EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)

Volume 40 (2021), Number 2

A Multiscale Microfacet Model Based on Inverse Bin Mapping

Asen Atanasov[1,2] Alexander Wilkie[1] Vladimir Koylazov[2] Jaroslav Křivánek[1,3]

[1]Charles University, Prague
[2]Chaos Software
[3]Chaos Czech a. s.

Figure 1: *A photograph of wing mirror (left) with pronounced glint from metallic flakes that served as an inspiration for our wing mirror scene (middle). The metallic flakes are modelled with a 2K normal map with flakes sampled from a GTR distribution (GTR gamma = 1.5, GTR alpha = 0.002) [Bur12]. Additionally, the roughness of the flakes is modelled with a Beckmann distribution with Beckmann alpha = 0.005. The flake roughness contributes to the overall appearance, and is a useful parameter for artistic control. To the right we provide three regions from the same scene rendered with different Beckmann flake roughness (0.0025, 0.01, 0.04). Small perturbations of the roughness of the flakes completely change the behaviour of the glints. The rendering of such nearly specular surfaces requires some form of filtering, the effect of which is shown in our accompanying video. All the renderings in this figure were done with our proposed normal map filtering algorithm.*

Abstract

Accurately controllable shading detail is a crucial aspect of realistic appearance modelling. Two fundamental building blocks for this are microfacet BRDFs, which describe the statistical behaviour of infinitely small facets, and normal maps, which provide user-controllable spatio-directional surface features. We analyse the filtering of the combined effect of a microfacet BRDF and a normal map. By partitioning the half-vector domain into bins we show that the filtering problem can be reduced to evaluation of an integral histogram (IH), a generalization of a summed-area table (SAT). Integral histograms are known for their large memory requirements, which are usually proportional to the number of bins. To alleviate this, we introduce Inverse Bin Maps, a specialised form of IH with a memory footprint that is practically independent of the number of bins. Based on these, we present a memory-efficient, production-ready approach for filtering of high resolution normal maps with arbitrary Beckmann flake roughness. In the corner case of specular normal maps (zero, or very small roughness values) our method shows similar convergence rates to the current state of the art, and is also more memory efficient.

CCS Concepts

• ***Computing methodologies*** → ***Rendering; Reflectance modeling;***

1. Introduction

Since its beginning, the quest for photorealistic computer-generated images has had researchers focus on physically based models of reflectance, both for entire surfaces, as well as for detailed structures. Microfacet theory was developed in the optics community before the advent of computer graphics [BS63; TS67], was only later introduced to graphics by Cook et al. [CT82], and became an essential tool in the field [WMLT07]. Microfacet theory assumes that surfaces are made up of a collection of statistically distributed reflective facets, and it describes their aggregate directional behaviour. However, the resulting purely homogeneous surface appearance only matches our visual experience when viewing objects from rather large distances. For the closeup and mid-range views, which are much more common in our everyday experience, the effect of light interacting with individual details of surface structure can often be resolved by the naked eye.

Independently of microfacet approaches, Blinn et al. [Bli78] presented bump mapping, a technique which adds fine surface detail by perturbing the surface normal according to a heightfield that is provided via a texture. When the tangent space normal is directly stored in the texture, this technique is referred as normal mapping.

Rendering the combined BRDF of a normal map used on a microfacet BRDF was investigated from the perspective of microfacet theory by Schüssler et al. [SHHD17]: this is a problem of high practical relevance. Virtually all rendering systems support normal mapping and the user can additionally control the roughness of the underlying surface. For example, a metal surfaces with different roughness can be represented by a microfacet BRDF while scratches can be added through a normal map. Normal-mapped surfaces with roughness close to zero produce spatially varying, illumination- and view- dependent micro-highlights referred to as *glints*, *sparkles*, coherent scratches, etc. Metallic car paint flakes can be modelled via normal map [GCG*05]. While their surface can be modelled as specular, measurements support that the roughness of the individual flakes has an important contribution to the overall appearance [SNM*02].

In Section 4, we investigate the problem of properly filtering such a combined microfacet BRDF. Filtering techniques like mip mapping [Wil83] and summed-area tables [Cro84] are available for diffuse color textures, but they do not work for normal maps due to the nonlinearity of the reflection operator [HSRG07]. We show that the normal map filtering problem can be solved using a generalized summed-area table known as *integral histogram (IH)* [Por05]. In section 5, we develop an accurate and efficient filtering algorithm for Beckmann flake roughness that can be implemented using IH. However, the memory requirements of standard IH implementations make them impractical to use for non-trivial scenes. Therefore, we introduce a new optimized form of IH, the Inverse Bin Map (IBM), which is very fast to build, and which has modest memory requirements, comparable with mip maps and SATs. The contributions of our work are:

- We show that the filtering problem of the combined effect of a normal map and a microfacet BRDF can be reduced to the evaluation of an Integral Histogram.
- We introduce the Inverse Bin Map (IBM) - a novel implementation of integral histograms with a memory footprint similar to the size of the input data and not proportional to the number of histogram bins. Additionally, our data structure is fast to build, and naturally supports arbitrary query regions.
- An accurate and efficient filtering algorithm for Beckmann microfacet BRDF based on IBMs.

2. Related work

2.1. Explicit micro-structure modeling.

Ershov et al. [EKK99; EKM01] simulated metallic car paint using a statistical model: the main drawback of their approach was that appearance was not consistent in animations. Günther et al. [GCG*05] simulated metallic car paint glitter based on non-filtered procedurally generated normal maps. We take the same approach to render car paint and show that filtering is crucial to resolve such sharp glints. Rump et al. [RMS*08] rendered car paint using measured data. Jakob et al. [JHY*14] developed a model based on microfacet theory which uses a stochastic process to compute temporally consistent sparkling. This method can be used to render metallic flakes, but flake roughness and size and transparency of the flake layer cannot easily be included in the model. Zirr et al. [ZK16] presented a real-time approach capable of rendering sparkling flakes and parallel scratches. Later, Chermain et al. [CSJD20] developed a real-time approach to render glitter that additionally converges to the microfacet BRDF for high flake densities. Methods specialized for efficient rendering of scratches have been recently developed [RGB16; WVJH17; VWH18]. Kuznetsov et al. [KHX*19] simulate materials with stochastic nature like flakes and scratches with a pre-trained neural network. These approaches are limited to specific spatial details, thus have limited expressiveness.

2.2. Normal map filtering.

Approaches based on normal maps are more flexible since they can represent the spatial and directional features given by the map. Therefore, efficient implementation of normal map filtering is a very important problem for production rendering systems. Approaches that approximate the actual distribution of normals inside the pixel with a single lobe [Tok05; OB10; DHI*13] offer artefact-free solutions, and are compatible with real-time graphics, but high frequency detail like sharp sparkling is lost [YHJ*14]. Han et al. [HSRG07] investigated the combined effect of an isotropic BRDF and a normal map and provided filtering techniques that approximate the distribution of normals by a small number of lobes. Notably, Wu et al. [WZYR19] developed a method for pre-filtering of displacement-mapped surfaces with isotropic BRDFs which accounts for accurate shadowing-masking and interreflections. The method does not support high directional resolution to render closely viewed specular surfaces.

