Eurographics Symposium on Rendering - Experimental Ideas & Implementations (2017)
P. Sander and M. Zwicker (Editors)

Temporal Coherence for Metropolis Light Transport

Joran Van de Woestijne, Roald Frederickx, Niels Billen and Philip Dutré

Department of Computer Science, KU Leuven, Belgium

Abstract

Metropolis Light Transport is a powerful global illumination algorithm, yet it has some issues that make it less suitable for animation rendering. Due to the algorithm's local exploration of path space, difficult light paths can appear very late in the rendering process or might be missing from the final image altogether. This unpredictable convergence behaviour is especially troublesome when rendering animations, since these paths need to be rediscovered for every frame. An inability to rediscover difficult light paths across the entire animation causes unpleasant flickering artefacts. Our algorithm tackles this issue by introducing temporal mutations for Metropolis Light Transport, which perturb a light path from one point in time to another, potentially across many frames. This allows us to propagate difficult paths through the entire animation. Our technique supports multiple animation types, such as camera motion and object motion, and can robustly handle different shutter setups. The convergence speed of individual frames with motion blur is increased and convergence variations between frames are diminished, especially for specular transport and difficult indirect lighting.

CCS Concepts

•***Computing methodologies*** → ***Rendering;*** *Ray tracing;*

1. Introduction

Computing global illumination efficiently while handling complex illumination, materials and geometry remains a significant problem in computer graphics. Most common rendering systems today rely on numerical integration techniques based on ray tracing to numerically evaluate the rendering equation.

The Metropolis Light Transport algorithm solves the rendering equation by using Markov chains which stochastically explore the neighbourhoods of paths based on their relative contribution to the image. While this technique makes efficient reuse of important paths, the initial discovery of these paths can still pose some difficulty. Within the context of animations, the difficult light paths have to be rediscovered in every frame of the animation, otherwise noticeable flickering noise can appear.

In this paper, we introduce a temporal mutation strategy to solve this problem. This mutation allows paths to jump between different time points in the animation, thereby sharing these paths over multiple frames and improving the temporal coherence.

The contributions of this paper are the following:

- We propose a new temporal perturbation which mutates a path in time;
- We allow difficult light paths to be propagated through an animation, thereby improving coherence and overall convergence between frames;
- We show that our algorithm achieves smoother motion blur in scenes with difficult moving specular paths, and it leads to a significant reduction in troublesome flickering artefacts.

2. Related Work

2.1. Markov Chain Monte Carlo Rendering

The seminal rendering technique that first used a Markov Chain Monte Carlo (MCMC) method [MRR*53, Has70] is Metropolis Light Transport (MLT) [VG97]. To explore path space, the MLT process iteratively samples a new proposed light path $\mathbf{X}^p$ by mutating the current path $\mathbf{X}^i$. Once a mutated sample has been proposed, it is probabilistically accepted or rejected in such a way to ensure that the resulting sampling density of the paths matches their relative contribution to the image.

The mutation strategies of Veach and Guibas [VG97] require path space knowledge and specifically designed transition probability functions for every mutation. Instead of applying mutations in path space, Kelemen et al. [KSKAC02] proposed to mutate the space of random numbers used to generate the paths, called the Primary Sample Space. While mutating the random numbers is conceptually simpler, one loses the ability to use path space knowledge in crafting specialised mutation strategies for particularly difficult light paths. Two such specialised mutation strategies that improve the handling of hard specular-diffuse-specular paths in classical MLT were developed by Jakob and Marschner [JM12] and Kaplanyan et al. [KHD14]. These methods use path space informa-

tion to efficiently explore the manifold imposed by the constraints of specular light transport.

Energy Redistribution Path Tracing [CTE05] is an MCMC technique which uses a different approach to spread out the energy through the path space. Instead of using bi-directional path tracing to generate seeds, it relies on a uni-directional path tracer to improve the stratification of the seeds in the image region. The employed mutation strategy uses a large number of short Markov Chains to distribute the energy through the path space, as compared to MLT, which uses a smaller number of very long Markov chains. Population Monte Carlo Energy Redistribution by Lai et al. [LFCD07] further improves upon this method by stopping the chains with diminishing returns and swapping them for new seeds.

2.2. Temporal MCMC Rendering

Li et al. [LLR*15] proposed to apply improved mutations which adapt to the local shape of the integrand in primary sample space. While their algorithm applies mutations in primary sample space, our algorithm applies mutations in path space, which allows us to leverage more information about the paths. The technique by Li et al. also considers the temporal aspect of the manifold to improve motion blur for these difficult specular light paths.

Lai et al. [LLD09] created a framework for animations and introduced a time perturbation, with the goal of adding temporal coherence to their Population Monte Carlo Energy Redistribution algorithm. Only the frames of the animation are considered in their technique and not the time between the frames, which can become problematic when strong temporal variations prohibit light paths from jumping over the duration of a full frame. We approach this differently by staying in the MLT framework and regarding time as a continuum, which allows us to adapt the perturbation size to the temporal complexity. Xu et al. [XCS09] also aim to render animations by making Energy Redistribution Path Tracing temporally coherent, but they only consider the re-use of light paths in neighbouring frames and no perturbation of the light path is discussed.

To our knowledge, no documented efforts have been made that extend the classical path space Metropolis Light Transport algorithm to exploit temporal coherence for animation rendering.

