Robust Light Transport Simulation via Metropolised Bidirectional Estimators - Supplemental Document

Martin Šik[1*] Hisanari Otsu[2] Toshiya Hachisuka[2] Jaroslav Křivánek[1]
[1]Charles University in Prague [2]The University of Tokyo

1 Introduction

One purpose of this document is to present implementation details that were omitted from the paper. These include a brief overview on how to compute different path sampling techniques used in our algorithm (Sec. 2) and equations used to mutate samples in the primary sample space (Sec. 3).

The other purpose is to discuss more closely the failure cases, which we encountered during our research on robust and practical combination of Markov chain Monte Carlo (MCMC) and vertex connection and merging/unified path sampling (VCM/UPS) (Sec. 4).

2 Computing path sampling techniques

In this section we give a brief overview on how to compute the pdfs of various path sampling techniques used in our algorithm. Please note that the used path sampling techniques are identical to those from the VCM/UPS algorithm and therefore their pdfs calculation details are also the same. For a more complete reference about the VCM/UPS sampling techniques please refer to other sources [Georgiev et al. 2012; Veach 1997]. We start this short overview by dividing the path sampling techniques into two groups: *connection techniques* and *merging techniques*. While the connection techniques were already used in bidirectional path tracing [Veach and Guibas 1994], merging techniques are specific to VCM/UPS. In the following text we describe how to compute the pdfs of both of these techniques.

2.1 Connection techniques

The connections techniques create a full path by connecting two vertices of a light subpath (subpath that starts on a light source) and an eye subpath (subpath that starts at the camera). Different connection techniques then vary in the length of the connected light subpath and eye subpath (including degenerated subpaths consisting of a single or zero vertices). In the following text we will note (s, t) connection technique as the technique that connects a light subpath of s vertices with and an eye subpath of t vertices, Fig. 1 shows examples of different connection techniques.

Since the connection is deterministic, the probability $p_C^{(s,t)}$ of generating a given full path $\bar{x} = x_0, \ldots, x_k$ using the (s, t) connection technique is a product of pdfs of sampling each subpath

$$p_C^{(s,t)}(\bar{x}) = p(x_0, \ldots, x_s)p(x_t, \ldots, x_k). \tag{1}$$

Probability of sampling a given subpath (light or eye subpath) is then computed as a product of probabilities of sampling each of its vertices given a preceding vertex

$$p(x_0, \ldots, x_s) = p(x_0)p(x_0 \to x_1) \cdots p(x_{s-1} \to x_s). \tag{2}$$

The probability $p(x_n \to x_{n+1})$ then depends on the type of the vertices x_n, x_{n+1} (a vertex on a specular/diffuse surface, a vertex on a light source etc.), for details refer to Veach's doctoral thesis [1997].

*e-mail:sik@cgg.mff.cuni.cz

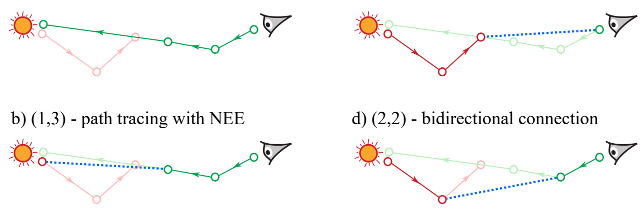

Figure 1: *This figure shows examples of different connection techniques using the same light (red) and eye (green) subpath to construct different full paths. The connection between the last eye vertex and light vertex is shown as a blue dashed line. For each technique we list in the parentheses the number of light and eye subpath vertices they use.*

2.2 Merging techniques

The merging techniques differ from the connection techniques in two ways. First, instead of connecting selected vertices of two subpaths, we instead merge these vertices into one vertex. This means that if the vertices x_s and x_t lie in the vicinity of each other, we assume they are in fact identical and thus the path is completed (see Fig. 2).

The other difference from connection techniques is brute force path reuse, where we try to merge one light subpath with all eye subpaths. Therefore one can generate much more full paths given one light subpath and thus the efficiency of the algorithm is increased. In practice, one then merges a given vertex of one light subpath with all vertices of all eye subpath that lie in its vicinity (usually within a sphere of a given radius).

As with the connection techniques, different merging techniques vary in the length of the used light subpath and eye subpath. The probability $p_M^{(s,t)}$ of sampling a full path $\bar{x}$ using the (s, t) merging technique is then

$$p_M^{(s,t)}(\bar{x}) = p_C^{(s-1,t)}(\bar{x})p(x_{s-1} \to x_s)\pi r^2. \tag{3}$$

Here we use the probability $p_C^{(s-1,t)}$ of sampling $\bar{x}$ using the $(s-1, t)$ connection technique, since it generates most of the vertices in the similar way. r is then a largest distance in which two vertices lie and are still assumed to be identical for the purpose of the merging technique. The derivation of the above equation as well as more details about the merging techniques can be found in the original work on VCM/UPS [Georgiev et al. 2012; Hachisuka et al. 2012].