A family of accurate approaches for specular normal maps inherits the mathematical framework of Yan et al. [YHJ*14] which represents the NDF as a convolution of a Gaussian footprint around the shading point and a Gaussian intrinsic roughness lobe around the normal map directions. This definition leads to 4D texture-direction Gaussian queries to evaluate the NDF. Later approaches improve performance [YHMR16], compute antialiasing for global illumination effects [BYRN17], derive accurate shadowing-masking factors

using approximation with anisotropic Beckmann lobes [CCM18], and introduce wave effects [YHW*18]. All of them have high memory requirements and are based on expensive 4D position-normal queries. Zhu et al. [ZXW19] developed a method based on the method of Yan et al. [YHMR16] which offers memory reduction for the special case of normal maps with a block structure. Wang et al. [WHHY20] generate an infinite surface from a small example map via by-example blending. Memory usage is 35MB for a 512^2 map, which can be reduced by at least 10% when flat Gaussian elements[YHMR16] are used. Both Zhu et al. [ZXW19] and Wang et al. [WHHY20] are intended to render textures with predominantly stationary structure, and without macroscopic features. Recently, Gamboa et al. [GGN18] explored the combined filtering of specular normal maps and an environment illumination. The additional pre-filtering of the incident illumination by projecting the environment to a spherical harmonics (SH) basis is a key advantage of this technique. The pre-filtering of the SH coefficients is achieved by a spherical histogram, which is an integral histogram, that is constructed over spherical bins. This aspect is similar to our solution, although the use of classical IH attributes to the large memory requirements of the method (2.3-2.7GB for 2K maps), and filtering queries are restricted to axis-aligned regions: our proposed method usually uses less than 40MB (for 2K maps), which is at least 60 times less memory. Practical disadvantages are a lack of support for area light sources, and typically at least minutes of pre-computation time that are needed for SH projections. Recently, specular manifold sampling (SMS) method was demonstrated to render glints with modest memory requirements and with similar convergence rates [ZGJ20]. The method also has a brief pre-computation: only a LEAN map [OB10] is built. However, SMS does not employ an acceleration data structure to find the glints in the pixel footprint, and instead relies on stochastic sampling. This strategy becomes inefficient for an increasing number of glints in the footprint.

2.3. Discussion

The method of Yan et al. [YHMR16] provides two modes of operation: flat Gaussian elements which represent the non-interpolated normal map, and curved Gaussian elements which represent a smooth interpolated surface. We support only flat un-interpolated normal maps, and therefore our surface is very similar to Yan's flat elements, see Figure 2. Our proposed technique aims to provide a practical filtering solution not only for specular surfaces, but also for surfaces with low roughness where the appearance changes dramatically, but filtering is still beneficial. Our solution exposes a single parameter, Beckmann roughness, which provides meaningful artistic control. This parameter, which we also refer to as *Beckmann flake roughness*, is conceptually similar to the intrinsic roughness of Yan et al. [YHMR16]. But the memory requirements of their method are very high, see Table 2. Furthermore, the intrinsic roughness is designed and demonstrated to work in a small operational range that represents specular surfaces. Surfaces with a slightly larger roughness are out of the scope of Yan et al., and the method quickly loses energy, see Figure 3. Note that in all comparisons we match our Beckmann flake roughness to Yan's intrinsic roughness using the relation $\alpha = \sqrt{2}\sigma_r$ [Hei14].

The method of Yan et al.[YHMR16] must locate all Gaussian elements inside the pixel that contribute to the reflection. Then contributing elements have to be processed individually and weighted against the Gaussian footprint. This could be inefficient for scenes with high texel-to-pixel ratios and many contributing elements. For example, such inefficiency is demonstrated in the convergence plots of Yan's curved elements in Figure 7 and we discuss it in Section 6.1. Note that altering the texel-to-pixel ratio is an extremely common scenario in practical renderer usage: when the object is moving away from the camera or the camera is zooming out, when the scene is rendered at lower resolution, or when the tiling of the texture is increased. In our method, the contributing texels in each pixel increase with Beckmann flake roughness. Therefore, our data structure provides pre-filtering: the aggregated projected area of texels with similar normals inside the pixel is efficiently computed for high texel-to-pixel ratios. Our data structure is extremely fast to build and uses less memory than Yan's flat elements.

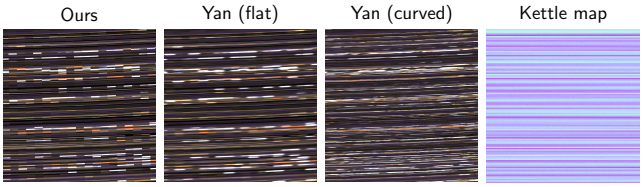

Figure 2: ***Kettle scene*** *[ZGJ20]: 50× zoom is applied to observe the surfaces of the three different models. From left to right: our method, Yan[2016] flat elements and Yan[2016] curved elements. The rightmost image is the normal map. Our method does not support normal map interpolation and renders a piecewise flat surface similar to the flat Gaussian elements of Yan et al. [YHMR16]. Yan's flat elements are blurrier due to the Gaussian footprint whereas our method uses pixel-wide "box" filter. The kettle scene can be seen in Figure 7.*

2.4. Integral histograms

Integral histograms (IH) were first introduced in the field of computer vision [Por05], and became a fundamental tool for image analysis and processing that has numerous applications[BP19]. Due to their high memory requirements WaveletSAT [LS13] was developed, offering lossless compression of IH at the cost of reducing the query complexity from constant to logarithmic. Compression rates are commonly 1:8[BP19]. Recently, Ballester-Ripoll and Pajarola [BP19] proposed a lossy compression scheme for IH based on tensor decomposition with higher compression rates and extended IH queries for arbitrary regions: however, the price for this are slower retrieval times. In principle, our algorithm and other variations based on the binned BRDF described in Section 4 can be implemented with any of these three data structures - classical IH, WaveletSAT and the tensor decomposition scheme. Due to the high number of bins required by our solution, and potentially high-resolution normal map, the memory requirements of the first two are too high for our practical application. The method of Ballester-Ripoll et al. [BP19] provides significantly more flexibility in terms of supported query regions, but it is not viable for our problem, due to its pre-processing time that can be up to several hours.

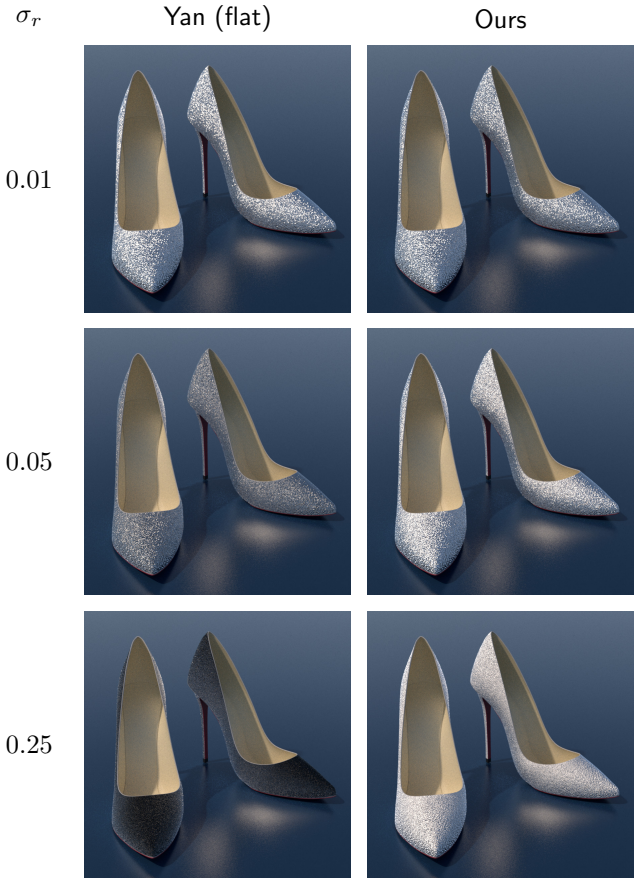

Figure 3: ***Shoes scene*** *[ZGJ20]: Renders with Gaussian flat elements (left column) [YHMR16] and with our method (right column). The three rows show increasing roughness* σ_r $(0.01, 0.05, 0.25)$. *Note that the method of Yan et al. is designed to render specular surfaces and does not conserve energy for increased roughness.*

3. Background

We present the main constructs that we build on to develop our filtering algorithms.