3. Temporal Perturbations

The mutation strategies introduced by Veach and Guibas [VG97] do not consider time parameter mutations of the paths generated by a Markov chain. In images containing motion blur, all paths generated by a single Markov chain will share the time parameter of the initial seed path (see Figure 1). Consequently, a large number of chains is required to adequately sample the entire path space over the full time range. When an insufficient number of Markov chains is used, the motion blur will not appear smooth due to the time correlation within the chains, as can be seen in Figure 7b. Increasing the number of chains for a fixed mutation budget, decreases the number of mutations per chain, defeating the purpose of the Markov chain exploration and degenerating towards the initial bidirectional seed samples in the limit. Moreover, the inability to mutate the time parameter is even more problematic when rendering animations. Since paths cannot be shared between frames, flickering artifacts occur when a difficult light path cannot be rediscovered across different frames, which as of yet has prohibited the use of MLT in production rendering [CJ16].

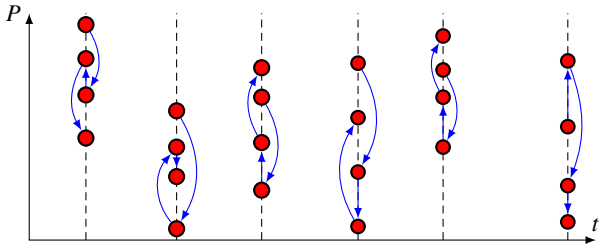


Figure 1: *Markov chains exploring the (spatial) path space P when rendering an image with motion blur over a time interval in t. The Markov chain samples are represented by red dots and the transitions (mutations) are represented by blue arrows. Without temporal mutations, each Markov chain explores the path space at a single time point.*

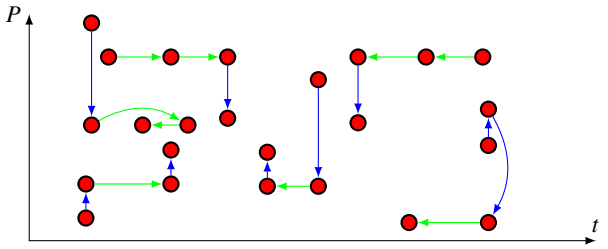


Figure 2: *Markov chains exploring the path space P when rendering an image with motion blur over a time interval t, while making use of temporal perturbations. The Markov chain samples are represented by red dots, spatial transitions (standard MLT mutations) are represented by blue arrows, and temporal transitions (temporal perturbations) are represented by green arrows.*

We propose to solve these issues by introducing a new mutation strategy, called *temporal perturbations*, which allow a Markov chain to jump from one time point to another (see Figure 2). A temporal perturbation changes the time parameter of the path and attempts to construct a new valid path at that new time point, which can potentially be situated in a different frame when rendering animations. Sections 3.1 discusses how such perturbed paths can be constructed. The transition probabilities of these mutations are derived in Section 3.2, and Section 3.3 shows how animations can be rendered efficiently.

3.1. Temporal Perturbation

Given an initial path $\mathbf{X}^{\mathrm{i}} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n)$ at time t^{i}, a temporal perturbation to a proposed time t^{p} will change the position of vertices located on animated objects from $\mathbf{x}_i$ to $\mathbf{x}'_i$ (see Figure 3). Furthermore, when such a 'moving' vertex $\mathbf{x}'_i$ is located on a specular surface, the subsequent specular path is followed, resulting in a chain of moved vertices $(\mathbf{x}'_i, \mathbf{x}'_{i+1}, \ldots, \mathbf{x}'_k)$ until a diffuse vertex at $\mathbf{x}'_{k+1}$ is intersected. Similarly as chosen in the (non-bidirectional)

perturbations of the original MLT technique [Vea98], we currently reject proposal paths if their length or configuration (i.e. the scattering interaction type: diffuse, specular transmission/reflection) have changed from the original path. To check the validity of a new proposal path $\mathbf{X}^{\mathrm{p}} = (\mathbf{x}_0^*, \mathbf{x}_1^*, \ldots, \mathbf{x}_n^*)$, we need to reconnect every edge, $\mathbf{x}_i^* \rightarrow \mathbf{x}_{i+1}^*$, between successive vertices of the new path (where $\mathbf{x}_i^*$ can be either be a stationary vertex $\mathbf{x}_i$ or a vertex $\mathbf{x}_i'$ that was moved because it is part of an animated object or because it is situated in the tail of a specular chain). Concretely, a full perturbation is constructed by iterating over the vertices $\mathbf{x}_i^*$ for $i = 0, 1, \ldots, n$ and distinguishing the following cases:

- $\mathbf{x}_i^*$ **is diffuse**: We can reconnect the edge between the diffuse vertex $\mathbf{x}_i^*$ and next vertex $\mathbf{x}_{i+1}^*$ by testing whether they remain mutually visible after the motion of the vertices and/or the motion of other objects.
- $\mathbf{x}_i^*$ **is specular**: The outgoing direction at $\mathbf{x}_i^*$ is fixed due to the constraint imposed by the specular interaction and the incoming direction $\overrightarrow{\mathbf{x}_{i-1}^* \mathbf{x}_i^*}$ from the previous edge. We follow the specular interactions $(\mathbf{x}_i^*, \mathbf{x}_{i+1}', \ldots, \mathbf{x}_d')$ until the specular path ends on a diffuse vertex $\mathbf{x}_d'$, and then check the visibility of this vertex with the next one at $\mathbf{x}_{d+1}^*$, from which we continue the iteration.

