new, flexible, strategy to incorporate hierarchical refinement paves the way for radiosity algorithms in which both the computational and the discretisation error will be dealt with adequately.

The Monte Carlo method promises to offer alternatives of great value for deterministic radiosity system solution.

15.2 Original contributions

The research performed in the context of this dissertation has led to the following original results:

- The analysis of the discretisation error in chapter 3 is a first original contribution. The derived algorithms for a-posteriori discretisation error computation and error-driven refinement generalise previous results for constant and bilinear basis functions. The algorithm for discretisation error control is new. Chapter 3 is a revised version of [P6][1];

- Chapter 5 presents a new Monte Carlo algorithm for computing patch-to-patch form factors. based on weighted sampling. It is almost as reliable as with directional sampling, although uniform area sampling is used.

[1] Citations like [P6] refer to the list of publications at the end of this chapter.

An analysis of the variance of Monte Carlo form factor estimators leads to a heuristic for adaptively choosing the number of samples needed in order to ensure a given accuracy for the light transport between a pair of patches.

These results can be integrated easily in existing hierarchical refinement radiosity implementations;

- Chapters 6 to 8 present a systematic overview of Monte Carlo methods for solving linear systems such as the system of radiosity equations. New results in these chapters include:
  - the analysis of the computational cost of stochastic Jacobi algorithms;
  - the result that stochastic adaptations of more advanced relaxation algorithms such as over-relaxation and the Chebyshev method, will not be significantly superior to the simple stochastic Jacobi algorithms because they will need approximately the same number of samples (rays in radiosity);
  - the result that the discretisation error in a discrete Monte Carlo radiosity algorithm will hardly be higher than in a continuous Monte Carlo radiosity algorithm;
  - the theoretical and empirical comparison of collision shooting random walk radiosity versus stochastic Jacobi radiosity, indicating that in practice both will be approximately equally efficient for constant approximations.
- The importance-driven stochastic Jacobi iterative method in chapter 9 is a new result. The present text is a completely revised and significantly extended version of [P7].
- In chapter 10, the use of a constant control variate in the context of random walk radiosity estimators is proposed. An improved constant radiosity step for stochastic Jacobi radiosity, based on variance minimisation, is presented;
- The combination of gathering and shooting random walk radiosity estimators in chapter 11 is another new result. The present text is an extended version of [P13]. The combination of gathering and shooting in stochastic Jacobi iterations is a small, but quite beneficial, new result;
- Chapter 12 presents new empirical results concerning the efficiency of low-discrepancy sampling in Monte Carlo radiosity algorithms. The present text is a revised version of [P12]. The main results are:
  - Low-discrepancy sampling appears to be more effective in discrete than in continuous Monte Carlo radiosity algorithms;
  - No low-discrepancy sample sequence is systematically superior to the other sequences.
- The extension of discrete Monte Carlo radiosity algorithms to higher order radiosity approximations in chapter 13 is a new result. Main conclusions:
  - Stochastic Jacobi iterations are significantly superior to discrete random walk methods for higher order approximations. They are not worse than continuous random walk methods for higher order approximations;

  - The amount of work required in order to compute a result of prescribed (computational) accuracy is proportional to the number of basis functions.

- Finally, chapter 14 presents a new, flexible, strategy in order to incorporate hierarchical refinement in Monte Carlo radiosity algorithms. Experiments with an inexpensive refinement criterion indicate that a speed-up of one order of magnitude can be obtained compared to deterministic hierarchical radiosity algorithms, while the storage requirements can be reduced by a factor 5. The present text includes some practical information that could not be published in [P10] because of space constraints.

15.3 Directions for future research

15.3.1 Better hierarchical refinement criteria

The main problems of previous radiosity methods concern meshing and form factor computation and storage. The study in this dissertation indicates that the Monte Carlo method leads to efficient radiosity algorithms in which no form factors need to be computed or stored explicitly. The framework in chapter 3 will be useful in order to develop effective automatic and adaptive meshing algorithms for higher order radiosity approximations. The current algorithms in chapter 3 however assume patch-to-patch form factor computation and have been worked out only with a very restricted form of refinement. Moreover, they are too expensive in order to be applied with success in a per-ray fashion as proposed in chapter 14.

The main area of future research therefore concerns the development of very flexible but inexpensive criteria and strategies for hierarchical refinement for constant as well as higher-order radiosity approximations. In particular, future refinement strategies should incorporate selective discontinuity meshing at a low storage and computational cost. Since typically only a very small fraction of the discontinuity curves in a scene actually causes noticeable artifacts, a possible area for improvement of discontinuity meshing algorithms may consist in resolving only those discontinuity lines that lead to visible artifacts. Current discontinuity meshing algorithms appear to be little selective, and require large amounts of computing time and/or computer storage. Also the reliable determination of which discontinuities will actually lead to visible artifacts and which ones will not, unfortunately is an unresolved problem. In the context of higher order approximations, efficient prediction of what approximation order will suffice for a receiver element, combined with p-refinement as explained in §3.3.2, will prevent unnecessary computation of higher order approximation terms. In order to successfully apply them in a per-ray fashion, the resulting refinement criteria and strategies should ideally be inexpensive compared to the cost of tracing a ray through the scene, without requiring large amounts of storage.

Experiments in this dissertation indicate that even without a solution to these problems, the presented algorithms will result in image quality that will be sufficient for many applications. In order to obtain images of very high quality without a per-pixel Monte Carlo final gather pass however, a solution to these problems is required. Preliminary work into this direction includes [104, 13, 73, 10].

15.3.2 More Monte Carlo

Some topics for future research concerning the Monte Carlo solution of linear systems are:

- The exotic random walk estimators in §7.2.6 need to be examined further. Preliminary experiments indicate that these estimators do not compete as presented with the collision estimator. Unlike the collision estimator however, they contribute scores to a fixed number of nodes, so that they may lead to perfect importance-sampling random walk estimators;
- In chapter 8, a number of more obscure Monte Carlo techniques for solving linear systems have been enumerated. They need to be studied in more detail;
- The development of efficient and reliable algorithms in order to sample ray directions according to incoming radiosity or importance, without an explicit representation of incoming radiosity, will be beneficial in the context of sequential adaptive importance sampling (§9.2.1), and view-importance sampling (§9.3.1);
- More reliable heuristics for combining gathering and shooting estimators (chapter 11) can be developed. Also the combination of different shooting (or gathering) random walk estimators can be a topic of further research;
- In this dissertation, only a limited number of variance reduction techniques has been examined. In particular, more advanced applications of stratified sampling in order to aim more rays to "difficult" regions of a scene [80] and weighted sampling are possible.
- Low-discrepancy sampling in random walk problems is a topic of active ongoing research [18, 117]. Application of emerging research results in this area to Monte Carlo radiosity is another interesting area for future research;
- Stochastic relaxation algorithms appear not to have received any attention in general Monte Carlo literature. Their application to other problems than radiosity needs to be investigated;
- The development of random walk methods for the solution of wavelet preconditioned systems of linear equations may lead to new efficient hierarchical Monte Carlo algorithms for solving problems like the radiosity problem.

15.3.3 Dynamic environments with general surface characteristics

The problem of computing the illumination in "glossy" environments can be described by a system of linear equations that is similar to the radiosity system. It is expected that the Monte Carlo techniques for linear systems that are described in this dissertation, can be applied in a quite straightforward manner to those systems for glossy illumination.

In order to take advantage of time-coherence in dynamic environments, algorithms similar to the view-importance driven stochastic Jacobi algorithms can be developed. The notion of importance will then reflect to what extent the illumination at a given spot changes as the result of a change of scene geometry or material properties.

Publications

[P1] Ph. Bekaert and Y. D. Willems. Ray-tracing 3d linear graftals. In *Winter School on Computer Graphics and CAD Systems '94, Plzen, Czech Republic*, pages 46–54, February 1994.

[P2] Ph. Bekaert and Y. D. Willems. A progressive importance-driven rendering algorithm. In *10th Spring School on Computer Graphics and its Applications, Comenius University, Bratislava, Slovakia*, pages 58–67, June 1994.

[P3] Ph. Bekaert, G. Uytterhoeven, and Y. D. Willems. An experiment with wavelet image coding. In *11th Spring School on Computer Graphics, Comenius University, Bratislava, Slovakia*, pages DM1–DM6, June 1995.

[P4] Ph. Bekaert and Y. D. Willems. Importance-driven progressive refinement radiosity. In *Rendering Techniques '95 (Proceedings of the 6th Eurographics Workshop on Rendering, Dublin, Ireland)*, pages 316–325. Springer Computer Science, June 1995.

[P5] Ph. Bekaert and Y. D. Willems. HIRAD: A hierarchical higher order radiosity implementation. In *12th Spring Conference on Computer Graphics, Comenius University, Bratislava, Slovakia*, pages 213–218, June 1996.

[P6] Ph. Bekaert and Y. D. Willems. Error control for radiosity. In *Rendering Techniques '96 (Proceedings of the 7th Eurographics Workshop on Rendering, Porto, Portugal)*, pages 153–164. Springer Computer Science, June 1996.

[P7] A. Neumann, L. Neumann, Ph. Bekaert, Y. D. Willems, and W. Purgathofer. Importance-driven stochastic ray radiosity. In *Rendering Techniques '96 (Proceedings of the 7th Eurographics Workshop on Rendering, Porto, Portugal)*, pages 111–122. Springer Computer Science, June 1996.

[P8] Ph. Dutré, Ph. Bekaert, F. Suykens, and Y. D. Willems. Bidirectional radiosity. In *Rendering Techniques '97 (Proceedings of the 8th Eurographics Workshop on Rendering, St. Etienne, France)*, pages 205–216. Springer Computer Science, June 1997.

[P9] L. Neumann, A. Neumann, and Ph. Bekaert. Radiosity with well distributed ray sets. *Computer Graphics Forum*, 16(3):C261–C270, 1997.