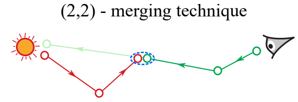

Figure 2: *We can apply the merging technique to construct a full path from two subpaths if two of their vertices are close enough to each other (these are marked by dashed blue ellipse).*

3 Mutating samples in the primary sample space

We base our algorithm on Primary Sample Space Metropolis Light Transport (PSSMLT) [Kelemen et al. 2002]. Our algorithm therefore applies mutations on a vector of random numbers (primary space samples) $u \in \mathcal{U}$, rather than directly on paths. We use each u to generate a light subpath, by applying standard photon tracing technique and using u as input random numbers. The eye subpaths used in our algorithm do not rely on MCMC and are generated using standard Monte Carlo (for more information see Sec. 5 of the main paper).

In order to mutate a primary sample u, we however do not apply the original method used in PSSMLT. Instead we use the adaptive mutation kernel introduced to light transport by Hachisuka et al. [2011]. While we observed our algorithm seems to generate noiseless images faster using the original PSSMLT's mutation kernel, it is also more prone to high sample correlation and the Markov chain also tends to get stuck more often in a local maxima of the target function, which results in fireflies and sometimes even non-uniform convergence. Therefore we have chosen the more robust adaptive mutation kernel.

The adaptive mutation of u_{i+1} given a previous primary sample u_i is defined as follows

$$u_{i+1} = u_i + \begin{cases} (2\xi)^{1/\theta_i} & \text{if } \xi < 0.5 \\ -(2\xi - 1)^{1/\theta_i} & \text{otherwise.} \end{cases} \tag{4}$$

Here ξ is a uniform random number within $[0; 1)$ and θ_i is the adaptation parameter. The adaptation parameter θ_i is updated after u_{i+1} has been accepted or rejected by the MCMC algorithm as follows

$$\theta_{i+1} = \theta_i + \frac{A_{i+1} - A^*}{i}. \tag{5}$$

Here A^* is our goal acceptance ratio (which we set to 0.234) and A_i is the current acceptance ratio computed by dividing the number of accepted mutations by the total number of mutations from the start of the algorithm up until the $(i+1)$-th mutation.

4 Our quest for a robust combination of MCMC and VCM/UPS

While our solution to combining MCMC and VCM/UPS may to some seem quite straightforward, we have in fact reached it after many failed attempts. For those who are interested, we present here how we arrived to the final solution and we briefly mention some of the more promising alternatives we considered along the way.

4.1 Straightforward combination of PSSMLT and VCM/UPS

We began our attempts by searching for the most easy to implement and yet effective combination of MCMC and VCM/UPS. Since PSSMLT is built on top of bidirectional path tracing, which only lacks the merging techniques compared to VCM/UPS, PSSMLT with added merging techniques (extended PSSMLT) seemed as a good starting point. Adding merging to PSSMLT was quite straightforward until we wanted to make it efficient by utilizing brute force path reuse (i.e. one subpath is merged with many others). Enabling brute force path reuse (BFPR) proved to be a major challenge that shaped our whole algorithm.

4.1.1 Generating subpaths for BFPR

In the usual implementation of VCM/UPS, a full set of light subpaths is generated prior to creating any eye subpath in order to enable BFPR. This is however not applicable in PSSMLT, since it uses MCMC to always generate a pair of eye and light subpath (which form full paths using connection techniques). A light subpath cannot be generated by PSSMLT without an eye subpath and vice versa. We solved this problem by using a separate set of light subpaths $\mathcal{L}$ used solely for merging. The MCMC algorithm used in our extended PSSMLT then looks as follows:

1. Propose a pair of eye and light subpath
2. Connect the pair of subpaths using connection techniques
3. Merge vertices of the eye subpath with all light subpaths in the set $\mathcal{L}$
4. Accept/reject the pair of subpaths given the total contribution

To obtain the required light subpath set $\mathcal{L}$, we came up with two different methods:

Method A: First, we run standard PSSMLT and store light subpaths that it generates into $\mathcal{L}$. Later on, when $\mathcal{L}$ is large enough, we can switch to the extended PSSMLT and utilize light subpaths from $\mathcal{L}$ for merging. We further use light subpaths generated by MCMC to update $\mathcal{L}$.

Method B: The second solution is to generate $\mathcal{L}$ by an independent algorithm specialized at light subpath tracing, such as the visibility-driven photon tracing of Hachisuka and Jensen [Hachisuka and Jensen 2011]. The whole algorithm then iterates over two steps: first we generate $\mathcal{L}$ and then we run extended PSSMLT with given number of samples using $\mathcal{L}$ for merging.

Unfortunately, none of the above methods proved to be an effective solution. We discuss problems of each method in the following text.

4.1.2 Method A - computing light subpath probability

In order to compute probability of the merging technique (see Sec. 2), we need to know the probability with which light subpaths in $\mathcal{L}$ have been generated. In the method A we use previous iterations of PSSMLT to generate $\mathcal{L}$, therefore we must compute the probability with which PSSMLT generates a given light subpath. Unfortunately, as we discuss in detail in Sec. 5.4 of the main paper, computing analytically such probability is impossible due to PSSMLT's complicated target function. Implementing extended PSSMLT with method A is therefore infeasible.