Note that the above reconnection strategy is described as starting from the camera vertex $\mathbf{x}_0^*$ (see Figure 3), but it can easily be reversed to start from the light source vertex $\mathbf{x}_n^*$ (see Figure 4).

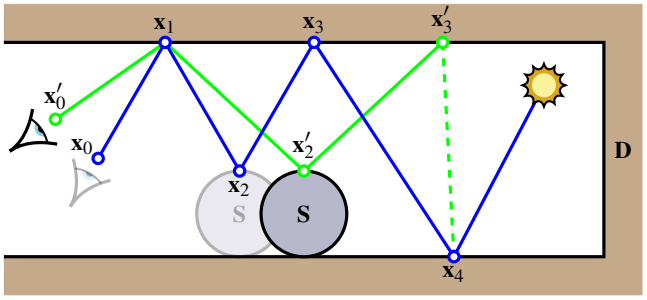


Figure 3: *A temporal perturbation with a moving camera and a moving specular sphere. The path is reconnected starting from the camera towards the light source. The recalculation of the path starts at the first diffuse intersection from the camera, where the connection with the displaced vertex on the sphere is made. As the sphere is specular, the outgoing direction is followed before finally reconnecting to the original light path at point* $\mathbf{x}_4$*, which is the second consecutive diffuse vertex.*

3.2. Transition Probability Functions

The MLT algorithm generates a Markov chain of sample paths by generating a tentative proposal path $\mathbf{X}^{\mathrm{p}}$ as a mutation of the current path $\mathbf{X}^{\mathrm{i}}$. The state of the Markov chain advances either to $\mathbf{X}^{\mathrm{p}}$ or remains in $\mathbf{X}^{\mathrm{i}}$ according to the following acceptance probability [VG97]

$$a(\mathbf{X}^{\mathrm{i}} \rightarrow \mathbf{X}^{\mathrm{p}}) = \min\left\{1, \frac{f(\mathbf{X}^{\mathrm{p}})T(\mathbf{X}^{\mathrm{p}} \rightarrow \mathbf{X}^{\mathrm{i}})}{f(\mathbf{X}^{\mathrm{i}})T(\mathbf{X}^{\mathrm{i}} \rightarrow \mathbf{X}^{\mathrm{p}})}\right\}, \tag{1}$$

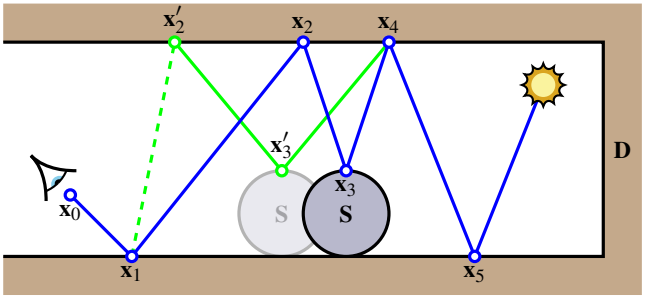


Figure 4: *A temporal perturbation with a moving specular sphere. The path is reconnected starting from the light source towards the camera. The reconnection is similar to Figure 3, except that the process starts at the second diffuse intersection from the light source and reconnects at point* $\mathbf{x}_1$*, which is the last diffuse vertex before the camera.*

with f the path contribution function and $T(\mathbf{X}^{\mathrm{i}} \rightarrow \mathbf{X}^{\mathrm{p}})$ the transition probability of generating a new sample $\mathbf{X}^{\mathrm{p}}$ from the sample $\mathbf{X}^{\mathrm{i}}$.

For our temporal perturbation, the transition probability function $T(\mathbf{X}^{\mathrm{i}} \rightarrow \mathbf{X}^{\mathrm{p}})$ is the product of two factors:

1. The time probability function, $p_{\mathrm{time}}(t^{\mathrm{i}} \rightarrow t^{\mathrm{p}})$, of changing the time parameter t^{i} of the original path $\mathbf{X}^{\mathrm{i}}$ to the time parameter t^{p} of the proposed path $\mathbf{X}^{\mathrm{p}}$.
2. The probability density of all possible ways of generating the proposed path $\mathbf{X}^{\mathrm{p}}$ from the previous configuration $\mathbf{X}^{\mathrm{i}}$.

The first factor, $p_{\mathrm{time}}(t^{\mathrm{i}} \rightarrow t^{\mathrm{p}})$, can have any distribution under the ergodicity condition that every point in time can eventually be sampled. When rendering a single image with motion blur, we can easily satisfy this condition by using a uniform distribution over the time range. For animations, a more elaborate perturbation strategy is described in Section 3.3.

The second factor, the probability density for all the possible ways of generating the proposal $\mathbf{X}^{\mathrm{p}}$, can be derived by noting that our reconnection strategy is a deterministic process. As there is only one possible way to generate the proposal path, only the probability density of this path needs to be taken into account, which is equal to product of the individual probability densities of the new vertices.