[P10] Ph. Bekaert, L. Neumann, A. Neumann, M. Sbert, and Y. D. Willems. Hierarchical Monte Carlo radiosity. In *Rendering Techniques '98 (Proceedings of the 9th. Eurographics Workshop on Rendering, Vienna, Austria)*, pages 259–268. Springer Computer Science, June 1998.

[P11] Ph. Bekaert, Ph. Dutré, and Y. D. Willems. Final radiosity gather step using a Monte Carlo technique with optimal importance sampling. Technical Report CW275, Department of Computer Science, K. U. Leuven, November 1998. 14 pages.

[P12] Ph. Bekaert, R. Cools, and Y. D. Willems. An empirical comparison of Monte Carlo radiosity algorithms. In *7th International Conference in Central Europe on Computer Graphics, Visualization and Digital Interactive Media, Plzen, Czech Republic*, pages 9–16, February 1999.

[P13] M. Sbert, A. Brusi, and Ph. Bekaert. Gathering for free in random walk radiosity. In *Rendering Techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering, Granada, Spain)*, pages 97–102. Springer Computer Science, June 1999.

[P14] M. Feixas, E. del Acebo, Ph. Bekaert, and M. Sbert. Information theory tools for scene discretization. In *Rendering Techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering, Granada, Spain)*, pages 103–114. Springer Computer Science, June 1999.

[P15] M. Feixas, E. del Acebo, Ph. Bekaert, and M. Sbert. An information theory framework for the analysis of scene complexity. *Computer Graphics Forum*, 18(3):95–106, 1999.

Bibliography

[1] G. E. Albert. A general theory of stochastic estimates of the Neumann series for the solution of certain Fredholm integral equations and related series. In H. A. Meyer, editor, *Symposium on Monte Carlo methods, Florida, March, 16 and 17, 1954*, pages 37 – 46. J. Wiley and sons, 1956.

[2] J. Arvo. Stratified sampling of spherical triangles. In *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, pages 437–438, August 1995.

[3] J. Arvo, K. Torrance, and B. Smits. A Framework for the Analysis of Error in Global Illumination Algorithms. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 75–84, 1994.

[4] L. Aupperle and P. Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 155–162, 1993.

[5] L. Aupperle and P. Hanrahan. Importance and discrete three point transport. In *Fourth Eurographics Workshop on Rendering*, pages 85–94, June 1993. held in Paris, France, 14–16 June 1993.

[6] G. Baranoski, R. Bramley, and P. Shirley. Fast radiosity solutions for high average reflectance environments. In *Eurographics Rendering Workshop 1995*, June 1995.

[7] G. Baranoski, R. Bramley, and P. Shirley. Iterative methods for fast radiosity solutions. Technical Report TR429, Department of Computer Science, Indiana University, Bloomington, U.S.A., April 1995.

[8] D. R. Baum, S. Mann, K. P. Smith, and J. M. Winget. Making radiosity usable: Automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 51–60, July 1991.

[9] D. R. Baum, H. E. Rushmeier, and J. M. Winget. Improving radiosity solutions through the use of analytically determined form-factors. In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 325–334, July 1989.

[10] P. Bekaert. Opsplitsen volgens discontinuiteitslijnen in de radiositeitsmethode (discontinuity meshing in radiosity), 1997. Masters Thesis, Department of Computer Science, Katholieke Universiteit Leuven.

[11] D. Blythe and T. McReynolds. Advanced graphics programming techniques using OpenGL, August 1999. ACM SIGGRAPH'99 Course 29.

[12] M. A. Bolin and G. W. Meyer. An error metric for Monte Carlo ray tracing. In *Eurographics Rendering Workshop 1997*, pages 57–68, June 1997.

[13] K. Bouatouch and S. N. Pattanaik. Discontinuity meshing and hierarchical multiwavelet radiosity. In *Graphics Interface '95*, pages 109–115, May 1995.

[14] K. Bouatouch, S. N. Pattanaik, and E. Zeghers. Computation of higher order illumination with a non-deterministic approach. *Computer Graphics Forum*, 15(3):327–338, August 1996.

[15] P. Bratley, B. Fox, and H. Niederreiter. Algorithm 738: Programs to generate Niederreiter's low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 20(4):494–495, December 1994.

[16] P. Bratley, B. L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy sequences. *ACM Transactions on Modelling and Computer Simulation*, 2(3):195–213, July 1992.

[17] N. P. Buslenko, D. I. Golenko, Yu. A. Shreider, I. M. Sobol, and V. G. Sragovich. *The Monte Carlo Method – The Method of Statistical Trials*. Pergamon Press, 1962. Translated from the Russian by G. J. Tee, 1966.

[18] R. E. Caflish. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, pages 1 – 49, 1998.

[19] A. T. Campbell, III. *Modeling Global Diffuse Illumination for Image Synthesis*. PhD thesis, Dept. of Computer Sciences, Univ. of Texas at Austin, December 1991.

[20] A. T. Campbell, III and D. S. Fussell. Adaptive mesh generation for global diffuse illumination. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 155–164, August 1990.

[21] F. Castro, R. Matinez, and M. Sbert. Quasi-Monte Carlo and extended first-shot improvement to the multi-path method. In *Proc. Spring Conference on Computer Graphics '98*, pages 91–102, Bratislava, Slovakia, April 1998. Comenius University.

[22] S. E. Chen. Incremental Radiosity: An Extension of Progressive Radiosity to an Interactive Image Synthesis System. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24, pages 135–144, August 1990.

[23] S. E. Chen, H. E. Rushmeier, G. Miller, and D. Turner. A progressive multi-pass method for global illumination. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 165–174, July 1991.

[24] P. H. Christensen. *Hierarchical Techniques for Glossy Global Illumination*. PhD thesis, University of Washington, 1995.

[25] P. H. Christensen, E. J. Stollnitz, D. H. Salesin, and T. D. DeRose. Wavelet Radiance. In *Fifth Eurographics Workshop on Rendering*, pages 287–302, Darmstadt, Germany, June 1994.

[26] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, August 1988.

[27] M. F. Cohen and D. P. Greenberg. The hemi-cube: A radiosity solution for complex environments. *Computer Graphics (SIGGRAPH '85 Proceedings)*, 19(3):31–40, July 1985.

[28] M. F. Cohen, D. P. Greenberg, D. S. Immel, and P. J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(3):26–35, March 1986.

[29] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press Professional, Boston, MA, 1993.

[30] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics*, 18(3):137–145, July 1984.

[31] R. Cools. Monomial cubature rules since "Stroud": a compilation – part 2. *Journal of Computational and Applied Mathematics*, 112(1-2):21–27, 1999.

[32] R. Cools and Ph. Rabinowitz. Monomial cubature rules since "Stroud": a compilation. *Journal of Computational and Applied Mathematics*, 48:309–326, 1993.

[33] R. R. Coveyou, V. R. Cain, and K. J. Yost. Adjoint and importance in Monte Carlo application. *Nuclear Science and Engineering*, 27:219 – 234, 1967.

[34] J. H. Curtiss. A theoretical comparison of the efficiencies of two classical methods and a Monte Carlo method for computing one component of the solution of a set of linear algebraic equations. In H. A. Meyer, editor, *Symposium on Monte Carlo methods, Florida, March, 16 and 17, 1954*, pages 191 – 233. J. Wiley and sons, 1956.

[35] L. M. Delves and J. L. Mohamed. *Computational methods for integral equations.* Cambridge University Press, 1985.

[36] I. T. Dimov and V. N. Alexandrov. A new highly convergent Monte Carlo method for matrix computations. *Mathematics and Computers in Simulation*, 47:165 – 181, 1998.

[37] I. T. Dimov and A. N. Karaivanova. A fast Monte Carlo method for matrix computations. In S. D. Margenov and P. S. Vassilevski, editors, *IMACS Series in Computational and Applied Mathematics Volume 3 (Iterative Methods in Linear Algebra II)*, pages 204 – 215. 1996.

[38] G. Drettakis and E. Fiume. A Fast Shadow Algorithm for Area Light Sources Using Backprojection. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 223–230, 1994.

[39] G. Drettakis and F. X. Sillion. Interactive update of global illumination using a line-space hierarchy. In *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, volume 31, pages 57–64, 1997.

[40] F. Durand, G. Drettakis, and C. Puech. The visibility skeleton: A powerful and efficient multi-purpose global visibility tool. In *SIGGRAPH 97 Conference Proceedings*, pages 89–100, August 1997.

[41] Ph. Dutré. *Mathematical Frameworks and Monte Carlo Algorithms for Global Illumination in Computer Graphics.* PhD thesis, Katholieke Universiteit Leuven, September 1996.

[42] Ph. Dutré, Ph. Bekaert, F. Suykens, and Y. D. Willems. Bidirectional radiosity. In *Eurographics Rendering Workshop 1997*, pages 205–216, June 1997.

[43] Ph. Dutré and Y. D. Willems. Importance-driven Monte Carlo light tracing. In *Fifth Eurographics Workshop on Rendering*, pages 185–194, Darmstadt, Germany, June 1994.

[44] Ph. Dutré and Y. D. Willems. Potential-driven Monte Carlo particle tracing for diffuse environments with adaptive probability density functions. In *Eurographics Rendering Workshop 1995*, June 1995.

[45] S. M. Ermakow. *Die Monte-Carlo-Methode und verwandte Fragen.* V.E.B. Deutscher Verlag der Wissenschaften, Berlin, 1975.

[46] C. J. Everett and E. D. Cashwell. A third Monte Carlo sampler. Technical Report LA-9721-MS, Los Alamos National Lab, 1983.

[47] M. Feda. A Monte Carlo approach for Galerkin radiosity. *The Visual Computer*, 12(8):390–405, 1996.

[48] M. Feda and W. Purgathofer. Accelerating radiosity by overshooting. *Third Eurographics Workshop on Rendering*, pages 21–32, May 1992.

[49] M. Feda and W. Purgathofer. Progressive ray refinement for Monte Carlo radiosity. In *Fourth Eurographics Workshop on Rendering*, pages 15–26, June 1993. held in Paris, France, 14–16 June 1993.