4.1.3 Method B - suboptimal MIS weights

In the method B, we used an independent MCMC algorithm with a simple visibility target function to generate $\mathcal{L}$, therefore we were able compute the required probabilities (see Sec. 5.4 of the main paper). While this time we could implement extended PSSMLT, the results were unsatisfactory. We can observe in Fig. 3 that the extended PSSMLT is often less effective than original PSSMLT.

The cause of this is that we used multiple importance sampling (MIS) weights [Veach and Guibas 1995] from VCM/UPS. We completely ignored the fact that merging and connection techniques use light subpaths generated by two different algorithms (visibility driven MCMC generates $\mathcal{L}$, while PSSMLT generates light subpaths for connections). In order to optimally combine the techniques we needed to project the probabilities with which these MCMC algorithms generated light subpaths into MIS weights. However, we already know that for light subpaths generated by

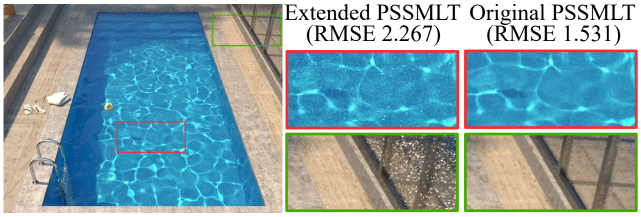

Figure 3: *Equal-time comparison of original PSSMLT and our extended version that uses visibility driven MCMC to generate light subpaths for merging. Due to the suboptimal combination of sampling techniques, our extended version yields worse results even for areas dominated by specular-diffuse-specular paths (these are usually handled best by the merging techniques).*

PSSMLT we can not compute the required probabilities. Thus we were forced to abandon extended PSSMLT and use a different approach.

4.2 Splitting generation of subpaths

First, to solve the problems of the method B, we decided to split the generation of light subpaths and eye subpaths into two independent MCMC algorithms. This way, we could use the same set of light subpaths for both merging and connection techniques. Furthermore, to effectively compute subpath probability we only used a visibility target function. The proposed algorithm worked as follows:

1. Generate eye subpaths using visibility-driven MCMC
2. Generate light subpaths using visibility-driven MCMC
3. Combine the two sets of subpaths using VCM/UPS techniques
4. Iterate the above steps until convergence

While we solved the problems of the extended PSSMLT, we were facing a new problem, the definition of visibility for eye subpaths.

4.2.1 Visibility definition

Original visibility-driven photon tracing accepts a proposed light subpath only if it contributes to the image (i.e. if it can be merged with or connected to any eye subpath). We applied this for generation of our light subpaths as well. We also wanted to use MCMC to generate only eye subpaths that contribute to the image. Unfortunately, when the algorithm above generates eye subpaths, it has not yet generated light subpaths and therefore we could not estimate the contribution to the image.

We solved this problem by defining the visibility for eye subpaths using *old* light subpaths generated in the previous iterations. More specifically, an eye subpath is visible if any of its vertices lie in the vicinity of any vertex from old light subpaths (i.e. it can be merged with any light vertex). While this approach worked, the eye subpaths were often generated in places where they could not reach any light subpath from the current iteration. Due to this fact, the resulting algorithm often suffered from non-uniform convergence.

4.3 Final algorithm

Since the MCMC used to generate eye subpaths was a source of problems, we decided to replace it with a simple stratified Monte Carlo generator. Sec. 4.3 of the main paper discusses that this decision has brought many advantages and resulted in a more effective algorithm. At this point our algorithm was already quite robust and performed better than state of the art algorithms on our test scenes. However, we wanted to push it even further.

We noticed that for light subpaths the visibility target function is not always optimal, since light subpaths are often distributed in visible areas with low contribution to the image. Therefore, we experimented with different target functions, including the original PSSMLT contribution target function. Even though, while using MCMC with the contribution target function, we had to settle with less optimal MIS weights (see Sec. 5.4 of the main paper), the results were often better than with the visibility target function. To utilize advantages of both target functions, we have combined them using replica exchange. Replica exchange was then the final piece of our new robust algorithm that combines the strengths of MCMC and VCM/UPS.

References

GEORGIEV, I., KŘIVÁNEK, J., DAVIDOVIČ, T., AND SLUSALLEK, P. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph. (SIGGRAPH Asia '12) 31*, 6.

HACHISUKA, T., AND JENSEN, H. W. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph. 30*, 5.

HACHISUKA, T., PANTALEONI, J., AND JENSEN, H. W. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph. (SIGGRAPH Asia '12) 31*, 6.

KELEMEN, C., SZIRMAY-KALOS, L., ANTAL, G., AND CSONKA, F. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Comp. Graph. Forum (Eurographics 2002) 21*, 3, 531–540.

VEACH, E., AND GUIBAS, L. 1994. Bidirectional estimators for light transport. In *Proc. Eurographics Rendering Workshop*.

VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH '95*.

VEACH, E. 1997. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University.