This allows us to write the transition probability function of our temporal perturbations as

$$T(\mathbf{X}^{\mathrm{i}} \rightarrow \mathbf{X}^{\mathrm{p}}) = p_{\mathrm{time}}(t^{\mathrm{i}} \rightarrow t^{\mathrm{p}}) \prod_{i=0}^{n-1} P_{\sigma^\perp}(x_i \rightarrow x_{i+1}) G(x_i \leftrightarrow x_{i+1}), \tag{2}$$

with n the length of the path, $P_{\sigma^\perp}(x_i \rightarrow x_{i+1})$ the probability density of sampling the direction from x_i to x_{i+1} and $G(x_i \leftrightarrow x_{i+1})$ the geometry factor between x_i and x_{i+1} [Vea98]. Note that the division in Equation (1) cancels the common factors arising from the part of the path before the first and after the last perturbed vertex.

3.3. Animation Rendering

Until now, we have considered the time range t to span across a single motion-blurred image since the frames of an animation are traditionally rendered separately. However, by interpreting t as a continuous domain spanning across multiple frames of an animation, all frames of an animation can be rendered simultaneously (see Figure 5). Because t is continuous over multiple frames, our temporal perturbation can share difficult light paths across multiple frames, without having to rediscover them from scratch.

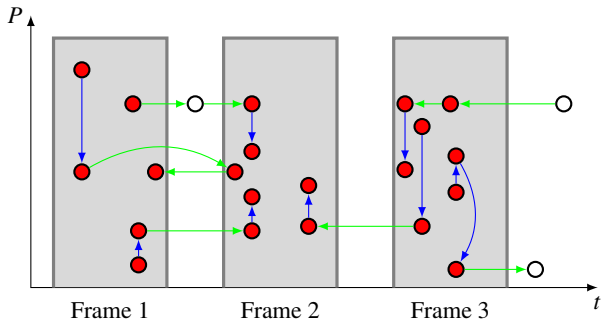


Figure 5: *Markov chains exploring the path space when rendering an animation, while making use of temporal perturbations. This allows chains to move across frames, increasing the likelihood that difficult light paths are found across all frames. Samples within an open shutter (red) will contribute to the image, while samples during a closed shutter (white) do not contribute to the image.*

Previously, we discussed a temporal perturbation that samples a new time point t^{p} from a uniform distribution over the duration of a single frame when rendering motion blur. However, a trivial generalisation to a uniform proposal distribution over the full span of an animation can introduce two problems. Firstly, the acceptance probability can drop severely for large jumps in the time domain t. As the temporal perturbation size increases, the probability that all nodes of a path can be successfully reconnected decreases due to the motion of the objects in the scene. Secondly, animations are usually rendered using a realistic camera model which emulates a shutter that is only open for a short amount of time in each frame. When the new point of time is chosen uniformly, a large number of samples will be generated at times when the shutter is closed and will therefore not contribute to the final image.

To solve the aforementioned problems, we propose the following two modifications to $p_{\mathrm{time}}(t^{\mathrm{i}} \to t^{\mathrm{p}})$. First, we limit the interval for the proposed time t^{p} to a small range Δ centered around the current time point t^{i} where $t^{\mathrm{p}} \in [t^{\mathrm{i}} - \Delta, t^{\mathrm{i}} + \Delta]$. This prevents large jumps in time, increasing the acceptance probability of our perturbation. The optimum choice of Δ is scene dependent, but a general rule of thumb would be to aim for an acceptance rate of roughly 23% for the temporal perturbation on the light paths that are of interest [RGG97].

Second, within this smaller time range, we use a piece-wise constant distribution to give a higher weight to the time intervals where the shutter is open. This reduces the amount of samples which are located in a closed-shutter interval and do not contribute to the image.

One might be tempted to assign zero weight to values of t when the shutter is closed. However, this can cause problems in fast moving scenes with difficult light transport. Here, the acceptance probability for a single 'large' temporal jump over the shutter-closed interval can become vanishingly small. To protect against this case, we assign different sampling probabilities to the open and closed shutter intervals which modulate the temporal sample density such that, on average, 10% of the samples fall in a closed shutter interval (see Figure 6). Effectively, 90% of the samples will then still be used for the final renders and the few samples that lie in between frame exposures are only used to propagate difficult light paths in between frames.

Note that the overall normalisation constant should also be calculated from paths that are distributed according to this same temporal distribution (e.g. Figure 6a). From this set of paths, the initial seed paths for the chains are then importance sampled as is customary [Vea98].

4. Results and Discussion

We implemented our temporal perturbation in Mitsuba [Jak10]. To demonstrate the results of our technique, we used the following scenes:

- **Glass egg scene:** the moving glass egg creates difficult moving specular light paths;
- **Bathroom scene:** a scene containing a large amount of specular and glossy materials.
- **Ajar door scene:** a light source which can only be reached through a slightly ajar door, combined with specular objects make this a particularly challenging scene;