3.1. Microfacet BRDF

The microfacet BRDF with specular microfacets [WMLT07] is:

$$f^\alpha(\mathbf{i},\mathbf{o},\mathbf{n}) = \frac{F(\mathbf{i},\mathbf{h})D^\alpha(\mathbf{h},\mathbf{n})G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{n})}{4(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})}. \tag{1}$$

We use similar notation to Walter et al. [WMLT07], see Table 1. However, their definitions of the microfacet distribution D and the shadowing-masking function G depend implicitly on the macrosurface normal $\mathbf{n}$ and the roughness parameter α - note that all angles are defined with respect to $\mathbf{n}$. For clarity of our derivation we make both dependences explicit. Importantly, microfacet BRDFs approach the specular BRDF [WMLT07] as the roughness diminishes

$$\lim_{\alpha\to 0} f^\alpha(\mathbf{i},\mathbf{o},\mathbf{n}) = \frac{F(\mathbf{i},\mathbf{h})\delta(\mathbf{h},\mathbf{n})}{4(\mathbf{i}\cdot\mathbf{h})^2}. \tag{2}$$

Table 1: *Table of notation.*

Macrosurface-related symbols	
$\mathcal{H}^2$	Unit hemisphere
$\mathcal{D}$	Unit disk
$\mathbf{i}$	Incoming light direction
$\mathbf{o}$	Outgoing light direction
$\mathbf{n}$	Surface normal at position x
$\mathbf{h}$	Half vector $\mathbf{h} = (\mathbf{i}+\mathbf{o})/\|\mathbf{i}+\mathbf{o}\|$
x	Position on a macrosurface
A	Finite region in texture space around x
f_x	Combined BRDF at position x
f_A	Filtered BRDF over A
F	Fresnel term for conductors or dielectrics
Microsurface-related symbols:	
$\mathbf{t}_k$	Normal of k-th texel
T_k	Texture space region corresponding to k-th texel
N	Total number of normal map texels
w_k	Texel weight $\|A \cap T_k\|/\|A\|$
H_j	Bin on the hemisphere with index j
B	Total number of bins
W_j	Bin weight $\sum_{k\|\mathbf{t}_k\in H_j} w_k$
β, β^{-1}	Binning strategy and its inverse
$\mathcal{B}$, $\mathcal{B}^{-1}$	Bin Map and Inverse Bin Map
D_x	Normal distribution function (NDF)
G_x	Shadowing-masking function of the normal map
α	Beckmann flake roughness
f^α	Microfacet micro-BRDF aligned with the microsurface
D^α	Microfacet distribution of the micro-BRDF
G^α	Shadowing-masking function of the micro-BRDF
C	Texel contribution function
Other symbols:	
$\|X\|$	Surface area of the region X
I_X	Indicator function of the region X
δ	Spherical delta function
σ_r	Yan's intrinsic roughness [YHJ*14; YHMR16]

Indeed microfacet materials without roughness have all facets aligned with the macrosurface: the microfacet distribution D^α becomes a Dirac delta distribution, there is no shadowing-masking $G^\alpha = 1$, and the expression is nonzero only when $\mathbf{h} = \mathbf{n}$.

3.2. Normal map

The normal map is a collection of N texels, each occupying an equal rectangular region T_k of unit texture space, and it is associated with a normal in tangent space $\mathbf{t}_k \in \mathcal{H}^2$. Alternatively, the normals can be defined as points on the unit disk $\mathcal{D}$ [YHJ*14].

The normal distribution function (NDF) based on the normal map is

$$D_x(\mathbf{m}) = \sum_{k=1}^{N} \delta(\mathbf{t}_k,\mathbf{m}) I_{T_k}(x), \tag{3}$$

which is dependent on the texture space position x and I_{T_k} is the

indicator function of the texel region T_k. Our definition is equivalent to the one in Han et al. [HSRG07], however we do not divide explicitly by N, because in our definition the texel area is $|T_k| = \frac{1}{N}$.

3.3. Combined BRDF

In order to use microfacet BRDFs and normal maps together in a way that is in agreement with assumptions of microfacet theory we follow a derivation similar to Schüssler et al. [SHHD17]. We substitute D_x for the microfacet distribution and f^α for the micro BRDF in the general formula for microfacet BRDF with a micro BRDF assigned to each microfacet [WMLT07]

$$\begin{aligned} f_x(\mathbf{i},\mathbf{o},\mathbf{n}) &= \int_{\mathcal{H}^2} \frac{(\mathbf{i}\cdot\mathbf{m})(\mathbf{o}\cdot\mathbf{m})}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} f^\alpha(\mathbf{i},\mathbf{o},\mathbf{m}) D_x(\mathbf{m}) G_x(\mathbf{i},\mathbf{o},\mathbf{m}) \mathrm{d}\omega_m \\ &= \sum_{k=1}^{N} \frac{(\mathbf{i}\cdot\mathbf{t}_k)(\mathbf{o}\cdot\mathbf{t}_k)}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} f^\alpha(\mathbf{i},\mathbf{o},\mathbf{t}_k) G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k) I_{T_k}(x), \end{aligned} \tag{4}$$

where $\mathbf{n}$ is the surface normal at position x, $\mathbf{m}$ is the microsurface normal and G_x is the shadowing-masking function corresponding to D_x. As previous work, we use Smith shadowing-masking function of the normal map G_x [YHMR16]. The integration is defined over the unit hemisphere $\mathcal{H}^2$ for an infinitesimal solid angle $\mathrm{d}\omega_m$ around the micronormal $\mathbf{m}$. The delta function in the definition of D_x breaks the integral into a sum over all texel normals $\mathbf{t}_k$. Furthermore, only the term for which $I_{T_k}(x)$ is nonzero, the term for which $x \in T_k$. When we expand f^α in this term we reach the desired combined BRDF of a normal map and a microfacet BRDF

$$\begin{aligned} f_x(\mathbf{i},\mathbf{o},\mathbf{n}) &= \frac{(\mathbf{i}\cdot\mathbf{t}_k)(\mathbf{o}\cdot\mathbf{t}_k)}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} \frac{F(\mathbf{i},\mathbf{h}) D^\alpha(\mathbf{h},\mathbf{t}_k) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{t}_k)}{4(\mathbf{i}\cdot\mathbf{t}_k)(\mathbf{o}\cdot\mathbf{t}_k)} G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k) \\ &= \frac{F(\mathbf{i},\mathbf{h}) D^\alpha(\mathbf{h},\mathbf{t}_k) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{t}_k)}{4(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k) \\ &= F(\mathbf{i},\mathbf{h}) C(\mathbf{i},\mathbf{o},\mathbf{t}_k,\mathbf{n}), \end{aligned} \tag{5}$$

where $C(\mathbf{i},\mathbf{o},\mathbf{t}_k,\mathbf{n}) = D^\alpha(\mathbf{h},\mathbf{t}_k) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{t}_k) G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k) / (4(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n}))$ is the texel contribution.

3.4. Integral histogram (IH)

For our exposition we describe 2D IHs, however they are directly generalized to higher dimensions. We start by defining the related concept of a summed-area table (SAT) [Cro84]. It is a cumulative table of a 2D array used for fast integral look-ups in arbitrary axis-aligned regions. Given a 2D array $I(i,j) \in \mathbb{R}$, the SAT of I is again a 2D array of the same size

$$\mathrm{SAT}_I(i,j) = \sum_{k=0}^{i} \sum_{l=0}^{j} I(k,l). \tag{6}$$

Given an arbitrary axis-aligned region $i_0 \le i \le i_1,\ j_0 \le j \le j_1$ in I, the SAT is used to efficiently look up the sum of I over it

$$\mathrm{SAT}_I(i_1,j_1) - \mathrm{SAT}_I(i_1,j_0) - \mathrm{SAT}_I(i_0,j_1) + \mathrm{SAT}_I(i_0,j_0). \tag{7}$$

Note that this is a discrete application of the Fundamental Theorem of Calculus in 2D.

Integral Histograms (IHs) are a natural extension of SATs. For a 2D data set with specified binning strategy, the idea is to build SATs on the indicator functions of the bins[BP19]. Specifically, for a data set I with binning β the indicator functions are

$$I_b(i,j) = \begin{cases} 1, & \beta(I(i,j)) = b \\ 0, & \text{otherwise} \end{cases}, \quad 0 \le b \le B \tag{8}$$

where B is the number of bins. The IH of the data set then is

$$\mathrm{IH}_{I,\beta} = \{\mathrm{SAT}_{I_0}, \cdots, \mathrm{SAT}_{I_{B-1}}\} \tag{9}$$

Consequently, the histogram of any axis-aligned subregion of I can be extracted by evaluating Equation 7 for each bin.

