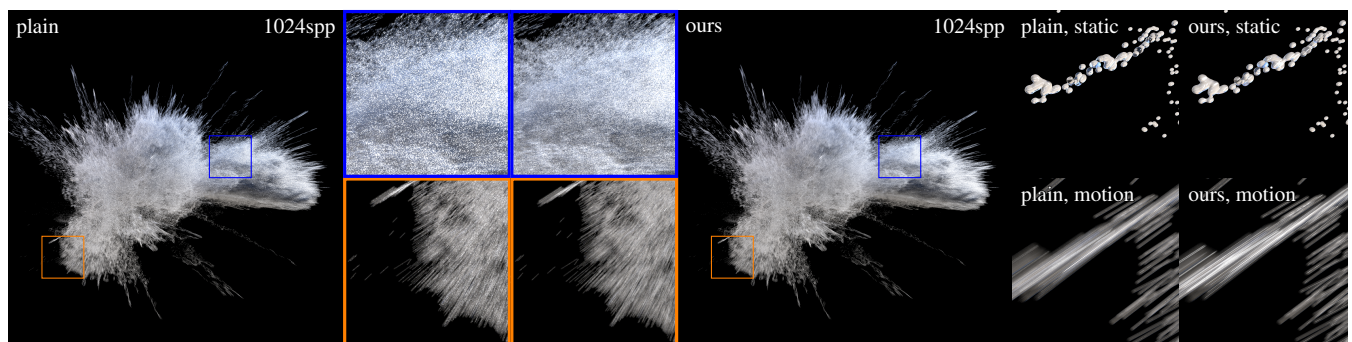# Temporal Normal Distribution Functions

Lorenzo Tessari[1,2], Johannes Hanika[1,2], Carsten Dachsbacher[1], Marc Droske[2]

Karlsruhe Institute of Technology[1], Weta Digital[2]

Figure 1: *A fast moving water splash, consisting of an isosurface generated by 12M particles. High curvature detail with highly glossy Bidirectional Scattering Distribution Function (BSDF) creates difficult, glint-like specular highlights (see details in right images). These are even harder to sample when the objects are fast moving. Our approach alters the roughness of the microfacet BSDF specifically to account for motion (compare the static crops to the right with the moving ones just below), introducing a temporal normal distribution function. This distribution captures the anisotropic motion of the highlight and allows us to render such cases efficiently.*

Abstract

Specular aliasing can make seemingly simple scenes notoriously hard to render efficiently: small geometric features with high curvature and near specular reflectance result in tiny lighting features which are difficult to resolve at low sample counts per pixel. LEAN and LEADR mapping can be used to convert geometric surface detail to anisotropic surface roughness in a preprocess. In scenes including fluid simulation this problem is particularly apparent with fast moving elements such as spray particles, which are typically represented as participating media in movie rendering. Both approaches, however, are only valid in the far-field regime where the geometric detail is much smaller than a pixel, while the challenge of resolving highlights remains in the meso-scale. Fast motion and the relatively long shutter intervals, commonly used in movie production, lead to strong variation of the surface normals seen under a pixel over time aggravating the problem. Recent specular anti aliasing approaches preintegrate geometric curvature under the pixel footprint for one specific ray to achieve noise free images at low sample counts. We extend these to anisotropic surface roughness and to account for the temporal surface normal variation due to motion blur. We use temporal derivatives to approximate the distribution of the surface normal seen under a pixel over the course of the shutter interval. Furthermore, we discuss how this can afterwards be combined with the surface BSDF in a practical way.

CCS Concepts
• Computing methodologies → Ray tracing; Reflectance modeling;

1. Introduction

Computing global illumination in computer generated imagery has become an indispensable tool in movie production in the last years. This is achieved by path tracing, a versatile method which traces random rays of light through a virtual scene and forms an image by collecting their contributions to the pixels. This can be more formally described as an integration problem over the space of light transport paths. In the search for more detail and realism, new movie shows routinely reveal weaknesses of sampling techniques,

however. In fact, often the most long standing and well known problems manifest themselves as noise in rendered images.

One such long standing problem is aliasing due to subsampling of the integrand. In principle, this can be alleviated by using prefiltering or postfiltering [MN88]. Because of the general applicability of reconstruction post filters, there is a large body of work in this subject. However, prefiltering, if possible, can reduce aliasing more effectively [MN88].

Consider the example in figure 1: tiny geometric features are inside each pixel, and their specular reflectance and high curvature compresses bright, glinty features to even smaller size as illustrated in figure 2. Fast moving geometry aggravates the problem by spreading out the highlight in time, often requiring many thousands of samples per pixel to converge [FHP*18, Sec. 6.2.6]. Postfiltering would take one sample that happens to land on such a bright glint and spread the contribution to the framebuffer out to a few pixels by using a reconstruction filter with large support, see e.g. [ETH*09]. However, the assumption is that such a small feature can be sampled in the first place. Also, such a reconstruction filter often knows nothing about geometric boundaries and will blur over them unless explicit checks [DSHL10] or heavy machinery [GLA*19] are used.

Prefiltering on the other hand attempts to "smooth out" the geometry and reflectance of the input, but there is no general solution. In practice, it is often implemented by adaptive tessellation in conjunction with LEAN/LEADR mapping [DHI*13]. On top of this, heuristics to artificially *roughen* the reflectance of the surfaces are commonly used, in the extreme cases making surfaces diffuse [HD09, JG19]. Recently, *specular anti aliasing* has been proposed to accumulate roughness of surfaces within the footprint of a pixel [KHPL16], and also consider their curvature. We extend this approach to anisotropic curvature, and additionally consider motion: fast moving geometry will extend the distribution of surface normals seen under a pixel over the course of the shutter interval. Anisotropy is important, since LEADR mapping on a coarse scale will lead to anisotropic scattering profiles even if the base surface is a simple isotropic smooth dielectric.

In summary, our contributions are

- the introduction of temporal normal distribution functions and the derivation of a corresponding microfacet model,
- using a derivation of temporal normal derivatives to approximate the temporal distribution by an anisotropic Beckmann distribution,
- the combination of the temporal distribution with the surface BSDF to devise a fully practical method suitable for high-quality offline rendering.

2. Background and previous work

Specular aliasing is particularly hard to reduce with highly-detailed and unstructured geometry with specular reflectance. In general, contributions to light transport are results of convolution of the pixel filter, shutter interval, BSDFs, light source areas etc., and computed by sampling the path space. Often the result remains quite similar even when the filtering along these individual events deviates from the actual configuration. This is, for example, ex-
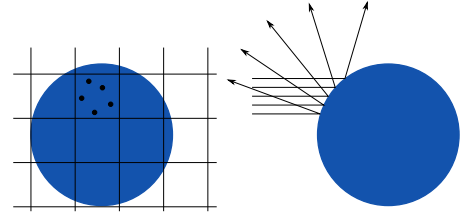
Figure 2: *Specular aliasing: high-curvature relative to pixel footprint blows up ray-space domain covered by specular bounces. A small light source is even less likely to be intersected then. This will introduce a lot of noise, even though the deterministic event at the surface does not technically introduce randomness.*

ploited with pixel reconstruction filters, which compensate for suboptimal sampling and filtering elsewhere. In our work, we also investigate filtering, but focus on altering microfacet BSDFs by convolving its normal distribution function to aid connection-based Monte Carlo techniques such as next event estimation. This has the advantage that the geometric silhouettes remain accurate.

Reconstruction filters Postfiltering is typically the least intrusive way to address aliasing. Best results are achieved when the sampling budget is steered according to properties of the reconstruction filter, e.g. for motion blur [ETH*09]. Zwicker et al. [ZJL*15] provide an extensive survey on adaptive sampling and reconstruction.

Altering the BSDFs Avoiding specular aliasing by simply reducing the specularity, e.g. changing the roughness or microfacet BSDF parameters [JG19], is minimally invasive and has been done in one form or the other in many situations where sampling becomes difficult. For example, roughening of the surface enables path-perturbation and reconnection techniques (cf. "Finding good paths" in [FHH*19]) to become effective. However, introducing roughness to specular surfaces does not necessarily decrease variance, as Dirac delta materials (e.g. perfect mirrors) do not cause variance yet possibly aliasing. There is also a line of work that includes multi modal glints under one pixel into a more complex normal distribution function [YHJ*14, JHY*14]. In our work, we are interested in geometry more similar to single water droplets, so we do not require this sophisticated machinery for this use case.