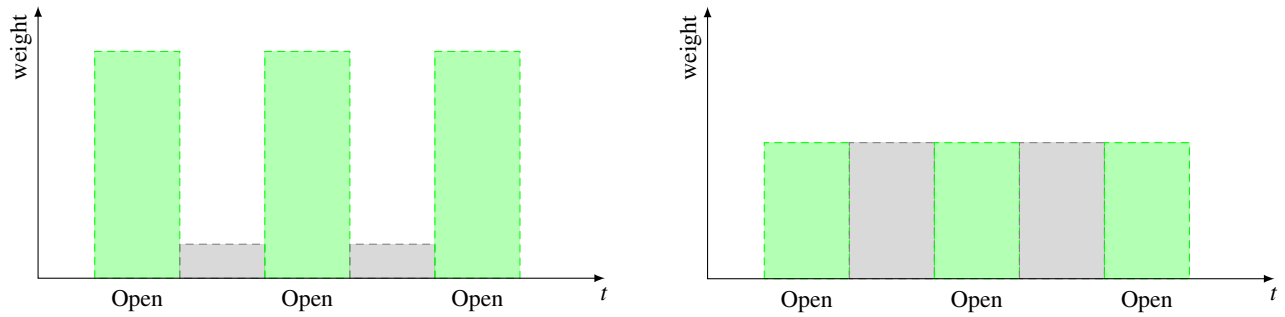
4.1. Moving object perturbation parameters

The reconnection strategy can be initialised either from the camera (forward direction) or from the light source (backward direction), as discussed in Section 3.1. When a path is perturbed using our temporal perturbation, we can choose the forward direction, the backward direction or both directions. When both directions are used, one of them is stochastically chosen to be tried first and only when that first connection strategy fails, the other direction is tried. We have illustrated the differences on the glass egg scene in Figure 7.

From Figure 7, we can see that the forward direction achieves better results than the backward direction. Indeed, the backward strategy can potentially fail to reconnect to the camera when the vertex which should connect to the (pinhole) camera is specular. Nonetheless, using both directions increases the robustness of the algorithm in general, especially when using depth of field cameras for which connections can me made more easily (note: all results that we present here are generated with simple, but more constraining, pinhole cameras).

4.2. Animation Experiments

To evaluate the performance of our technique when rendering animations, we compare it against standard 'frame by frame' MLT and



(a) Piecewise constant distribution with weight ratio 1/10

(c) Uniform distribution

Figure 6: *Volume of samples for the piecewise constant distribution used for our renderings (a), compared to the volume of samples for the uniform distribution (b). Our weighting aims to place 10% of the temporal samples in the region where the shutter is closed (gray), in order to facilitate mutations of paths in between times where the shutter is open (green), without being overly wasteful.*

PSSMLT. The random number stream which is used in PSSMLT also determines the time sample of the light paths. As a result, PSSMLT also includes a form of temporal coherence, which makes it an ideal technique to compare against. We use two test scenarios for the animation experiments: the ajar door scene and the bathroom scene. For both experiments, an animation consisting of 24 frames is rendered, with the render time fixed to one minute per frame (except for the reference).

Ajar door scene The ajar door scene is lit almost exclusively by indirect lighting, as the only light entering the scene is through the door opening. The floor and some of the teapots are specular, making it even harder to reach the light source. This scene has originally been introduced by Eric Veach to showcase Metropolis Light Transport, as MLT makes use of the path space knowledge to retain the connection with the light source. The results of the experiment are visualised in Figure 8. The standard MLT frames still contain a large number of bright spots due to the difficulty of exploring the specular paths through the animation. This is particularly visible where the back wall meets the floor and on the bottom of the left (reflective) teapot. As standard MLT does not exploit temporal coherence, these light paths need to be re-discovered at every point in time. These bright spots also change heavily from frame to frame, which causes flickering in the animation. Our technique noticeably improves upon standard MLT in these regions of high noise. Moreover, our motion blur appears much smoother, as light paths do not have to be re-discovered in every point of time. Although the rightmost (glass) teapot still shows some dark regions due to missing light paths with these low sample count renders, its appearance stays much more consistent in time with our technique than with standard MLT (cfr. the bottom and top of the teapot). PSSMLT is somewhat more capable in correctly capturing the hard specular paths of the glass teapot on the right, yet it has a hard time reconnecting the paths to the light source. Indeed, PSSMLT can easily perturb light paths through specular materials, because the perturbations on the random number stream never invalidate the specular constraints, but it does not have the path space knowledge to re-locate the light source.