In their original form, IHs are ideal due to their fast look-ups, but they have three considerable disadvantages:

- **High memory requirements:** the higher the number of bins the sparser the bin indicator functions are. This redundancy increases the memory footprint proportionally to the number of bins. Therefore, it is not unusual for IHs to be unviable due to them exceeding the available memory for a given task[BP19].
- **Slow construction:** the construction speed can be too slow for some applications, especially for large number of bins.
- **Only axis-aligned look-ups:** traditional IHs with fast look-up are restricted to axis-aligned rectangle regions. In turn, this would imply that we would have to use axis-aligned pixel footprints $A^\perp$, which is undesirable.

4. Normal map filtering

Proper filtering of normal maps is a challenging problem. In this section, we analyse the filtering of our combined microfacet BRDF f_x.

4.1. Filtered BRDF

We define the filtered BRDF f_A by averaging a spatially varying BRDF f_x over a finite texture space region A around x:

$$f_A(\mathbf{i},\mathbf{o},\mathbf{n}) = \frac{1}{|A|} \int_A f_x(\mathbf{i},\mathbf{o},\mathbf{n}) \mathrm{d}x. \tag{10}$$

Note that on both sides of the equation, $\mathbf{n}$ is the normal at position x. Like all previous filtering techniques we rely on the assumption of a locally flat geometry. In practice this is a source of bias, which is however sufficiently small when the surface normal $\mathbf{n}$ does not change considerably over the region A. The filtered BRDF can be defined more generally using an arbitrary filter kernel, but we use the constant "box" filter $\frac{1}{|A|}$, because this will lead to an efficient implementation later.

Using Equation 4 for f_x we expand f_A by substituting f^α and distributing the integration over the normal map texels T_k

$$f_A(\mathbf{i},\mathbf{o},\mathbf{n}) = F(\mathbf{i},\mathbf{h}) \sum_{k=1}^{N} C(\mathbf{i},\mathbf{o},\mathbf{t}_k,\mathbf{n}) w_k, \tag{11}$$

where

$$w_k = \frac{|A \cap T_k|}{|A|}. \tag{12}$$

Texel weights w_k sum up to 1 and they are nonzero when the texels T_k cover the filtering region A.

Therefore, evaluation of this filtered BRDF requires a loop over all normal map texels which happen to be fully or partially contained in the filtering region A. For a small number of texels in A this formulation is actually the most efficient way to compute the filtered BRDF, but as A grows to cover more texels, this formula becomes impractical. For such scenarios we derive alternative formulation for f_A.

4.2. Binned BRDF

We partition the hemisphere $\mathcal{H}^2$ into directional bins $H_j \subset \mathcal{H}^2$ such that $H_i \cap H_j = \emptyset$, for $i \neq j$ and $\cup_{j=1}^{B} H_j = \mathcal{H}^2$, where B is the total number of bins. Each normal from the map belongs to a bin $\mathbf{t}_k \in H_j$ and therefore for a sufficiently fine binning all normals inside a bin are nearly identical. This allow us to group the terms in f_A by bin index j

$$f_A(\mathbf{i},\mathbf{o},\mathbf{n}) \approx F(\mathbf{i},\mathbf{h}) \sum_{j=1}^{B} C(\mathbf{i},\mathbf{o},\mathbf{b}_j,\mathbf{n}) W_j, \tag{13}$$

where $\mathbf{b}_j \in H_j$ is a normal in bin j and the bin weights are $W_j = \sum_{k|\mathbf{t}_k \in H_j} w_k$. Consequently, the bin weights W_j also sum up to one like the texel weights w_k. The bin weight represent what portion of the surface in the region A has normal in a given bin.

Note that if we build an IH of the normal map with a given binning we can compute the bin weights W_j in Equation 13 efficiently for any axis-aligned region in texture space $A^{\perp}$, that additionally do not split texels.

5. Our solution

There are practical concerns regarding direct use of Equation 13 with an IH. Although, a single bin can be queried cheaply using Equation 7, large number of bins can lead to a significant overhead. The other three practical challenges stem from the classical IH and are discussed in Section 3.4. In this section we address these issues and describe our filtering algorithm.

5.1. Beckmann flake roughness

First, we define the micro BRDF f^{α}. We use the Beckmann microfacet BRDF with Smith shadowing-masking function [WMLT07] for G^{α}. The Beckmann microfacet distribution is

$$D^{\alpha}(\mathbf{h},\mathbf{t}_k) = \frac{\chi^+(\mathbf{h}\cdot\mathbf{t}_k)}{\pi\alpha^2\cos^4(\theta)} \exp\frac{-\tan^2\theta}{\alpha^2}, \tag{14}$$

where θ is the angle between $\mathbf{h}$ and $\mathbf{t}_k$, and $\chi^+(a)$ is one for $a > 0$ and zero for $a \leq 0$. It is a Gaussian distribution of slopes with standard deviation $\sigma = \frac{\alpha}{\sqrt{2}}$ [Hei14]. The tails of D^{α} are exponentially bounded, and therefore, 95% of its microfacets are within 2σ and nearly all of them are contained within 3σ. We chose the Beckmann distribution with this property in mind, because the bin weights W_j in Equation 13 do not need to be computed for all bins outside of this region. These terms of the equation will be multiplied by the tails of D^{α} and will have tiny contribution to the BRDF value. Given a threshold 3σ where we cut the tails, the truncated part of D^{α} lies in a cone of angle $\theta_0 = \arctan(3\sigma)$ with radius $\sin(\theta_0)$. Subsequently,

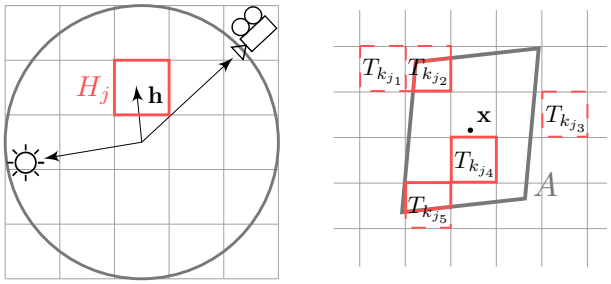

Figure 4: *Left: The unit disk partitioned into 5×5 bins. The half vector $\mathbf{h}$ is inside bin H_j. Right: The pixel footprint A in the texture space. All 5 texels corresponding to the j-th bin $\mathcal{B}^{-1}(j) = \{k_{j_1},\ldots,k_{j_5}\}$ are drawn in red. Three of them overlap with A, are in k_{A_j}, and contribute to W_j.*

we can evaluate Equation 13 by only considering a smaller number of bins B_0 that sufficiently cover this cone:

$$f_A(\mathbf{i},\mathbf{o},\mathbf{n}) \approx F(\mathbf{i},\mathbf{h}) \sum_{j=1}^{B_0} C(\mathbf{i},\mathbf{o},\mathbf{b}_j,\mathbf{n}) W_j \tag{15}$$

5.2. Bin strategy

We partition the bounding square $[-1,1]^2$ of the unit disk $\mathcal{D}$ uniformly into $b \times b$ bins, each with index $j \in [0, b^2-1]$, see Figure 4 (right). Each normal on the unit disk belongs to a bin and the bins that do not overlap with $\mathcal{D}$ are always empty ($B \leq b^2$). Then we can efficiently implement the binning function β and its inverse:

- $\beta(\mathbf{m}) = j$: Given a normal $\mathbf{m} \in \mathcal{H}^2$, we can find the index of the bin j which contains it: $j = \lfloor b(0.5\mathbf{m}_x + 0.5)\rfloor + b\lfloor b(0.5\mathbf{m}_y + 0.5)\rfloor$.
- $\beta^{-1}(j,\xi_0,\xi_1) = \mathbf{m}$: We can sample a random normal inside a given bin j: $(\mathbf{m}_x,\ \mathbf{m}_y) = (2(\lfloor j\%b\rfloor + \xi_0)/b - 1,\ 2(\lfloor j/b\rfloor + \xi_1)/b - 1)$, where ξ_0 and ξ_1 are uniform random variables.