[50] M. Feixas, E. del Acebo, Ph. Bekaert, and M. Sbert. An information theory framework for the analysis of scene complexity. *Computer Graphics Forum*, 18(3):C95 – C106, September 1999. Proceedings of Eurgraphics'99, Milan, Italy.

[51] G. E. Forsythe and R. A. Leibler. Matrix inversion by a Monte Carlo method. *Math. Tabl. Aids. Comput.*, 4:127 – 129, 1950.

[52] D. W. George, F. X. Sillion, and D. P. Greenberg. Radiosity Redistribution for Dynamic Environments. *IEEE Computer Graphics and Applications*, 10(4):26–34, July 1990.

[53] R. Gershbein. A study of integration methods for couplings of galerkin radiosity systems. In *6 th Eurographics Rendering Workshop, Dublin, Ireland*, June 1995.

[54] S. Gibson and R. J. Hubbold. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5):297–310, 1996.

[55] A. S. Glassner, editor. *An Introduction to Ray Tracing*. Academic Press, 1989.

[56] A. S. Glassner. *Principles of Digital Image Synthesis.* Morgan Kaufmann, San Francisco, CA, 1995.

[57] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2nd edition, 1989.

[58] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. In *SIGGRAPH '84 Conference Proceedings (Minneapolis, MN, July 23-27, 1984)*, pages 213–222, July 1984.

[59] S. Gortler, M. F. Cohen, and Ph. Slusallek. Radiosity and relaxation methods. *IEEE Computer Graphics and Applications*, 14(6):48–58, November 1994.

[60] S. Gortler, P. Schröder, M. Cohen, and P. Hanrahan. Wavelet radiosity. In *SIGGRAPH '93 Proceedings*, pages 221–230, 1993.

[61] E. Haines and J. Wallace. Shaft culling for efficient ray-traced radiosity. In *Eurographics Workshop on Rendering*, 1991.

[62] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

[63] J. H. Halton. Sequential Monte Carlo. *Proc. Cambridge Phil. Soc.*, 58:57 – 78, 1962.

[64] J. H. Halton. A restrospective and prospective survey of the Monte Carlo method. *SIAM Review*, 12(1):1 – 63, January 1970.

[65] J. H. Halton. Sequential Monte Carlo techniques for the solution of linear systems. *Journal of Scientific Computing*, 9(2):213 – 257, 1994.

[66] J. M. Hammersley and D. C. Handscomb. *The Monte Carlo method*. Cambridge Univ. Press, 1964.

[67] D. C. Handscomb. Remarks on a Monte Carlo integration method. *Numerische Mathematik*, 6:261 – 268, 1964.

[68] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.

[69] S. Hardt and S. Teller. High-fidelity radiosity rendering at interactive rates. In *Eurographics Rendering Workshop 1996*, pages 71–80. Springer Wien, June 1996.

[70] P. S. Heckbert. Simulating global illumination using adaptive meshing. Technical Report UCB/CSD 91/636, Computer Science Division (EECS), University of California, Berkeley, California, USA, June 1991.

[71] P. S. Heckbert. Discontinuity meshing for radiosity. *Third Eurographics Workshop on Rendering*, pages 203–226, May 1992.

[72] P. S. Heckbert and J. Winget. Finite element methods for global illumination. Technical Report UCB/CSD 91/643, Computer Science Division (EECS), University of California, Berkeley, California, USA, July 1991.

[73] D. Hedley, A. Worrall, and D. Paddon. Selective culling of discontinuity lines. In *Rendering Techniques '97 (Proceedings of the Eight Eurographics Workshop on Rendering)*, pages 69–80, June 1997.

[74] E. Hlawka. Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. *Annali di matematica pura ed applicada*, 54:325–333, 1961.

[75] N. Holzschuch. *Le contrôle de l'erreur dans la méthode de radiosité hiérarchique*. Ph.D. thesis, Université Joseph Fourier — Grenoble I, France, March 1996.

[76] N. Holzschuch and F. Sillion. Accurate computation of the radiosity gradient for constant and linear emitters. In *6 th Eurographics Rendering Workshop, Dublin, Ireland*, June 1995.

[77] D. S. Immel, M. F. Cohen, and D. P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics*, 20(4):133–142, August 1986.

[78] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Review*, 36(3):377 – 412, 1994.

[79] H. W. Jensen. Global illumination using photon maps. In *Eurographics Rendering Workshop 1996*, pages 21–30. Eurographics, June 1996.

[80] V. Jolivet, D. Plemenos, and M. Sbert. A pyramidal hemisphere subdivision method for Monte Carlo radiosity. In *Eurographics '99 Conference Proceedings (Short Papers)*, pages 63–69, September 1999.

[81] H. Kahn. Use of different Monte Carlo sampling techniques. In H. A. Meyer, editor, *Symposium on Monte Carlo methods, Florida, March, 16 and 17, 1954*, pages 146 – 190. J. Wiley and sons, 1956.

[82] H. Kahn and T. E. Harris. Estimation of particle transmission by random sampling. In *Monte Carlo method, National Bureau of Standards, Appl. Math. Series*, volume 12, pages 27 – 30, 1949.

[83] J. T. Kajiya. The rendering equation. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):143–150, August 1986.

[84] M. H. Kalos. Importance sampling in Monte Carlo shielding calculations. *Nucl. Sci. Eng.*, 16:227 – 234, 1963.

[85] M. H. Kalos and P. Whitlock. *The Monte Carlo method.* J. Wiley and sons, 1986.

[86] J. K. Kawai, J. S. Painter, and M. F. Cohen. Radiooptimization - Goal Based Rendering. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 147–154, 1993.

[87] A. Keller. The Fast Calculation of Form Factors Using Low Discrepancy Sequences. In *Proceedings of the Spring Conference on Computer Graphics (SCCG '96)*, pages 195–204, Bratislava, Slovakia, June 1996. Comenius University Press.

[88] A. Keller. Quasi-Monte Carlo radiosity. In *Eurographics Rendering Workshop 1996*, pages 101–110, June 1996.

[89] A. Keller. *Quasi-Monte Carlo methods for photorealistic image synthesis*. PhD thesis, Universität Kaiserslautern, Germany, June 1997.

[90] A. Kok. Grouping of patches in progressive radiosity. In *Fourth Eurographics Workshop on Rendering*, pages 221–232. Eurographics, June 1993. held in Paris, France, 14–16 June 1993.

[91] A. Kok. *Ray Tracing and Radiosity Algorithms for Photorealistic Images Synthesis*. PhD thesis, Technische Universiteit Delft, The Netherlands, 1994.

[92] J. F. Koksma. Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1. *Mathematica; Tijdschrift voor studeerenden voor de acten wiskunde M.O. en voor studeerenden aan universiteiten. Afdeeling B*, 11:7–11, 1942/43.

[93] C. Kollman, K. Baggerley, D. Cox, and R. Picard. Adaptive importance sampling on discrete markov chains. Technical Report LA-UR-96-3998, Los Alamos National Lab, November 1996.

[94] R. Kress. *Linear Integral Equations.* Springer Verlag, 1989.

[95] E. P. Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Katholieke Universiteit Leuven, February 1996.

[96] E. P. Lafortune and Y. D. Willems. Bi-directional Path Tracing. In H. P. Santo, editor, *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal, December 1993.

[97] E. P. Lafortune and Y. D. Willems. The ambient term as a variance reducing technique for Monte Carlo ray tracing. In *Fifth Eurographics Workshop on Rendering*, pages 163–171, Darmstadt, Germany, June 1994.

[98] E. P. Lafortune and Y. D. Willems. A theoretical framework for physically based rendering. *Computer Graphics Forum*, 13(2):97–107, June 1994.

[99] E. P. Lafortune and Y. D. Willems. Using the Modified Phong BRDF for Physically Based Rendering. Technical Report CW197, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, November 1994.

[100] Y. Lai and J. Spanier. Adaptive importance sampling algorithms for transport problems. In H. Niederreiter and J. Spanier, editors, *Lecture Comments in Computational Science and Engineering (Proceedings of Monte Carlo and Quasi-Monte Carlo Methods, Claremont, California, USA, June 1998)*. Springer-Verlag, 1999.

[101] E. Languenou, K. Bouatouch, and P. Tellier. An adaptive discretization method for radiosity. *Computer Graphics Forum*, 11(3):C205–C216, 1992.

[102] D. Lischinski, B. Smits, and D. P. Greenberg. Bounds and error estimates for radiosity. In *Proceedings of SIGGRAPH '94*, pages 67–74, July 1994.

[103] D. Lischinski, F. Tampieri, and D. P. Greenberg. Discontinuity meshing for accurate radiosity. *IEEE Computer Graphics and Applications*, 12(6):25–39, November 1992.

[104] D. Lischinski, F. Tampieri, and D. P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *SIGGRAPH '93 Proceedings*, pages 199–208, 1993.

[105] K. Matkovic, L. Neumann, and W. Purgathofer. A survey of tone mapping techniques. In *13th Spring Conference on Computer Graphics*, pages 163–170. Comenius University, Bratislava, Slovakia, June 1997.

[106] G. A. Mikhailov. On a class on Monte Carlo estimators. *Theory of Probability and its Applications*, pages 137 – 138, 1970.

[107] G. A. Mikhailov. *Optimization of Weighted Monte Carlo Methods.* Springer Series in Computational Physics, 1992.

[108] A. Neumann, L. Neumann, Ph. Bekaert, Y. D. Willems, and W. Purgathofer. Importance-driven stochastic ray radiosity. In *Eurographics Rendering Workshop 1996*, pages 111–122, June 1996.

[109] L. Neumann. New Efficient Algorithms with Positive Definite Radiosity Matrix. In *Fifth Eurographics Workshop on Rendering*, pages 219–237, Darmstadt, Germany, June 1994.

[110] L. Neumann. Monte Carlo radiosity. *Computing*, 55(1):23–42, 1995.

