DOI: 10.1111/cgf.12679
Eurographics Symposium on Rendering 2015
J. Lehtinen and D. Nowrouzezahrai
(Guest Editors)

Volume 34 (2015), Number 4

Improved Half Vector Space Light Transport

Johannes Hanika, Anton Kaplanyan and Carsten Dachsbacher

Karlsruhe Institute of Technology

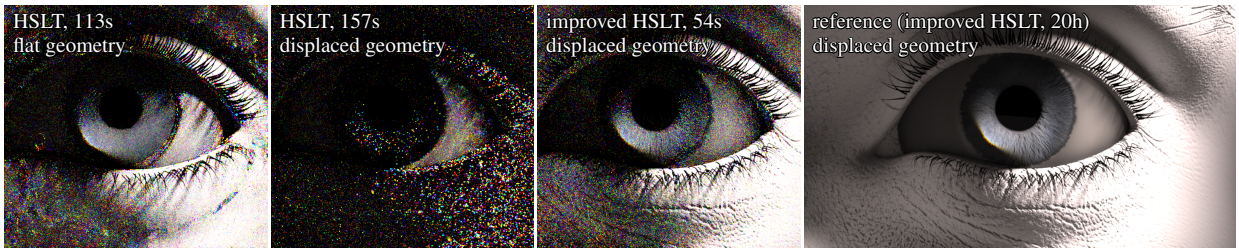


Figure 1: Images computed with half vector space light transport (HSLT) (64 samples per pixel, spectral renderer). Left: original HSLT [KHD14] on smooth geometry (113 sec) and with displacement textures switched on (157 sec, center left). Center right: our new *breakup mutation strategy* is more robust to fine displacements and faster (54 sec). Both original and improved HSLT use our new ray differentials which provide a more uniform stratification on the image. Right: reference (20 hours).

Abstract

In this paper, we present improvements to half vector space light transport (HSLT) [KHD14], which make this approach more practical, robust for difficult input geometry, and faster. Our first contribution is the computation of half vector space ray differentials in a different domain than the original work. This enables a more uniform stratification over the image plane during Markov chain exploration. Furthermore, we introduce a new multi chain perturbation in half vector space, which, if combined appropriately with half vector perturbation, makes the mutation strategy both more robust to geometric configurations with fine displacements and faster due to reduced number of ray casts. We provide and analyze the results of improved HSLT and discuss possible applications of our new half vector ray differentials.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Physically based simulation of light transport is one of the core ingredients for rendering photorealistic images. The striving of researchers for the long sought-for goal of developing an efficient and robust simulation method has led to considerable progress in recent years. Nevertheless, many challenges in computing light transport still remain, e.g. when complex geometry, materials, and illumination meet.

Modern simulation methods build on the path integral formulation of light transport [Vea98] and numerical integration and sampling techniques, such as (Markov chain) Monte Carlo methods, to stochastically create paths connecting the sensor to light sources. The path sampling strategy is crucial for the efficiency. Veach and Guibas [VG97] were the first to apply Markov chain Monte Carlo (MCMC) sampling. Their Metropolis light transport (MLT) method mutates paths, sampling the path space proportional to the paths' contributions to the image. MLT spawned further work, which led to a variety of MCMC methods, including methods working in primary sample space [KSKAC02] and, presented recently, half vector space light transport (HSLT) [KHD14]. HSLT represents the interactions along a path by the halfway vectors, which has benefits for sampling paths (by perturbing half vectors) within sub-spaces of the path space.

This paper presents improvements to HSLT making this approach more practical, more robust in the presence of complex, i.e. finely tessellated and displaced, geometry, and also improves its performance. Note that, as in the original HSLT, we focus on inter-surface light transport and do not consider participating media. Our main contributions are:

- A new formulation of half vector space ray differentials, which provides uniform stratification on the image plane even for paths with extreme angular configurations.
- A new multi chain perturbation in half vector space which is simpler and faster than the previous half vector perturbation.
- A "breakup" strategy to combine multi chain and half vector perturbation on a single path, in the same spirit as combining lens perturbation and manifold walks [Jak13]. In contrast to Jakob [Jak13], our strategy does not depend on a classification in "diffuse" and "specular" interactions which is known to be problematic [KD13].

After discussing previous work in Sec. 2, we introduce our new ray differentials (Sec. 3) and the new mutation strategy for half vector light transport (Sec. 4). In Sec. 5 we describe how to combine half vector perturbation and the new strategy to jointly work on the same path. We present and analyze results (Sec. 6 and 7) and discuss limitations (Sec. 8).

2. Background and Previous Work

Notation Our work extends half vector space light transport (HSLT) [KHD14], and thus we keep the notation as similar as possible and only recapitulate the most important aspects. We denote transport paths as $\mathbf{X}$; they connect a vertex on a light source $\mathbf{x}_0$ with a sensor vertex $\mathbf{x}_k$. Every inner vertex $\mathbf{x}_i$ has an incoming edge direction $\mathbf{i}_i$ pointing towards the previous vertex $\mathbf{x}_{i-1}$ (closer to the light) and an outgoing edge $\mathbf{o}_i$ towards $\mathbf{x}_{i+1}$. The half vector $\mathbf{h}_i$ bisects $\mathbf{i}_i$ and $\mathbf{o}_i$.

Metropolis Light Transport (MLT) MLT samples transport paths by choosing a tentative proposal path $\mathbf{X}^t$ using perturbations of the current path $\mathbf{X}^i$. The next state of the Markov chain $\mathbf{X}^{i+1}$ advances to either $\mathbf{X}^t$ or $\mathbf{X}^i$ with the acceptance probability [MRR*53, Has70]

$$a = \min\left\{1, \frac{f(\mathbf{X}^t)/T(\mathbf{X}^i \to \mathbf{X}^t)}{f(\mathbf{X}^i)/T(\mathbf{X}^t \to \mathbf{X}^i)}\right\},$$

where $f(\mathbf{X})$ is the *measurement contribution function* [Vea98] and T denotes the transition probabilities; both are in product vertex area measure $\mathrm{d}\mathbf{X}$.

The lens perturbation [Vea98] randomly changes the incoming direction $\mathbf{i}_k$ at the camera aperture $\mathbf{x}_k$. It then ray traces to find the first intersection $\mathbf{x}_{k-1}$, and in case of specular interfaces continues further until a deterministic connection is possible, i.e. a vertex $\mathbf{x}_b$ with a diffuse BSDF is found. Manifold Exploration (ME) [Jak13] adopts this idea, but the connection is performed through specular chains if the next vertex is not diffuse. This procedure, however, depends on a classification of vertices as either 'diffuse' or 'specular', and glossy interactions have to be randomly classified as one or the other. This has been shown to be problematic [KD13]. HSLT [KHD14] can be superior to ME in highly glossy scenes even without the possibility of deterministic connections between sub-paths; this connection, however, gives ME a speed edge over HSLT. We extend this idea to a more flexible and general construction: first, we sample the "breakup vertex" $\mathbf{x}_b$ freely, independent of the notion of diffuse or specular. Second, we extent the lens perturbation to also mutate half vectors, resembling the multi chain perturbation [Vea98] known from MLT.