Our key idea is to choose the bin resolution b depending on the flake roughness α in such way that the number of contributing bins B_0 is a small constant. Thus, surfaces with lower roughness will have higher bin resolutions. This can be done in a number of ways, but we found that the following approach works well in practice. We take a square neighbourhood of bins centered at the bin which contains the half vector by taking two bins in each direction, a total of 25 bins. Let the 3σ cone of radius $\sin(\theta_0)$ is centered around the half vector $(0,0,1)$. Our goal is to choose b so that the neighbourhood of bins sufficiently covers this cone. We set b in such a way that the ratio between the 25 neighbourhood bins and the total number of b^2 bins approximates the ratio between the area of the cone's bounding square $4\sin^2(\theta_0)$ and the square of area 4 which bounds the unit disk:

$$b = \left\lfloor \frac{5}{\sin(\theta_0)} \right\rfloor. \tag{16}$$

Lastly, we omit the four corners of this square neighbourhood, be-

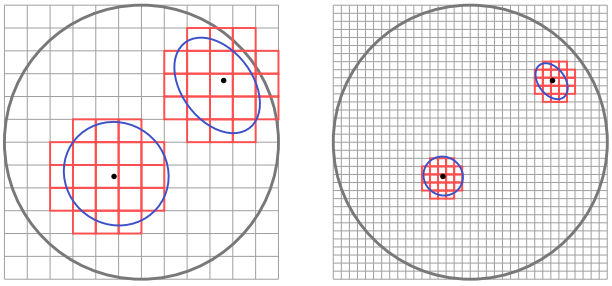

Figure 5: *Bin size vs. Beckmann flake roughness. Shown is a unit disk with entries for two half vectors (black dots), and their corresponding* 3σ *cones (blue). Left: bin resolution* 12×12 *for Beckmann flake roughness* $\alpha = 0.2$ *and Right:* 34×34 *for* $\alpha = 0.07$*. See Table 4 for the memory usage of our method for different bin resolutions.*

cause they are mostly outside the 3σ cone and end up with $B_0 = 21$, see Figure 5.

5.3. Evaluation and sampling

Using β we transform the input normal map into a *bin map* $\mathcal{B}$, which is an integer map of bin indices that correspond to each normal map direction:

$$\mathcal{B}(k) = \beta(\mathbf{t}_k) = j, \quad \mathbf{t}_k \in H_j. \tag{17}$$

Once the bin map is computed we do not store the original normal map. All necessary components of our algorithm are based on $\mathcal{B}$. We evaluate the BRDF as follows:

- Find the bin of the half vector $j = \beta(\mathbf{h})$
- Find all nearby $B_0 - 1$ bins that belong to the neighbourhood of j, see Figure 5
- Compute all B_0 bin weights W_j of these bins
- For all nonzero bin weights compute the texel contribution function C at bin centers and accumulate the result, see Equation 15
- Apply Fresnel term for conductors or dielectrics

In order to use this BRDF in the multiple importance sampling (MIS) framework [VG95], we also provide a sampling technique for this BRDF:

- Sample random point in A
- Find the corresponding texel T_k
- Look up its bin index from the bin map $j = \mathcal{B}(k)$
- Compute the center normal of this bin $\mathbf{b}_j = \beta^{-1}(j, 0.5, 0.5)$
- Generate a random normal with Beckmann distribution $\mathbf{m} \propto D^\alpha$ centered at $\mathbf{b}_j$
- Compute the reflected direction $\mathbf{i} = \text{reflect}(\mathbf{o}, \mathbf{m})$

The corresponding probability is

$$p(\mathbf{i}) = \frac{1}{4(\mathbf{i}\cdot\mathbf{h})} \sum_{j=1}^{B_0} D^\alpha(\mathbf{h}, \mathbf{b}_j) W_j. \tag{18}$$

The rest of this section describes how we compute the bin weights W_j.

5.4. Inverse Bin Map

We developed a novel variant of IH data structures that we call *Inverse Bin Map*. Its memory footprint and construction time are practically independent of the number of bins. Additionally, it natively supports arbitrary-shaped query regions. These critical advantages come at the price of raising the look-up cost to logarithmic, as with WaveletSAT.

A key observation is that in our context sampling and evaluation are inverse operations. Sampling finds the incoming light direction $\mathbf{i}$ of a given microsurface normal, while the evaluation finds all contributing microsurface normals given $\mathbf{i}$. As usual, the inverse problem is the harder of the two. We notice that sampling based on the bin map $\mathcal{B}$ is a very efficient operation, while the evaluation would require to loop over all bin map texels inside the filtering region A. Potentially, this is inefficient since the number of texels could be arbitrary large. Therefore, we designed a data structure that can act as the inverse of the bin map, in order to make the reverse operation efficient. Since the bin map $\mathcal{B}$ maps a texel position 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 positions of all texels with the given bin index:

$$\mathcal{B}^{-1}(j) = \{k_{j_1}, k_{j_2}, \cdots, k_{j_n}\}, \quad \mathcal{B}(k_{j_i}) = j, \quad i = 1..n \tag{19}$$

where n is the total number of normal map normals in bin j. All lists of positions $\mathcal{B}^{-1}(j)$ are concatenated in a single array of of size N (note that $|\mathcal{B}^{-1}| = |\mathcal{B}|$). Essentially, a map of all bin (normal) map positions, sorted by bin index.

We use $\mathcal{B}^{-1}$ to compute all bin weights W_j by selecting the subset k_{A_j} of texels $\mathcal{B}^{-1}(j)$ which lie in A: in Figure 4, only $\{k_{j_2}, k_{j_4}, k_{j_5}\}$ are in A. Note that in a typical normal map for some bins j $|\mathcal{B}^{-1}(j)|$ is too large for linear traversal. Therefore, we construct a 2D hierarchy for each bin j such that $|\mathcal{B}^{-1}(j)| > L$, where L is a leaf size ($L = 10$ is our implementation), provided during construction.

Two additional data structures accompany $\mathcal{B}^{-1}$:
Index $\mathcal{I}$: Given a bin index it returns the size of $\mathcal{B}^{-1}(j)$ and an offset. For $|\mathcal{B}^{-1}(j)| <= L$ the offset is the start of the list $\mathcal{B}^{-1}(j)$ in $\mathcal{B}^{-1}$. This list is a single leaf and is queried linearly: all texels $k \in \mathcal{B}^{-1}(j)$ are tested directly for intersection with A and w_k are computed (Equation 12). If $|\mathcal{B}^{-1}(j)| > L$ the offset is the start of the 2D hierarchy in the forest $\mathcal{F}$. $\mathcal{I}$ is implemented as an array of size B.

Forest $\mathcal{F}$: A forest of 2D kd-trees, one for each bin j such that $|\mathcal{B}^{-1}(j)| > L$. 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 serialization, we pack the whole forest topology in a single integer list.

Usually, IH are used to query the whole histogram with all B bins, or as in our case sub-histogram of B_0 bins together. The inverse bin map is designed with this proposition in mind. Since all kd-trees have the same splitting planes, hence the same node sizes. We implement traversal for a number of bins that computes the intersections between tree nodes and the region A once for all queried bins. While the tree nodes are axis-aligned, we use parallelogram approximation of the pixel footprint A based on ray differentials [Ige99]. In principle, IBM can work with arbitrarily shaped query regions by providing the proper node-region intersection procedures.

6. Results

We demonstrate our method with different normal maps and lighting scenarios. We have implemented a car paint material with metallic flakes, modelled with a normal map. Flakes orientations are sampled from a GTR distribution [Bur12], and the corresponding Smith shadowing-masking function is applied for G_x [Dim15]. The paint consists of three layers: specular coat layer, metallic flakes layer filtered with our algorithm and base layer with diffuse and glossy term. The transparency of the flake layer is computed using a single channel mip map; when we report the memory for our car paint material we include this data structure in the total. Additionally, we provide a control for the roughness of the individual flakes, as we show in the **Wing mirror scene**, Figure 1. This scene is lit by a sun and an environment light. In our video we show that for a range of small Beckmann flake roughness values the filtering is crucial in order to achieve converged result: renders with equal time stochastic sampling exhibit severe flickering.

Our second scene, the **Car wheel** (Figure 6) has 1K map with scratches that are tiled over the surface to achieve a high texel-to-pixel ratio. The scene is lit by 9 small area lights and an environment with large light sources. In our video we observe that the portions of the surface lit by the high frequency illumination (the 9 small lights) benefit from our filtering technique for low roughness values. As the roughness increases the filtered version is still more stable and some "boiling" can be seen in the stochastic version, however the benefit of our technique for these scenarios is smaller.