[111] L. Neumann, M. Feda, M. Kopp, and W. Purgathofer. A New Stochastic Radiosity Method for Highly Complex Scenes. In *Fifth Eurographics Workshop on Rendering*, pages 195–206, Darmstadt, Germany, June 1994.

[112] L. Neumann, A. Neumann, and Ph. Bekaert. Radiosity with well distributed ray sets. *Computer Graphics Forum*, 16(3), 1997.

[113] L. Neumann, W. Purgathofer, R. Tobler, A. Neumann, P. Elias, M. Feda, and X. Pueyo. The stochastic ray method for radiosity. In P. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, July 1995.

[114] L. Neumann, R. F. Tobler, and P. Elias. The Constant Radiosity Step. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 336–344. Springer-Verlag, 1995.

[115] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *CBMS-NSF regional conference series in Appl. Math.* SIAM, Philadelphia, 1992.

[116] T. Nishita and E. Nakamae. Continuous tone representation of 3-D objects taking account of shadows and interreflection. *Computer Graphics (SIGGRAPH '85 Proceedings)*, 19(3):23–30, July 1985.

[117] A. B. Owen. Monte Carlo extension of quasi-Monte Carlo. In D. J. Medieiros, E. F. Watson, M. Manivannan, and J. Carson, editors, *Winter Simulation Conference Proceedings*, pages 571–577, 1998.

[118] A. B. Owen and Y. Zhou. Safe and effective importance sampling. Technical report, Stanford University, Department of Statistics, 1999.

[119] S. N. Pattanaik and K. Bouatouch. Linear radiosity with error estimation. In *6 th Eurographics Rendering Workshop, Dublin, Ireland*, June 1995.

[120] S. N. Pattanaik and S. P. Mudur. Computation of global illumination by Monte Carlo simulation of the particle model of light. *Third Eurographics Workshop on Rendering*, pages 71–83, May 1992.

[121] S. N. Pattanaik and S. P. Mudur. The potential equation and importance in illumination computations. *Computer Graphics Forum*, 12(2):131–136, 1993.

[122] S. N. Pattanaik and S. P. Mudur. Adjoint equations and random walks for illumination computation. *ACM Transactions on Graphics*, 14(1):77–102, January 1995.

[123] M. Pellegrini. Monte Carlo approximation of form factors with error bounded a priori. In *Proc. of the 11th. annual symposium on Computational Geometry*, pages 287 – 296. ACM Press, 1995.

[124] M. J. D. Powell and J. Swann. Weighted uniform sampling – a Monte Carlo technique for reducing variance. *J. Inst. Maths. Applics.*, 2:228 – 236, 1966.

[125] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in* FORTRAN. Cambridge University Press, 2nd edition edition, 1992.

[126] M. Ramasubramanian, S. N. Pattanaik, and Donald P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *Computer Graphics (ACM SIGGRAPH'99 proceedings)*, August 1999.

[127] R. Y. Rubinstein. *Simulation and the Monte Carlo method*. J. Wiley and sons, 1981.

[128] H. Rushmeier, C. Patterson, and A. Veerasamy. Geometric simplification for indirect illumination calculations. In *Proceedings of Graphics Interface '93*, pages 227–236, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.

[129] L. Santaló. *Integral Geometry and Geometric Probability*. Addison-Welsey, Reading, Mass, 1976.

[130] M. Sbert. An integral geometry based method for fast form-factor computation. *Computer Graphics Forum*, 12(3):C409–C420, 1993.

[131] M. Sbert. *The use of global random directions to compute radiosity — Global Monte Carlo techniques*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, November 1996.

[132] M. Sbert. Error and complexity of random walk Monte Carlo radiosity. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):23–38, March 1997.

[133] M. Sbert. Optimal source selection in shooting random walk Monte Carlo radiosity. *Computer Graphics Forum*, 16(3):301–308, August 1997.

[134] M. Sbert, A. Brusi, and Ph. Bekaert. Gathering for free in random walk radiosity. In *Rendering Techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering, Granada, Spain)*, pages 97–102. Springer Computer Science, June 1999.

[135] M. Sbert, A. Brusi, R. Tobler, and W. Purgathofer. Random walk radiosity with generalized transition probabilities. Technical Report IIiA-98-07-RR, Institut d'Informàtica i Aplicacions, Universitat de Girona, January 1998.

[136] M. Sbert, F. Perez, and X. Pueyo. Global Monte Carlo: A Progressive Solution. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 231–239, 1995.

[137] M. Sbert, X. Pueyo, L. Neumann, and W. Purgathofer. Global multipath Monte Carlo algorithms for radiosity. *The Visual Computer*, 12(2):47–61, 1996.

[138] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, and D. Greenberg. Painting with light. In *Computer Graphics (ACM SIGGRAPH'93 proceedings)*, pages 143–146, August 1993.

[139] P. Schröder. Numerical Integration for Radiosity in the Presence of Singularities. In *Fourth Eurographics Workshop on Rendering*, pages 177–184, June 1993.

[140] P. Schröder. *Wavelet algorithms for illumination computations*. PhD thesis, Princeton University, November 1994.

[141] P. Schröder and P. Hanrahan. On the form factor between two polygons. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 163–164, 1993.

[142] P. Shirley. A ray tracing method for illumination calculation in diffuse–specular scenes. In *Graphics Interface '90*, pages 205–212, May 1990.

[143] P. Shirley. Radiosity via ray tracing. In J. Arvo, editor, *Graphics Gems II*, pages 306–310. Academic Press, San Diego, 1991.

[144] P. Shirley. Time complexity of Monte Carlo radiosity. In *Eurographics '91*, pages 459–465. North-Holland, September 1991.

[145] P. Shirley, B. Wade, Ph. M. Hubbard, D. Zareski, B. Walter, and Donald P. Greenberg. Global Illumination via Density Estimation. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 219–230, 1995.

[146] Yu. A Shreider. The solution of linear algebraic equations by a Monte Carlo method. In Yu. Ya. Bazilevskii, editor, *The Theory of Mathematical Machines*, pages 198–204. Pergamon Press Ltd., 1958. Translated from the Russian by C. A. R. Hoare, 1963.

[147] F. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, September 1995.

[148] F. Sillion, J. Arvo, S. Westin, and D. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):187–196, July 1991.

[149] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 335–344, July 1989.

[150] F. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, 1994.

[151] B. Smits, J. Arvo, and D. Greenberg. A clustering algorithm for radiosity in complex environments. In *SIGGRAPH '94 Proceedings*, pages 435–442, July 1994.

[152] B. Smits, J. Arvo, and D. Salesin. An importance-driven radiosity algorithm. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 273–282, July 1992.

[153] J. Spanier. A new family of estimators for random walk problems. *Journal of the Institute of Mathematics and its Applications*, 23:1 – 31, 1979.

[154] J. Spanier. Geometrically convergent learning algorithms for global solutions of transport problems. In H. Niederreiter and J. Spanier, editors, *Lecture Comments in Computational Science and Engineering (Proceedings of Monte Carlo and Quasi-Monte Carlo Methods, Claremont, California, USA, June 1998)*. Springer-Verlag, 1999.

[155] J. Spanier. Monte Carlo methods for flux expansion solution of transport problems. *Nuclear Science and Engineering*, to appear.

[156] J. Spanier and E. M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, 1969.

[157] J. Spanier and L. Li. Quasi-Monte Carlo methods for integral equations. In H. Niederreiter et al., editor, *Proceedings of Monte Carlo and quasi-Monte Carlo methods in scientific computing, Salzburg, Austria, July 1996*, number 127 in Lecture Notes in Statistics, pages 398–414. Springer-Verlag, 1997.

[158] J. Spanier and E. H. Maize. Quasi-random methods for estimating integrals using relatively small samples. *SIAM Review*, 36(1):18–44, March 1994.

[159] M. Stamminger, H. Schirmacher, and Ph. Slusallek. Getting rid of links in hierarchical radiosity. In *Computer Graphics Forum (Proceedings EUROGRAPHICS '98)*, 1998.

[160] M. Stamminger, P. Slusallek, and H-P Seidel. Bounded radiosity — illumination on general surfaces and clusters. *Computer Graphics Forum*, 16(3), 1997.

[161] M. Stamminger, Ph. Slusallek, and H.-P. Seidel. Three point clustering for radiance computations. In *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)*, pages 211–222, 1998.

[162] A. Stewart and S. Ghali. Fast computation of shadow boundaries using spatial coherence and backprojections. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 231–238, July 1994.

[163] W. Stürzlinger. Adaptive Mesh Refinement with Discontinuities for the Radiosity Method. In *Fifth Eurographics Workshop on Rendering*, pages 239–248, June 1994.

[164] F. Suykens and Y. D. Willems. Weighted multipass methods for global illumination. *Computer Graphics Forum*, 18(3):C209–C220, 1999. Proceedings of Eurographics '99, Milan, Italy.

[165] L. Szirmay-Kalos, T. Foris, L. Neumann, and C. Balasz. An analysis of quasi-Monte Carlo integration applied to the transillumination radiosity method. *Computer Graphics Forum (Eurographics '97 Proceedings)*, 16(3), 1997. C271–C281.

[166] L. Szirmay-Kalos, T. Foris, and W. Purgathofer. Quasi-Monte Carlo global light tracing with infinite number of rays. In *WSCG '98 (Sixth European Conference in Central Europe on Computer Graphics and Visualization)*, pages 386–393, Plzen, Czech Republic, 1998. University of West Bohemia.

[167] L. Szirmay-Kalos and W. Purgathofer. Global ray-bundle tracing with hardware acceleration. In *Ninth Eurographics Workshop on Rendering*, Vienna, Austria, June 1998.

[168] L. Szirmay-Kalos and W. Purgathofer. Analysis of the quasi-Monte Carlo integration of the rendering equation. In *WSCG '99 (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media)*, pages 281–288, Plzen-Bory, Czech Republic, February 1999. University of West Bohemia.