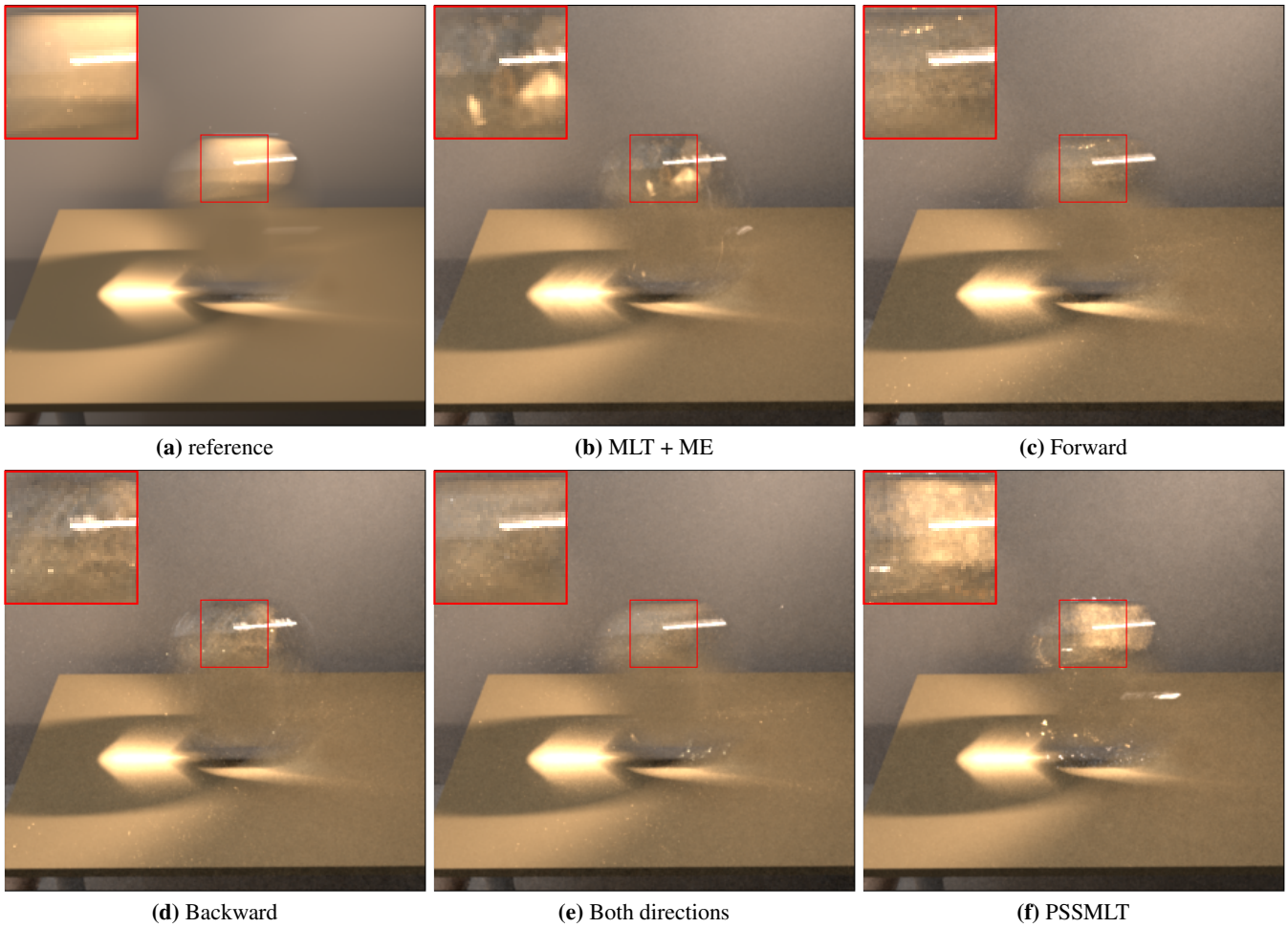
Bathroom scene The bathroom scene contains a myriad of complex specular geometry, such as taps, mirrors, a trash can and a picture frame. The scene is lit by an area light in the back of the room. The results of the experiment are visualised in Figure 9. Standard MLT again has issues with the specular materials through the animation, causing a number of bright spots in the frames. This effect is mostly visible on the taps and picture frame, which have a very blotchy appearance. Our technique performs much better on the specular materials and greatly improves the motion blur in the scene. PSSMLT shows even better results on the taps and picture frame, but at the cost of a slightly increased noise overall. Because the light source 'target' is somewhat bigger in this scene than in the ajar door scene, the additional noise (e.g. on the white diffuse wall) is less pronounced for this bathroom scene compared to the door scene.

Ajar door animation As the appearance of flickering artefacts is best judged from a video, the supplementary material contains an additional animation for the ajar door setup that compares our temporal mutations to the standard MLT algorithm. The improvement with respect to small scale noise is readily apparent. Additionally, pay particular attention to variations in the overall brightness of successive frames, which arises due to an imprecise estimate of the overall normalization constant. Our temporal framework estimates a single overall normalization constant for the full animation, whereas the 'frame by frame' standard MLT algorithm estimates the normalization factor separately for each frame (in this case: for each of the 98 frames, with $1/98$ the number of samples per frame as compared to our single, 'global' normalization constant).

5. Limitations and Future Work

The interval width to generate a new time sample for the temporal perturbation is scene dependent and requires some fine-tuning, which is common for MCMC algorithms. While a good rule of thumb is to set the interval width such that the acceptance rate of the perturbation reaches around 23% [RGG97], it should ideally be adaptive based on the actual difficulty of the path under consideration. Finding an optimal strategy remains future work,

When rendering animations where the shutter is not open during



(a) reference **(b)** MLT + ME **(c)** Forward

(d) Backward **(e)** Both directions **(f)** PSSMLT

Figure 7: *The glass egg scene demonstrating the different moving object perturbation directions. Plate (a) shows the reference, rendered in five hours using MLT with Manifold Exploration and our time perturbation with both directions. The other images are all rendered in three minutes. Plate (b) shows MLT with Manifold Exploration which does not perform time perturbations, and therefore lacks most of the motion blur. Plate (c) shows our algorithm using only time perturbations from the camera to the light source. This results in a smoother image than (b) thanks to the addition of the time perturbation. Plate (d) shows our algorithm using only time perturbations starting from the light source. The results are worse than (c), which demonstrates that more perturbations can be performed from the camera in this scene. Plate (e) shows our algorithm using perturbations from both directions, which is more robust than (c) and (d). Plate (f) shows PSSMLT which, for this scene, is more capable in properly capturing the brightness on the top of the egg. Note that the apparent lack in brightness on the top of the egg in the MLT renders with low numbers of mutations [(b), (c), (d), (e)] is a result of the low number of seed paths and mutations (all algorithms are initialised with the same seed paths). The reference in image (a) was computed with our temporal MLT algorithm and indeed converges to a bidirectionally path traced solution that was rendered as a consistency check, but this bidirectional solution requires significantly more time to converge.*