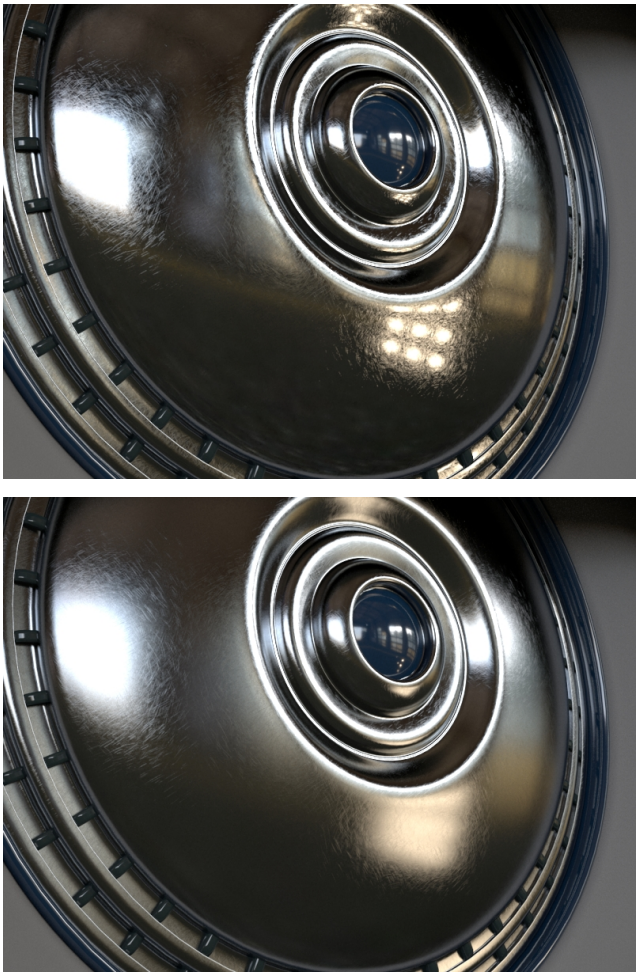

Figure 6: ***Car wheel scene*** *with scratch normal map and Beckmann flake roughness 0.01 (top) and 0.04 (bottom), and filtered with our algorithm. The behaviour of this surface in an animation can be seen in the accompanying video: due to our pre-filtering, object appearance is temporally stable across frames.*

6.1. Comparison with previous work

We compare our method against Yan et al. [YHMR16], the current state of the art for rendering specular high-resolution normal maps. All comparison results in this subsection are based on the original code, scenes and scripts provided by Zeltner et al. [ZGJ20], which includes the original implementation of Yan et al. [YHMR16]. We also implemented our own method as a Mitsuba 2 BRDF [NVZJ19]. All comparisons were rendered on an AMD Ryzen Threadripper 3970X machine. As discussed in Section 2, we want to investigate the scenario of increasing the number of texels that fall within a typical image pixel. We do this by rendering the original scenes from Zeltner et al.[ZGJ20], and by increasing the tiling of the input texture 8 times which results in increasing the texel-to-pixel ratio 8^2 times. The results from the comparisons are presented in Figure 7. Our method demonstrates very similar convergence rates to Yan's flat elements for equal matched roughness $\sigma_r = 0.005$. The convergence plots for both scenes clearly show a tendency: as the number of texels per pixel grows, Yan's curved elements convergence declines. Note that both Yan flat and curved elements use the same hierarchy and intrinsic roughness. The difference in the performance in $\times 8$ scenes (second and fourth row in Figure 7) is because in the curved elements case there are 1-2 orders more contributing elements to be processed in comparison with the flat elements case. We also provide additional results from our method with increased roughness to demonstrate that our convergence is not impeded by growing number of contributing texels. In fact, it improves slightly. Additionally, the **Shoes scene** demonstrates that for some normal maps the appearance of Yan's curved elements for low intrinsic roughness can be matched with our method with increased roughness, see Figure 7 (the first two rows, **Yan (curved)** and **Ours (rough)**). Our method is $90\times$ more memory efficient and the pre-processing is nearly $30\times$ faster than Yan's method with flat Gaussian elements, see Tables 2 and 3. Our method has low memory requirements for wide range of

Table 2: *Memory usage for the methods in Figure 7.*

	Ours	Yan (flat)	Yan (curved)	Ours (rough)
Shoes scene	36MB	3.2GB	19.3GB	36MB
Kettle scene	36MB	3.4GB	59GB	36MB

Table 3: *Pre-processing times of the methods in Figure 7.*

	Ours	Yan (flat)	Yan (curved)	Ours (rough)
Shoes scene	0.2s	8.2s	56.6s	0.3s
Kettle scene	0.3s	8.4s	241.3s	0.3s

Beckmann flake roughness values, and therefore wide range of bin resolutions, see Table 4. For higher resolutions the index $\mathcal{I}$ takes more memory, but the trees in the forest $\mathcal{F}$ are shallow and take less memory. For lower resolutions it is the opposite.

7. Conclusion and future work

In this paper, we present a production-ready normal map filtering method: there are no noticeable pre-computation times, and its memory requirements are very low. Due to pre-filtering being applied, our technique does not slow down if large numbers of normal map texels fall within a single pixel: zooming out from a surface with glints does not cause performance issues.

The algorithm we propose filters direct illumination. Our filtering solution is based on an *Inverse Bin Map*: a specialized integral histogram implementation that enables us to perform the necessary lookups with low memory requirements, and at reasonable speeds. We believe that this data structure has the potential to replace IH in some applications where construction speed, memory efficiency or regions with arbitrary shapes are important. Extending the *Inverse Bin Map* to Gaussian queries would also benefit some applications[BP19].

8. Acknowledgements

We dedicate this work to our friend and colleague Jaroslav. We are grateful to Tashko Zashev for preparing the wing mirror scene and to the anonymous reviewers for the insightful remarks, and acknowledge academic funding from the following sources: GAČR grant number 19-07626S, Charles University grant SVV-260588.

Table 4: *The bin resolution and memory usage of our method for varying Beckmann flake roughness.*

Beckmann roughness α	0.0025	0.01	0.04	0.16
Bin resolution b^2	942^2	235^2	59^2	15^2
Wing mirror scene (2K)	43MB	37MB	36MB	36MB
Car wheel scene (1K)	15MB	9MB	9MB	8MB
Shoes scene (2K)	40MB	36MB	36MB	36MB
Kettle scene (2K)	42MB	36MB	36MB	36MB

References

[Bli78] BLINN, JAMES F. "Simulation of Wrinkled Surfaces". *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '78. New York, NY, USA: ACM, 1978, 286–292. DOI: 10.1145/800248.507101. URL: http://doi.acm.org/10.1145/800248.507101.

[BP19] BALLESTER-RIPOLL, R. and PAJAROLA, R. "Tensor Decompositions for Integral Histogram Compression and Look-Up". *IEEE Transactions on Visualization and Computer Graphics* 25.2 (Feb. 2019), 1435–1446. DOI: 10.1109/TVCG.2018.2802521.

[BS63] BECKMANN, PETR and SPIZZICHINO, ANDRE. *The Scattering of Electromagnetic Waves from Rough Surfaces*. New York: Pergamon, 1963.

[Bur12] BURLEY, BRENT. "Physically-Based Shading at Disney". 2012.

[BYRN17] BELCOUR, LAURENT, YAN, LING-QI, RAMAMOORTHI, RAVI, and NOWROUZEZAHRAI, DEREK. "Antialiasing Complex Global Illumination Effects in Path-Space". *ACM Trans. Graph.* 36.4 (Jan. 2017). ISSN: 0730-0301. DOI: 10.1145/3072959.2990495. URL: http://doi.acm.org/10.1145/3072959.2990495.

[CCM18] CHERMAIN, XAVIER, CLAUX, FRÉDÉRIC, and MÉRILLOU, STÉPHANE. "A microfacet-based BRDF for the accurate and efficient rendering of high-definition specular normal maps". *The Visual Computer* (Oct. 2018). DOI: 10.1007/s00371-018-1606-7.