[169] S. Teller and P. Hanrahan. Global visibility algorithms for illumination computations. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 239–246, 1993.

[170] S. Tezuka. *Uniform Random Numbers: Theory and Practice*. Kluwer Academic Publishers, 1995.

[171] R. Tobler, A. Wilkie, M. Feda, and W. Purgathofer. A hierarchical subdivision algorithm for stochastic radiosity methods. In *Eurographics Rendering Workshop 1997*, pages 193–204, June 1997.

[172] R. Troutman and N. L. Max. Radiosity algorithms using higher order finite element methods. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 209–212, 1993.

[173] J. Tumblin and H. E. Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48, November 1993.

[174] E. Veach. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford university, Department of Computer Science, December 1997.

[175] E. Veach and L. J. Guibas. Bidirectional Estimators for Light Transport. In *Fifth Eurographics Workshop on Rendering*, pages 147–162, Darmstadt, Germany, June 1994.

[176] E. Veach and L. J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH 95 Conference Proceedings*, pages 419–428, August 1995.

[177] J. R. Wallace, M. F. Cohen, and D. P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *Computer Graphics (SIGGRAPH '87 Proceedings)*, 21(4):311–320, July 1987.

[178] J. R. Wallace, K. A. Elmquist, and E. A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics (SIGGRAPH '89 Proceedings)*, 23(3):315–324, July 1989.

[179] G. J. Ward. Measuring and Modeling Anisotropic Reflection. In *Computer Graphics (ACM SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.

[180] G. J. Ward and P. Heckbert. Irradiance gradients. In *Third Eurographics Workshop on Rendering*, pages 85–98, May 1992.

[181] W. Wasow. A comments on the inversion of matrices by random walks. *Math. Tabl. Aids. Comput.*, 6:78 – 81, 1952.

[182] T. Whitted. An improved illumination model for shaded display. *Computer Graphics*, 13(2):14–14, August 1979.

[183] A. Willmott and P. Heckbert. An Empirical Comparison of Radiosity Algorithms. Technical report, Carnegie Mellon university, department of Computer Science, May 1997.

[184] W. Xu and D. S. Fussell. Constructing solvers for radiosity equation systems. In *Fifth Eurographics Workshop on Rendering*, pages 207–217, Darmstadt, Germany, June 1994.

[185] H. R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 213–220, 1993.

A Basis Functions for Quadrilaterals and Triangles

The basis functions $\psi_{i,\alpha}$ in our implementation of higher-order radiosity are obtained by uniform mapping of canonical basis functions ψ_α on the unit square or the standard triangle. These canonical basis functions have been obtained by orthonormalisation of the polynomial functions $1, u, v, uv, u^2, v^2, u^3, u^2v, uv^2, v^3, \ldots$ on the unit square or the standard triangle by means of the Gram-Schmidt orthonormalisation procedure.

The Gram-Schmidt orthonormalisation procedure is a general procedure that transforms a given set of n independent vectors $v_i, i = 1, \ldots, n$ into a set of n orthonormal vectors e_i spanning the same vector space. The procedure is outlined in algorithm 21.

Algorithm 21: Gram-Schmidt procedure: transforms a set of n input vectors $v_i, i = 1, \ldots, n$ into a set of n orthonormal vectors e_i spanning the same vector space.

---

1. $e_1 \leftarrow v_1/\|v_1\|$;
2. For $i = 2, \ldots, n$,
   (a) $v_i \leftarrow v_i - \sum_{j=1}^{i-1} < v_i, e_j > e_j$;
   (b) $e_i \leftarrow v_i/\|v_i\|$.

In step 2a of this algorithm, the i-th input vector is replaced by its orthogonal complement w.r.t. the vector space spanned by the first $i-1$ vectors. The orthogonal complement is the difference between a vector and its projection on a sub-space. The projection is obtained as explained in §2.2.2. The vectors $e_j, j < i$ form an orthonormal set, so that the dual vectors in the scalar products for obtaining the projection coefficients, are equal to the primary vectors e_j themselves.

This procedure can be applied to polynomial functions on the unit square or the standard triangle as follows:

- Polynomial functions $F(u, v)$ are represented by their coefficients f_i in $F(u, v) = \sum_i f_i U_i(u, v)$, where U_i denotes the monomials $1, u, v, uv, \ldots$;
- Linear combinations of functions correspond to linear combinations of their coefficients:

$$(F + G)(x) = F(x) + G(x) = \sum_i (f_i + g_i) U_i(x)$$

$$(a \cdot F)(x) = a \cdot F(x) = \sum_i (a \cdot f_i) U_i(x) \quad (a \text{ is a real number})$$

- Scalar products correspond to:

$$< F, G > = < \sum_i f_i U_i, \sum_j g_j U_j > = \sum_{i,j} f_i g_j < U_i, U_j >$$

The scalar products $< U_i, U_j >$ of the monomials U_i are integrals over the unit square or the standard triangle. They are pre-computed once and stored in a table;

- The norm $\|F\|$ is the square root of a scalar product:

$$\|F\| = \sqrt{< F, F >}.$$

The resulting orthonormal polynomials up to degree 3 on the unit square and standard triangle are shown in table A.1. They are plotted in figure 2.3.

	Unit square
ψ_1	1
ψ_2	$\sqrt{3}(2u-1)$
ψ_3	$\sqrt{3}(2v-1)$
ψ_4	$3(1-2u-2v+4uv)$
ψ_5	$\sqrt{5}(1-6u+6u^2)$
ψ_6	$\sqrt{5}(1-6v+6v^2)$
ψ_7	$\sqrt{7}(-1+12u-30u^2+20u^3)$
ψ_8	$\sqrt{15}(-1+6u+2v-12uv-6u^2+12u^2v)$
ψ_9	$\sqrt{15}(-1+2u+6v-12uv-6v^2+12uv^2)$
ψ_{10}	$\sqrt{7}(-1+12v-30v^2+20v^3)$

	Standard triangle
ψ_1	1
ψ_2	$\sqrt{2}(3u-1)$
ψ_3	$\sqrt{6}(-1+u+2v)$
ψ_4	$\sqrt{9/7}(1-4u-4v+20uv)$
ψ_5	$\sqrt{75/7}(1-6.8u-1.2v+6uv+7u^2)$
ψ_6	$\sqrt{15}(1-2u-6v+6uv+u^2+6v^2)$
ψ_7	$-2+30u-90u^2+70u^3$
ψ_8	$\sqrt{12}(-1+13u+2v-24uv-33u^2+21u^3+42u^2v)$
ψ_9	$\sqrt{20}(-1+9u+6v-48uv-15u^2-6v^2+7u^3+42u^2v+42uv^2)$
ψ_{10}	$\sqrt{28}(-1+3u+12v-24uv-3u^2-30v^2+u^3+12u^2v+30uv^2+20v^3)$

Table A.1: Orthonormal polynomials ψ_α on the unit square and the standard triangle.

B Uniform Parametrisation of Convex Quadrilaterals

The implementation of higher-order radiosity algorithms is significantly simplified if a uniform mapping between the standard domain and each 3D patch is used. Integrals on a 3D patch then correspond to integrals on the unit square or standard triangle, multiplied with the patch area (twice the patch area for triangles). A barycentric mapping between the standard triangle and a triangular 3D patch is always uniform. A bilinear mapping between the unit square and a convex 3D quadrilateral is however only uniform if the quadrilateral is a parallelogram. This appendix describes a uniform mapping between the unit square and an arbitrary convex quadrilateral.

The basic idea is to transform bilinear coordinates (s,t) of a point $\vec{p}$ into uniform coordinates (u,v) (and vice versa) in such a way that u denotes the relative area left of the point in the quadrilateral and v denotes the relative area below the point (see figure B.1).

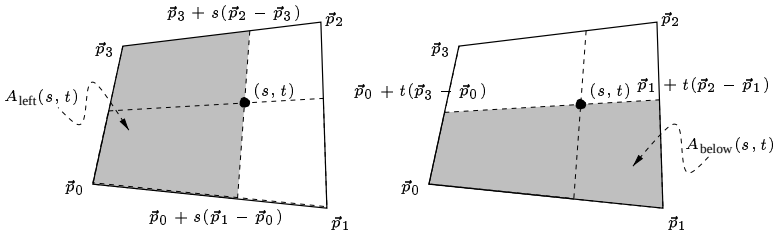


Figure B.1: The uniform coordinates (u,v) of a point $\vec{p}$ with bilinear coordinates (s,t) correspond with the relative area $A_{\text{left}}/A_{\text{total}}$ and $A_{\text{below}}/A_{\text{total}}$ left and below the point.

Consider the Jacobian of the bilinear mapping from the unit square to a convex quadrilateral with vertices $\vec{p}_0, \vec{p}_1, \vec{p}_2, \vec{p}_3$ and normal $\vec{n}$. It can be shown that the Jacobian is of the form $a + bs + ct$:

$$dA(s,t) = (a + bs + ct)\, dsdt$$

where (s,t) are the bilinear coordinates of a point, and:

$$\begin{aligned} a &= ((\vec{p}_1 - \vec{p}_0) \times (\vec{p}_3 - \vec{p}_0)) \cdot \vec{n} \\ b &= ((\vec{p}_2 - \vec{p}_3) \times (\vec{p}_1 - \vec{p}_0)) \cdot \vec{n} \\ c &= ((\vec{p}_2 - \vec{p}_1) \times (\vec{p}_3 - \vec{p}_0)) \cdot \vec{n}. \end{aligned}$$

a, b and c are the surface area of certain parallelograms related with the quadrilateral, with appropriate sign indicating whether the width or height of the quadrilateral increases or decreases as a function of s and t (see figure B.2). The total area of the quadrilateral is given by

$$A_{\text{total}} = \int_0^1 \int_0^1 (a + bs + ct)\, dsdt = a + \frac{1}{2}(b + c).$$

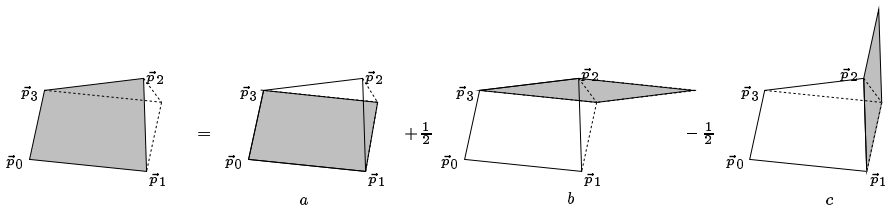


Figure B.2: The surface area of a convex quadrilateral is the sum of the surface area of the parallelogram in (a) and half the surface area of the parallelograms in (b) and (c), with appropriate sign. The coefficients a, b and c in the Jacobian of the bilinear transformation correspond to the signed surface area of these parallelograms.