Geometry and Textures Specular aliasing due to detailed geometry is often addressed by prefiltering displacement or normal maps using techniques such as LEAN [OB10] or LEADR [DHI*13] mapping, where geometric detail is interpreted as statistical variation inside a microsurface in the context of microfacet theory [CT82]. By this, geometry is simplified and analytical evaluation of the distribution of micronormals makes, for instance, next event estimation a lot more efficient. Motion blur on textures has been addressed by anisotropic filtering [Lov05], which involves similar considerations about derivatives as our approach.

These techniques are costly to apply in path space where the integration domain is high dimensional and irregular: samples from indirect bounces usually are not spaced on regular grids and texture footprints needs to be derived per sample. Simplifying geometry

$t=0$ $t=0.5$ $t=1$ $t \in [0,1]$
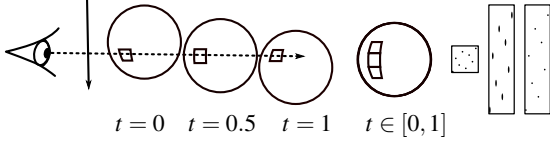
Figure 3: *A sphere plotted for three time instants $t = 0, 0.5, 1.0$. The normal under the pixel changes over time. If we overlay the pixel footprints for $t \in [0,1]$ on the surface of the sphere, we see an elongated pixel footprint, which spreads apart any sampling pattern vertically in this example (right two boxes). The longer this footprint becomes, the less likely is it to intersect a small light source in this area. Intuitively, we compensate for this spread by convolving the BSDF with a kernel that elongates it in the direction of motion (left of these two boxes), shrinking the gaps between the samples.*

P	intersection point on surface
N	surface normal
$\partial_u \mathbf{P}$	$\frac{\partial \mathbf{P}}{\partial u}$ tangent vector in u direction
$\partial_v \mathbf{P}$	$\frac{\partial \mathbf{P}}{\partial v}$ tangent vector in v direction
Σ	covariance matrix of the Beckmann surface BSDF
α^t	temporal (macro) Beckmann roughness along motion
α^μ	base (micro) BSDF Beckmann roughness
α_1, α_2	anisotropic roughness, major and minor BSDF axes
P	covariance matrix of geometric curvature in slope space
$\mathbf{o}$	outgoing direction from **P** to camera
$\mathbf{i}$	incident direction from **P** to light
$\mathbf{h}$	3D half vector $= \frac{\mathbf{i}+\eta\mathbf{o}}{\|\mathbf{i}+\eta\mathbf{o}\|}$
$\mathbf{h}_s$	2D half vector in slope space
$D(\mathbf{h})$	distribution of slopes of micronormals
$D^t(\mathbf{h})$	distribution of slopes of normal over time
G	geometric shadowing/masking term
M	Jacobian from slope space $\mathbf{h}_s$ to on-surface measure

Table 1: *List of symbols used in our math notation.*

per ray accordingly would require to (re)build an acceleration structure for ray tracing and is thus done once, or cached and reused for batches of similar rays, resulting in sub-optimal geometry for individual samples. Still, after conservatively considering geometric simplifications, the remaining curvature fans out specular reflections under the ray footprint (figure 2). This causes high variance if the light source is small compared to this footprint.

Ray differentials Ray footprints are obtained by ray differentials [Ige99], tracking the size of a pixel through a number of interactions. This concept has been extended to full paths [SW01], the temporal domain [SSE09], the wavelength domain [EBR*14] for spectral rendering, as well as the adjoint case for photon mapping [SFES11]. Thus we assume the equations for particular derivatives to be readily available in this paper. A more advanced way of determining pixel spread in higher dimensions is, for example, covariance tracing [BSS*13]. Pencil tracing [STN87] is an early work which traces 4D spread along with a path. We also track differentials through reflection, refraction, and transfer operators. Recently, fully general differentiable rendering has been discussed [LADL18, LHJ19], but is overly generic and slow for our use case.

Filtering emitters Alternate approaches filter the light source, e.g. [KVHS00], using the footprint when arriving at the light. The extent of the differential, however, needs to be clamped to limit the error as the resulting approximation is only valid in a local neighbourhood of the main path as it does not account for changes in visibility and high frequency geometry. While image-based lighting is straightforward to filter (e.g. using an *elliptically weighted average*, EWA approach), geometric lights are not easily filterable as ray casting is a point-sampling operation and does not retrieve lights in the neighbourhood of a ray. For surface interactions that are not perfectly specular, combining the technique with next-event estimation is difficult: while the filtering region depends on the direction being sampled, the sampled direction would depend on the amount of filtering to be applied in order to maintain high quality importance sampling of the filtered distribution.

We are aiming for an approach which operates on the material level only, and thus is easy and minimally invasive to integrate, and works well together with path-tracing algorithms. in particular the modification of the material at the hit-point does not depend on ver-

tices that are constructed further towards the light and the approach works for all light sources, not only filterable image-based lights.

Precisely resolving highlights Manifold walks [HDF15, JM12] on the surface geometry can be used to find exactly the paths that fulfill the specular constraint. Outside the Markov chain context, this has so far only been demonstrated for transmissive caustics, not for reflective highlights [WZHB09], using exhaustive search.

Specular anti aliasing The same basic underlying calculus as for manifold walks has been used to compute an approximation of the highlight integrated over the pixel [KHPL16, TK19] by filtering the BSDF in the half vector domain.

Motion blur widens the integration domain further (Fig. 3), making it even harder to sample. Similar problems arise with other distribution effects like depth-of-field. We follow along the same line as specular anti aliasing, but are interested in accurate anisotropy, application for indirect bounces in path space and motion blur.

Our approach is also based on the observation, that in practice, specular surfaces often have some amount of roughness due to LEAN/LEADR mapping. This requires us to handle "temporal roughness" on top of anisotropic microfacet BSDF roughness.

3. Filtering microfacet distributions

In this section we describe our approach by first summarising from [KHPL16] the foundations of specular anti aliasing and derivation of precise expressions to handle anisotropic BSDF and geometric curvature (section 3.1). We will then proceed with our contribution to extending this to the time domain (section 3.2). This is for the most part a straightforward extension of the spatial domain technique, but has a different geometric shadowing/masking term since microfacets at different time steps do not shadow each other.
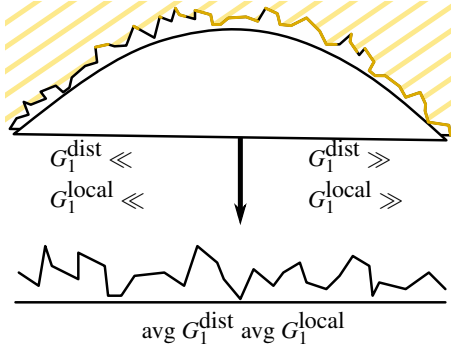
Figure 4: *When spatial curvature (top) is baked into surface roughness (bottom), we ignore the correlation between local and distant shadowing (G_1^{local} and G_1^{dist}), and replace an integration over both by the product of their respective averages.*

3.1. Pixel filtering

On a high level, specular anti aliasing analytically describes geometric detail under a pixel footprint and evaluates it closed-form along with a modified BSDF. We will revise the two causes of normal variation in a pixel, first, roughness of the BSDF and then geometric curvature. We will use the notation as outlined in table 1.

