Survey of Markov Chain Monte Carlo Methods in Light Transport Simulation: Supplemental Material

Martin Šik and Jaroslav Křivánek

1 Introduction

In this supplemental material we discuss how to break detailed balance condition (see Sec. 2.3 in the main text) and still ensure that MCMC algorithm converges (Sec. 2). We also discuss details of some algorithms that did not fit into the main text, more specifically the general Hamiltonian Monte Carlo [1] (Sec. 3) and general Multiple-try Metropolis [2] (Sec. 4). In Sec. 5 we provide our personal recommendations as to which algorithm converges faster in which scene.

Besides this, we list (Tab. 1 and Tab. 2) all the light transport algorithms discussed in the main text, while pointing out their main features. For readers interested in testing different MCMC algorithms themselves, we have compiled a list of publicly available implementations (Tab. 3).

2 Breaking detailed balance condition

While the detailed balance condition (Eq. (8) in the main text) ensures that a Markov chain is invariant to a given distribution, the condition is not necessary [3], [35]. It is sufficient to satisfy a less strict *global* balance condition, which states that for all $u, v \in \mathcal{U}$:

$$\int_{\mathcal{U}} \pi(u)q(u \to v)\alpha(u \to v)dv = \int_{\mathcal{U}} \pi(v)q(v \to u)\alpha(v \to u)dv \tag{1}$$

The above equation can be translated as how much energy (i.e. target function value) flows out of any $u \in \mathcal{U}$ must also flow back from all $v \in \mathcal{U}$. It is straightforward to see that the detailed balance (Eq. (8) in the main text) implies the above balance condition.

Breaking detailed balance may potentially result in more efficient mutations [3], [4], however it is then also more challenging to ensure Eq. (1) holds. Most methods therefore rather ensure that the detailed balance holds.

• *Martin Šik and Jaroslav Křivánek are with the Charles University, Prague, Czech Republic*
E-mail: sik@cgg.mff.cuni.cz, jaroslav.krivanek@mff.cuni.cz

3 Hamiltonian Monte Carlo

In this section we discuss *Hamiltonian Monte Carlo* (HMC) [1] in more detail. We first describe the general HMC algorithm and then we describe the details of *Hessian-Hamiltonian Monte Carlo* light transport algorithm [5] that were left out from the main text.

3.1 General Hamiltonian Monte Carlo

HMC generates a new proposal by simulating Hamiltonian dynamics over the landscape of the graph of the target function. Hamiltonian dynamics is a system of differential equations defined on Hamiltonian energy $E(u, m)$. Here u represents the position of a point mass and HMC interprets it as the Markov Chain state, while m is the mass momentum. The Hamiltonian energy is the sum of potential energy (gravity) $E_U(u)$ and kinetic energy $E_K(m)$

$$E(u, m) = E_U(u) + E_K(m). \tag{2}$$

In HMC, the potential energy is defined using the target function π as

$$E_U(u) = -\log(\pi(u)). \tag{3}$$

The negative sign ensures that the point mass will be pulled by the gravity towards peaks of the target function π, while $\log$ better captures the high dynamic range of π and allows for simpler derivatives of E_U. The kinetic energy is equal to $\frac{1}{2}mAm^T$, where the matrix A transforms the shape of the HMC proposal distribution (A should be optimally set to the covariance of the target function [6]).

To generate a proposal using HMC, we first consider the current state as the position of a point mass and randomly assign to it a momentum m from a Gaussian distribution $\exp(-E_K(m))$ (note that $E_K(m)$ is quadratic in m, hence the previous formula is a Gaussian). Hamiltonian dynamics are then simulated for a fixed time and a new position v and momentum l is obtained. Note that simulating Hamiltonian dynamics for arbitrary target function usually requires time consuming numerical integration. The new state equal to the new position v is then accepted with the probability

$$\alpha(u \to v) = \left(\frac{\pi(v)\exp(-E_K(l))}{\pi(u)\exp(-E_K(m))}, 1\right). \tag{4}$$

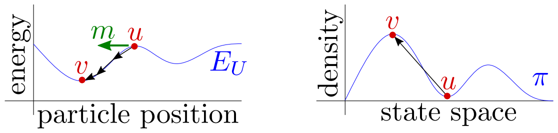

Fig. 1: In Hamiltonian Monte Carlo (Sec. 3), the potential energy E_U (left) is equal to negative logarithm of the target function π (right). Given a current position (state) u and a randomly sampled momentum m of a point mass, Hamiltonian dynamics are simulated and a new position (proposal) v is obtained. The gravity defined by E_U pulls the point mass downwards, thus the proposal v is pulled towards the peaks of the target function.