The area left of a point with bilinear coordinates (s,t) is (see figure B.1, left):

$$A_{\text{left}}(s,t) = \int_0^s \int_0^1 (a + bx + cy)\, dydx = (a + \frac{1}{2}c)s + \frac{1}{2}bs^2.$$

The area below a point with bilinear coordinates (s,t) is (see figure B.1, right):

$$A_{\text{below}}(s,t) = \int_0^1 \int_0^t (a + bx + cy)\, dydx = (a + \frac{1}{2}b)t + \frac{1}{2}ct^2.$$

The uniform coordinates (u,v) corresponding with bilinear coordinates (s,t) can now be obtained as

$$\begin{aligned} u &= \frac{A_{\text{left}}(s,t)}{A_{\text{total}}} = \frac{(a + \frac{1}{2}c)s + \frac{1}{2}bs^2}{a + \frac{1}{2}(b+c)} \\ v &= \frac{A_{\text{below}}(s,t)}{A_{\text{total}}} = \frac{(a + \frac{1}{2}b)t + \frac{1}{2}ct^2}{a + \frac{1}{2}(b+c)}. \end{aligned}$$

These relations allow to map bilinear coordinates to uniform coordinates. In order to map uniform coordinates (u,v) into corresponding bilinear coordinates, the following quadratic equations in s and t need to be solved:

$$\begin{aligned} (a + \frac{1}{2}c)s + \frac{1}{2}bs^2 &= u(a + \frac{1}{2}(b+c)) \\ (a + \frac{1}{2}b)t + \frac{1}{2}ct^2 &= v(a + \frac{1}{2}(b+c)). \end{aligned}$$

These equations have one root in the range $[0,1]$.

In the implementation, a, $\frac{1}{2}b$ and $\frac{1}{2}c$ are pre-computed once and stored for each irregular convex quadrilateral in the scene.

C A Selection of Numerical Integration Rules

This appendix presents a compilation of non-product numerical integration rules for the unit square and the standard triangle. Use of these integration rules is recommended for smooth functions on the unit square or standard triangle. They are useful for 2D-functions that are not the product of two 1D-functions, such as the non-product basis functions in our higher-order radiosity implementation.

Approximate numerical integration is performed by evaluating the integrand at a number of well-chosen nodes, with coordinates (u_k, v_k), where $k = 1, \ldots, K$. K denotes the number of nodes. A weighted sum of the evaluations yields an approximation for the integral:

$$\sum_{k=1}^{K} w_k f(u_k, v_k) \approx \int_D f(u, v)\, du dv.$$

The weights w_k and the node coordinates are carefully chosen so that an as wide as possible class of functions will be integrated exactly with as few nodes as possible. The rules enumerated below are exact for polynomials up to a given degree. The maximum order of a polynomial that is integrated exactly by an integration rule, is called the degree of the rule.

The rules presented below are a selection of rules found in [32, 31]. For degrees 4 to 9, one rule is retained according to the following criteria:

- lowest number of nodes for given degree;
- all nodes lay strictly inside the integration domain;
- the nodes have positive weights;
- the convex hull of the nodes is maximal.

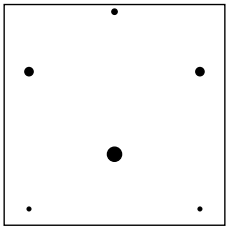
The latter criterion allows better occlusion detection if the integration rules are used in order to compute form factors in radiosity, with visibility testing by ray-tracing.

For all rules below, the sum of the weights has been normalised to 1. The sum of weights should equal the surface area of the domain, and thus be $1/2$ for triangles. Normalising the weights to sum to 1 is convenient because an integral over a 3D patch will then always require multiplication with the patch area when a uniform mapping is used, regardless of the kind of domain. If the weights were normalised to $1/2$ for triangles, a multiplication with twice the patch area would be required for triangles.

The images left of the tables show the location of the nodes. The size of the dots is proportional to the weight of the corresponding node.

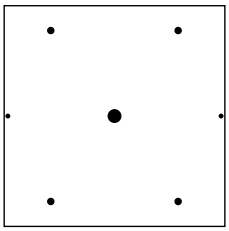
C.1 Numerical integration rules for the unit square

Degree 4, 6 nodes [C6, C4]:



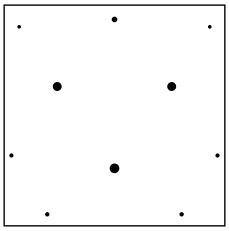
u_k	v_k	w_k
0.5	0.32158895511345498	0.32160302122221301
0.5	0.967086179481358	0.1228414232222315
0.88729833462074148	0.69544258126503555	0.19047092727140325
0.11270166537925852	0.69544258126503555	0.19047092727140325
0.88729833462074148	0.07361731105911451	0.087306850506374503
0.11270166537925852	0.07361731105911451	0.087306850506374503

Degree 5, 7 nodes, Radon's rule [C5]:



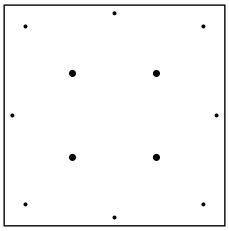
u_k	v_k	w_k
0.5	0.5	0.2857142857142857
0.78867513459481287	0.8872983346207417	0.1388888888888889
0.78867513459481287	0.1127016653792583	0.1388888888888889
0.21132486540518713	0.8872983346207417	0.1388888888888889
0.21132486540518713	0.1127016653792583	0.1388888888888889
0.98304589153964794	0.5	0.079365079365079361
0.016954108460352058	0.5	0.079365079365079361

Degree 6, 10 nodes [C6]:



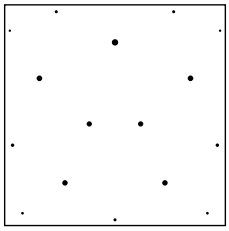
u_k	v_k	w_k
0.5	0.93491668762502944	0.098187647741087
0.5	0.26029682419394451	0.18869072031065251
0.93187141317307698	0.90141875810382843	0.05154151264706975
0.068128586826923021	0.90141875810382843	0.05154151264706975
0.75934526069629604	0.63107183275402901	0.17249803462246599
0.24065473930370401	0.63107183275402901	0.17249803462246599
0.96698627248642444	0.31845170842596704	0.065129372183079251
0.033013727513575508	0.31845170842596704	0.065129372183079251
0.80448876800817803	0.051695683618773525	0.067391896521515254
0.19551123199182202	0.051695683618773525	0.067391896521515254

Degree 7, 12 nodes [C5]:



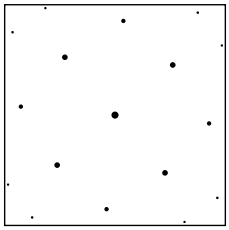
u_k	v_k	w_k
0.96291004988627571	0.5	0.060493827160493827
0.03708995011372429	0.5	0.060493827160493827
0.5	0.96291004988627571	0.060493827160493827
0.5	0.03708995011372429	0.060493827160493827
0.69027721660415775	0.69027721660415775	0.13014822916684862
0.69027721660415775	0.30972278339584219	0.13014822916684862
0.30972278339584219	0.69027721660415775	0.13014822916684862
0.30972278339584219	0.30972278339584219	0.13014822916684862
0.90298989145929942	0.90298989145929942	0.059357943672657558
0.90298989145929942	0.097010108540700579	0.059357943672657558
0.097010108540700579	0.90298989145929942	0.059357943672657558
0.097010108540700579	0.097010108540700579	0.059357943672657558

Degree 8, 16 nodes [C6]:



u_k	v_k	w_k
0.5	0.82978006598017107	0.1125691940763975
0.5	0.025428538478437512	0.041642606694453251
0.97625473303578092	0.88252590977884204	0.024717364983357751
0.023745266964219025	0.88252590977884204	0.024717364983357751
0.766163727037103	0.96848799054420798	0.038424186785202999
0.233836272962897	0.96848799054420798	0.038424186785202999
0.84236814897586743	0.66682835886787351	0.099171744018225746
0.15763185102413252	0.66682835886787351	0.099171744018225746
0.6165716204007024 6	0.46020836381130148	0.088003591986423754
0.38342837959929749	0.46020836381130148	0.088003591986423754
0.96384165965305857	0.36387995969373299	0.047397263644449503
0.036158340346941487	0.36387995969373299	0.047397263644449503
0.7265603437018745	0.19313232330098601	0.093775250286896747
0.2734396562981255	0.19313232330098601	0.093775250286896747
0.918751820211406	0.055761174732014473	0.031404697910018001
0.081248179788593999	0.055761174732014473	0.031404697910018001

