Efficient Multiscale Rendering of Specular Microstructure

Asen Atanasov
Charles University, Prague
Chaos Group

Jaroslav Křivánek
Charles University, Prague
Chaos Czech a. s.

Vladimir Koylazov
Chaos Group

Alexander Wilkie
Charles University, Prague

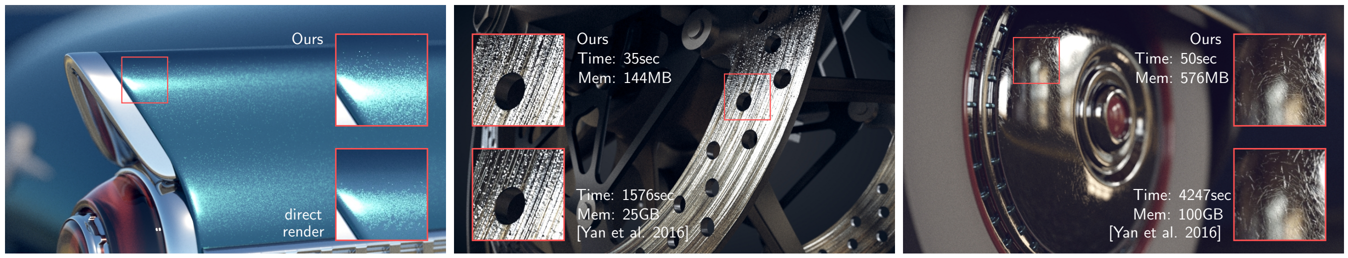

Figure 1: Our filtering method on surfaces such as car paint and scratched metal: left to right, texture resolution is 4k, 4k, and 8k and number of bins B is 127^2, 63^2 and 255^2. In the leftmost figure, the insets show the difference between our method (top) and unfiltered direct normal map evaluation (bottom). Unfiltered rendering misses many sparkles, and would flicker in an animation. The insets of the middle and right images are comparisons between our method (top) and that of Yan et al. [2016] (bottom) which is rendered with one Gaussian element per texel and intrinsic roughness $\sigma_r = 0.005$.

ABSTRACT

Texturing is a ubiquitous technique used to enrich surface appearance with fine detail. While standard filtering approaches, such as mipmapping or summed area tables, are available for rendering diffuse reflectance textures at different levels of detail, no widely accepted filtering solution exists for multiresolution rendering of surfaces with fine specular normal maps. The current state of the art offers accurate filtering solutions for specular reflection at the cost of very high memory requirements and expensive 4D queries. We propose a novel normal map filtering solution for specular surfaces which supports data pre-filtering, and with an evaluation speed that is roughly independent of the filtering footprint size. Its memory usage and evaluation speed are significantly more favorable than for existing methods. Our solution is based on high-resolution binning in the half-vector domain, which allows us to quickly build a very memory efficient data structure.

CCS CONCEPTS

• **Computing methodologies → Reflectance modeling**.

KEYWORDS

specular reflection, normal map, car paint, glints

ACM Reference Format:
Asen Atanasov, Jaroslav Křivánek, Vladimir Koylazov, and Alexander Wilkie. 2020. Efficient Multiscale Rendering of Specular Microstructure. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '20 Talks), August 17, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388767.3407338

1 RELATED WORK

Accurate rendering techniques for specular normal maps are based on microfacet theory [Walter et al. 2007]. The smooth distribution of a classical microfacet BRDF $D(\mathbf{h})$ defined for the half vector $\mathbf{h}$ is replaced by a *normal distribution function* (NDF) defined by the normal map [Yan et al. 2016]. Given a normal map with N texels, each texel has a normal $\mathbf{t}_k$ and occupies region T_k in the texture space: the NDF is defined as $D(\mathbf{h}, \mathbf{x}) = \sum_{k=1}^{N} \delta(\mathbf{t}_k, \mathbf{h}) \mathbf{I}_{T_k}(\mathbf{x})$, where δ is a spherical delta function and $\mathbf{I}_S$ is the indicator function on the set S. Note that the NDF depends on the texture space position $\mathbf{x}$. In order to facilitate efficient evaluation of the sharp spikes typical for glitter NDF, Yan et al. [2016] approximate the normal map positions and normals with 4D Gaussians referred to as *elements*. Effectively, this approximation introduces Gaussian filters for both location and normal information. These elements are sorted in a 4D hierarchy, and a traversal is required to evaluate the NDF or to sample normal proportional to it. Compared to brute force rendering, this approach offers significant improvements for point and small area lights. It is the approach closest to our own work, but we improve in several important aspects: memory efficiency, performance and pre-filtering. This allows our solution to scale well for distant views, and very high resolution normal maps.