Here we have just plugged $\exp(-E_K(m))$ as the mutation kernel into the acceptance probability (Eq. (2) in the main text). Now from Eq. (2) and Eq. (3) we directly see that $\pi(u)\exp(-E_K(m)) = \exp(-E(u,m))$ and thus if Hamiltonian dynamics are simulated accurately, the energy remains constant and the proposed v will be always accepted. Due to the potential energy definition in Eq. (3), Hamiltonian dynamics will create more proposals at places with higher target function value (Fig. 1) and thus effectively explore local neighborhood of a current sample. For a more comprehensive description of HMC we refer readers to other sources [6].

3.2 Approximating target function in Hessian-Hamiltonian Monte Carlo

As described in the main text (Sec. 5.3.2) *Hessian-Hamiltonian Monte Carlo* (H^{2}MC) by Li et al. [5] does not apply HMC directly to avoid the costly integration, which is required to simulate Hamiltonian dynamics. Instead H^{2}MC locally approximates the target function using a Taylor series

$$\log(\pi(v)) \approx \frac{1}{2}v^T Hv + Gv + \log(\pi(u)). \tag{5}$$

Here u is the current state and v is a proposal. The Hessian matrix H and the gradient vector G of $\log(\pi(u))$ are computed using automatic differentiation [7]. This approximation allows Hamiltonian dynamics to be solved analytically.

The authors show that the analytical solution of Hamiltonian dynamics for a new position is actually a linear mapping from momentum m sampled from a Gaussian distribution. Thus a new position (new proposal) is also distributed according to a Gaussian distribution (Fig. 2).

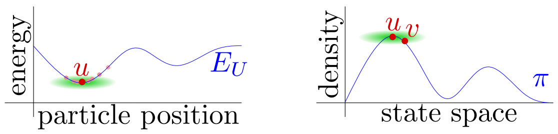

Fig. 2: In Hessian-Hamiltonian Monte Carlo (H^{2}MC, Sec. 3.2), the positions generated using Hamiltonian dynamics (left) from the same position u with a momentum sampled from a Gaussian distribution are also distributed according to a Gaussian distribution (green). This distribution is used to generate proposals v in H^{2}MC (right).

```
1:  Select $u_0$
2:  for $i = 0$ to the number of samples do
3:      Generate $k$ proposals $V = \{v_1, \ldots, v_k\}$
4:      Sample a proposal $v \in V$ according to $\hat{w}(u_i \to v)$
5:      Mutate $v$ to $k-1$ competitors $u_1^*, \ldots, u_{k-1}^*$
6:      Set: $U^* = \{u_1^*, \ldots, u_{k-1}^*, u_k^* = u_i\}$
7:      Compute acceptance probability $\hat{\alpha}(U^* \to V)$
8:      Generate random number $\xi \sim \mathrm{U}(0,1)$
9:      if $\hat{\alpha}(U^* \to V) > \xi$ then
10:         Accept proposal: set $u_{i+1} = v$
11:     else
12:         Reject proposal: set $u_{i+1} = u_i$
```

Fig. 3: Mutliple-try Metropolis algorithm.

4 Multiple-try Metropolis

In this section we describe the general Multiple-try Metropolis (MTM) [2] algorithm in more detail. Note that unlike standard Metropolis-Hastings (MH) [8], the Multiple-try Metropolis algorithm tries several proposals before accepting or rejecting a single one of them. The selected proposal should have theoretically higher chance of acceptance compared to MH and thus should decrease the chance of the chain getting stuck in a local maximum.

As in the standard Metropolis-Hastings algorithm, all proposals v are sampled using a mutation Q with the conditional probability $q(u \to v)$. Given a target function π we define the proposal importance weight $\hat{w}$

$$\hat{w}(u \to v) = \pi(v)q(v \to u)\hat{\lambda}(v \leftrightarrow u), \tag{6}$$

where $\hat{\lambda}(v \leftrightarrow u)$ is a non-negative symmetric function, which can be chosen by the user. The only requirement is that $\hat{\lambda}(v \leftrightarrow u) > 0$ whenever $q(v \to u) > 0$. $\hat{\lambda}$ influences the probability of accepting a proposed state, however the authors show that its impact is minimal.