Sampling domains Light transport integration can be performed in different domains, e.g. outgoing projected solid angle [Kaj86, Arv86], surface area measure [Vea98], primary sampling space [KSKAC02], gradient domain [LKL*13], and half vector space [JM12, KHD14].

Outgoing projected solid angle domain $\mathrm{d}\mathbf{o}^\perp$ is a natural domain for surface scattering processes and constructs a path by sampling outgoing directions sequentially from one end. The end point of a path is thus hard to control or predict.

Surface area measure $\mathrm{d}\mathbf{x}$ is an impractical domain for path sampling, but is used to compare paths (possibly constructed with different sampling strategies and of different length) and their sampling probabilities. This measure, however, exhibits singularities when two path vertices get close [Kel97].

Primary sample space [KSKAC02] emphasizes more important regions of the projected solid angle domain by warping the domain, thus inheriting all the construction disadvantages of the outgoing projected solid angle domain.

Gradient domain transport [LKL*13] samples path differences in a domain defined by a suitable shift mapping. The resulting gradient image is then reconstructed after rendering using a Poisson solver. The success of the method depends greatly on the applied shift mapping and has been shown to work well if the resulting sampling domain is smooth with sparse edges, i.e. if it follows the statistics of natural images.

None of these domains provide a practical method for controlling both end points (e.g. sensor and light) of a path being sampled, which is a usual requirement for rendering an image. The half vector domain notably allows one to sample paths, while keeping both end points fixed. Moreover, all scattering events along a path can be importance-sampled [KHD14], which enables the efficient sampling of chains of highly glossy scattering events. However, sampling in this domain requires a more expensive path construction (using an iterative predictor-corrector method).

The half vector domain has further important properties: it can be easily transformed into an outgoing projected solid angle domain where scattering is defined; it has stable numerical behavior (compared to area measure); it is an important bridge to lens design, allowing extension of its transfer (or ABCD) matrices to transport with scattering.

Differential geometry Similarly to manifold exploration [Jak13] and HSLT [KHD14], our method relies on differential geometry for HSLT perturbation of the algorithm. We refer the reader to their discussion of previous work on differential geometry for more information.

3. An Improved Half Vector Space Mutation

In this section, we introduce the plane-plane parameterization for half vectors to HSLT. This parameterization is known from microfacet BRDF models [BS63, CT82], where it is also called the space of slopes. We will derive the affected transformations and constraint derivatives for HSLT.

3.1. Ray Differentials in Plane-Plane Parameterization

Solving the path space integral via Metropolis light transport (MLT) can be very efficient, but stratification on the image plane is typically not achieved. The original lens perturbation [VG97]—only one out of many perturbation strategies required to efficiently explore the path space—facilitates stratification by perturbing paths by explicitly moving the pixel coordinate, and constructing the rest of the path to follow. For more advanced perturbations, however, ray differentials need to be taken into account.

The HSLT perturbation explicitly models ray differentials in half vector space to perform predictable steps in image space [KHD14, Sec. 6.2]. This is achieved by first projecting the pixel differential to the tangent space of the first vertex from the camera $\mathbf{x}_{k-1}$ yielding $\Delta\mathbf{x}_{k-1}$. Then, we consult the tridiagonal constraint derivative matrix [Jak13]:

$$\begin{pmatrix} \mathbf{B}_1 & \mathbf{C}_1 & & & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & & \\ & \dots & \dots & \dots & \\ & & \mathbf{A}_{k-2} & \mathbf{B}_{k-2} & \mathbf{C}_{k-2} \\ \mathbf{0} & & & \mathbf{A}_{k-1} & \mathbf{B}_{k-1} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ . \\ \Delta\mathbf{h}_i \\ . \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ . \\ . \\ \mathbf{0} \\ \Delta\mathbf{x}_{k-1} \end{pmatrix}$$

to map $\Delta\mathbf{x}_{k-1}$ to a half vector offset $\Delta\mathbf{h}_i$ at every vertex $\mathbf{x}_i$ by extracting a 2×2 block D_i from this inverted matrix such that $\Delta\mathbf{h}_i = D_i^{-1}\cdot\Delta\mathbf{x}_{k-1}$ [KHD14, Eq. (9)]. This formalism is a powerful tool to map pixel offsets or ray differentials to half vector space and further to vertex offsets at all vertices $\mathbf{x}_i, i < k-1$ of the path.

In its original form, however, the choice of transforming ray differentials to the domain of half vectors (projected solid angle $d\mathbf{h}^\perp$) leads to problems: the domain has finite support (the unit disk, see Fig. 3) and since the ray differentials are only a first-order expression, the resulting transformed $\Delta\mathbf{h}$ will not account for the compression near the boundary of the domain and extend across it for some configurations of angles along a path.

This is a problem which limits the practical use of these ray differentials as some form of clamping is required not to leave the valid domain of half vectors. Previously this has been achieved by sampling an anisotropic Phong lobe on the hemisphere, and further distorting this using a Moebius transform [KHD14]. This ultimately distorts the pixel offsets in an uncontrolled manner as we will see below.

A better domain to transform the first-order differences

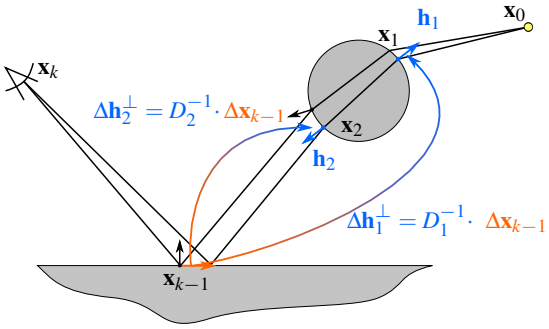


Figure 2: Transforming pixel offsets to half vector space ray differentials. The matrix D_i is computed from the constraint derivative matrix.

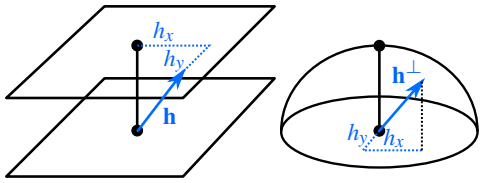


Figure 3: Different parameterizations. Left: the parallel plane parameterization ($\mathbf{h}_x^\parallel, \mathbf{h}_y^\parallel \in (-\infty,\infty)$) which we use for ray differentials. Right: hemispherical parameterization (${\mathbf{h}_x^\perp}^2 + {\mathbf{h}_y^\perp}^2 \leq 1$) used in the original HSLT method.

