Real-Time Path Guiding Using Bounding Voxel Sampling: Supplemental Material

HAOLIN LU, University of California San Diego, USA
WESLEY CHANG, University of California San Diego, USA
TREVOR HEDSTROM, University of California San Diego, USA
TZU-MAO LI, University of California San Diego, USA

In this supplemental document, we provide pipeline design and implementation details and further discuss bias.

ACM Reference Format:
Haolin Lu, Wesley Chang, Trevor Hedstrom, and Tzu-Mao Li. 2024. Real-Time Path Guiding Using Bounding Voxel Sampling: Supplemental Material. *ACM Trans. Graph.* 43, 4, Article 125 (July 2024), 3 pages. https://doi.org/10.1145/3658203

1 DETAILS OF PIPELINE DESIGN

1.1 Advantages of Voxels

The main advantage of using voxels over surface-based representations is its increased robustness. Directly estimating irradiance per scene primitive has drawbacks for both large and small primitives. When a large primitive has only a small portion exposed to illumination (e.g., the walls in VeachAjar), a single irradiance value is unable to capture the variation within the primitive. Conversely, estimating irradiance for very small primitives is expensive and often unnecessary. In contrast, voxels naturally subdivide large primitives and cluster smaller ones, robustly expressing irradiance variation across space, as observed in previous work [Crassin et al. 2011; Fridovich-Keil* et al. 2022; Keller et al. 2014].

1.2 Light Injection v.s. Reverse Instant Radiosity

Reverse Instant Radiosity [Segovia et al. 2006] traces paths from the camera and places virtual point lights at the first hit of the indirect bounce. These points, which we denote as x_2, are guaranteed to contribute to the image, since they are visible from some shading points. Unfortunately, reverse instant radiosity requires the estimation of the probability density for generating each point x_2:

$$p(x_2) = \int_{\mathcal{M}_c} p(x_1)p(x_1 \to x_2)\mathrm{d}x_1, \tag{1}$$

where, $\mathcal{M}_c$ is the set of all shading points. Estimating this integral introduces substantial overhead.

Fortunately, since we do not perform any path reconnection and our path PDF is independent of the light injection, we do not need to compute the integral above. This enables us to inject light information very efficiently compared with Reverse Instant Radiosity.

1.3 Superpixel Clustering.

The method for clustering shading points by a top-down kd-tree partition [Ou and Pellacini 2011; Walter et al. 2006; Wu and Chuang 2013] is too slow for real-time applications, and so we base our method off of the SLIC superpixel algorithm [Achanta et al. 2012], which simply clusters based on local similarity and can be parallelized [Ren et al. 2015].

The pixel clustering process aims to minimize the within-cluster sum of squared distances. The K-means algorithm accomplishes this optimization by iteratively assigning each pixel to the nearest cluster and updating the centroids of each cluster accordingly. SLIC further simplifies the problem by only searching for nearby clusters within a small region, typically twice the initial tile size, in the assignment stage. In our implementation, we initialize the cluster centroids using the data from the center pixel of each 32×32 tile and only execute the pixel assignment stage once.

To define the distance, we assess geometric similarity instead of the color similarity used in SLIC. We assign each shading point x_1 a data tuple $d = (p, n, u)$, which includes world space position p, shading normal n, and pixel coordinates u. Then, we define the distance between two pixels as:

$$\mathrm{dist}_{pix}(d_1, d_2) = |p_1 - p_2|^2 + w_u \cdot |u_1 - u_2|^2 + t_{n_1,n_2}. \tag{2}$$

w_u is a scene-dependent parameter that controls the influence of screen-space distance, and t_{n_1,n_2} penalizes significant disparity between normals:

$$t_{n_1,n_2} = \begin{cases} 0, & \text{if } (n_1 \cdot n_2) > 0.1, \\ 1000000, & \text{otherwise.} \end{cases} \tag{3}$$

1.4 Supervoxel Clustering.

We aim to group voxels with similar visibility conditions into clusters. To represent the visibility features for each voxel, we randomly select 128 representative shading points from the image using stratified sampling. For each voxel with non-zero irradiance, we evaluate the visibility between the voxel and each representative point, packing the binary visibilities in a 128-bit feature vector R. We define our distance metric as

$$\mathrm{dist}_{\mathrm{R}}(p, q) = \mathrm{countbits}(\mathrm{R}_p \oplus \mathrm{R}_q) + w_E \cdot |E_p - E_q|, \tag{4}$$

where $\oplus$ is bitwise XOR, E_p/E_q are the irradiance of each voxel, and w_E is a parameter controlling the weight of irradiance similarity.

For clustering, we adopt a simplified version of K-means clustering with a fixed cluster count. We use 32 in our experiments.

Authors' addresses: Haolin Lu, University of California San Diego, CA, USA, hal128@ucsd.edu; Wesley Chang, University of California San Diego, CA, USA, wec022@ucsd.edu; Trevor Hedstrom, University of California San Diego, CA, USA, tjhedstr@ucsd.edu; Tzu-Mao Li, University of California San Diego, CA, USA, tzli@ucsd.edu.