BSDF roughness A Beckmann roughness of an anisotropic BSDF is defined in the tangent frame $\mathbf{N}, \partial_u \mathbf{P}, \partial_v \mathbf{P}$ at a surface point $\mathbf{P}$ via the covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_u^2 & r_{uv}\sigma_u\sigma_v \\ r_{uv}\sigma_u\sigma_v & \sigma_v^2 \end{pmatrix}, \tag{1}$$

where $\sigma_u^2, \sigma_v^2, r_{uv}$ are respectively the original roughness on the major and minor axis and their correlation coefficient. This formulation expresses the shape of a Gaussian normal distribution function (NDF) in slope space and $r_{uv} \in [-1, 1]$ is simply its rotation coefficient. The NDF is defined as [Hei14, eq (68)]:

$$D(\mathbf{h}) = \frac{\chi^+(\mathbf{h} \cdot \mathbf{N})}{\pi |\Sigma| h_n^4} \exp\left(\frac{1}{2}\mathbf{h}_s^T \Sigma^{-1} \mathbf{h}_s\right), \tag{2}$$

with $\mathbf{h}_s = \left(h_u/h_n, h_v/h_n\right)^T$ being the half vector in slope space. Here, χ^+ clamps the dot product to be positive. Note that we use a standard Gaussian covariance matrix rather than a Beckmann roughness, because the latter factors in the $\frac{1}{2}$ in the exponent (i.e. Beckmann $\alpha = \sqrt{2}\sigma$), making convolutions of Gaussians more cumbersome to express later on.

Surface curvature We follow [KHPL16, eq. (4)] and define the Jacobian of the change of local half vector in slope space $\mathbf{h}_s$ with respect to the surface position $\mathbf{P}$ as

$$M(\mathbf{P}) = \partial \mathbf{h}_s / \partial \mathbf{P}. \tag{3}$$

This expression includes surface curvature, and thus depends on the normal derivatives with respect to the surface parameterisation u, v. To express the half vector dependency, the partial derivatives employed here in general require knowledge of the position of the next

path vertex (closer to the light) [JM12]. Since we do not always know the outgoing vertex beforehand, we assume perfect specular reflection to derive the required quantities for equation (3). This is in line with our assumption that the base roughness of the surface is relatively specular.

Combined NDF We now approximate a combined NDF D' as the convolution of the microfacet NDF and the distribution caused by spatial curvature under the pixel footprint [KHPL16, eq. (12)]:

$$\Sigma' = \Sigma + P. \tag{4}$$

Let the pixel footprint be defined in screen space as a Gaussian where σ_p matches the half pixel width. We compute the footprint in half vector space by projecting the pixel to the local tangent frame by mapping the pixel coordinates $(i, j), (i + 0.5, j), (i, j + 0.5)$ forward through the shot camera and intersecting the resulting rays with the plane around $\mathbf{P}$ with normal $\mathbf{N}$. From these three points, $\mathbf{P}, \mathbf{P}_i, \mathbf{P}_j$, we derive the covariance matrix of an anisotropic Gaussian in tangent space $(\partial_u \mathbf{P}, \partial_v \mathbf{P}, \mathbf{N})$. This matrix is then mapped through $M(\mathbf{P})$ to obtain a covariance matrix in slope space.

Discussion We can predict D with high accuracy by assuming a convolution of the two causes of normal variation. This assumption is reasonable, since the materials we are interested in are very specular. When it comes to the geometric shadowing/masking of the microsurface, we just depend on the formulation of the microfacet BSDF to correctly include the added roughness into the G term. Conceptually this disregards spatial correlation between the surface position and the micronormals even though we know they are correlated (see figure 4 for an illustration where G_1^{local} refers to shadowing-masking due to BSDF roughness and G_1^{dist} due to curvature). Intuitively we average normals from behind the horizon of the visible surface into the result. In fact we probably overestimate G, because the Cauchy-Schwartz inequality states that the absolute value of the covariance (avg $G_1^{local} \cdot G_1^{dist}$ [Hei14, Sec. 4.1]) is at most as high as the product of variances (avg G_1^{local} times avg G_1^{dist}). This means there is less shadowing, and thus brightness too high.

Increasing the roughness for the G computation results in more shadowing, so whatever was indirect light scattering between geometry in path tracing before now becomes shadowed because the BSDF only considers single scattering. This has the opposite effect: it tends to overshadow a little. The two effects partly cancel each other, and become less relevant in practice if the BSDF model contains a term to recover energy from multiple scattering within the microsurface [HHdD16,Tur19]. Such a model is essential when using prefiltering using LEADR maps to maintain a consistent look in the distance.

The shadowing function normalises the distribution of visible normals, i.e. $G \cdot D/(4\cos)$ is normalised. The slope induced by the curvature also self-shadows microfacets and introduces masking. We decorrelate the spatial location and the height by forming one microsurface with more roughness but without curvature. This new microsurface naturally passes the weak white furnace test for the masking induced by the curvature, since we alter both the normal
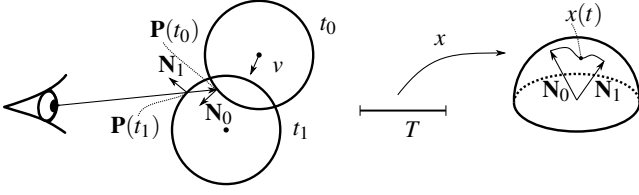
Figure 5: Left: *Change of normals* **N** *at intersection point* **P** *depending on time t.* Right: *Resulting map x(t) from time domain T to reference unit sphere.*

distribution and the masking term according to the new covariance matrix.

3.2. Temporal filtering

In this section we will formulate a temporal normal distribution function as a natural extension of spatial NDFs to the time domain and discuss analogies and differences. We refer to [Hei14] for an exhaustive introduction of the fundamental concepts of microfacet theory we are building upon. We will then use this concept to combine a micro-NDF (the base roughness of the BSDF and spatial specular anti aliasing) with a macro-NDF (due to motion blur). We propose to approximate the temporal distribution of normals in a local neighborhoord based on the first-order approximation around a reference normal based the derivative of the apparent normal and discuss some sources of errors.

Temporal NDF We begin with extending the notion of a NDF to describe the surface's micro detail to the temporal domain. The BRDF for a specular reflection to the perfect reflection direction **r** is given by [WMLT07, eq. (15)]

$$f_r^\mu(\mathbf{o}, \mathbf{i}) = \frac{F \cdot \delta_\mathbf{r}(\mathbf{i})}{|\cos\theta_\mathbf{o}|} = \frac{F \cdot \delta_{\mathbf{h}_r}(\mathbf{h}(\mathbf{i}))}{4|\cos\theta_\mathbf{o}| \cdot |\cos\theta_\mathbf{i}|} \quad (5)$$

using the well known relation

$$\left|\frac{d\mathbf{h}}{d\mathbf{i}}\right| = \frac{1}{4|\cos\theta_\mathbf{h}|} = \left|\frac{d\mathbf{h}}{d\mathbf{o}}\right|. \quad (6)$$

In [WMLT07, eq. (8)] the spatial transformation from micro to macro surface requires a ratio of cosines (respectively $\cos\theta$ and $\cos\hat\theta$) that takes into account this change while integrating radiance over solid angle.

$$L_o(r) \approx \int_{\Omega^+} \frac{|\cos\theta_\mathbf{o}|}{|\cos\hat\theta_\mathbf{o}|} f_r^\mu(\mathbf{o}, \mathbf{i}) \frac{|\cos\theta_\mathbf{i}|}{|\cos\hat\theta_\mathbf{i}|} L_i(\mathbf{i}) \, d\mathbf{i}^\perp \quad (7)$$