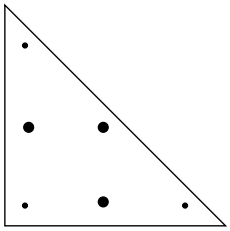
Degree 9, 17 nodes [C3]:



u_k	v_k	w_k
0.5	0.5	0.13168724279835392
0.98442498318098881	0.81534005986583447	0.022219844542549678
0.015575016819011134	0.18465994013416559	0.022219844542549678
0.18465994013416559	0.98442498318098881	0.022219844542549678
0.81534005986583447	0.015575016819011134	0.022219844542549678
0.87513854998945029	0.96398082297978482	0.02802490053239912
0.12486145001054971	0.036019177020215176	0.02802490053239912
0.036019177020215176	0.87513854998945029	0.02802490053239912
0.96398082297978482	0.12486145001054971	0.02802490053239912
0.76186791010721466	0.7266699105678236	0.099570609815517519
0.23813208989278534	0.2733300894321764	0.099570609815517519
0.2733300894321764	0.76186791010721466	0.099570609815517519
0.7266699105678236	0.23813208989278534	0.099570609815517519
0.53810416409630857	0.92630786466683113	0.067262834409945196
0.46189583590369143	0.073692135333168873	0.067262834409945196
0.073692135333168873	0.53810416409630857	0.067262834409945196
0.92630786466683113	0.46189583590369143	0.067262834409945196

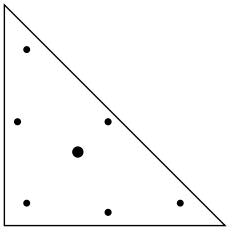
C.2 Numerical integration rules for the standard triangle

Degree 4, 6 nodes [C2]:



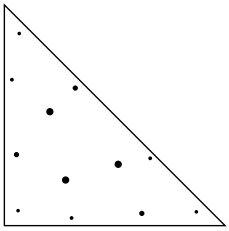
u_k	v_k	w_k
0.81684757298045851	0.091576213509770729	0.10995174365532183
0.091576213509770729	0.091576213509770729	0.10995174365532183
0.091576213509770729	0.81684757298045851	0.10995174365532183
0.10810301816807021	0.44594849091596489	0.2233815896780115
0.44594849091596489	0.44594849091596489	0.2233815896780115
0.44594849091596489	0.10810301816807021	0.2233815896780115

Degree 5, 7 nodes [C5]:



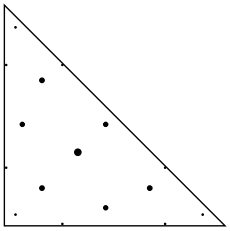
u_k	v_k	w_k
0.33333333333333331	0.33333333333333331	0.22500000000000001
0.1012865073234563	0.1012865073234563	0.12593918054482711
0.1012865073234563	0.79742698535308731	0.12593918054482711
0.79742698535308731	0.1012865073234563	0.12593918054482711
0.47014206410511511	0.47014206410511511	0.13239415278850619
0.47014206410511511	0.059715871789769809	0.13239415278850619
0.059715871789769809	0.47014206410511511	0.13239415278850619

Degree 6 and 7, 12 nodes [C1]:



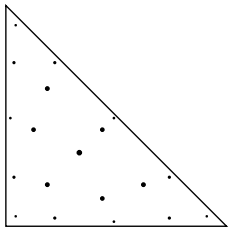
u_k	v_k	w_k
0.062382265094390842	0.067517867073924362	0.053034056314869002
0.067517867073924362	0.8700998678316848	0.053034056314869002
0.8700998678316848	0.062382265094390842	0.053034056314869002
0.055225456656919997	0.32150249385201563	0.087762817428896217
0.32150249385201563	0.62327204949106441	0.087762817428896217
0.62327204949106441	0.055225456656919997	0.087762817428896217
0.034324302945094878	0.66094919618679804	0.05755008556995056
0.66094919618679804	0.30472650086810721	0.05755008556995056
0.30472650086810721	0.034324302945094878	0.05755008556995056
0.51584233435360005	0.277716166976405	0.13498637401961758
0.277716166976405	0.20644149866999489	0.13498637401961758
0.20644149866999489	0.51584233435360005	0.13498637401961758

Degree 8, 16 nodes [C2]:



u_k	v_k	w_k
0.33333333333333331	0.33333333333333331	0.1443156076777862
0.081414823414554124	0.4592925882927229	0.095091634267284966
0.4592925882927229	0.4592925882927229	0.095091634267284966
0.4592925882927229	0.081414823414554124	0.095091634267284966
0.89890554336593786	0.050547228317031033	0.032458497623198135
0.050547228317031033	0.050547228317031033	0.032458497623198135
0.050547228317031033	0.89890554336593786	0.032458497623198135
0.65886138449647969	0.1705693077517601	0.1032173705347184
0.1705693077517601	0.1705693077517601	0.1032173705347184
0.1705693077517601	0.65886138449647969	0.1032173705347184
0.0083947774099572114	0.72849239295540413	0.027230314174434864
0.0083947774099572114	0.26311282963463872	0.027230314174434864
0.72849239295540413	0.26311282963463872	0.027230314174434864
0.72849239295540413	0.0083947774099572114	0.027230314174434864
0.26311282963463872	0.0083947774099572114	0.027230314174434864
0.26311282963463872	0.72849239295540413	0.027230314174434864

Degree 9, 19 nodes [C2]:



u_k	v_k	w_k
0.33333333333333331	0.33333333333333331	0.097135796282796102
0.020634961602525929	0.48968251919873701	0.031334700227139835
0.48968251919873701	0.48968251919873701	0.031334700227139835
0.48968251919873701	0.020634961602525929	0.031334700227139835
0.125820817014129	0.43708959149293553	0.07782754100477543
0.43708959149293553	0.43708959149293553	0.07782754100477543
0.43708959149293553	0.125820817014129	0.07782754100477543
0.62359292876193562	0.18820353561903219	0.079647738927209097
0.18820353561903219	0.18820353561903219	0.079647738927209097
0.18820353561903219	0.62359292876193562	0.079647738927209097
0.91054097321109406	0.044729513394452969	0.025577675658698101
0.044729513394452969	0.044729513394452969	0.025577675658698101
0.044729513394452969	0.91054097321109406	0.025577675658698101
0.036838412054736258	0.74119859878449801	0.043283539377289404
0.036838412054736258	0.22196298916076571	0.043283539377289404
0.74119859878449801	0.22196298916076571	0.043283539377289404
0.74119859878449801	0.036838412054736258	0.043283539377289404
0.22196298916076571	0.036838412054736258	0.043283539377289404
0.22196298916076571	0.74119859878449801	0.043283539377289404

Bibliography

[C1] Gatermann. The construction of symmetric cubature formulas for the square and the triangle. *Computing*, 40:229–240, 1988.

[C2] Lyness and Jespersen. Moderate degree symmetric quadrature rules for the triangle. *J. Inst. Maths. Applics.*, 15:19–32, 1975.

[C3] Möller. Kubaturformeln mit minimaler Knotenzahl. *Numerische Mathematik*, 25(2):185–200, 1976.

[C4] Schmid. On cubature formulae with a minimal number of knots. *Numerische Mathematik*, 31(3):281–297, 1978.

[C5] Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, 1972.

[C6] Wissmann and Becker. Partially symmetric cubature formulas for even degrees of exactness. *SIAM Journal on Numerical Analysis*, 23(3):676–685, June 1986.

D Low-discrepancy sampling of points on a triangle

This appendix describes a uniform mapping from the unit square to the standard triangle $(0,0),(1,0),(0,1)$ that transforms dyadic boxes in the unit square into dyadic triangles. A dyadic box is a sub-square defined by coordinates $u \in [i/2^k,(i+1)/2^k), v \in [j/2^k,(j+1)/2^k)$, where $i,j = 1,\ldots,2^k-1$. A dyadic triangle is a sub-triangle that results by connecting the mid-points of the sides of a parent dyadic triangle.

Such a transformation is useful in the context of hierarchical Monte Carlo radiosity (chapter 14) with regular quadtree subdivision of quadrilaterals and triangles. Indeed: sub-elements on quadrilaterals correspond to dyadic boxes in the unit square in this case, and sub-elements on triangular patches correspond to dyadic triangles. The mapping proposed here simplifies the implementation of hierarchical Monte Carlo radiosity because it allows the same sampling and sample counting routines to be used on triangles and quadrilaterals.

Also for other problems however, this mapping will be useful in connection with low-discrepancy sampling with a base-2 2D Niederreiter sequence [16, 15]. With this sequence, sample points on the unit square are placed in dyadic boxes in a particular order that ensures low discrepancy. The transformed points will be placed in corresponding (isomorphic) dyadic triangles in the same order, so that low-discrepancy sampling on the standard triangle can be expected[1].

The mapping assumes that the points to be transformed lay on a regular grid with resolution 2^k in each dimension, in other words that the coordinates are $i/2^k$ with $i = 0,\ldots,2^k-1$. Here, k is the number of bits in the base-2 representation of the point coordinates. This is the case with the base-2 Niederreiter sequences described in [16, 15]. It is straightforward to generalise the mapping being presented here to an arbitrary (homogeneous) base.

First, in order to avoid that different points on the unit square will be mapped to the same destination point in the standard triangle, the points in the square are shifted by 1/4-th of the grid resolution in each direction (see figure D.1). Now consider the parent grid cells at resolution 2^{k-1}. Each such parent grid cell contains four points, one on the lower-left of each quadrant of the cell. Three of these points lay below the main cell diagonal. One point lays above the diagonal. The point above the diagonal is the point in the upper-right quadrant of the parent cell. By "folding" the cell along the main diagonal, the point above the diagonal is transformed into a point below the diagonal. The three points below the diagonal do not change (see figure D.1, rightmost illustration). Subsequently, the process of folding cells along their main diagonal is repeated at lower resolutions, until eventually all points lay below the main diagonal of the unit square (figure D.2). The triangle below the main diagonal of the unit square is the standard triangle.

All needed operations can be performed very efficiently by considering the bit-

[1] An in-depth theoretical study of the properties of this transformation is a topic for future research.