0730-0301/2024/7-ART125
https://doi.org/10.1145/3658203

This begins with a serialized seeding process, K-Means++ [Arthur and Vassilvitskii 2007], to improve clustering quality and temporal stability. This is followed by a single pass in which each voxel is assigned to the cluster whose centroid has the closest distance dist_R to the voxel.

1.5 Sampling with Fuzzy Clustering

The voxel selection process assigns a distinct voxel selection distribution p_{vs} for each superpixel. However, in practice, this discretization of superpixels results in discontinuities along superpixel boundaries. This issue arises because each superpixel has different estimates of its average visibility, especially in penumbra regions, as illustrated in Fig. 1(a), leading to different supervoxel selection probabilities across different superpixels. This causes pixels across the superpixel boundaries to have a high likelihood of selecting significantly different voxels.

Several prior path guiding works [Dittebrandt et al. 2020; Müller 2019] encounter the same issue, as they discretize the distribution across octree leaves. One common solution is to introduce random jittering to the shading point's position, allowing for stochastic selection of a neighboring distribution [Binder et al. 2018]. Inspired by Colom et al. [Colom et al. 2022], we also stochastically determine which distribution to use for each shading point. However, unlike blind jittering, our approach utilizes fuzzy clustering [Bezdek et al. 1984] to wisely determine the probabilities.

To accomplish this, in the superpixel clustering stage, we also identify the four nearest superpixels for each shading point. During the sampling process, we randomly select one of these four superpixels with the probability $p(\text{SP}_i|x)$ defined as:

$$p(\text{SP}_i|x) = \frac{w_i}{\sum w_j}, \text{where } w_j = \frac{1}{\text{dist}_{pix}(d_x, d_{\text{SP}_j})} \tag{5}$$

and dist_{pix} is given in Equation 2. d_x is the data tuple of point x and d_{SP_j} is the centroid of SP_j. This effectively blurs the superpixel assignment so that there are no discontinuities across superpixel boundaries, without selecting superpixels with significantly different geometric characteristics, as illustrated in Fig. 1(d).

2 BIAS WHEN COMBINING BSDF SAMPLING AND PATH GUIDING

Consider the evaluation of indirect illumination with two samples coming from two sampling strategies: z_0 with BSDF sampling (with probability density p_0), and z_1 with path guiding. If we use the BSDF sample z_0 to fit the path guiding distribution (in our case, this is done in the light injection pass), the resulting probability distribution $p_1(z_1|z_0)$ is condition on z_0. As a result, the naïve balance heuristic MIS estimator

$$\langle L \rangle = \frac{f(z_0)}{p_0(z_0) + p_1(z_0|z_0)} + \frac{f(z_1)}{p_0(z_1) + p_1(z_1|z_0)} \tag{6}$$

is not an unbiased estimator of the integral $\int f(z)\mathrm{d}z$. The expectation of the estimator, $\int\int \langle L \rangle\, p(z_0, z_1)\mathrm{d}z_0\mathrm{d}z_1$, does not simplify to the integral.

Instead, we need to compute the marginalized probability density:

$$p_1^*(z) = \int p_1(z|z_0)p_0(z_0)\mathrm{d}z_0, \tag{7}$$

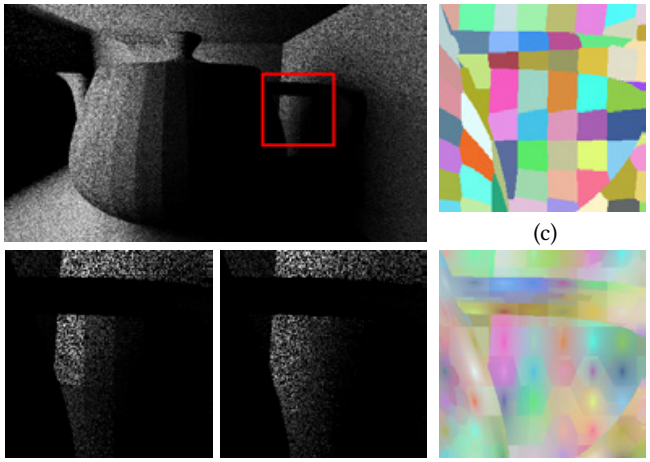

Fig. 1. Teapot scene rendered within one frame. (a) Distracting discontinuities are observed on the superpixel boundaries. (b) Fuzzy clustering effectively removes the artifacts. (c) Hard clustering visualization. (d) Fuzzy clustering visualization. Notice the edge between the handle and the background is preserved.

which does not admit a close-form solution. Note that simply providing an unbiased estimator of the integral above does not lead to an overall unbiased estimator, as expectation does not commute with division.

Practical solutions. Given the aforementioned challenges, in practice, it would be better to sidestep the computation of the marginalized probability density.