to is the plane-plane parameterization of half vectors (denoted as $\mathbf{h}^\parallel$). It also better matches the definition of microfacet BSDFs, e.g. a Beckmann lobe is a normal distribution in plane-plane parameterization [BS63]. The transform from projected solid angle to plane-plane and its Jacobian are:

$$f : \mathbf{h}^\perp \mapsto \mathbf{h}^\parallel = \left(\frac{h_\mathbf{x}}{\sqrt{1-h_\mathbf{x}^2-h_\mathbf{y}^2}}, \frac{h_\mathbf{y}}{\sqrt{1-h_\mathbf{x}^2-h_\mathbf{y}^2}} \right)$$

$$J_f = \begin{pmatrix} c+h_\mathbf{x}^2 c^3 & h_\mathbf{x} h_\mathbf{y} c^3 \\ h_\mathbf{x} h_\mathbf{y} c^3 & c+h_\mathbf{y}^2 c^3 \end{pmatrix},$$

$$\text{where } c := 1/\cos\theta = 1/\sqrt{1-h_\mathbf{x}^2-h_\mathbf{y}^2},$$

$$\text{and } |J_f| = (1-h_\mathbf{x}^2-h_\mathbf{y}^2)^2 = \cos^4\theta$$

is the Jacobian determinant of the transform. Note that this domain is unbounded, i.e. $\mathbf{h}^\parallel \equiv \mathbb{R}^2$. Here, all h scalars denote components of $\mathbf{h}^\perp$. It is possible to daisy-chain the transforms, in particular to multiply the two Jacobians from vertex area measure to projected half vector domain, and further to plane-plane parameterization. We can use this result for the first-order transformation of ray differentials, and the determinant to evaluate the transition probability densities in vertex area measure for the acceptance probability.

3.2. Plane-Plane Constraint Derivatives

In practice, instead of daisy-chaining the Jacobians as described above, we directly compute the constraint derivatives $\mathbf{A}_i\mathbf{B}_i\mathbf{C}_i$ in the plane-plane space $\mathrm{d}\mathbf{h}^{\parallel}$ for the simplified measurement contribution and the predictor-corrector method.

Consider a surface point $\mathbf{x}$ with geometric normal $\mathbf{n}_\mathrm{g}$ and the tangent space $(\mathbf{n}_\mathrm{g}, \partial_u\mathbf{x}, \partial_v\mathbf{x})$, and a shading normal $\mathbf{n} = \mathbf{n}(u,v)$ parameterized along the geometric tangent vectors $\partial_u\mathbf{x}$ and $\partial_v\mathbf{x}$ with scalars (u,v). Then a half vector constraint in tangent space can be written as [Jak13]:

$$T_\mathbf{x}^\mathrm{T}(u,v)\mathbf{h}_\mathbf{x}^{\parallel}(u,v) = \mathbf{0},$$

where $T_\mathbf{x}$ is a 3×2 matrix formed by the $\mathbf{s}$ and $\mathbf{t}$ vectors of a shading tangent frame, and $\mathbf{h}^{\parallel}$ is a half vector in the plane-plane domain. We construct the shading frame matrix by setting the vector $\mathbf{s}$ to follow the direction of $\partial_u\mathbf{x}$ and write the shading frame vectors $\mathbf{s}$ and $\mathbf{t}$ of $T_\mathbf{x}$ as

$$T_\mathbf{x}(u,v) = \begin{pmatrix} \mathbf{s}(u,v) \\ \mathbf{n}(u,v) \times \mathbf{s}(u,v) \end{pmatrix},$$

where $\mathbf{s}(u,v) = \mathcal{N}\left(\mathcal{G}\left(\partial_u\mathbf{x}, \mathbf{n}(u,v)\right)\right)$ is a normalized Gram-Schmidt orthogonalization ($\mathcal{G}(\cdot,\cdot)$ operator) of $\partial_u\mathbf{x}$ with respect to $\mathbf{n}(u,v)$, here $\mathcal{N}(\cdot)$ is a normalization operator. Note that $T_\mathbf{x}$ does not depend on the vector $\partial_v\mathbf{x}$. The tensor derivative of this matrix can be computed in spirit of Jakob [Jak13].

A half vector $\mathbf{h}^{\parallel}$ at the surface point $\mathbf{x} \equiv \mathbf{x}_i$ with predecessor vertex $\mathbf{x}_\mathrm{p} \equiv \mathbf{x}_{i-1}$, a successor vertex $\mathbf{x}_\mathrm{s} \equiv \mathbf{x}_{i+1}$, and the same parameterization (u,v) at $\mathbf{x}$ can be written as

$$\mathbf{h}^{\parallel}(u,v) = \mathbf{h}_\mathrm{un}(u,v)/|\langle\mathbf{h}_\mathrm{un}(u,v), \mathbf{n}(u,v)\rangle|, \text{where}$$
$$\mathbf{h}_\mathrm{un}(u,v) = \mathcal{N}\left(\mathbf{x}_\mathrm{p} - \mathbf{x}(u,v)\right) + \eta(\mathbf{x},u,v)\mathcal{N}\left(\mathbf{x}_\mathrm{s} - \mathbf{x}(u,v)\right)$$

is an unnormalized half vector, and $\mathbf{x}(u,v) = \mathbf{x} + u\partial_u\mathbf{x} + v\partial_v\mathbf{x}$ is moving on the geometric tangent plane of the surface and η is the ratio of indices of refraction at the interface.

Exemplarily we show the full derivative of $\mathbf{h}^{\parallel}$ by $\mathrm{d}u$ at $\mathbf{x}$ (all other derivatives are derived analogously):

$$-\frac{\mathrm{d}\mathbf{h}^{\parallel}}{\mathrm{d}u} = -\frac{-\frac{\partial_u\mathbf{x}}{|\mathbf{e_o}|} + \frac{\mathbf{o}\partial_u\mathbf{x}\cdot\mathbf{o}}{|\mathbf{e_o}|} - \frac{\partial_u\mathbf{x}}{|\mathbf{e_i}|} + \frac{\mathbf{i}\partial_u\mathbf{x}\cdot\mathbf{i}}{|\mathbf{e_i}|}}{\mathbf{n}\cdot(\mathbf{o}+\mathbf{i})}$$
$$\frac{(\mathbf{o}+\mathbf{i})\left(\partial_u\mathbf{n}\cdot(\mathbf{o}+\mathbf{i}) + \mathbf{n}\cdot\left(-\frac{\partial_u\mathbf{x}}{|\mathbf{e_o}|} + \frac{\mathbf{o}\partial_u\mathbf{x}\cdot\mathbf{o}}{|\mathbf{e_o}|} - \frac{\partial_u\mathbf{x}}{|\mathbf{e_i}|} + \frac{\mathbf{i}\partial_u\mathbf{x}\cdot\mathbf{i}}{|\mathbf{e_i}|}\right)\right)}{\left(\mathbf{n}\cdot(\mathbf{o}+\mathbf{i})\right)^2},$$

where the length of the incident edge $|\mathbf{e_i}| = \sqrt{(\mathbf{x}_\mathrm{p}-\mathbf{x})\cdot(\mathbf{x}_\mathrm{p}-\mathbf{x})}$ and for the outgoing edge $|\mathbf{e_o}| = \sqrt{(\mathbf{x}_\mathrm{s}-\mathbf{x})\cdot(\mathbf{x}_\mathrm{s}-\mathbf{x})}$. One can easily obtain the above result by applying the derivative of multiplication and the derivative of normalization.

