Double Hierarchies for Efficient Sampling in Monte Carlo Rendering

Norbert Bus
Tamy Boubekeur
LTCI, Telecom ParisTech, Paris-Saclay University

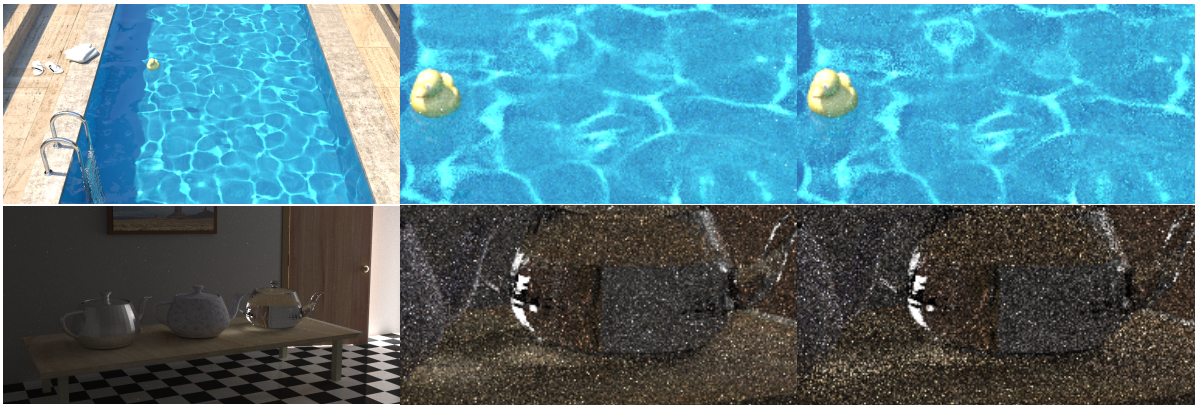

Figure 1: Renderings of difficult scenarios. From left to right: reference, GMMs [Vorba et al. 2014] and our method.

ABSTRACT

We propose a novel representation of the light field tailored to improve importance sampling for Monte Carlo rendering. The domain of the light field i.e., the product space of spatial positions and directions is hierarchically subdivided into subsets on which local models characterize the light transport. The data structure is based on *double trees*, and only approximates the exact light field, but enables efficient queries for importance sampling and easy setup by tracing photons in the scene. The framework is simple yet flexible, supports any type of local model for representing the light field, provided it can be efficiently importance sampled, and progressive refinement with an arbitrary number of photons. Last, we provide a reference open source implementation of our method.

CCS CONCEPTS

• **Computing methodologies** → **Ray tracing**; • **Mathematics of computing** → **Markov-chain Monte Carlo methods**;

KEYWORDS

Monte Carlo rendering, importance sampling

SIGGRAPH '17 Talks, July 30 - August 03, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5008-2/17/07.
https://doi.org/http://dx.doi.org/10.1145/3084363.3085063

ACM Reference format:
Norbert Bus and Tamy Boubekeur. 2017. Double Hierarchies for Efficient Sampling in Monte Carlo Rendering. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
https://doi.org/http://dx.doi.org/10.1145/3084363.3085063

1 BACKGROUND

Importance sampling for incoming light. One of the most recent approaches for representing incoming light is proposed by [Vorba et al. 2014]. It starts by creating a dense sampling of distributions by associating a hemispherical distribution with points in the scene. Then, during sampling, the distribution closest to the rendered point is queried and used to sample the incoming light. The method uses GMMs (gaussian mixture models) as distributions, but they can be replaced by any model. As the method only models the incoming light, one has to combine it with BRDF sampling using multiple importance sampling.

Representing the light field. [Ren et al. 2013] achieve real-time global illumination by utilizing a similar light field domain subdivision strategy, training neural network models that approximate the light field in each subspace. Our method shares this idea of training local models but additionally enables efficient sampling by exploiting *double tree hierarchies* [Bus et al. 2015]. Our proposed technique develops further this data structure to enable efficient sampling and training with photons.

2 METHOD

Let D be the space of hemispherical distributions and let the mapping $m : \mathbb{R}^3 \rightarrow D$ assign a distribution of incoming light to each point in a scene. In the Monte-Carlo setting $m(\cdot)$ is used for an

estimator of the integral in the rendering equation, *i.e.*, $L(x,\omega) = \frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i,\omega)L(x,\omega_i)/m(x,\omega_i)$ where $m(x,\omega_i) = m(x)(\omega_i)$ for simplicity, $L(x,\omega)$ is the radiance at $x \in \mathbb{R}^3$ in direction $\omega \in \mathbb{S}^2$, ω_i is the sampled direction and $f(\cdot,\cdot,\cdot)$ is the BRDF including the cos term. This simple notation reveals that what we seek is a function $m : \mathbb{R}^3 \times \mathbb{S}^2 \to D$, such that given any x, $m(x,\cdot)$ can be easily sampled and the marginal integral, $\int_{\Omega_x} m(x,\omega)d\omega$, is known, therefore $m(x,\cdot)$ can be normalized to obtain a distribution. Note that $m(\cdot,\cdot)$ is practically the representation of the light field in such a way that for a given x we can easily sample the corresponding distribution.

Overview. In this paper, we propose a construction that approximates the light field and possesses the previous property. The idea is very simple: we subdivide the space into subspaces and for each of the subspaces we create a local model that approximates the light field on this restricted domain. The underlying model is arbitrary, e.g., simple constant function or Gaussian mixture model, the only restriction on it is that it enables efficient sampling and marginal integral calculation. The only difficulty is how to actually obtain a hemispherical distribution – or more precisely sample it – for any point in the scene as one would have to join several local models. Our key idea is to address this issue by structuring these local distributions over a *double hierarchy* of the position-normal product space, such that sampling proper hemispherical distributions is efficient. Similarly to [Vorba et al. 2014], these local models are created from multiple batches of photons in a preprocessing phase.