Recently, Gamboa et al. [2018] presented a pre-filtered solution to the problem, which offers memory and speed improvements over Yan et al. [2016] via pre-filtering of the environment. However, their solution is not directly suitable for integration into a general rendering engine, due to a heavy pre-computation step, and lack of support for area lights.

In this paper, we propose a novel filtering technique: our technique is accurate, efficient at all scales, and requires only a brief preparation phase. Also, our memory requirements are comparable

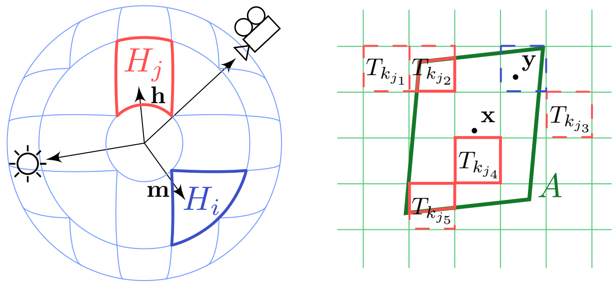

Figure 2: Left: The hemisphere partitioned into 5×5 bins seen from above. The half vector h is inside bin j. Right: The pixel footprint A in the texture space. All 5 texels corresponding to the j-th bin $\mathcal{B}^{-1}(j) = \{k_{j_1}, \dots, k_{j_5}\}$ are drawn in red. Three of them overlap with A, are in k_{A_j}, and contribute to $D_A^{H_j}$.

to mipmapping, which is at least 60 times less than current state of the art [Gamboa et al. 2018].

2 OUR SOLUTION

To achieve these improvements we apply a constant filter in both texture and half-vector spaces. Specifically, we average the NDF over a finite area A around $\mathbf{x}$ and a finite solid angle H around $\mathbf{h}$ rendering the filtered NDF $D_A^H = \frac{1}{|H||A|}\int_H \int_A D(\mathbf{h}, \mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\omega_{\mathbf{h}} \approx \frac{1}{|H||A|}\sum_{k=1}^{N} \mathbf{I}_H(\mathbf{t}_k)|A \cap T_k|$, where $|S|$ is the area of the region S. The approximation is accurate for small H. This formulation suggests that D_A^H is evaluated by summing the area of all texels inside the filtering area A whose normals are in the solid angle H. To define finite solid angles for different half vectors we partition the hemisphere into B equiangular bins H_j, hence the bin solid angle is $|H_j| = \frac{2\pi}{B}$, see projected hemisphere in Figure 2. Furthermore, we set the filtering region A to the pixel footprint defined by the ray differentials [Igehy 1999].

Next we describe how we sample a normal from the NDF and how we evaluate the NDF. Using these two procedures we implement sampling and evaluation of a microfacet BRDF [Walter et al. 2007].

2.1 Sample D_A^H

As preparation step, we quantise the normal map by assigning each of its k texels $\mathbf{t}_k$ to a directional bin H_j in a *bin map* $\mathcal{B}(k) = j$. Given a pixel footprint A, we sample a normal proportional to the NDF by sampling a random point $\mathbf{y}$ in A, and finding its corresponding bin index i from the bin map. We then sample a random normal $\mathbf{m}$ inside the bin H_i, see the blue regions in Figure 2. The probability to sample this normal is equal to the NDF $D_A^{H_i}$.