Differentiating the 2D half vector constraint by u and v at every vertex results in a 2×2 matrix. These matrices for each half vector are then stored along with every vertex as

$$\mathbf{A}_i = \mathrm{d}\mathbf{h}_i^{\parallel}/\mathrm{d}\mathbf{x}_{i-1}, \quad \mathbf{B}_i = \mathrm{d}\mathbf{h}_i^{\parallel}/\mathrm{d}\mathbf{x}_i, \quad \mathbf{C}_i = \mathrm{d}\mathbf{h}_i^{\parallel}/\mathrm{d}\mathbf{x}_{i+1},$$

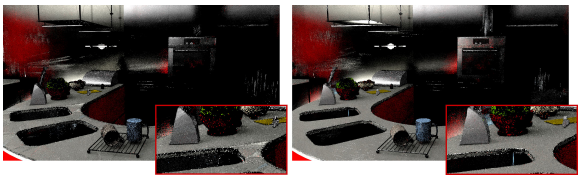


Figure 4: Non-spectral renderings with our method integrated into Mitsuba *without* orthonormalized tangent frames. Left: half vector perturbations (Sec. 3) without compensation for cross-primitive rotation of the tangent frame. Right: HVMCP (Sec. 4) where half vectors are rotated.

where the differentiation is with respect to $\mathrm{d}\mathbf{x}_i$, i.e. a 2D differential of $\mathbf{x}_i$ parameterized along its u and v axes.

Measurement Note that the optimized measurement contribution [KHD14, Eq. (5)] now has the Jacobian $|\mathrm{d}\mathbf{o}_i^{\perp}/\mathrm{d}\mathbf{h}_i^{\parallel}|$ in the product for every vertex instead of $|\mathrm{d}\mathbf{o}_i^{\perp}/\mathrm{d}\mathbf{h}_i^{\perp}|$. This means that we need to multiply by $(\mathbf{h}_i\cdot\mathbf{n}_i)^3$ instead of dividing by $\mathbf{h}_i\cdot\mathbf{n}_i$; this is numerically more well-behaved.

Non-orthonormal Tangent Frames On triangle meshes with vertex normals, the curvature can be most precisely expressed with barycentric coordinates, i.e. without aligning $\partial_u\mathbf{x}$ and $\partial_v\mathbf{x}$ to the texture coordinates of the mesh and without orthonormalization. In fact, the measurement contribution in half vector space also contains the curvature in the transfer matrix [KHD14, Eq.(5)]. A more accurate curvature can also reduce number of the predictor-corrector steps, however, the effect in our tests (for simple geometry such as planes and spheres) was negligible.

In principle, we can derive the constraint derivatives in this non-orthogonal space and account for the area measure change due to the area spanned by the basis vectors. However, in the case of rough surfaces, the rapid change of parameterization due to the rotation of the tangent frame between triangles leads to inconsistent orientations. To correct this, the tangent bundle must be kept continuous on the surface by means of parallel transport. In general, this requires keeping a constant angle to a geodesic connecting the base points of two tangent spaces. Finding such geodesics on generic triangle meshes [SSK*05] is impractical for HSLT. According to our experiments tessellating the geometry is a pragmatic solution yielding sufficiently accurate results. Differences are still visible in unconverged images, see Fig. 4. Note that our half vector multi chain perturbation introduced in Sec. 4 can easily avoid this problem.

3.3. Sampling with Plane-Plane Ray Differentials

Next we build on the new ray differentials and detail how we account for the BSDF when sampling half vectors.

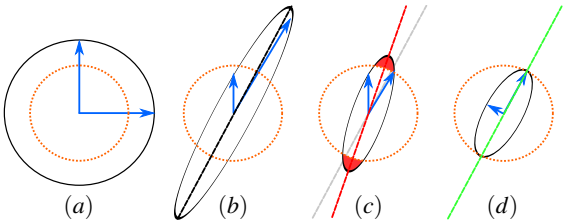


Figure 5: Illustration of ray differentials in half vector space: (a) isotropic half vector space and optimal BSDF step size dashed in orange. (b) an anisotropic Gaussian determined by the ray differential matrix D_i. (c) clipping the basis vectors individually leads to two problems (marked in red): a wrong direction of the main axis and step sizes outside the BSDF circle. (d) first performing a singular value decomposition and then clipping the resulting singular values yields the correct result.

Sampling Domain of Half Vector Offsets Kaplanyan et al. [KHD14] sampled half vector offsets using a zero-centered anisotropic Phong distribution on the disk and using a Möbius transform to align it with the requested offset. This ensures that the sampled offset remains within the domain of projected solid angle (the unit disk). In contrast, the plane-plane parameterization for ray differentials constitutes an unbounded domain and thus allows us to directly sample an anisotropic Gaussian distribution, which is a natural match for Beckmann half vector distributions.

Directional Alignment when Combining with BSDF Another issue in the original HSLT method is that the pixel footprint basis vectors in half vector domain ($\mathbf{h}_\mathbf{u}^\perp$ and $\mathbf{h}_\mathbf{v}^\perp$) were orthonormalized—fixing the direction of the longer vector—to align the sampling scheme with the ray differential before clipping with the BSDF step size [KHD14, Sec. 6.2 and 7]. This, however, typically results in a slight rotation of the pixel footprint in half vector domain. Here $\mathbf{h}_\mathbf{u}$ denotes the change of $\mathbf{h}$ mapped from the pixel step size along the horizontal $\mathbf{u}$ axis on the image plane, as opposed to the aforementioned scalar u that parameterizes the tangent space of a vertex.

Fig. 5 depicts the alignment problem. Fig. 5a shows the isotropic plane-plane half vector domain and the dashed orange circle indicates a proposed BSDF step size computed from roughness (the center does not necessarily mean $\mathbf{h} = \mathbf{0}$). Fig. 5b shows the transformed pixel footprint vectors $\mathbf{h}_\mathbf{u}$ and $\mathbf{h}_\mathbf{v}$ in blue with the corresponding ellipse drawn at one standard deviation of the respective Gaussian. Fig. 5c illustrates the clipping of these vectors at the BSDF step size: the main axes of the resulting ellipse rotate, resulting in an undesired angular offset as well as causing a part of the sampling domain to lie outside the requested BSDF step size. Note that orthogonalization of $\mathbf{h}_\mathbf{u}$ and $\mathbf{h}_\mathbf{v}$ before clipping does not improve the result as the longer vector remains unchanged.

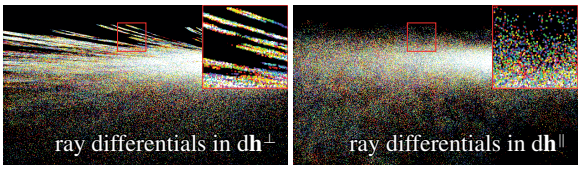


Figure 6: Comparison of ray differentials in projected solid angle d$\mathbf{h}^\perp$ and plane-plane parameterization d$\mathbf{h}^\parallel$ (spectral rendering). The projected half vector ray differentials show non-uniform exploration of the image plane and distracting line artifacts.