[Cro84] CROW, FRANKLIN C. "Summed-area Tables for Texture Mapping". *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '84. New York, NY, USA: ACM, 1984, 207–212. ISBN: 0-89791-138-5. DOI: 10.1145/800031.808600. URL: http://doi.acm.org/10.1145/800031.808600.

[CSJD20] CHERMAIN, XAVIER, SAUVAGE, BASILE, JEAN-MICHEL, DISCHLER, and DACHSBACHER, CARSTEN. "Procedural Physically-based BRDF for Real-Time Rendering of Glints". *Comput. Graph. Forum (Proc. Pacific Graphics)* 39.7 (2020), 243–253.

[CT82] COOK, R. L. and TORRANCE, K. E. "A Reflectance Model for Computer Graphics". *ACM Trans. Graph.* 1.1 (Jan. 1982), 7–24. ISSN: 0730-0301. DOI: 10.1145/357290.357293. URL: http://doi.acm.org/10.1145/357290.357293.

[DHI*13] DUPUY, JONATHAN, HEITZ, ERIC, IEHL, JEAN-CLAUDE, et al. "Linear Efficient Antialiased Displacement and Reflectance Mapping". *ACM Transactions on Graphics*. Proceedings of Siggraph Asia 2013 32.6 (Nov. 2013), Article No. 211. DOI: 10.1145/2508363.2508422. URL: https://hal.inria.fr/hal-00858220.

[Dim15] DIMOV, ROSSEN. "Deriving the Smith shadowing function G_1 for $\gamma \in (0,4]$". 2015.

[EKK99] ERSHOV, SERGEY V., KHODULEV, ANDREI B., and KOLCHIN, KONSTANTIN V. "Simulation of sparkles in metallic paints". *Proceedings of Graphicon'99*. 1999, 121–128.

[EKM01] ERSHOV, SERGEY, KOLCHIN, KONSTANTIN, and MYSZKOWSKI, KAROL. "Rendering Pearlescent Appearance Based On Paint-Composition Modelling". *Comput. Graph. Forum* 20 (Sept. 2001). DOI: 10.1111/1467-8659.00515.

[GCG*05] GÜNTHER, JOHANNES, CHEN, TONGBO, GOESELE, MICHAEL, et al. "Efficient Acquisition and Realistic Rendering of Car Paint". 2005.

[GGN18] GAMBOA, LUIS E., GUERTIN, JEAN-PHILIPPE, and NOWROUZEZAHRAI, DEREK. "Scalable Appearance Filtering for Complex Lighting Effects". *ACM Trans. Graph.* 37.6 (Dec. 2018), 277:1–277:13. ISSN: 0730-0301. DOI: 10.1145/3272127.3275058. URL: http://doi.acm.org/10.1145/3272127.3275058.

[Hei14] HEITZ, ERIC. "Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs". *Journal of Computer Graphics Techniques (JCGT)* 3.2 (June 2014), 48–107. ISSN: 2331-7418. URL: http://jcgt.org/published/0003/02/03/.

[HSRG07] HAN, CHARLES, SUN, BO, RAMAMOORTHI, RAVI, and GRINSPUN, EITAN. "Frequency domain normal map filtering". *ACM Trans. Graph.* 26 (July 2007), 28. DOI: 10.1145/1275808.1276412.

[Ige99] IGEHY, HOMAN. "Tracing Ray Differentials". *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, 179–186. ISBN: 0-201-48560-5. DOI: 10.1145/311535.311555. URL: http://dx.doi.org/10.1145/311535.311555.

[JHY*14] JAKOB, WENZEL, HAŠAN, MILOŠ, YAN, LING-QI, et al. "Discrete Stochastic Microfacet Models". *ACM Trans. Graph.* 33.4 (July 2014), 115:1–115:10. ISSN: 0730-0301. DOI: 10.1145/2601097.2601186. URL: http://doi.acm.org/10.1145/2601097.2601186.

[KHX*19] KUZNETSOV, ALEXANDR, HAŠAN, MILOŠ, XU, ZEXIANG, et al. "Learning Generative Models for Rendering Specular Microgeometry". *ACM Trans. Graph.* 38.6 (Nov. 2019), 225:1–225:14. ISSN: 0730-0301. DOI: 10.1145/3355089.3356525. URL: http://doi.acm.org/10.1145/3355089.3356525.

[LS13] LEE, T. and SHEN, H. "Efficient Local Statistical Analysis via Integral Histograms with Discrete Wavelet Transform". *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), 2693–2702. DOI: 10.1109/TVCG.2013.152.

[NVZJ19] NIMIER-DAVID, MERLIN, VICINI, DELIO, ZELTNER, TIZIAN, and JAKOB, WENZEL. "Mitsuba 2: A Retargetable Forward and Inverse Renderer". *ACM Trans. Graph.* 38.6 (Nov. 2019). ISSN: 0730-0301. DOI: 10.1145/3355089.3356498. URL: https://doi.org/10.1145/3355089.3356498.

[OB10] OLANO, MARC and BAKER, DAN. "LEAN Mapping". *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. I3D '10. Washington, D.C.: ACM, 2010, 181–188. ISBN: 978-1-60558-939-8. DOI: 10.1145/1730804.1730834. URL: http://doi.acm.org/10.1145/1730804.1730834.

[Por05] PORIKLI, FATIH. "Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces". *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*. Vol. 1. CVPR '05. USA: IEEE Computer Society, July 2005, 829–836. ISBN: 0769523722. DOI: 10.1109/CVPR.2005.188. URL: https://doi.org/10.1109/CVPR.2005.188.

[RGB16] RAYMOND, BORIS, GUENNEBAUD, GAEL, and BARLA, PASCAL. "Multi-Scale Rendering of Scratched Materials using a Structured SV-BRDF Model". *ACM Transactions on Graphics* (July 2016). DOI: 10.1145/2897824.2925945. URL: https://hal.inria.fr/hal-01321289.

[RMS*08] RUMP, MARTIN, MÜLLER, GERO, SARLETTE, RALF, et al. "Photo-realistic Rendering of Metallic Car Paint from Image-Based Measurements". *Computer Graphics Forum* 27.2 (Apr. 2008). Ed. by SCOPIGNO, R. and GRÖLLER, E., 527–536.

[SHHD17] SCHÜSSLER, VINCENT, HEITZ, ERIC, HANIKA, JOHANNES, and DACHSBACHER, CARSTEN. "Microfacet-Based Normal Mapping for Robust Monte Carlo Path Tracing". *ACM Trans. Graph.* 36.6 (Nov. 2017). ISSN: 0730-0301. DOI: 10.1145/3130800.3130806. URL: https://doi.org/10.1145/3130800.3130806.

[SNM*02] SUNG, LI-PIIN, NADAL, MARIA E., MCKNIGHT, MARY E., et al. "Optical reflectance of metallic coatings: Effect of aluminum flake orientation". *Journal of Coatings Technology* 74 (2002). ISSN: 1935-3804. URL: https://doi.org/10.1007/BF02697975.

[Tok05] TOKSVIG, MICHAEL. "Mipmapping Normal Maps". *Journal Graphics Tools* 10 (2005), 65–71.

[TS67] TORRANCE, K.E. and SPARROW, E.M. "Theory for Off-Specular Reflection from Roughened Surfaces". *Journal of the Optical Society of America (JOSA)* 57.9 (Sept. 1967), 1105–1114. URL: http://www.graphics.cornell.edu/~westin/pubs/TorranceSparrowJOSA1967.pdf.

[VG95] VEACH, ERIC and GUIBAS, LEONIDAS J. "Optimally Combining Sampling Techniques for Monte Carlo Rendering". *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: ACM, 1995, 419–428. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218498. URL: http://doi.acm.org/10.1145/218380.218498.

[VWH18] VELINOV, ZDRAVKO, WERNER, SEBASTIAN, and HULLIN, MATTHIAS B. "Real-Time Rendering of Wave-Optical Effects on Scratched Surfaces". *Computer Graphics Forum 37 (2) (Proc. EUROGRAPHICS)* 37.2 (2018).

[WHHY20] WANG, BEIBEI, HAŠAN, MILOŠ, HOLZSCHUCH, NICOLAS, and YAN, LING-QI. "Example-Based Microstructure Rendering with Constant Storage". *ACM Trans. Graph.* 39.5 (Aug. 2020). ISSN: 0730-0301. DOI: 10.1145/3406836. URL: https://doi.org/10.1145/3406836.