The pseudo-code of MTM is outlined in Fig. 3. The algorithm starts as MH with an initial sample (line 1) from which it generates a Markov chain. Given a sample u_i the algorithm generates k proposals $V = \{v_1, \ldots, v_k\}$ with probability $q(u_i \to v_j)$ (line 3). The final proposal v is then sampled from V with probability proportional to $\hat{w}(u_i \to v)$ (line 4). Next, from the proposal v we sample $k-1$ *competitors* $u_1^*, \ldots, u_{k-1}^*$ with probability $q(v \to u_j^*)$ (line 5). We complete the set of competitors U^* with the current sample $u_k^* = u_i$ (line 6). The acceptance probability with which we will accept the proposal (line 7) is defined as

$$\hat{\alpha}(U^* \to V) = \min\left(1, \frac{\hat{w}(u_i \to v_1) + \ldots + \hat{w}(u_i \to v_k)}{\hat{w}(v \to u_1^*) + \ldots + \hat{w}(v \to u_k^*)}\right) \tag{7}$$

Then the proposal is either accepted (line 10) or rejected (line 12) as in standard MH.

From the algorithm description one can immediately observe that for generating a single sample from the Markov chain, $2k-1$ tentative samples must be generated. Thus the better distribution of samples is payed for by far more expensive sample generation. MTM is therefore mainly useful in applications, where generating fewer but better distributed samples is more advantageous.

5 Recommendations

In this section we briefly discuss which of the existing algorithms is better at handling a given scene. Note that this discussion is based purely on our personal experience when using these algorithms and not on a thorough analysis. We discuss here only the algorithms that generate full light transport and thus we do not include here algorithms based on stochastic progressive photon mapping (they do not generate pure specular/highly glossy paths).

The original Metropolis Light Transport (MLT, see Sec. 3.3 in the main paper) is, according to our own experience, the best choice when one wants to render a scene with only diffuse materials. MLT can generate paths very quickly and due to absence of glossy/specular transport it can easily sample all paths and efficiently reconnect mutated sub-path to the existing paths. MLT also handles well scenes with difficult visibility (where light has to bounce several times before reaching the camera).

For scenes that contain difficult specular/highly glossy transport and where this difficult transport is not separated to too many individual peaks (i.e. there is not a problem with global exploration), we recommend using either MLT with mutations from both Manifold Exploration Light Transport (MELT, see Sec. 5.2 in the main paper) and Half-vector Space Light Transport (HLST, see Sec. 5.2 in the main paper) or using Hessian-Hamiltonian Monte Carlo (H^2MC see Sec. 5.3 in the main paper). While H^2MC has better global exploration due to its inherent importance sampling (H^2MC uses the primary sample space), MELT/HSLT can better utilize geometry features of the scene and also does not require computation of target function differentials (and thus it maybe more practical to implement in a production renderer).

For scenes that feature specular/highly glossy transport separated to many individual peaks (and/or contain difficult visibility), we recommend using Metropolised Bidirectional Estimators (MBE, see Sec. 10.2 in the main paper). While MBE seems to converge rather slowly compared to algorithms like HSLT, it discovers all the peaks due to its better global exploration and is able to efficiently sample them due to its use of brute-force path reusal with spatial regularization. However, MBE is also very costly in terms of memory (it requires storing all camera subpaths in each iteration). In the case when this is an issue we recommend using algorithms that can utilize both primary sample space and path space mutations (see Sec. 4.3 in the main paper) while enhancing these algorithm with Replica exchange to achieve better global exploration.

For a very special case of a scene, which contains mainly pure specular paths, the best solution seems to be using just Primary sample Space MLT (PSSMLT, see Sec. 4.1 in the main paper) with a single mapping which corresponds to unidirectional path tracing. While this algorithm is simple, it can generate paths very fast and thus handle such scenes by brute-force.

References

[1] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters B*, vol. 195, no. 2, pp. 216 – 222, 1987.

[2] J. S. Liu, F. Liang, and W. H. Wong, "The Multiple-Try method and local optimization in Metropolis sampling," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 121–134, 2000.