Fig. 5d illustrates our solution: we compute a 2×2 singular value decomposition (SVD) [Bli96] of the matrix consisting of $\mathbf{h}_\mathbf{u}$ and $\mathbf{h}_\mathbf{v}$ as column vectors for every vertex $\mathbf{x}_i$:

$$D_i^{-1} = R_i \cdot diag(s_x, s_y) \cdot V$$

The SVD decomposes the matrix into rotation matrices R and V and two singular values (s_x, s_y) defining the extent of the ellipse along the orthogonal main axes. By this, the radii of the ellipse can be scaled without introducing undesired rotation.

As a rotation matrix has a unity Jacobian determinant, sampling in this rotated space does not require an additional correction term. Note that the projection of pixel offsets to the tangent space of the vertex $\mathbf{x}_{k-1}$ (first from the camera) is non-linear. However, we are only interested in first-order derivatives which can be represented with a matrix.

Fig. 6 compares ray differentials in projected solid angle d$\mathbf{h}^\perp$ to plane-plane parameterization d$\mathbf{h}^\parallel$ on a diffuse plane with light and camera at grazing angles. As in the original work, we distribute the step sizes among all vertices of a path [KHD14, Sec. 6.3]. Note that both sets of ray differentials are unbiased and converge to the same image. However, ray differentials in d$\mathbf{h}^\perp$ exhibit line-like clumping artifacts with smoothly changing direction. Ray differentials in d$\mathbf{h}^\parallel$ avoid the aforementioned problems and result in uniform step sizes throughout the whole domain.

BSDF Scale Factor We adapt step sizes to BSDFs by scaling the ray differential shape (preserving anisotropy) according to the bandwidth of the material, which is determined by the Beckmann-equivalent roughness of the surface [Jak10] (essentially the standard deviation of a Gaussian in plane-plane space, scaled by a factor of two). By this we obtain the expected step size in plane-plane (cf. [KHD14, Sec. 6.1]):

$$(\sigma_{x,i}, \sigma_{y,i}) = \min\{\mathbf{s}_i, \alpha_i/\sqrt{\pi}\},$$

where $\mathbf{s}_i = (s_{x,i}, s_{y,i})$ are the singular values of the ray differential matrix. Limiting the step size is important to avoid "jumping off" glossy highlights. Taking the minimum with the singular values ensures that step sizes remain small when recommended by the ray differentials. The half vector step

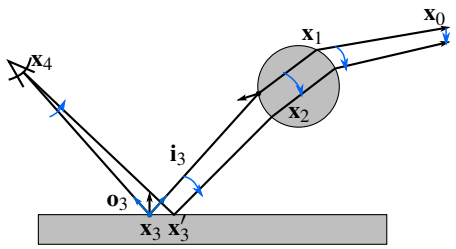


Figure 7: Illustration of the half vector multi chain perturbation. The vertices typically diverge farther away from the current path the more the half vectors along the path are perturbed.

$\Delta\mathbf{h}_i$ is then computed by sampling two 1D Gaussians for the two main axes and the standard deviations $\sigma_{x,i}$ and $\sigma_{y,i}$.

4. Half Vector Multi Chain Perturbation

In this section we introduce the *half vector multi chain perturbation* (HVMCP) to complement the half vector space perturbation introduced in [Jak13, KHD14]. The latter can only perturb the path in a local environment around an existing path (a *sub-space*), in which the constraint derivatives are meaningful. HVMCP facilitates jumps between sub-spaces in a much simpler way than in the original HSLT method [KHD14, Sec. 5.2]. In fact our HVMCP is similar to lens and multichain perturbations [VG97]: it begins with perturbing the pixel coordinate and the point on the aperture, then tracing a ray to find the updated position of $\mathbf{x}_{k-1}$. Its half vector $\mathbf{h}_{k-1}$ is perturbed in plane-plane space by a small symmetric jump (see below), and the updated direction towards the light $\mathbf{i}_{k-1}$ is computed from the half vector and the direction to the eye $\mathbf{o}_{k-1}$ (Fig. 7). Tracing in the direction towards the light yields the next vertex, for which we proceed analogously.

For each half vector we achieve a symmetric perturbation by using the average of the BSDF roughnesses at the respective vertex of the current and the tentative path for the mutation (e.g. $\mathbf{x}_3$ and $\mathbf{x}'_3$ in Fig. 7). In contrast to the half vector space perturbation, both are already known at this point and the tentative vertex locations do not change anymore once they have been determined by ray tracing.

Creating these tentative paths is similar to using primary sample space MLT [KSKAC02] with a simple path tracer as underlying sampler. One consequence is that HVMCP shares the problem of moving the vertex on the light $\mathbf{x}_0$ off the geometry of (small) luminaires. This results in invalid paths that have to be rejected.

However, as we are mainly interested in providing means to facilitate sub-space jumps, we perform very small, yet sufficient, half vector steps. This increases the acceptance rates of HVMCP. We use $0.01 \cdot \alpha$ as the standard deviation of a Gaussian random variable, where α is the mean

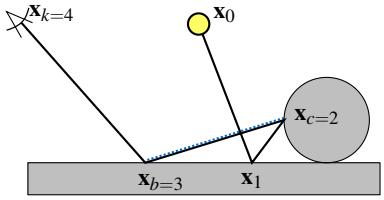


Figure 8: Notation used for the breakup strategy: $\mathbf{x}_b$ and $\mathbf{x}_c$ in a configuration where the mutation strategy acts very similar to many-light methods: the multi chain perturbation is performed between $\mathbf{x}_k$ and $\mathbf{x}_b$ (first hit in the scene), and a connection is performed to $\mathbf{x}_c$. In this example, the sub-path $\mathbf{x}_b \dots \mathbf{x}_c$ contains no half vectors and consequently the half vector perturbation sub-path degenerates to a simple deterministic connection.

BSDF roughness of the pair of current and tentative vertex. In Sec. 5 we combine this strategy with half vector perturbation to increase the flexibility for path space exploration, and also to alleviate the problem of missing the light source.

Determining the acceptance probability for the perturbed path requires computing the transition probabilities in vertex area measure. To achieve this with HVMCP, we first convert the (symmetric) half vector transition probability to projected outgoing solid angle $d\mathbf{i}^{\perp}$ by applying the appropriate Jacobian $|d\mathbf{i}^{\perp} / d\mathbf{h}^{\parallel}|$, which is known from microfacet theory [WMLT07, DHI*13], and then apply geometry terms to convert to vertex area measure. In fact the geometry terms cancel out with those in the measurement contribution and thus do not need to be computed explicitly.

Non-orthonormal Tangent Frames As mentioned above, HVMCP can elegantly avoid the problem of non-orthonormal tangent frames: in HSLT, we cannot guarantee that the construction of the proposal path $\mathbf{X}^t$ from the current path $\mathbf{X}^i$ will touch the same surface points as the reverse construction of $\mathbf{X}^i$ from $\mathbf{X}^t$ due to the predictor-corrector Newtonian walk. Unlike HSLT, HVMCP does not iteratively fix paths and the construction is reversible. In this case, global parallel transport on the mesh is not required. We only need to ensure that the point-to-point tangent space rotation from $\mathbf{X}^i$ to $\mathbf{X}^t$ is reversible, i.e. exactly compensated by its inverse for the backwards walk from $\mathbf{X}^t$ to $\mathbf{X}^i$. In our implementation, we project the shared line from a source vertex to a proposal vertex to the tangent frame of both vertices, and rotate the half vector such that it keeps a constant angle with respect to the projection of the line in each tangent frame. This is achieved by applying a simple 2D rotation matrix when carrying the half vector from one tangent frame to another.