Similarly with time, we need to add the change in orientation between different time instants. For a time-dependent ray we denote by **P**(t) the surface hitpoint at time t; the filtered radiance in a temporal neighbourhood $T = [t_0, t_1]$ around a reference geometry with normal **N** (say, the normal at $\mathbf{P}(\frac{t_0+t_1}{2})$) can be written as:

$$L_o(r) \approx \frac{1}{|T|} \int_{t_0}^{t_1} \int_{\Omega^+} \frac{|\cos\theta_\mathbf{o}^t|}{|\cos\hat\theta_\mathbf{o}|} \frac{F \cdot \delta_{\mathbf{h}(\mathbf{P}(t))}(\mathbf{h})}{4|\cos\theta_\mathbf{o}^t| \cdot |\cos\theta_\mathbf{i}^t|} L_i^t(\mathbf{i}) \, d\mathbf{i}^\perp \, dt \quad (8)$$

$$= \frac{1}{|T|} \int_{t_0}^{t_1} \int_{\Omega^+} \frac{F \cdot \delta_{\mathbf{h}(\mathbf{P}(t))}(\mathbf{h})}{4|\cos\hat\theta_\mathbf{o}| \cdot |\cos\hat\theta_\mathbf{i}|} \hat{L}_i(\mathbf{i}) \, d\mathbf{i}^\perp \, dt \quad (9)$$

using the transformation of radiance from reference space to temporal micro space $\hat{L}_\mathbf{i}^t = \frac{|\cos\theta_\mathbf{i}^t|}{|\cos\hat\theta_\mathbf{i}|} \hat{L}_\mathbf{i}$. Here, the hat notation $\hat\theta$, $\hat{L}$ indicate that quantities are with respect to the fixed reference geometry. This plays a similar role as the geometric frame in spatial microfacet theory, i.e., microscopic variation is expressed relative to a fixed local coordinate system. The formulation is based on the apparent change of the material response in the neighborhood, moving with the time dependant hitpoint. Here, we need to assume a sufficiently small neighborhood to disregard discontinuities of the closest-point transform and such that we can, similar to the spatial case, assume that L_i does not contain significant far-field variation. We now define

$$D^t(\mathbf{v}) := \frac{1}{|T|} \int_{t_0}^{t_1} \delta_\mathbf{v}(\mathbf{h}(\mathbf{P}(t))) \, dt \quad [s], \quad (10)$$

which we call *temporal NDF*. If $x : T = [t_0, t_1] \to \Omega, t \mapsto \mathbf{N}(\mathbf{P}(t))$ defines the forward mapping from time to normal domain according to the ray-intersection (cf. figure 5), then

$$\mu := |T|^{-1}\lambda \circ x^{-1} \quad (11)$$

is the push-forward measure of the Lebesgue measure λ in the time domain to the normal domain, hence D^t is the *Radon-Nikodym* derivative $\frac{d\mu}{d\omega}$ of μ. We have

$$\frac{1}{|T|} \int_{\Omega^+} D^t(\mathbf{m}) \, d\mathbf{m} = 1 \quad [s]. \quad (12)$$

Using D^t equation (9) simplifies to

$$L_o(r) = \int_{\Omega^+} \underbrace{\frac{F \cdot D^t(\mathbf{h}(\mathbf{P}(t)))}{4|\cos\hat\theta_\mathbf{o}| \cdot |\cos\hat\theta_\mathbf{i}|}}_{\text{temporal BRDF}} \cdot L_i(\mathbf{i}) \, d\mathbf{i}^\perp. \quad (13)$$

Note that in comparison to spatial microfacet theory, there is no masking term, since radiance is not influenced by the local geometry at a different time. Hence, the analogue of the distribution of visible normals [Hei14]

$$D_{\text{vis}}^t(\mathbf{h}) := \frac{\cos\theta_\mathbf{h}}{\cos\theta_\mathbf{o}} D^t(\mathbf{h}) \quad (14)$$

is not normalised but still remains physically plausible (see figure 6): The shadowing/masking terms G of the Smith surface are composed of local and distant terms [Hei14, Sec. 4.1]. The local term is the self shadowing of the microfacet due to backfacing normals. This part remains valid at a different time, while the distant shadowing part does not: microfacets at different instants in time do not shadow each other. Depending on the geometric complexity and the temporal dependence of the incoming ray, D^t can exhibit a very complex behaviour. Let us, for the moment, assume that we can locally approximate D^t by a highly anisotropic Beckmann NDF, which will be described later in more detail. This makes the distinction between local and distant G terms very simple, since the local shadowing part is contained in the clamped cosine in the definition of D in equation (2). This means we can simply disregard the shadowing due to the increased roughness, i.e. we evaluate G using only the spatial roughness from base BSDF and curvature.

Handling glossy micro-NDF In practice materials are rarely perfectly specular, since micro detail that is no longer represented by actual geometry during render time needs to be accounted for. If
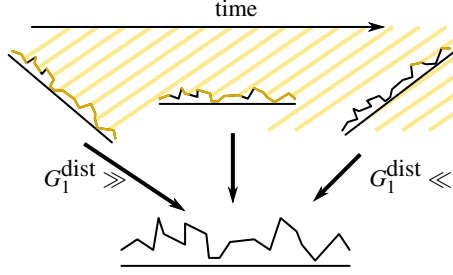
Figure 6: *Aggregating temporal changes of the macro normal* **N** *(top) into surface roughness (bottom) looks very similar to aggregating surface curvature (figure 4). Note, while we want to average the effect of local shadowing G_1^{local}, microfacets do not shadow each other at different time steps. Thus, we compute the distant shadowing term G_1^{dist} from the base roughness only, i.e. without accounting for the convolution with the temporal roughness P.*
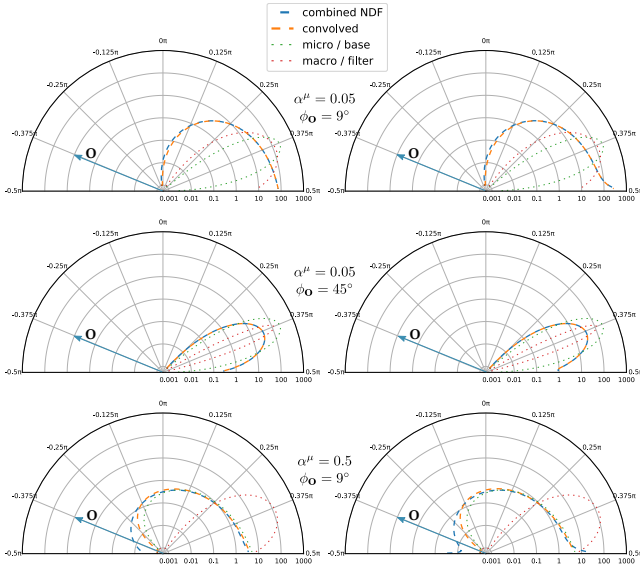
Figure 7: *Comparison of ground truth f_{comb} (eq. (15)), combined BSDF (numerically integrated) and approximation f_{conv} (eq. (17)). The plots show the angular BSDF response for a fixed outgoing direction* **o**. *For all of them the macro NDF is a Beckmann distribution with $\alpha_1^t = 0.3$ along the motion direction and $\alpha_2^t = 0.015$ perpendicular to that. In the first row the base roughness is $\alpha^\mu = 0.05$ and the azimuthal angle ϕ between* **o** *and major axis of filter anisotropy is $9°$. In the second row $\alpha^\mu = 0.05$ and $\phi = 45°$ and in the last row, $\alpha^\mu = 0.5$ and $\phi = 0$. In the left column the full G-term is used for the introduced filtering roughness, which matches fairly well with the approximation if also therein the full G_2 term is used. As shown in the right column, setting the G_{conv} to G^μ is able to capture the temporal case in which the macro G is 1.*

the micro-NDF belongs to a Cook-Torrance microfacet BSDF, the combined BSDF

$$f_{comb}(\mathbf{o},\mathbf{i},\mathbf{N}) = \int_{\Omega^+} \frac{|\langle\mathbf{m},\mathbf{o}\rangle|}{|\langle\mathbf{N},\mathbf{o}\rangle|} f^\mu(\mathbf{o},\mathbf{i},\mathbf{m}) \frac{|\langle\mathbf{m},\mathbf{i}\rangle|}{|\langle\mathbf{N},\mathbf{i}\rangle|} D_\mathbf{N}^t(\mathbf{m})\, d\mathbf{m}$$

$$= \int_{\Omega^+} \frac{F(\mathbf{o},\mathbf{i}) D_m^\mu(\mathbf{h}(\mathbf{i},\mathbf{o})) G^\mu(\mathbf{i},\mathbf{o},\mathbf{h})}{4|\langle\mathbf{N},\mathbf{o}\rangle||\langle\mathbf{N},\mathbf{i}\rangle|} D_\mathbf{N}^t(\mathbf{m})\, d\mathbf{m} \quad (15)$$

becomes an integral over all half-vectors **m** of the temporal macro-NDF, (similarly to the spatial case of [WMLT07, eq. (8)]), which makes it impractical to sample and evaluate during rendering. This aggregation of the spatial micro and temporal macro NDFs has the structure of a convolution [KHPL16, eq. (12)]. Beckmann NDFs seem most amenable to this structure, since they are Gaussians in the slope domain where the convolution of Gaussians with other Gaussians is closed. Unfortunately, this property does not carry over to the spherical domain, as the projection from slope space does not result in a Gaussian but to an approximation of this convolution. However, as long as the micro NDF only has up to moderate roughness, the addition of covariance matrices similar to equation (4), i.e.