2.2 Evaluate D_A^H

A key observation is that in our context sampling and evaluation are inverse operations. Sampling computes an NDF normal $\mathbf{m}$ given a map texel, while evaluation finds all contributing map texels given a half vector normal $\mathbf{h}$. We notice that sampling based on the bin map $\mathcal{B}$ is a very efficient operation, while the evaluation is potentially very inefficient, as the number of texels in A can be almost arbitrarily large. Therefore, we design the inverse map of the bin map to facilitate the evaluation. Since the bin map $\mathcal{B}$ maps a texel index to a bin index, we define the inverse bin map $\mathcal{B}^{-1}$ as the mapping from a bin index to the list of texel indices for a given bin index: $\mathcal{B}^{-1}(j) = \{k_{j_1}, k_{j_2}, \cdots, k_{j_b}\}$, where j_b is the number of normals in bin j. For a half vector $\mathbf{h}$, we evaluate the NDF based on the inverse bin map $\mathcal{B}^{-1}$ as follows: first, we find the bin index j, such that $\mathbf{h} \in H_j$. Then we select the subset k_{A_j} of texels $\mathcal{B}^{-1}(j)$ which lie in A: in Figure 2, only $\{k_{j_2}, k_{j_4}, k_{j_5}\}$ are in A.

2.3 Implementation details

During the preparation step we compute and store the two N-element integer maps $\mathcal{B}$ and $\mathcal{B}^{-1}$ based on the input normal map which we do not keep afterwards. Note that in a typical normal map often $|\mathcal{B}^{-1}(j)|$ is too large for linear traversal, therefore, we devise a combination of two auxiliary data structures that make access to $\mathcal{B}^{-1}$ efficient. The first observation which led us to this design was that for a large number of bins, the nonempty entries are sparse. Therefore, we use a hash map that maps a bin index to its texel list in $\mathcal{B}^{-1}$. Furthermore, some bin lists in $\mathcal{B}^{-1}$ are too large to be traversed linearly. To optimise their traversal, we build a forest of 2D kd trees, one for each texel list in $\mathcal{B}^{-1}$ larger than a given value L (16 in our implementation). L is the maximum leaf size of the forest. During the construction of the forest we follow several conventions. First, we always split the larger side of the node in the middle, so that split dimension and split position do not need to be stored. Second, we sort each texel list so that for each tree node its texels are consecutive in $\mathcal{B}^{-1}$. This serves two purposes for the traversal: cache coherence is improved, and the size of each node is implicitly propagated as offsets to the node start and end in $\mathcal{B}^{-1}$. This property provides pre-filtering data, so if a node is entirely inside the filtering region A its total area is immediately returned. Lastly, to achieve memory efficiency and to favour serialisation, we pack the whole forest topology in a single integer list.

Our benchmarks confirmed that in contrast to existing techniques [Yan et al. 2016], the memory and run time requirements of our method are compatible with use in a production renderer. Sample images rendered with our technique are presented in Figure 1, along with timing and memory usage.

ACKNOWLEDGMENTS

We are grateful to Veselin Mihaylov for preparing the scenes and to Lingqi Yan for providing us with his code.

REFERENCES

Luis E. Gamboa, Jean-Philippe Guertin, and Derek Nowrouzezahrai. 2018. Scalable Appearance Filtering for Complex Lighting Effects. *ACM Trans. Graph.* 37, 6, Article 277 (Dec. 2018), 13 pages. https://doi.org/10.1145/3272127.3275058

Homan Igehy. 1999. Tracing Ray Differentials. In *Proceedings of SIGGRAPH '99*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 179–186. https://doi.org/10.1145/311535.311555

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques (EGSR'07)*. Eurographics Association, Aire-la-Ville, Switzerland, 195–206. https://doi.org/10.2312/EGWR/EGSR07/195-206

Ling-Qi Yan, Miloš Hašan, Steve Marschner, and Ravi Ramamoorthi. 2016. Position-Normal Distributions for Efficient Rendering of Specular Microstructure. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2016)* 35, 4 (2016).