5. Breakup Strategy for Paths

The idea of HSLT is to satisfy the constraints along a path for which it relies on the underlying differential geometry.

However, in many cases it is not advisable or necessary to satisfy them, e.g. when they are very soft (diffuse materials) or when the differential geometry is not reliable (high-frequency displacement). Especially in such cases, an option to split the path at one of these vertices and connect to it from both sides can be a remedy. In order to improve image stratification we propose to first mutate the camera sub-path by perturbing and tracing the outgoing direction (HVMCP), and then performing half vector mutation for the next sub-path. To this end, we describe how to combine the two mutation strategies from Sec. 3 and 4.

For this, we will choose vertices $\mathbf{x}_a, \mathbf{x}_b$ and $\mathbf{x}_c, k \geq a \geq b \geq c \geq 0$ along the path (the notation is analogous to manifold exploration [Jak13], see Fig. 8). This breaks up the path into three sub-paths (determining the indices b and c is important and will then be discussed in the remainder of this section):

- **Multi Chain Sub-Path** $(\mathbf{x}_k, \ldots, \mathbf{x}_b)$: the multi chain perturbation starts at $\mathbf{x}_a$, and creates all vertices up to including $\mathbf{x}_b$. We choose $a \equiv k$ fixed at the camera, as we always want to mutate the pixel coordinates to improve image stratification. $\mathbf{x}_b$ is determined by ray tracing the perturbed directions $\mathbf{i}_k, \ldots, \mathbf{i}_{b+1}$ from the eye with HVMCP.
- The **Half Vector Space Sub-Path** $(\mathbf{x}_{b-1}, \ldots, \mathbf{x}_{c+1})$ is updated via the half vector space predictor-corrector method. The vertices $\mathbf{x}_b$ and $\mathbf{x}_c$ are fixed and the half vectors determining the sub-path in between are perturbed: $(\mathbf{x}_b, \mathbf{h}_{b-1}, \ldots, \mathbf{h}_{c+1}, \mathbf{x}_c)$.
- The **Fixed Sub-Path** $(\mathbf{x}_c, \ldots, \mathbf{x}_1, \mathbf{x}_0)$ always remains unchanged and thus neither recomputing transition probabilities and measurement, nor tracing any rays is required.

In order to increase the efficiency in terms of mutations per CPU cycle, we want to achieve large b and c, with the constraint that $a \equiv k \geq b \geq c \geq 0$. Using $b = k$ results in half vector mutation only, and $b = 0$ means full multi chain perturbation; note that $b = c$ is only possible if $b = c = 0$. Choosing $c = 0$ and mutating half vectors between $\mathbf{x}_b$ and $\mathbf{x}_0$ reduces autocorrelation, while $c > 0$ increases the performance due to the fixed sub-path. An important aspect of a good choice of b is to avoid half vector mutations on too fine differential geometry, as otherwise the mutation may get stuck in small highlights for too long (Fig. 1).

Essentially, we determine b and c by sampling a probability mass function (PMF) determined by the roughness of the material at the vertex locations. We resort to full half vector mutation or full HVMCP if no suitable breakup points can be found as described next.

Choosing b (Length of HVMCP) We use a simple approach to sample b. We construct a PMF where $P(b = i) = \alpha_i / N$ for $i \in [1, k-1]$ where α_i is the roughness Beckmann equivalent at vertex $\mathbf{x}_i$. Alternatively, we also tried using the 90-percentile of the roughness [Jak13] to form the probability, however, in our experiments the difference between the two heuristics was negligible. To also include full HVMCP and full half vector mutation we assign constant probabilities of $P(b = 0) = 0.1/N$ and $P(b = k) = 0.1/N$. The normalization factor is $N = 0.2 + \sum_{i=1}^{k-1} \alpha_i$. The PMF is sampled using the inversion method. In Sec. 6 we analyze this strategy and show experiments biasing b more towards the camera.

The choice of b is more important to achieve stratification on the image plane, while c is rather a trade off between correlation and computation time. Therefore we determine c after fixing b as described next.

Choosing c (Half Vector Sub-Path) The location of $\mathbf{x}_c$ along the path affects the mutation speed as well as the autocorrelation which can lead to visible artifacts (temporal flickering) in animations. To reduce these artifacts while choosing $c > 0$, we want to sample c such that the Jacobian resulting from this choice is as flat as possible: we observe that the half vector space sub-path $(\mathbf{x}_b, \mathbf{h}_{b-1}, \ldots, \mathbf{h}_{c+1}, \mathbf{x}_c)$ incurs a computation of the geometry factor $G(\mathbf{x}_b, \mathbf{x}_{b-1})$ or $G(\mathbf{x}_c, \mathbf{x}_{c+1})$ (depending on the tracing direction used to compute the derivatives) when computing the (optimized) half vector space measurement contribution. We thus choose c with the goal of minimizing the geometry factor $G(\mathbf{x}_c, \mathbf{x}_{c+1})$ at $\mathbf{x}_c$ (analogous for the opposite tracing direction). This lines up well with observations from many-lights rendering, where small distances (and thus large geometry terms) pose problems in form of correlated blotches [DKH*14]. In fact the path configuration $c = b - 1$ is exactly the same as using a virtual point light. Similarly, singular and highly glossy BSDFs at $\mathbf{x}_c$ need to be avoided. Thus, for sampling c we construct a PMF as $P(c = i) = \alpha_i \cdot \|\mathbf{x}_i - \mathbf{x}_{i+1}\|^2 / M$ with $M = \sum_{j=0}^{b-1} \alpha_i \cdot \|\mathbf{x}_i - \mathbf{x}_{i+1}\|^2$.

6. Results

We implemented the original HSLT and our improved HSLT in our custom renderer, and integrated it into Mitsuba (which we plan to release) for comparison with other methods.

Spectral Rendering Our renderer also extends HSLT to spectral ray tracing. We found that it is not necessary to introduce the wavelength to the derivatives (akin to Elek et al. [EBR*14]) as the spectral dispersion introduces quite small vertex offsets only. Instead, in case of transmission we recompute the constraint derivatives after mutating the wavelength and let the predictor-corrector method update the vertex offset. The transition probability needs to account for the wavelength mutation only for asymmetric perturbations. All images, except for Fig. 4 are spectral renderings with uniform sampling of wavelengths.

Mutating Endpoints We also integrated mutation of the endpoints of a path on the aperture (for rendering depth of field) and on the light source. We account for the change in the transition probability in vertex area measure, and again the predictor-corrector method fixes the path as long as the offsets remain small enough.