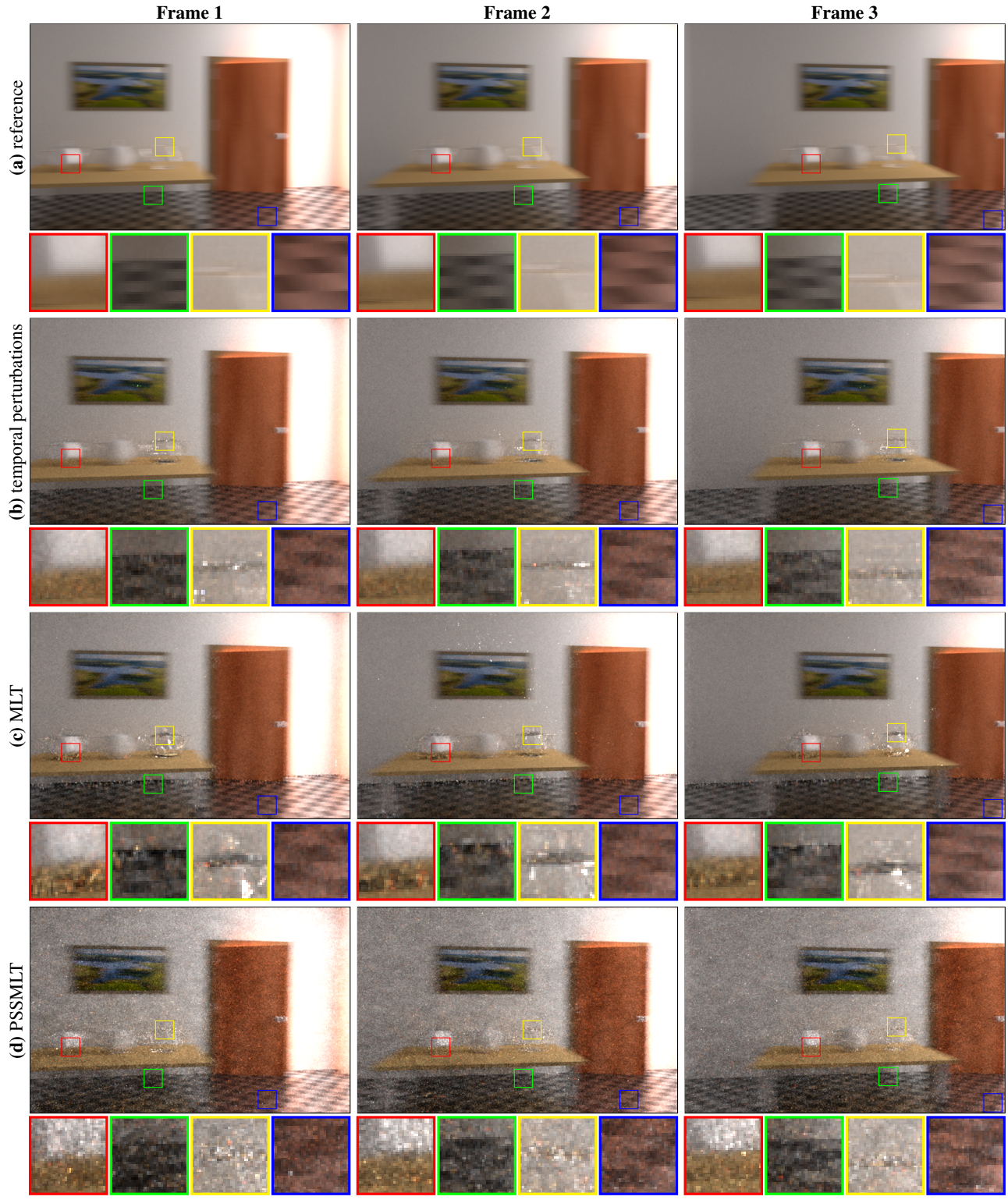


Figure 8: *Equal time comparison of frames 8, 11 and 14 for the ajar door scene with continuously open shutter. Our technique considerably improves upon standard MLT by retaining most of the hard light paths through the animation. PSSMLT is more capable in capturing the specular light transport of the glass teapot but introduces a lot of noise due to the inability of retaining the connection with the light source (cfr. the diffuse white wall).*

Figure 9: *Equal time comparison of frames 6, 12 and 18 of the bathroom scene with a shutter that is open for half of the frame duration. For clarity, only the paths which would be sampled by an MLT renderer are shown (paths traced separately with a specialized direct lighting renderer are ignored, which explains the black appearance of the light source). Our technique manages to retain a lot of the hard specular paths through the animation, making it a considerable improvement over standard MLT. PSSMLT is more suited to capture the specular materials through the animation, but it comes at the cost of slightly increased noise overall, as can be seen on the right wall as a whole (albeit less pronounced then in Figure 8, where the path space knowledge of MLT has a greater advantage).*

the entire frame, a number of samples get lost depending on the weight given to the piecewise constant distribution. While this allows samples to cross between frames, it reduces the convergence speed somewhat since the fraction of lost samples do not contribute to the image. Although we found a loss of 10% to be both sufficiently robust and acceptable, there is certainly room for a better analysis.