[3] V. I. Manousiouthakis and M. W. Deem, "Strict detailed balance is unnecessary in Monte Carlo simulation," *The Journal of Chemical Physics*, vol. 110, no. 6, pp. 2753–2756, 1999.

[4] H. Suwa and S. Todo, "Markov chain Monte Carlo method without detailed balance," *Phys. Rev. Lett.*, vol. 105, p. 120603, Sep 2010.

[5] T.-M. Li, J. Lehtinen, R. Ramamoorthi, W. Jakob, and F. Durand, "Anisotropic Gaussian mutations for Metropolis light transport through Hessian-Hamiltonian dynamics," *ACM Trans. Graph. (SIGGRAPH Asia 2015)*, vol. 34, no. 6, 2015.

[6] R. M. Neal, "MCMC using Hamiltonian dynamics," *ArXiv e-prints*, Jun. 2012.

[7] D. Piponi, "Automatic differentiation, C++ templates, and photogrammetry," *Journal of Graphics Tools*, vol. 9, no. 4, 2004.

[8] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, 1970.

[9] E. Veach and L. J. Guibas, "Metropolis light transport," in *SIGGRAPH '97*, 1997.

[10] C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka, "A simple and robust mutation strategy for the Metropolis light transport algorithm," *Comp. Graph. Forum (Eurographics)*, vol. 21, no. 3, 2002.

[11] T. Hachisuka, A. S. Kaplanyan, and C. Dachsbacher, "Multiplexed Metropolis light transport," *ACM Trans. Graph.*, vol. 33, no. 4, 2014.

[12] H. Otsu, A. S. Kaplanyan, J. Hanika, C. Dachsbacher, and T. Hachisuka, "Fusing state spaces for Markov chain Monte Carlo rendering," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017.

[13] J. Pantaleoni, "Charted Metropolis light transport," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 75:1–75:14, Jul. 2017.

[14] B. Bitterli, W. Jakob, J. Novák, and W. Jarosz, "Reversible jump Metropolis light transport using inverse mappings," *ACM Trans. Graph.*, vol. 37, no. 1, pp. 1:1–1:12, Oct. 2017.

[15] W. Jakob and S. Marschner, "Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport," *ACM Trans. Graph.*, vol. 31, no. 4, 2012.

[16] A. S. Kaplanyan, J. Hanika, and C. Dachsbacher, "The natural-constraint representation of the path space for efficient light transport simulation," *ACM Transactions on Graphics*, vol. 33, no. 4, 2014.

[17] J. Hanika, A. Kaplanyan, and C. Dachsbacher, "Improved half vector space light transport," *Comp. Graph. Forum*, vol. 34, no. 4, 2015.

[18] T. Hachisuka and H. W. Jensen, "Robust adaptive photon tracing using photon path visibility," *ACM Tr. Graph.*, vol. 30, no. 5, 2011.

[19] S. Kitaoka, Y. Kitamura, and F. Kishino, "Replica exchange light transport," *Computer Graphics Forum*, vol. 28, no. 8, 2009.

[20] A. S. Kaplanyan and C. Dachsbacher, "Path space regularization for holistic and robust light transport," *Comput. Graph. Forum (Eurographics 2013)*, vol. 32, no. 2, pp. 63–72, 2013.

[21] M. Šik and J. Křivánek, "Improving global exploration of MCMC light transport simulation," in *ACM SIGGRAPH 2016 Posters*, 2016.

[22] B. Segovia, J.-C. Iehl, and B. Péroche, "Metropolis instant radiosity," *Comput. Graph. Forum (Eurographics 2007)*, vol. 26, no. 3, 2007.

[23] A. Keller, "Instant Radiosity," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997.

[24] B. Segovia, J.-C. Iehl, and B. Proche, "Coherent Metropolis light transport with Multiple-Try mutations," LIRIS UMR 5205 CNRS/INSA de Lyon, Tech. Rep. RR-LIRIS-2007-015, Apr. 2007.

[25] E. Veach, "Robust Monte Carlo methods for light transport simulation," Ph.D. dissertation, Stanford University, 1997.

[26] J. Hoberock and J. C. Hart, "Arbitrary importance functions for Metropolis light transport." *Com. Graph. Forum*, vol. 29, no. 6, 2010.