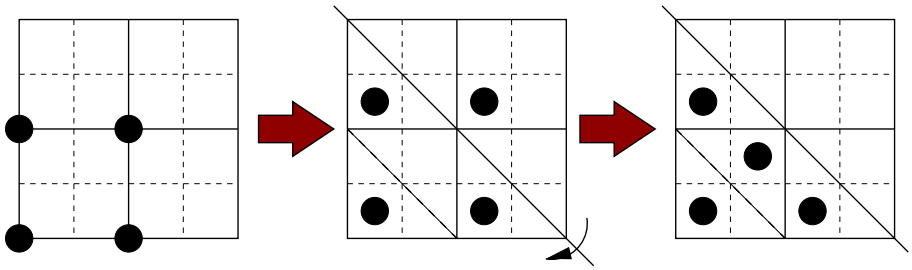


Figure D.1: Before applying the new mapping, the points to be mapped are slightly displaced in order to prevent coinciding results. The right-most illustration shows the result of the first folding step.

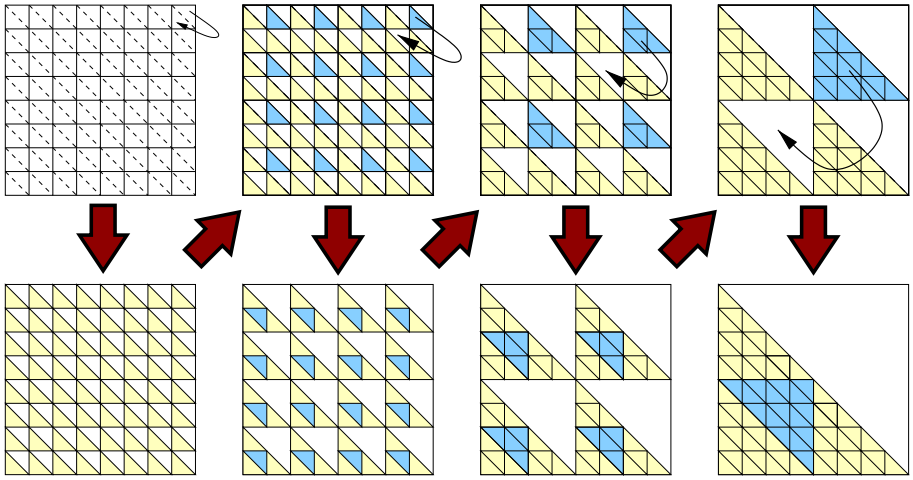


Figure D.2: The new mapping transforms the unit square into the standard triangle by repeated folding. First, the triangles above the diagonal in dyadic cells at the highest resolution are folded along the diagonal (top-left). The result of this step in one cell is shown in detail in the right-most illustration of figure D.1. Next, the folding process is repeated at lower resolutions. Eventually, all points in the unit square are mapped to the standard triangle (bottom-right).

representation of the integer numbers i in the point coordinates $i/2^k$:

- Displacement of the input points can be implemented by shifting the bit representation 2 bits to the left and next setting the least significant bit to 1;
- Whether or not folding at a given resolution is necessary can be determined by inspecting the bits at corresponding positions in the bit-representation of each coordinate. If the corresponding bits both are 1, the point lays in the upper-right quadrant and folding is necessary. No folding is necessary if the corresponding bits are both 0, or one is 0 and the other is 1, in which case the point lays in the lower-left quadrant or the lower-right or upper-left quadrant respectively;
- Rather than folding, the same effect is obtained more efficiently by mirroring

around the centre of the containing cell. If the cell were the full unit square, mirroring would be obtained by replacing both coordinates by their 1-complement: $(x, y) \to (1 - x, 1 - y)$. Folding at higher resolutions is obtained by replacing only the least significant bits of the coordinates. The most significant bits determine the cell in which the point is to be folded. Since folding leaves the point inside the containing cell, the most significant bits shall be left unchanged.

- We found it convenient to start folding at the lowest resolution level (the whole unit square) and proceed with higher resolutions rather than vice versa. The result is identical, since the number of times that each bit in the coordinate representation is toggled remains the same.

The resulting C-programming language code fragment is shown below. `u` and `v` are unsigned integers containing the integer representation of the coordinates of the point to be transformed: $u = \text{integer}(x \times 2^k), v = \text{integer}(y \times 2^k)$. At the end, they contain the integer representation of the coordinates of the transformed point: $x' = u/2^{k+2}, y' = v/2^{k+2}$. `NBITS` is $k + 2$: the number of bits in the coordinate representation, plus two (because of the displacement). Care must be taken that `NBITS` is less than the number of bits in the representation of an unsigned integer, or an overflow error will occur.

```
/* unsigned u, v;     (given) */
   unsigned m, d;

   u = (u<<2) | 1; v = (v<<2) | 1;  /* displace */

   /* d contains 1's where folding is needed */
   d = (u & v) & ~1;

   /* mask marking most significant bits */
   m = 1<<NBITS;

   while (d) {
     if (d&(1<<(NBITS-1))) {        /* need to fold */
       u = (u & m) | (~(u-1) & ~m); /* fold */
       v = (v & m) | (~(v-1) & ~m);
     }
     d <<= 1;
     m |= m>>1;
   }
```

Notations

$< X, Y >$	scalar product of X and Y (functions or vectors)
$\|X\|$	norm of X (functions, vectors, matrices)
$\overline{X}$	overestimate for a quantity X
$\underline{X}$	underestimate for a quantity X
$\tilde{X}$	estimate or approximation for X, or modified quantity X
$\hat{X}$	Monte Carlo estimator for a quantity X
$\hat{X}_N$	secondary estimator for X: $\hat{X}$ with N samples
$\lfloor x \rfloor$	largest integer smaller of equal to x
$\mathbf{A}, \mathbf{C}$	matrices with elements a_{ij}, c_{ij}
$\mathbf{a}, \mathbf{b}, \mathbf{e}, \mathbf{r}, \mathbf{x}, \mathbf{w}$	vectors with components $a_i, b_i, \ldots$
A_i	surface area of patch i
α_i	absorption probability of a random walk at state i
B_i	radiosity emitted by patch i (constant approximations)
$B_{i,\alpha}$	radiosity emitted "under" basis function $\psi_{i,\alpha}$
$B(x)$	"true" radiosity function at point x
$\tilde{B}(x)$	$\sum_{i,\alpha} B_{i,\alpha}\psi_{i,\alpha}(x)$
b_i	$B_i - E_i$: non-selfemitted radiosity at patch i
$B[\hat{X}]$	bias of the Monte Carlo estimator $\hat{X}$ for X
$\mathrm{Cov}[\hat{X}, \hat{Y}]$	co-variance of Monte Carlo estimators $\hat{X}$ and $\hat{Y}$
$\mathbf{C}^\top$	transpose of the matrix $\mathbf{C}$: elements $c_{ij}^\top = c_{ji}$
$\chi_j(x)$	characteristic function of S_j: 1 if $x \in S_j$ and 0 if not
δ_{ij}	Kronecker's delta: 1 if $i = j$, 0 if $i \neq j$
E_i	self-emitted radiosity by patch i (constant approximations)
$E_{i,\alpha}$	self-emitted radiosity "under" basis function $\psi_{i,\alpha}$
$E(x)$	"true" self-emitted radiosity function at x
$\tilde{E}(x)$	$\sum_{i,\alpha} E_{i,\alpha}\psi_{i,\alpha}(x)$
$E[\hat{X}]$	expectation of the Monte Carlo estimator $\hat{X}$ for X
$\varepsilon(x)$	error at a point x
ε_i	error on a patch i
F_{ij}	patch-i-to-patch-j form factor
$F_{dA_x,j}$	point-x-to-patch-j form factor
$f(P_i)$	maps a spectral representation of P_i to a single floating point value

$G(x,y)$	geometric radiosity kernel
$G_j(x)$	$\int_{S_j} G(x,y)dA_y$: point-x-to-patch-j form factor
$G_{j,\beta}(x)$	$\int_{S_j} G(x,y)\psi_{j,\beta}(y)dA_y$: generalised point-to-patch form factor
$h(x,\Theta)$	first point on a surface visible from x in the direction Θ_x
I_i	importance at patch i (radiosity-like)
J	random walk $j_0, j_1, \ldots, j_\tau$
$K(x,y)$	$\rho(x)G(x,y)$
$K_j(x)$	$\rho(x)G_j(x)$
$K_{j,\beta}(x)$	$\rho(x)G_{j,\beta}(x)$
$K_{i,\alpha;j,\beta}$	generalised form factor between $\psi_{i,\alpha}$ and $\psi_{j,\beta}$
$\tilde{K}_{j,\beta}(x)$	approximation of $K_{j,\beta}(x)$ as a linear combination of basis functions $\psi_{i,\alpha}(x)$ at x
N	number of samples in a Monte Carlo computation
n	size of a problem, for instance number of equations in a linear system
$\Omega(x)$	hemisphere of directions above x
$\Omega_j(x)$	directions from x pointing to patch or element j
$d\omega_\Theta$	infinitesimal solid angle containing direction Θ
P_i	A_iB_i: power emitted by patch i
$p(x)$	(continuous) probability density at a point x
p_i	(discrete) probability at a patch or state i
$p(J)$	probability associated by random walk J
p_{ij}	transition probability from i to j
π_i	random walk birth probability at i
Φ_i	A_iE_i: self-emitted power at patch i
$\psi_{i,\alpha}$	α-th basis function on patch i
ψ_α	α-th canonical basis function (unit square or standard triangle)
$\tilde{\psi}_{i,\alpha}$	α-th dual basis function on patch i
r_{xy}	distance between points x and y
ρ_i	reflectivity of patch i
$\rho(x)$	reflectivity at point x
S_i	surface of patch i (set of points)
$s(J)$	score of a random walk J
σ_i	survival probability of a random walk at state i
Θ_x	out-going direction at point x
θ_x	angle between Θ_x and the surface normal at x
Y_i	A_iI_i: importance at patch i (power-like)
V_i	source-importance at patch i (radiosity-like)
$V[\hat{X}]$	variance of the Monte Carlo estimator $\hat{X}$ for X
$\text{vis}(x,y)$	visibility predicate: 1 if x and y are mutually visible, 0 if not
W_i	A_iV_i: source-importance at patch i (power-like)
ζ_i	recurrent radiosity: fraction of radiosity on i due to itself