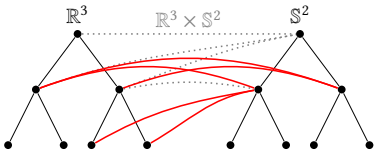

Figure 2: Double hierarchy for the product space.

2.1 Data structure

Consider a hierarchical subdivision of both $\mathbb{R}^3$ and $\mathbb{S}^2$ to a sufficient depth. Let's denote these structures by $\mathcal{R} = \{R_i^j \subseteq \mathbb{R}^3\}$ and $\mathcal{Q} = \{Q_i^j \subseteq \mathbb{S}^2\}$ where the indices are as follows: R_i^j is the jth subspace on the ith level. In practice, these structures might be any hierarchical clustering data structure, e.g., octrees. Each subspace (hereafter called node) in $\mathcal{R}$ stores an initially empty list of product space nodes (see below for their definition) and an RGB value.

We denote a hierarchical subdivision of the product space $\mathbb{R}^3 \times \mathbb{S}^2$ as $\mathcal{P}$ and restrict the subspaces to have the form $R_i^j \times Q_k^l$ with their children being formed by either subdividing R_i^j or Q_k^l into its children. Hence each product space subspace can be represented by a link between two nodes in $\mathcal{R}$ and $\mathcal{Q}$. $\mathcal{P}$ is stored as a tree structure with the same arity as $\mathcal{R}$ and $\mathcal{Q}$. See Figure 2 and note that the links between the two trees also form a tree.

A node in $\mathcal{P}$ represents a spatial subspace coupled with a subset of directions. We associate each of these nodes with a local model, $m_{R_i^j \times Q_k^l}$, representing $m(\cdot,\cdot)$ on this subspace. Since our local models need to provide quick access to their marginal integral, we restrict them to be constant along the spatial dimension.

Scene	Method	L1	RMSE	SSIM	Memory	Preproc.	Render	Total(s)
Pool	Gauss	0.213967	1.537650	0.663241	897.57	444.00	354.00	805.00
	Double	0.257429	1.132270	0.611492	1456.75	282.00	282.00	570.00
Door	Gauss	0.000226	0.002333	0.997712	938.51	816.00	528.00	1341.00
	Double	0.000255	0.001881	0.997360	2299.05	336.00	510.00	845.00

Table 1: Numerical results comparing the GMM approach by Vorba et al. (Gauss) with our method (Double).

2.2 Primitives

Our data structure is equipped with four basic primitives. *(i) Building:* depending on the exact realization, it is straightforward to build $\mathcal{R}$ and $\mathcal{Q}$. To build $\mathcal{P}$ one simply recursively subdivides product spaces clusters starting with $P_0^0 = R_0^0 \times Q_0^0$. *(ii) Refining:* each of these structures can be adaptively refined by simply adding new leaf nodes. *(iii) Training:* for each training photon we can easily descend to a leaf node of $\mathcal{P}$ and update the local distribution. Then for each node $R_i^j \times Q_k^l$ in $\mathcal{P}$ we calculate the marginal integrand of $m_{R_i^j \times Q_k^l}(\cdot,\cdot)$ and accumulate it in the RGB value of R_i^j. *(iv) Sampling:* for a given x we descend in $\mathcal{R}$ to the leaf containing x and we sum up the marginal integrands of the product space nodes stored along the path. Clearly, this sum is the marginal integrand on the whole $\mathbb{S}^2$ for x. Using this value as the normalization constant we can simply pick one product space cluster (from the links along the path of the descent) and sample its local distribution.

Given these primitives, our method works as follows: (i) prior to rendering we *build* shallow hierarchies and *train* the local models with the photons in each batch while after each batch we *refine* the structures; (ii) during rendering we *sample* the data structure.

3 EXPERIMENTS

We implemented our method as an extension of the publicly available source code of [Vorba et al. 2014] to enable easy comparison. We used an Intel Xeon E5-1660v3 CPU, and utilized two scenes (Fig. 1), providing examples of difficult scenarios. More examples are provided as supplemental materials. We have opted for simplicity hence we used octrees as hierarchical structures and simple constants as local models. We make our source code publicly available[1]. Table 1 gives error, memory and timing results. The reference has been created by the GMM based method of [Vorba et al. 2014] using 65K samples per pixel, while the test images use 256 samples. The preprocessing includes training the data structures with 30 batches of 300K photons. Even with the very simple constant model, our method performs comparably to the GMM based method in terms of quality. In terms of performance, it achieves a significant speed-up in the processing time, with an additional speed-up in the rendering time. We believe that the simplicity, flexibility and robustness of our method makes it a valuable alternative to other methods for importance sampling in Monte Carlo rendering.

REFERENCES

Norbert Bus, Nabil H. Mustafa, and Venceslas Biri. 2015. IlluminationCut. *Computer Graphics Forum (Proceedings of Eurographics 2015)* 34, 2 (2015), 561–573.

Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. 2013. Global Illumination with Radiance Regression Functions. *ACM Trans. Graph.* 32, 4 (July 2013), 130:1–130:12.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. Graph.* 33, 4, Article 101 (July 2014), 11 pages.

[1] Source code available at http://www.telecom-paristech.fr/ boubek/papers/DHS/