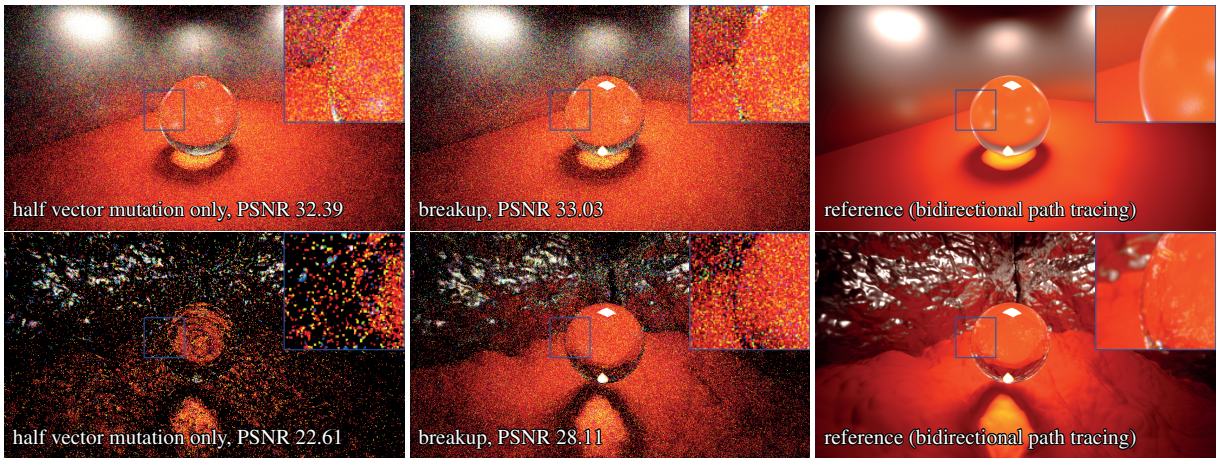


Figure 9: Equal time comparison (30 sec): with flat surfaces (top row) half vector mutation works well as the derivatives are well-behaved; the PSNR for breakup is slightly better as more samples per second are computed (shooting less rays). With fine displaced geometry (bottom row), half vector mutations get stuck in exploring small features, resulting in non-uniform exploration of the image plane.

Our test renders have been performed in our custom renderer on a hexacore Intel Core i7-3930K CPU @ 3.20GHz using 12 threads. All images but Fig. 6 (top) use the plane-plane ray differentials to enable a meaningful analysis of the breakup strategy.

Fig. 1 shows an equal sample count comparison of original HSLT and the breakup strategy ("improved HSLT"). Note that all images have been computed using the plane-plane ray differentials. The breakup strategy efficiently handles displaced diffuse surfaces, and also preserves the caustic on the iris which is itself a highly displaced surface. Due to less ray tracing, the half vector multi chain perturbation results in a large speed up. In principle, full half vector mutation also works on finely displaced surfaces, however, it gets stuck in small highlights as the BSDF roughness always clips the proposal of the ray differentials and the bidirectional mutation fails to insert enough independent paths into the Markov chain to achieve a uniform distribution.

Fig. 9 shows an equal time comparison of half vector mutation only (as in HSLT, but with plane-plane ray differentials) vs. the breakup strategy combining half vector and multi chain perturbations. HSLT performs well in regions of smooth geometry where the derivatives are well-behaved, but still computes slightly less mutations than the breakup strategy as more rays need to be traced for the predictor-corrector method. Note that the flat geometry has only very few polygons and ray tracing is thus a lot faster than with fine displaced geometry.

Fig. 10 shows the effect of choosing the vertex $\mathbf{x}_c$ with $c = b - 1$. For still images, we can prefer large c to increase speed, at the expense of increasing correlation. In this example, sampling b uniformly (all surfaces have the same roughness) sufficiently decorrelates the paths, i.e. large c can be chosen to reduce the number of traced rays without creating visible artifacts.

Figure 10: A diffuse scene with 256 samples per pixel: comparison of selecting c using $c = b - 1$ (left) to selecting based on (uniform) roughness (center). The sampling of b randomly decorrelates the walk and, perhaps counterintuitively, no splotchy correlation artifacts are visible. This means we can save some computation time (left) and get roughly the same image quality as for uniform sampling of c (center).

Figure 11: Comparison of choosing $b = k - 1$ and $c = k - 2$ (left) to roughness sampling as described in Sec. 5 (both rendered with 1024 samples per pixel). Choosing b and c close to the camera introduces splotch-like correlation artifacts, but increases the overall PSNR as dimensions closer to the camera are more thoroughly explored.

Fig. 11 shows the result of choosing $b = k - 1$ and $c = k - 2$ (close to the camera) rather than sampling them by rough-

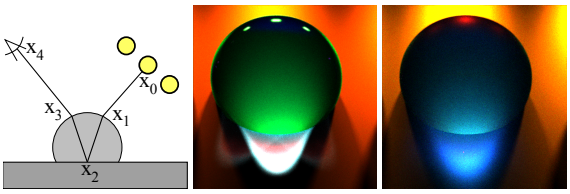


Figure 12: Schematic view of the scene with a sphere partly intersecting a ground plane and false-color images indicate the breakup vertex $\mathbf{x}_b$: magenta $b = 0$, red $b = 1$, green $b = 2$, blue $b = 3$, cyan $b = 4$. Center: false-color image obtained for a specular sphere and glossy ground plane. Right: roughness $\alpha = 0.2$ for all surfaces. Curvature clearly influences the success we can expect from a technique.

ness. In this example (with a maximum path length of 10), this results in a speedup and improved PSNR, however, at the expense of slightly non-uniform exploration in image space: small cavities in corners show correlation artifacts, which is a known problem of many-light methods [DKH*14]. Its appearance is no surprise, as the path from the light to $\mathbf{x}_c$ is kept and resembles a virtual point light. In this case, this strategy (at least for static images) works well as the diffuse BSDF at $\mathbf{x}_b$ hides the correlation of the signal at $\mathbf{x}_c$ behind a low-frequency convolution.

7. Analysis and Discussion of Breakup Criteria

In our prototypal implementation we sample b and c depending on surface roughness and vertex distance only (except for the comparisons above). Table 1 qualitatively compares the resulting breakup strategy for different configurations.

Beyond this approach, half vector ray differentials carry a lot of information about a path, e.g. the expected change of a half vector when moving on the image plane, or changes in world space distances. We believe that such information can lead to more sophisticated breakup heuristics. As an example, Fig. 12 shows false-color images of a sphere partly intersecting a ground plane to illustrate how curvature can affect the optimal choice of b and c. The images are created by splatting each path with the correct intensity, but using the color coding depending on the breakup vertex index b as indicated in the caption. In the center image, the green color inside the sphere indicates that the breakup vertex is predominantly at $b = 2$ (on the ground plane). In the right image all surfaces have equal roughness, e.g. all values of b are chosen with equal probability. However, from the coloring we can observe that mostly paths with $b = 3$ and $b = 4$ contribute. Note that with optimal choice of b, we would observe an equal contribution of all color codings. We could not proof sub-optimal behaviour in a practical rendering, even with long glossy chains (see Fig. 13).