Memory can become an issue when rendering long animations, as every frame of the animation has to be kept available. This can be combated by splitting up the animations in shorter sections which are rendered separately. Temporal coherence could possibly be preserved through the section boundaries through some modulated overlap in the spirit of Figure 6, where previous paths in the overlapping part could be shared as new seeds for the next section.

As our technique builds upon path space Metropolis Light Transport, specular transport remains a hard problem to solve. Even though the addition of the temporal perturbation already improves specular transport with regards to standard MLT, our technique remains outclassed by PSSMLT when it comes to this aspect. This is partly because or path space MLT perturbations preserve the original path configuration (length and type of interaction), which is not required by PSSMLT. This constraint could certainly be relaxed in future work.

Because light paths can be obstructed by objects in motion, every edge of the light path needs to re-checked for visibility when performing temporal perturbations. This could be avoided by checking for every edge whether it has intersected with the animated bounding box of any of the moving objects. If it has not intersected with any of them, the edge does not have to be re-checked.

6. Conclusion

We have presented a new animation rendering approach that introduces temporal coherence for Metropolis Light Transport. With our animation rendering framework, we can robustly perturb light paths through the entire animation, even when a direct transition from one frame to a neighbouring frame becomes infeasible for difficult light transport. The handling of motion blur in individual frames is improved, as well as the coherence between the frames in the animation, due to the retention of path space knowledge through time. Moreover, because we only need a single overall normalization constant, frame-to-frame brightness variations are minimized, further adding to the reduction in overall flickering artefacts.

Acknowledgements

We would like to thank Benedict Bitterli for providing the scenes [Bit16]. Roald Frederickx is a predoctoral fellow of the Fund for Scientific Research (FWO) of Flanders. Niels Billen is funded by the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[Bit16] BITTERLI B.: Rendering resources, 2016. URL: https://benedikt-bitterli.me/resources/. 9

[CJ16] CHRISTENSEN P. H., JAROSZ W.: The path to path-traced movies. *Foundations and Trends®in Computer Graphics and Vision 10*, 2 (Oct. 2016), 103–175. doi:10.1561/0600000073. 2

[CTE05] CLINE D., TALBOT J., EGBERT P.: Energy redistribution path tracing. *ACM Transactions on Graphics 24*, 3 (jul 2005), 1186 – 1195. doi:10.1145/1073204.1073330. 2

[Has70] HASTINGS W. K.: Monte carlo sampling methods using markov chains and their applications. *Biometrika 57*, 1 (1970), 97–109. doi:10.1093/biomet/57.1.97. 1

[Jak10] JAKOB W.: Mitsuba renderer, 2010. URL: http://www.mitsuba-renderer.org. 4

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics 31*, 4 (jul 2012), 58:1 – 58:13. doi:10.1145/2185520.2185554. 1

[KHD14] KAPLANYAN A. S., HANIKA J., DACHSBACHER C.: The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics 33*, 4 (jul 2014), 102:1 – 102:13. doi:10.1145/2601097.2601108. 1

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the metropolis light transport algorithm. *Computer Graphics Forum 21*, 3 (2002), 531 – 540. doi:10.1111/1467-8659.t01-1-00703. 1

[LFCD07] LAI Y.-C., FAN S. H., CHENNEY S., DYER C.: Photorealistic image rendering with population monte carlo energy redistribution. In *Rendering Techniques* (2007), Kautz J., Pattanaik S., (Eds.), The Eurographics Association. doi:10.2312/EGWR/EGSR07/287-295. 2

[LLD09] LAI Y.-C., LIU F., DYER C.: *Physically-based Animation Rendering with Markov Chain Monte Carlo*. Tech. Rep. 1653, University of Wisconsin-Madison, 2009. 2

[LLR*15] LI T.-M., LEHTINEN J., RAMAMOORTHI R., JAKOB W., DURAND F.: Anisotropic gaussian mutations for metropolis light transport through hessian-hamiltonian dynamics. *ACM Transactions on Graphics 34*, 6 (oct 2015), 209:1 – 209:13. doi:10.1145/2816795.2818084. 2

[MRR*53] METROPOLIS N., ROSENBLUTH A. W., ROSENBLUTH M. N., TELLER A. H., TELLER E.: Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*, 6 (1953), 1087–1092. doi:10.1063/1.1699114. 1

[RGG97] ROBERTS G. O., GELMAN A., GILKS W. R.: Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability 7*, 1 (February 1997), 110–120. doi:10.1214/aoap/1034625254. 4, 5

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1998. AAI9837162. 3, 4

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 65 – 76. doi:10.1145/258734.258775. 1, 2, 3

[XCS09] XU Q., CHEN H., SBERT M.: Efficient animation rendering based on spatio-temporal coherence. In *2009 First International Conference on Information Science and Engineering* (December 2009), pp. 1223 – 1226. doi:10.1109/ICISE.2009.537. 2