[27] J. Lehtinen, T. Karras, S. Laine, M. Aittala, F. Durand, and T. Aila, "Gradient-domain Metropolis light transport," *ACM Trans. Graph. (SIGGRAPH 2013)*, vol. 32, no. 4, 2013.

[28] M. Manzi, F. Rousselle, M. Kettunen, J. Lehtinen, and M. Zwicker, "Improved sampling for gradient-domain Metropolis light transport," *ACM Trans. Graph.*, vol. 33, no. 6, Nov. 2014.

[29] T. Hachisuka and H. W. Jensen, "Stochastic progressive photon mapping," *ACM Trans. Graph. (SIGGRAPH Asia 2009)*, vol. 28, no. 5, 2009.

[30] S. Fan, S. Chenney, and Y.-c. Lai, "Metropolis photon sampling with optional user guidance," in *Eurographics Symposium on Rendering (EGSR '05)*, 2005, pp. 127–138.

[31] J. Chen, B. Wang, and J.-H. Yong, "Improved stochastic progressive photon mapping with Metropolis sampling," *Comput. Graph. Forum (EGSR 2011)*, vol. 30, no. 4, 2011.

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
MLT [9]	3.1	N/A	Path space	Metropolis-Hastings	Introduced MCMC to light transport

Simplified state space: simplifying the algorithm, improving local exploitation/global exploration via importance sampling

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
PSSMLT [10]	4.1	N/A	Primary sample space	Metropolis-Hastings	Introduced primary sample space
MMLT [11]	4.2	PSSMLT	Primary sample space	Metropolis-Hastings	Improves PSSMLT by sampling one mapping to the path space.
Fusing state spaces [12] [13] [14]	4.3	MMLT	Primary sample space + Path space	Metropolis-Hastings	Introduced framework for using mutations from both spaces

Differential methods: improving local exploitation by utilizing geometry/target function differentials

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
MELT [15]	5.1	MLT	Path space	Metropolis-Hastings	Efficiently samples specular manifolds
HSLT [16]	5.2	MLT	Path space + Natural constraints space	Metropolis-Hastings	Introduced the natural constraints space that allows for efficient mutation of glossy/specular paths
Improved HSLT [17]	5.2.3	HSLT	Path space + Natural constraints space	Metropolis-Hastings	Improves the efficiency of the HSLT mutation
H^{2}MC [5]	5.3	MMLT	Primary sample space	Metropolis-Hastings	Applies ideas from Hamiltonian Monte Carlo to compute Gaussian proposal distribution for superior local exploitation.

Adaptive Markov chain Monte Carlo: allows for adaptive tuning of mutation parameters

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
Robust Adaptive Photon Tracing [18]	6.2	PSSMLT	Primary sample space	Replica exchange + Adaptivity	Introduced adaptive MCMC to light transport, improves stochastic progressive photon sampling by generating photons based on visibility target function

Tempering and Replica exchange: improves global exploration by applying Replica exchange algorithm or tempering/regularization

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
RELT [19]	7.3	PSSMLT	Primary sample space	Replica exchange	Introduced Replica exchange to light transport, different chains use different mappings to the path space
Path space regularization [20]	7.4	N/A	Any	Any	Discusses the idea of path space regularization that enables sampling of previously impossible to sample paths
Tempering of path contribution [21]	7.5	MMLT	Primary sample space	Replica exchange + Adaptivity	Combines the idea of regularization with Replica exchange, introduces new replica exchange moves

Multiple-try Metropolis: improves global exploration by applying Multiple-try Metropolis

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
MIR [22]	8.1	MLT	Path space	Multiple-try Metropolis	Introduced Multiple-try Metropolis to light transport, uses it to generate good distribution of virtual point lights (for instant radiosity [23])
Coherent MLT [24]	8.2	MLT	Path space	Multiple-try Metropolis	Does not improve global exploration, uses Multiple-try Metropolis to generate coherent paths in parallel

Modifying target function for uniform image error I: algorithms with full light transport

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
Two-stage MLT [25]	9.1	MLT	Path space	Metropolis-Hastings	Introduced the concept of modifying target function to achieve uniform image error
Multi-stage MLT [26]	9.1.1	MLT	Path space	Metropolis-Hastings	Improves two-stage MLT by using more stages and perceptual error metric
GDMLT [27]	9.2	MELT	Path space	Metropolis-Hastings	Apart from the main image, GDMLT samples also difference images (of neighboring paths contributions) and constructs the final image using Poisson reconstruction
Improved GDMLT [28]	9.2.3	GDMLT	Path space	Metropolis-Hastings	Improves global exploration of GDMLT by reducing the target function variation

TABLE 1: A list of MCMC algorithms used in light transport simulation (continues on the next page).