$$\Sigma_{conv} = \Sigma^\mu + P \quad (16)$$

fortunately provides a good approximation: the combined NDF D_{comb} can be approximated well by a Beckmann distribution D_{conv} that corresponds to the sum Σ_{conv} of the covariance matrices in a common local frame. Therefore the combined BSDF has the form

$$f_{comb}(\mathbf{o},\mathbf{i},\mathbf{N}) \approx f_{conv} = \frac{F(\mathbf{o},\mathbf{i}) D_{conv}(\mathbf{h}(\mathbf{i},\mathbf{o})) G_{conv}(\mathbf{i},\mathbf{o},\mathbf{h})}{4|\langle\mathbf{N},\mathbf{o}\rangle||\langle\mathbf{N},\mathbf{i}\rangle|}, \quad (17)$$

where G_{conv} denotes the combined G term, which still needs to be chosen appropriately. The G term of the Beckmann distribution implicitly depends on its roughness parameter, so a naive approach would be to applying the standard shadowing/masking term, i.e., the one corresponding to Σ_{conv}. We cannot expect this to be an accurate approximation, since this masking and shadowing would also account for temporal roughness.

In figure 7 we compared f_{conv} (dashed orange) with the ground-truth f_{comb} (dashed blue) which we obtain by brute-force numerical evaluation of equation (15). Indeed, the left column using the standard G term for the combined BSDF provides a good match as long as G is used for both micro and macro NDF.

The right column indicates a different behaviour of the ground truth at grazing angles if we omit the G term for the temporal NDF. Setting $G_{conv} := G^\mu$ recovers this behaviour well also at grazing angles and significant differences only start to exhibit at a high base roughness (bottom row, $\alpha^\mu = 0.5$).

Approximating D^t There is in no hope to describe D^t accurately in the full shutter range. Due to the complex interplay of the time-dependent intersection point, potentially high geometric detail moving with high velocity, the behaviour of **N** in a neighbourhood of a reference point is highly non-linear. Since the apparent normal variation depends on ray direction, and because of changes in visibility it would be infeasible to precompute and store this data.

To make use of the temporal NDF in practice, we approximate D_t only in a small neighbourhood in time, thereby limiting these
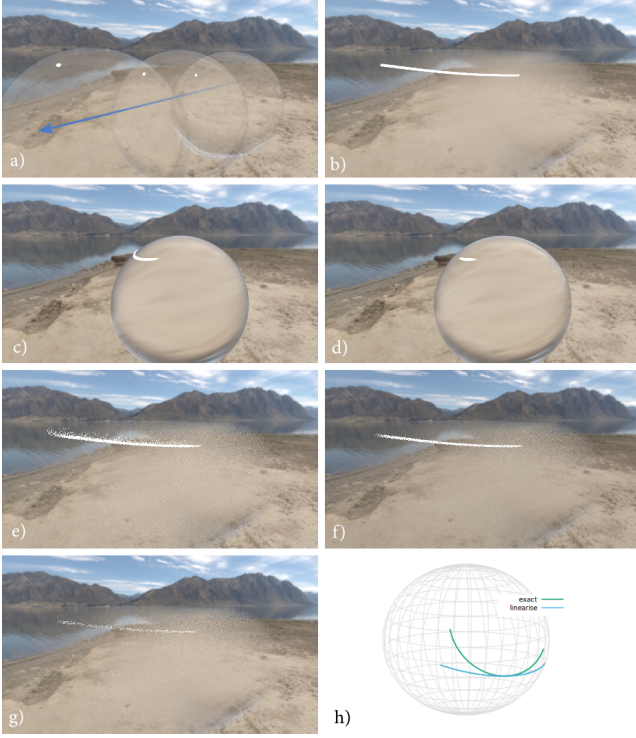
Figure 8: *Linearisation error: a) sphere moving from back to front. b) the highlight moves along a curved trajectory due to change of view angle (reference). c) applying anisotropic temporal roughness based on $\partial_t \mathbf{N}$. d) ours Δt weighted down by cosine with viewing direction. e) motion blurred version of c). f) motion blurred version of d). g) classic motion blurred render without temporal roughening, all rendered at 16 samples. h) collected normals over time determined by ray tracing (exact) and the first-oder approximation using just the first derivative (linearise). [TK19] perform a similar cosine downweighting effectively, since they use the projected solid angle half vector instead of the slope space half vector.*

higher-order errors. The apparent temporal variation of the Gauss map as a seen by the ray, i.e., $\mathbf{N}(t, \mathbf{P}(t))$ (in short $\mathbf{N}(t)$) is the central quantity that remains to estimate. We use the normal derivative $\partial_t \mathbf{N}(t)$ as a first-order approximation to extrapolate the normal away from the center, which provides a representative slope m that is then mapped onto the roughness parameter of a Beckmann distribution.

Since $\partial_t \mathbf{N}(t) \perp \mathbf{N}$, the derivative can also be used to define the major principal tangent direction. The temporal variation is one-dimensional so we set the roughness in the minor axis to 0, or to a small value in case the base NDF is perfectly specular.

More precisely, after choosing Δt to only a subset of the shutter, $\Delta t \partial_t \mathbf{N}(t)$ directly corresponds to the plane-parallel slope relating to the opening angle α between $\mathbf{N}$ and $\tilde{\mathbf{N}}$ parallel to $\mathbf{N} + \Delta t \partial_t \mathbf{N}(t)$, or in other words, the slope is
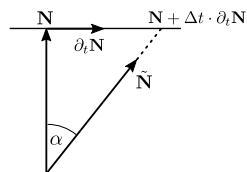
Figure 9: *Close up of moving spray particles from Fig. 1 without (left) and with temporal filtering (right).*

$m = \tan \alpha = \|\Delta t \cdot \partial_t \mathbf{N}(t)\|$. We map this value directly to the roughness value of the Beckmann distribution.

The next section describes how to compute this derivative. As the apparent normal depends both on the motion of the intersected object as well as the motion of the incoming ray, it naturally adapts the relative motion. In particular, a camera movement tracked with an object will cancel each other out and therefore avoid overblurring. This approach therefore depends on temporal ray differentials, which are a rather straightforward adaptation from classic ray differentials [Ige99]. The main difference is that the derivative of the intersection distance $s(t)$ is now also dependent on time. As all methods that utilise ray differentials, this limits the approach to unidirectional path tracing.

Added roughness helps reducing variance in conjunction with next-event estimation which is effective for sampling to small, bright light sources and high-frequency IBLs.

Figure 8 shows how the low-order approximation of the normal variation can lead to undesired shapes of the highlight. In order for the blurred version of the highlight to align perfectly with the correct trajectory (8b), the distribution would need to be curved on the sphere (green line in 8h). Due to linearisation we map to a geodesic trajectory (blue line in 8h), which results in the overly curved highlight in 8c and 8e. To account for this, we would require more information such as higher-order derivatives, which might be impractical to achieve. To avoid this effect, we observe that these errors usually happen at grazing angles where the directional changes are most significant and apply a simple weighting term, by scaling Δt with the cosine of the normal and incident direction (8c and 8f).

As shown in figure 9, the temporal NDF cannot handle occlusion boundaries. The highlight trajectories are broken up by occlusion of other particles. When temporal filtering is enabled these highlights can cause some overblurring when the velocity is large.

Indirect rays Since we only modify the roughness of the BSDF, we can easily apply our technique to indirect bounces as well. The ray footprint has to be derived by propagating the ray differential through all previous bounces. This can widen the footprint beyond the range in which a local approximation of the NDF is valid, so it

has to be clamped. The local environment around a transport path in path space contributes the result of the convolution of all events to the framebuffer. In particular, the temporal integration is only performed once per path, so all BSDF events along a path have to share the extra roughness introduced by the temporal NDF. Applying the full convolution to every vertex would introduce overblur. Since a convolution equates to summing variances, this means the temporal Beckmann roughness α^t should be distributed among path vertices i such that $\sum_i (\alpha_i^t)^2 = (\alpha^t)^2$.