[Wil83] WILLIAMS, LANCE. "Pyramidal Parametrics". *SIGGRAPH Comput. Graph.* 17.3 (July 1983), 1–11. ISSN: 0097-8930. DOI: 10.1145/964967.801126. URL: http://doi.acm.org/10.1145/964967.801126.

[WMLT07] WALTER, BRUCE, MARSCHNER, STEPHEN R., LI, HONGSONG, and TORRANCE, KENNETH E. "Microfacet Models for Refraction Through Rough Surfaces". *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. EGSR'07. Grenoble, France: Eurographics Association, 2007, 195–206. ISBN: 978-3-905673-52-4. DOI: 10.2312/EGWR/EGSR07/195-206. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206.

[WVJH17] WERNER, SEBASTIAN, VELINOV, ZDRAVKO, JAKOB, WENZEL, and HULLIN, MATTHIAS B. "Scratch Iridescence: Wave-optical Rendering of Diffractive Surface Structure". *ACM Trans. Graph.* 36.6 (Nov. 2017), 207:1–207:14. ISSN: 0730-0301. DOI: 10.1145/3130800.3130840. URL: http://doi.acm.org/10.1145/3130800.3130840.

[WZYR19] WU, LIFAN, ZHAO, SHUANG, YAN, LING-QI, and RAMAMOORTHI, RAVI. "Accurate Appearance Preserving Prefiltering for Rendering Displacement-Mapped Surfaces". *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3322936. URL: https://doi.org/10.1145/3306346.3322936.

[YHJ*14] YAN, LING-QI, HAŠAN, MILOŠ, JAKOB, WENZEL, et al. "Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33.4 (July 2014), 116:1–116:9. DOI: 10.1145/2601097.2601155.

[YHMR16] YAN, LING-QI, HAŠAN, MILOŠ, MARSCHNER, STEVE, and RAMAMOORTHI, RAVI. "Position-Normal Distributions for Efficient Rendering of Specular Microstructure". *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2016)* 35.4 (2016).

[YHW*18] YAN, LING-QI, HAŠAN, MILOŠ, WALTER, BRUCE, et al. "Rendering Specular Microgeometry with Wave Optics". *ACM Trans. Graph.* 37.4 (July 2018), 75:1–75:10. ISSN: 0730-0301. DOI: 10.1145/3197517.3201351. URL: http://doi.acm.org/10.1145/3197517.3201351.

[ZGJ20] ZELTNER, TIZIAN, GEORGIEV, ILIYAN, and JAKOB, WENZEL. "Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints". *ACM Trans. Graph.* 39.4 (July 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392408. URL: https://doi.org/10.1145/3386569.3392408.

[ZK16] ZIRR, TOBIAS and KAPLANYAN, ANTON S. "Real-time Rendering of Procedural Multiscale Materials". *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. I3D '16. Redmond, Washington: ACM, 2016, 139–148. ISBN: 978-1-4503-4043-4. DOI: 10.1145/2856400.2856409. URL: http://doi.acm.org/10.1145/2856400.2856409.

[ZXW19] ZHU, JUNQIU, XU, YANNING, and WANG, LU. "A Stationary SVBRDF Material Modeling Method Based on Discrete Microsurface". *Computer Graphics Forum* 38.7 (2019), 745–754. DOI: 10.1111/cgf.13876. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13876.

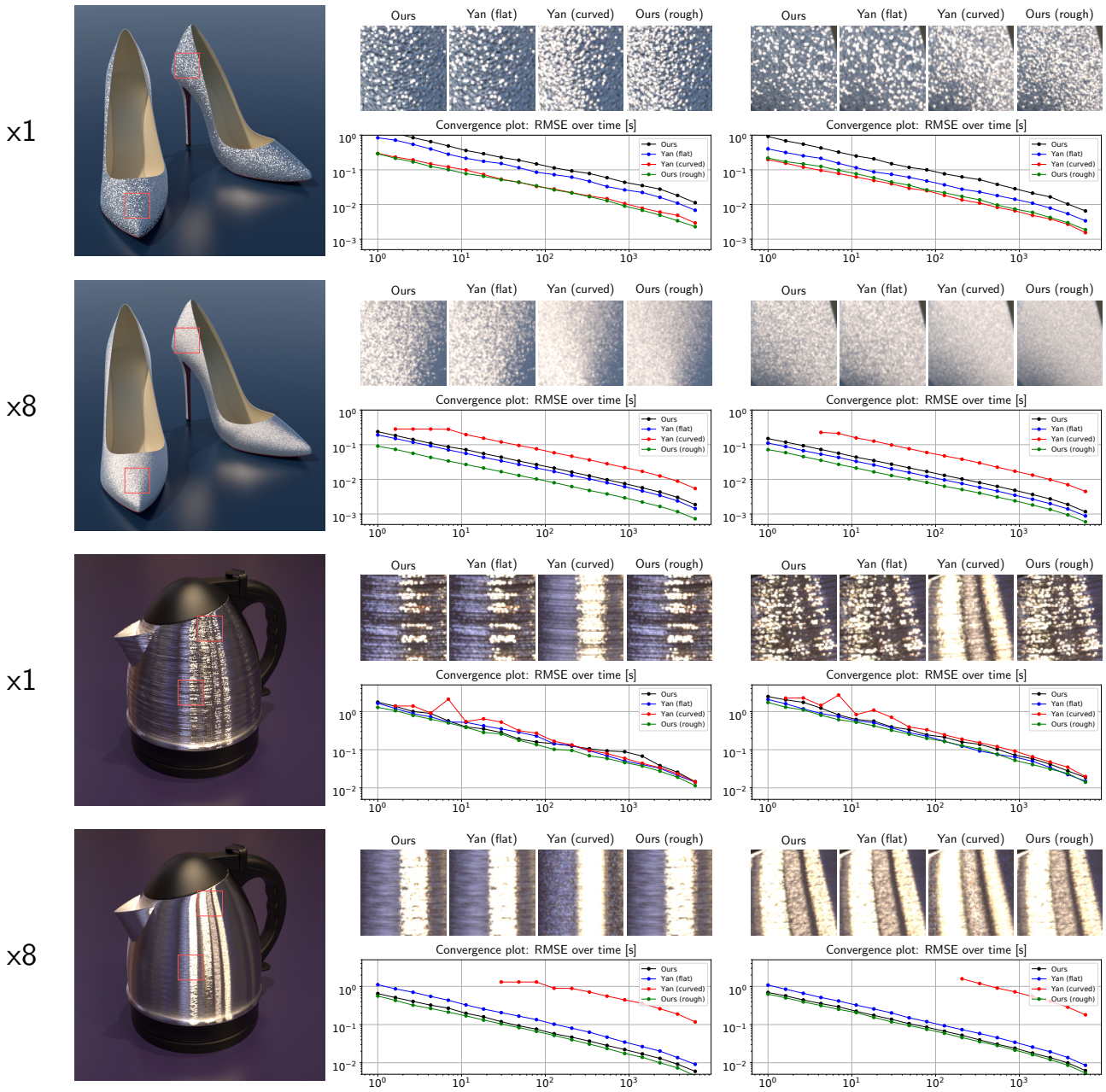

Figure 7: ***Shoes and Kettle scenes*** (800×800) *[ZGJ20]: We render two different texture tiles - the original scenes in the first and third rows and then* $\times 8$ *tiles in the second and fourth rows. All of these scenarios were rendered with four variations: our method, Yan et al. flat elements, Yan et al. curved elements, and our method with slightly increased roughness. The first three variations have equal roughness values* $\sigma_r = 0.005$*, while the last demonstrates our method with higher Beckmann flake roughness* $\alpha = 0.05$ *(*$\sigma_r = \frac{\alpha}{\sqrt{2}}$*) that is not supported by Yan's method. The large images in the left column are rendered with our low roughness variant (the first variant labelled* ***Ours****). Two regions of size 80x80 are selected from both scenes, and convergence plots are computed for each of them (middle and right column). We use the script provided by Zeltner et al. [ZGJ20] to compute the plots.*