Modifying target function for uniform image error II: algorithms based on stochastic progressive photon mapping [29]

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
Metropolis photon sampling [30]	9.3	MLT	Path space	Metropolis-Hastings	Uses vertices of paths generated by MLT as photons
Improved SPPM [31]	9.3	PSSMLT	Primary sample space	Metropolis-Hastings	Target function is based on the approximate density of photons contributing to each image pixel.
Visual importance SPPM [32]	9.3	PSSMLT	Primary sample space	Replica exchange	Utilizes precomputed visual importance [33] as a target function to guide the photons.
Spatial Target Function [34]	9.3.1	PSSMLT	Primary sample space	Replica exchange + Adaptivity	Derives a target function that achieves uniform error (under simplifying conditions) based on probability of a uniformly generated photon contributing to the image pixel.

Improving stratification: ensuring better global exploration by improving stratification of MCMC samples

Name or acronym	Section	Based on	State space	MCMC algorithm	Description
ERPT [35]	10.1	MLT	Path space	Metropolis-Hastings	Runs several independent chains (each initialized by stratified path tracer) per each pixel using MLT local perturbations
MBE [36]	10.2	PSSMLT	Primary sample space	Replica exchange + Adaptivity	Combines camera subpaths generated by stratified path tracer with light subpaths generated by MCMC, while utilizing all path sampling techniques from vertex connection and merging [37], [38]

TABLE 2: A list of MCMC algorithms used in light transport simulation (starts on the previous page).

Algorithms	Renderer	Author	Url
MLT, PSSMLT, ERPT, MELT, HSLT	Mitsuba	W. Jakob	https://www.mitsuba-renderer.org
HSLT	Mitsuba	A. S. Kaplanyan, J. Hanika	https://www.mitsuba-renderer.org
H^2MC	dpt	T.-M. Li	https://github.com/BachiLi/dpt
Fusing state spaces, MMLT, PSSMLT, MLT, MELT	Lightmetrica	H. Otsu	http://lightmetrica.org
RELT	xyz renderer	S. Kitaoka	https://github.com/skitaoka/xyz
Spatial target function, Improved SPPM, Robust Adaptive Photon Tracing, Two-stage/Multi-stage MLT	Mitsuba (branch)	A. Gruson, M. Šik, J. Vorba	https://github.com/beltegeuse/spatialTF_code
MBE	Mitsuba (branch)	M. Šik	http://cgg.mff.cuni.cz/~sik/meb/index.html

TABLE 3: A list of publicly available implementations of MCMC algorithms that are known to us.

[32] Q. Zheng and C.-W. Zheng, "Visual importance-based adaptive photon tracing," *The Visual Computer*, vol. 31, no. 6, 2015.

[33] P. H. Christensen, "Adjoints and importance in rendering: An overview," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 3, 2003.

[34] A. Gruson, M. Ribardière, M. Šik, J. Vorba, R. Cozot, K. Bouatouch, and J. Křivánek, "A spatial target function for Metropolis photon tracing," *ACM Trans. Graph.*, vol. 36, no. 1, pp. 4:1–4:13, Nov. 2016.

[35] D. Cline, J. Talbot, and P. Egbert, "Energy redistribution path tracing," *ACM Trans. Graph. (SIGGRAPH 2005)*, vol. 24, no. 3, 2005.

[36] M. Šik, H. Otsu, T. Hachisuka, and J. Křivánek, "Robust light transport simulation via Metropolised bidirectional estimators," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 245:1–245:12, Nov. 2016.

[37] I. Georgiev, J. Křivánek, T. Davidovič, and P. Slusallek, "Light transport simulation with vertex connection and merging," *ACM Trans. Graph. (SIGGRAPH Asia '12)*, vol. 31, no. 6, 2012.

[38] T. Hachisuka, J. Pantaleoni, and H. W. Jensen, "A path space extension for robust light transport simulation," *ACM Trans. Graph. (SIGGRAPH Asia '12)*, vol. 31, no. 6, 2012.