From our observations (also Tab. 1), we identify interesting directions for future work:

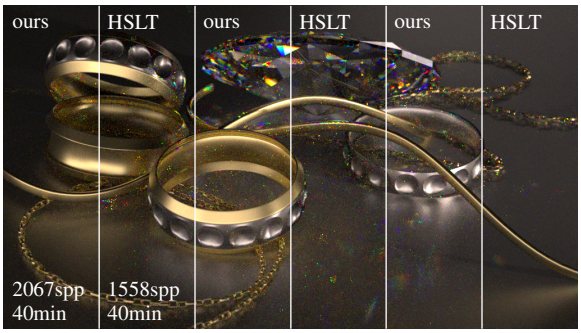


Figure 13: The necklace scene: in this case with relatively simple and smooth geometry and predominantly glossy materials, a full half vector perturbation is nearly always more efficient than breaking the path. Our simple heuristic detects this and degenerates almost always to pure half vector space perturbations.

- The breakup strategy avoids highly glossy surfaces. However, ray differentials can be used to augment the roughness-based breakups: we can estimate the maximum half vector perturbation at a vertex for a given step size on the image plane. Evaluating the BSDF Gaussian for the maximal perturbed half vector yields an expected change to the measurement contribution and thus the acceptance rate, which provides a well-founded basis for decision making.
- The original HSLT gets stuck in small highlights. However, we can compute the change to $\mathbf{x}_{k-1}$ for a given perturbation $\Delta\mathbf{h}_i$ as $\Delta\mathbf{x}_{k-1} = D_i \cdot \Delta\mathbf{h}_i$ and compute the pixel offset by projecting to the sensor. It would allow us to detect the occurrence of many consecutive small steps, and in this case use the breakup strategy to "jump off" the feature.

8. Limitations and Future Work

We have shown how to improve HSLT for highly displaced scenes where the details reside on surfaces with sufficiently wide glossy or diffuse BSDFs. In Fig. 1, for example, the cornea of the eye is smooth and has meaningful geometric derivatives at a large scale, while the other surfaces are displaced. However, HSLT still gets stuck in very small caustics if the surface casting them is finely displaced (which may or may not be desirable), but we believe that half vector ray differentials can help to make mutations more robust in the future.

In principle participating media can easily be incorporated in our work by making sure that all medium interactions lie in the multi chain sub-path of the mutation. However, this workaround results in small b (short half vector sub-paths) or even results in $b = 0$ (no half vector mutations at all).

The multi chain perturbation is a simple means to per-

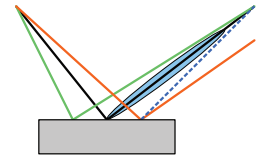
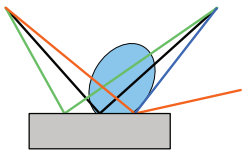
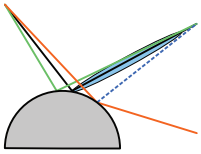
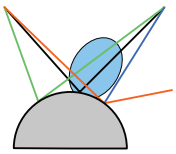
	 specular, flat surface	 rough, flat surface	 specular, curved surface	 rough, curved surface
Half vector [KHD14]	✓ meaningful differential geometry	✓ meaningful differential geometry	✓/✗ can stick to small highlights for too long if differential geometry is meaningless on ray differential scale	✓/✗ may be inefficient, can stick to too small features, differential geometry too fine wrt. ray differentials
HVMCP	✓ will make small changes to the path	✓/✗ sometimes inefficient: large steps due to high roughness, problems with small lights and intricate geometry at connecting vertex	✗ curvature will lead to a very different next vertex	✗ inefficient due to the very different next vertex (worse than in case of the specular surfaces) due to larger steps
Breakup	✓ usually keeps the half vector and does not perform a breakup at this vertex	✓ works well	✓ usually keeps the half vector; jumps off small highlights	✓/✗ may work well, depending on the surroundings; walks off the geometry of the feature

Table 1: A qualitative comparison of mutation strategies for specular/rough materials and flat/curved surfaces. Color coding: current path (black), half vector mutation (green), multi chain perturbation (orange), and breakup with connection (blue).

form sub-space jumps [KHD14, Sec. 5.2] and to explore wide parts of the path space. However, it cannot replace the (inefficient) bidirectional mutation, as it does not change the path configuration (i.e. it does not add or remove vertices).

We have demonstrated that there is potential to find more optimal breakup strategies to better combine multi chain and half vector perturbations potentially using ray differentials. We are confident that this tool will play an important role in making Metropolis light transport more aware of, and more uniformly exploring, the image plane, resulting in better noise characteristics and improving temporal stability.

References

[Arv86] ARVO J.: Backward ray tracing. In *SIGGRAPH Course Notes* (1986), pp. 259–263. 2

[Bli96] BLINN J.: Consider the lowly 2 x 2 matrix. *Computer Graphics and Applications, IEEE 16*, 2 (March 1996), 82–88. 5

[BS63] BECKMANN P., SPIZZICHINO A.: *The scattering of electromagnetic waves from rough surfaces*. MacMillan, New York, 1963. 3

[CT82] COOK R., TORRANCE K.: A reflectance model for computer graphics. *ACM Trans. on Graphics 1*, 1 (1982), 7–24. 3

[DHI*13] DUPUY J., HEITZ E., IEHL J.-C., POULIN P., NEYRET F., OSTROMOUKHOV V.: Linear efficient antialiased displacement and reflectance mapping. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia) 32*, 6 (2013). 6

[DKH*14] DACHSBACHER C., KRIVÁNEK J., HASAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum 33*, 1 (2014), 88–104. 7, 9

[EBR*14] ELEK O., BAUSZAT P., RITSCHEL T., MAGNOR M., SEIDEL H.-P.: Spectral ray differentials. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 33*, 4 (2014). 7

[Has70] HASTINGS W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 1 (1970), 97–109. 2

[Jak10] JAKOB W.: Mitsuba renderer, 2010. URL: http://www.mitsuba-renderer.org. 5

[Jak13] JAKOB W.: *Light transport on path-space manifolds*. PhD thesis, Cornell University, 2013. 2, 3, 4, 6, 7

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. on Graphics (Proc. SIGGRAPH) 31*, 4 (2012), 58:1–58:13. 2

[Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proc. of SIGGRAPH)* (1986), 143–150. 2

[KD13] KAPLANYAN A. S., DACHSBACHER C.: Adaptive progressive photon mapping. *ACM Trans. on Graphics 32*, 2 (2013), 16:1–16:13. 2

[Kel97] KELLER A.: Instant radiosity. *Proc. SIGGRAPH* (1997), 49–56. 2

[KHD14] KAPLANYAN A. S., HANIKA J., DACHSBACHER C.: The natural-constraint representation of the path space for efficient light transport simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 33*, 4 (2014). 1, 2, 3, 4, 5, 6, 10

[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. 1, 2, 6

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain Metropolis light transport. *ACM Trans. on Graphics (Proc. SIGGRAPH) 32*, 4 (2013), 95:1–95:12. 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. 2

[SSK*05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GORTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 553–560. 4

[Vea98] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1998. AAI9837162. 1, 2

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. *Proc. SIGGRAPH* (1997), 65–76. 1, 3, 6

[WMLT07] WALTER B., MARSCHNER S., LI H., TORRANCE K.: Microfacet models for refraction through rough surfaces. In *Proc. Eurographics Symposium on Rendering* (2007), pp. 195–206. 6