4. Computing derivatives

This section discusses how to compute the derivatives $\partial_t s(t)$ and $\partial_t \mathbf{N}(t)$ that are required to compute the temporal ray-differentials.

4.1. Implicit surfaces

We focus our discussion on implicit surfaces (cf. [OF03]), because they are ubiquitous in FX elements for movie production in the form of grid-based level set functions, blobbies, and ellipsoids. Elements such as water splashes and spray particles often feature high velocities and complex geometry which are good candidates to apply our approach. Furthermore, implicits provide a convenient mathematical toolset to compute the derivatives that we need as main ingredient to estimate the temporal normal variation.

For a level set function $\phi : \mathbb{R} \times \Omega \to \mathbb{R}$ (where $\Omega \subseteq \mathbb{R}^d$) we set $\mathcal{M}_{t,c}[\phi] := \{\phi(t, \cdot) = c\}$. By convention we often use the zero level set $\mathcal{M}(t) := \mathcal{M}_{t,0}$ and denote by $\mathbf{N}_\phi(t, x)$ its normal, i.e.,

$$\mathbf{N}_\phi(t,x) := \frac{\nabla_x \phi(t,x)}{\|\nabla_x \phi(t,x)\|}. \tag{18}$$

Temporal derivative of intersection For a time-dependent ray $r(t) = (\mathbf{o}(t), \mathbf{d}(t))$ the intersection distance s is implicitly defined by

$$f(t,s) := \phi(t, \mathbf{o}(t) + s\mathbf{d}(t)) = 0. \tag{19}$$

By the implicit function theorem there exists a function $s : U \to \mathbb{R}$ in a neighbourhood U around t_0 such that $f(t, s(t)) = 0$ and

$$\partial_t s(t) = -\frac{\partial_t f(t, s(t))}{\partial_s f(t, s(t))} = -\frac{\partial_t \phi + \langle \nabla_x \phi, \partial_t \mathbf{o} + s\partial_t \mathbf{d}\rangle}{\langle \nabla_x \phi, \mathbf{d}\rangle}. \tag{20}$$

Note, that in the implicit form this is a simpler and more general derivation of (12) in [Ige99].

Normal derivative on ray intersection Using equations (20), (27) and (28) (see Appendix A), the normal derivative at the ray intersection is

$$\partial_t \mathbf{N}(t) = \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{N}_\phi(t, \mathbf{o}(t) + s(t)\mathbf{d}(t))$$
$$= \partial_t \mathbf{N}_\phi + \nabla_x \mathbf{N}_\phi \cdot (\partial_t \mathbf{o} + \partial_t s\mathbf{d} + s\partial_t \mathbf{d}). \tag{21}$$

4.2. Parametric surfaces

For parametric surfaces $\mathbf{x} : \mathbb{R} \times V \to \mathbb{R}^3$ with $V \subset \mathbb{R}^2$ the analogue of the derivative in equation (21) is more complicated to obtain. In particular going through local coordinates to obtain $\partial_t s(t)$ becomes

unwieldy. We note that with regard to motion blur the rigid motion component in particular translation and rotation by far dominate higher-order changes such as deformation and bending. Therefore, we confine ourselves to expressing rigid motion of the local geometry. Since we only need to compute derivatives of normals, the second fundamental form is a sufficient description [dC76].

In the parametric form of $\mathbf{x}(u,v)$ and the Gauss map $\mathbf{N}(u,v)$ the principal curvatures κ_1 and κ_2 at a surface point $\mathbf{P}$ and their corresponding principal curvature directions $\mathbf{e}_1$ and $\mathbf{e}_2$ can be computed. The quadric approximation to the surface is then given by:

$$\mathbf{x}(u,v) = \frac{1}{2}u^2 \kappa_1 \mathbf{e}_1 + \frac{1}{2}v^2 \kappa_2 \mathbf{e}_2 + \mathbf{P} \tag{22}$$

and can be expressed in implicit form as

$$\hat{\phi}(x) = \frac{1}{2}\kappa_1 \langle x - \mathbf{P}, \mathbf{e}_1\rangle^2 + \frac{1}{2}\kappa_2 \langle x - \mathbf{P}, \mathbf{e}_2\rangle^2 + \langle x - \mathbf{P}, \mathbf{N}\rangle \overset{!}{=} 0. \tag{23}$$

Finally, we compose $\hat{\phi}$ with a motion transformation $M : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ which accounts for translation and rotation, i.e.,

$$\phi(t, \mathbf{x}) := \hat{\phi}(M(t, \mathbf{x})), \tag{24}$$

which allows us to apply the computations for implicits fom the previous section and Appendix A.

5. Results

Figure 1 shows a production FX asset with an implicit surface defined by particles. This illustrates the practical applicability and scalability of the approach. We always compare equal sample images, because there is virtually no performance impact of the method (i.e. a little bit of compute is added and can be hidden between memory fetches).

Comparison to Specular Anti Aliasing We implemented previous work for spatial specular anti aliasing of high curvature [KHPL16, TK19] in a GPU path tracer. To facilitate comparisons, we implemented our technique in this framework, too, and use publicly available implicit ray tracing scenes: the Gears scene (figure 10, https://www.shadertoy.com/view/tt2XzG), the Chain scene (figure 11, https://www.shadertoy.com/view/wlXSD7) and the Splash scene (figure 12, https://www.shadertoy.com/view/4lSSRW). The lighting setup is a relatively distant spherical light behind the camera and the BSDF is a reflective dielectric microfacet model with isotropic roughness of 0.02. Anisotropy is only introduced from curvature and motion, to be fair towards all methods ([TK19] only supports the isotropic case). Along with the images showing only the highlights, we provide a false color image (roughness AOV) where the red and green channels map the final roughness along the tangent and bitangent respectively.

Gears Figure 10 shows a ball made of rotating gears and proves to be a good test for motion vectors going in all directions. The fast changing normals around the gears makes it hard to resolve efficiently. KHPL16 increase the roughness around edges, which does not help with fast motion, but keeps the highlight sharp (orange insets); TK19 on the other hand, being isotropic in nature, blurs
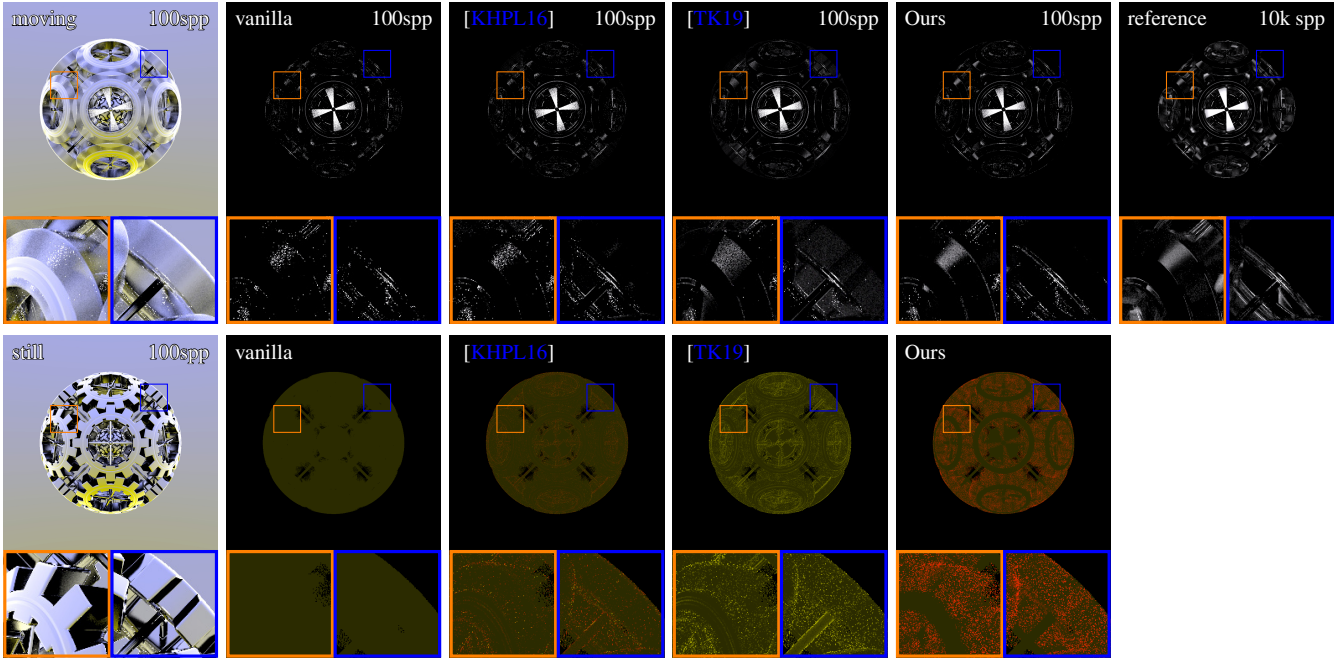
Figure 10: *A ball made of rotating gears, moving in different directions. Above: showing only highlights for comparisons. Below: roughness mapped to red and green channel for tangent and bitangent respectively. Notice the hightlight (orange inset) and the side of the gear (blue inset).*