If we use a different sample z_2 with probability density $p(z_2)$ that is not used for estimating the integral to fit the path guiding distribution, then the expectation of the second term simplifies:

$$\begin{aligned} &\int\int \frac{f(z_1)}{p_0(z_1) + p_1(z_1|z_2)} p_1(z_1|z_2)p(z_2)\mathrm{d}z_1\mathrm{d}z_2 \\ &= \int p(z_2)\mathrm{d}z_2 \int \frac{f(z_1)}{p_0(z_1) + p_1(z_1|z_2)} p_1(z_1|z_2)\mathrm{d}z_1 \\ &= \int \frac{f(z_1)}{p_0(z_1) + p_1(z_1|z_2)} p_1(z_1|z_2)\mathrm{d}z_1, \end{aligned} \tag{8}$$

and the expectation of the estimator would match the integral. Therefore, if the samples used for fitting is not used in the MIS estimator, the estimator remains unbiased.

A simple solution is to do light injection using the BSDF samples from last frame. As previous path guiding methods also face the same problem, it is also their common solution [Derevyannykh 2021; Dittebrandt et al. 2020] to use samples from last frame to update the distribution. This usually introduce a one-frame time lag for adaptation but completely remove the bias. Another possible solution is trace two sets of BSDF samples for light injection and MIS respectively, but this would increase the cost.

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. IEEE Trans. Pattern Anal. Mach. Intell. 34, 11 (nov 2012), 2274–2282.

David Arthur and Sergei Vassilvitskii. 2007. K-Means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (New Orleans, Louisiana) (SODA '07). Society for Industrial and Applied Mathematics, USA, 1027–1035.

James C. Bezdek, Robert Ehrlich, and William Full. 1984. FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences 10, 2 (1984), 191–203.

Nikolaus Binder, Sascha Fricke, and Alexander Keller. 2018. Fast Path Space Filtering by Jittered Spatial Hashing. In ACM SIGGRAPH Talks. Article 71, 2 pages.

Arnau Colom, Ricardo Marques, and Luís Paulo Santos. 2022. Interactive VPL-based global illumination on the GPU using fuzzy clustering. Computers & Graphics 108 (2022), 74–85. https://doi.org/10.1016/j.cag.2022.09.008

Cyril Crassin, Fabrice Neyret, Miguel Sainz, Simon Green, and Elmar Eisemann. 2011. Interactive Indirect Illumination Using Voxel Cone Tracing: A Preview. In Symposium on Interactive 3D Graphics and Games (San Francisco, California) (I3D '11). Association for Computing Machinery, New York, NY, USA, 207. https://doi.org/10.1145/1944745.1944787

Mikhail Derevyannykh. 2021. Real-Time Path-Guiding Based on Parametric Mixture Models. arXiv:2112.09728 [cs.GR]

Addis Dittebrandt, Johannes Hanika, and Carsten Dachsbacher. 2020. Temporal Sample Reuse for Next Event Estimation and Path Guiding for Real-Time Path Tracing. In Eurographics Symposium on Rendering - DL-only Track, Carsten Dachsbacher and Matt Pharr (Eds.). The Eurographics Association. https://doi.org/10.2312/sr.20201135

Sara Fridovich-Keil*, Alex Yu*, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In CVPR.

Alexander Keller, Ken Dahm, and Nikolaus Binder. 2014. Path Space Filtering. In ACM SIGGRAPH 2014 Talks (Vancouver, Canada) (SIGGRAPH '14). Association for Computing Machinery, New York, NY, USA, Article 68, 1 pages. https://doi.org/10.1145/2614106.2614149

Thomas Müller. 2019. "Practical Path Guiding" in Production. In ACM SIGGRAPH Courses: Path Guiding in Production, Chapter 10 (Los Angeles, California). ACM, New York, NY, USA, 18:35–18:48. https://doi.org/10.1145/3305366.3328091

Jiawei Ou and Fabio Pellacini. 2011. LightSlice: matrix slice sampling for the many-lights problem. ACM Trans. Graph. (Proc. SIGGRAPH Asia) 30, 6 (2011), 179:1–179:8.

C. Y Ren, V. A. Prisacariu, and I. D Reid. 2015. gSLICr: SLIC superpixels at over 250Hz. ArXiv e-prints (Sept. 2015). arXiv:1509.04232

B. Segovia, J. C. Iehl, R. Mitanchey, and B. Péroche. 2006. Bidirectional Instant Radiosity. In Proceedings of the 17th Eurographics Conference on Rendering Techniques (Nicosia, Cyprus) (EGSR '06). Eurographics Association, Goslar, DEU, 389–397.

Bruce Walter, Adam Arbree, Kavita Bala, and Donald P Greenberg. 2006. Multidimensional lightcuts. ACM Trans. Graph. (Proc. SIGGRAPH) 25, 3 (2006), 1081–1088.

Yu-Ting Wu and Yung-Yu Chuang. 2013. VisibilityCluster: Average directional visibility for many-light rendering. IEEE Trans. Vis. Comput. Graph. 19, 9 (2013), 1566–1578.