not only around the moving axis but also on its side, resulting in a blurred highlight and a rougher side of the gear (blue insets). This can be easily seen in the roughness AOV. Our method increases the roughness only along the motion vector, resulting in a sharper and more converged image.

Chain Figure 11 shows a chain moving upwards, a simple scene with many smooth and round edges, in an axis aligned motion. As expected, no method has particular issues in converging to the right image, but does not add much roughness either. In our case we increase it along the moving direction, obtaining a faster convergence compared to the others.

Splash Figure 12 shows a pool of blobs, rotating and twisting in a vortex of turbulent motion. This scene is particularly hard for all techniques: it presents both circular motion (orange inset) and straight fast movement (blue inset). In the first case, the revolution smooths well with TK19 but it also create very blurry highlights; KHPL16 instead, being anisotropic, captures better the spatial change of the normal but lacks in convergence. Adding roughness along the motion keeps the highlights sharp with minimal blur.

6. Conclusion and Future Work

We introduced temporal normal distribution functions for prefiltering microfacet BSDF for fast moving near specular surfaces. This is especially useful for cinematic movie production where often long shutter intervals meet high speed content, such as curved metal surfaces or FX elements for fluids. Analytically handling the convolution of the signal with the shutter interval, by pushing it into

a microfacet normal distribution, enables us to render such cases more efficiently. This integrates well into existing path tracers and handles edge cases more robustly, such as static geometry seen in a moving mirror or a camera tracking a fast moving object.

A few open questions and possible variations of this approach remain. We briefly touched upon indirect bounces. Filtering secondary interactions requires path differentials and potentially also temporal ray differentials to determine the indirect pixel footprint. Such footprints can easily become very large and then have to be clamped to be useful. How to do this in a rigorous way is still an open topic. Temporal ray differentials themselves have unsolved problems, for example when a bounce and a consecutive bounce both move in the same frame of reference: naive application will account for the same movement twice and lead to overblur.

This filtering idea can very likely be employed for depth-of-field as well. The motivation and problem statement are equivalent. It requires a mapping of the circle of confusion to roughness via the shape operator. Such an extension would be in line with the further development a sheared postfiltering approach received in the past [MYRD14].

An extension to micro NDFs other than Beckmann's is another open question with practical relevance, as for instance the GGX NDF is very popular. In a different setting, the convolution of GGX lobes has been approximated by Gaussian convolution and a correction factor [WHD17]. We also approximate the local neighbourhood in time with a Gaussian, which may profit from a better match to the actual shutter envelope function.
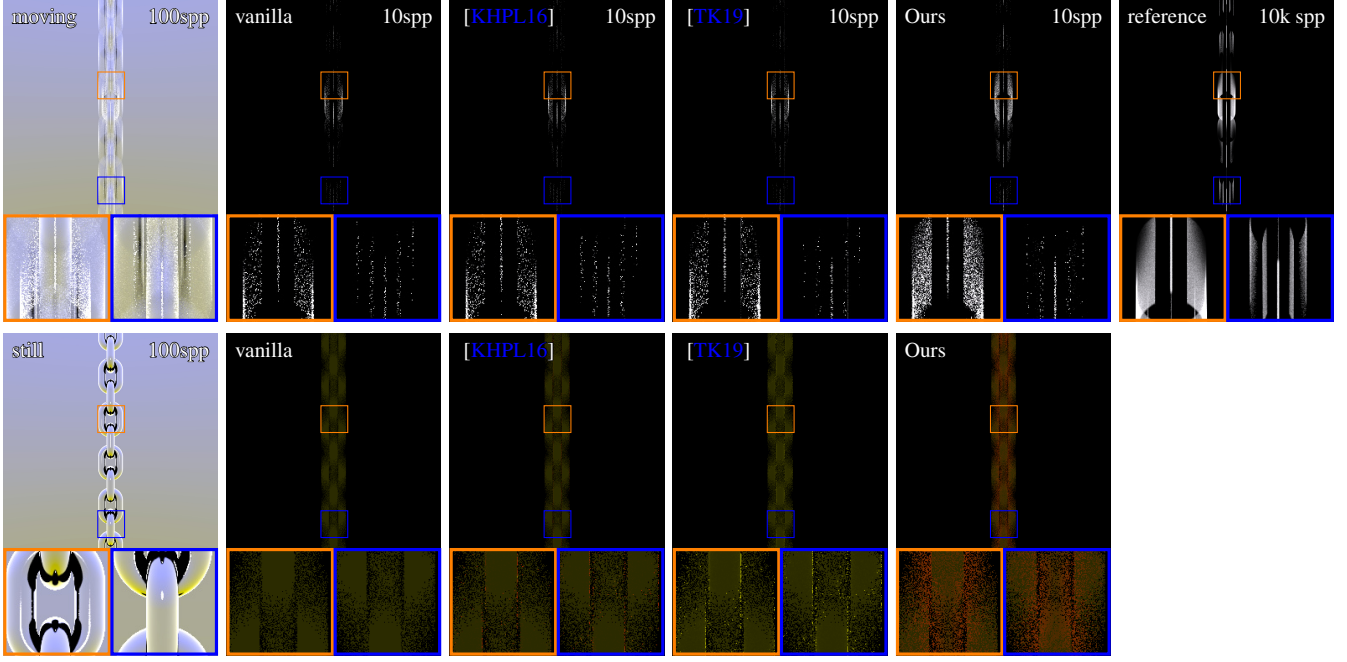
Figure 11: *A chain moving upwards. Above: showing only highlights for comparisons. Below: roughness mapped to red and green channel for tangent and bitangent respectively. Adding more roughness along the moving axis results in faster convergence.*
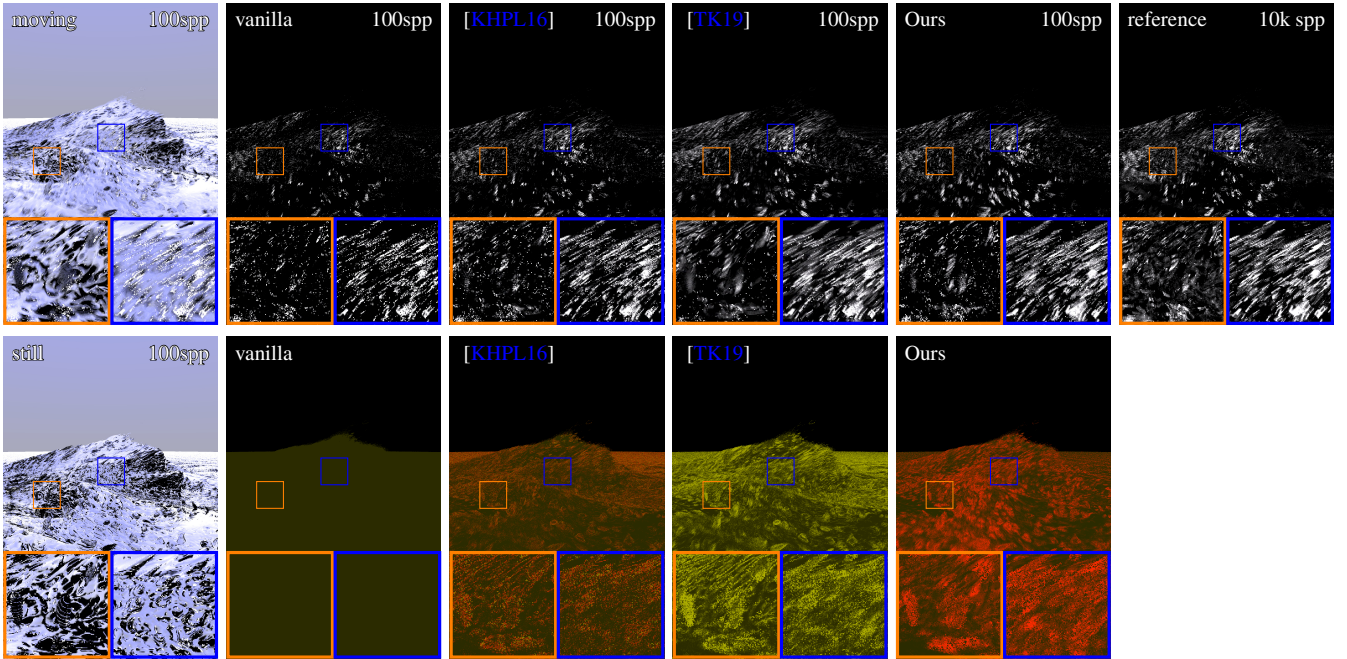
Figure 12: *A vortex of blobs rotates and twists around itself, with many small particles in turbulent motion. Above: showing only highlights for comparisons. Below: roughness mapped to red and green channel for tangent and bitangent respectively. We can see some concentric (and thus more isotropic) movement (orange inset) and fast straight motion (blue inset).*

Acknowledgements

References

[BSS*13] BELCOUR L., SOLER C., SUBR K., HOLZSCHUCH N., DURAND F.: 5d covariance tracing for efficient defocus and motion blur. *ACM Transactions on Graphics 32*, 3 (July 2013). 3

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Transactions on Graphics 1*, 1 (Jan. 1982), 7–24. 2

[dC76] DO CARMO M. P.: *Differential geometry of curves and surfaces.* Prentice Hall, 1976. 8

[DHI*13] DUPUY J., HEITZ E., IEHL J.-C., POULIN P., NEYRET F., OSTROMOUKHOV V.: Linear efficient antialiased displacement and reflectance mapping. *ACM Transactions on Graphics 32*, 6 (Nov. 2013). 2

[DSHL10] DAMMERTZ H., SEWTZ D., HANIKA J., LENSCH H.: Edge-avoiding à-trous wavelet transform for fast global illumination filtering. In *Proc. of ACM SIGGRAPH / Eurographics conference on High Performance Graphics* (2010), pp. 67–75. 2

[EBR*14] ELEK O., BAUSZAT P., RITSCHEL T., MAGNOR M., SEIDEL H.-P.: Spectral Ray Differentials. *Computer Graphics Forum* (2014). 3

[ETH*09] EGAN K., TSENG Y.-T., HOLZSCHUCH N., DURAND F., RAMAMOORTHI R.: Frequency analysis and sheared reconstruction for rendering motion blur. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* (New York, NY, USA, 2009), SIGGRAPH '09, Association for Computing Machinery. 2

[FHH*19] FASCIONE L., HANIKA J., HECKENBERG D., KULLA C. D., DROSKE M., SCHWARZHAUPT J.: Path tracing in production: part 1: modern path tracing. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH 2019, SIGGRAPH 2019 Courses, Los Angeles, CA, USA, July 28 - August 01, 2019* (2019), pp. 19:1–19:113. 2

[FHP*18] FASCIONE L., HANIKA J., PIEKÉ R., VILLEMIN R., HERY C., GAMITO M., EMROSE L., MAZZONE A.: Path tracing in production. In *SIGGRAPH 2018 Courses* (2018), pp. 15:1–15:79. 2

[GLA*19] GHARBI M., LI T.-M., AITTALA M., LEHTINEN J., DURAND F.: Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics (Proc. SIGGRAPH) 38*, 4 (2019). 2

[HD09] HERBST K., DIMIAN D.: Practical uses of a ray tracer for "Cloudy With a Chance of Meatballs". SIGGRAPH 2009 Talks, 2009. 2

[HDF15] HANIKA J., DROSKE M., FASCIONE L.: Manifold next event estimation. *Computer Graphics Forum 34*, 4 (July 2015), 87–97. 3

[Hei14] HEITZ E.: Understanding the masking-shadowing function in microfacet-based BRDFs. *Journal of Computer Graphics Techniques (JCGT) 3*, 2 (June 2014), 48–107. 4, 5

[HHdD16] HEITZ E., HANIKA J., d'EON E., DACHSBACHER C.: Multiple scattering microfacet BSDFs with the Smith model. *ACM Transactions on Graphics (Proc. SIGGRAPH) 35*, 4 (Aug. 2016), 58:1–58:14. 4

[Ige99] IGEHY H.: Tracing ray differentials. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* (1999). 3, 7, 8

[JG19] JENDERSIE J., GROSCH T.: Microfacet model regularization for robust light transport. *Computer Graphics Forum 38*, 4 (2019), 39–47. 2

[JHY*14] JAKOB W., HASAN M., YAN L.-Q., LAWRENCE J., RAMAMOORTHI R., MARSCHNER S.: Discrete stochastic microfacet models. *ACM Transactions on Graphics 33*, 4 (July 2014). 2

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 31*, 4 (July 2012), 58:1–58:13. 3, 4

[KHPL16] KAPLANYAN A. S., HILL S., PATNEY A., LEFOHN A.: Filtering distributions of normals for shading antialiasing. In *Proc. of ACM SIGGRAPH / Eurographics conference on High Performance Graphics* (2016). 2, 3, 4, 6, 8, 9, 10, 11

[KVHS00] KAUTZ J., VÁZQUEZ P.-P., HEIDRICH W., SEIDEL H.-P.: A unified approach to prefiltered environment maps. In *Rendering Techniques 2000* (Vienna, 2000), Péroche B., Rushmeier H., (Eds.), Springer Vienna, pp. 185–196. 3

[LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable Monte Carlo ray tracing through edge sampling. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 37*, 6 (2018), 222:1–222:11. 3

[LHJ19] LOUBET G., HOLZSCHUCH N., JAKOB W.: Reparameterizing discontinuous integrands for differentiable rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 38*, 6 (Dec. 2019). 3

[Lov05] LOVISCACH J.: Motion Blur for Textures by Means of Anisotropic Filtering. In *Proc. Eurographics Symposium on Rendering* (2005), Bala K., Dutre P., (Eds.), The Eurographics Association. 2

[MN88] MITCHELL D. P., NETRAVALI A. N.: Reconstruction filters in computer-graphics. *Computer Graphics (Proc. SIGGRAPH) 22*, 4 (June 1988), 221–228. 2

[MYRD14] MEHTA S. U., YAO J., RAMAMOORTHI R., DURAND F.: Factored axis-aligned filtering for rendering multiple distribution effects. *ACM Transactions on Graphics 33*, 4 (July 2014). 10

[OB10] OLANO M., BAKER D.: Lean mapping. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, Association for Computing Machinery, pp. 181–188. 2

[OF03] OSHER S., FEDKIW R.: *Level Set Methods and Dynamic Implicit Surfaces.* Springer Verlag, 2003. 8

[SFES11] SCHJØTH L., FRISVAD J. R., ERLEBEN K., SPORRING J.: Photon differentials in space and time. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications* (Berlin, Heidelberg, 2011), Richard P., Braz J., (Eds.), Springer Berlin Heidelberg, pp. 274–286. 3

[SS11] STAM J., SCHMIDT R.: On the velocity of an implicit surface. In *ACM Transactions on Graphics* (2011), vol. 30. 12

[SSE09] SPORRING J., SCHJØTH L., ERLEBEN K.: *Spatial and Temporal Ray Differentials.* Tech. Rep. 09/04, Københavns Universitet, 2009. 3

[STN87] SHINYA M., TAKAHASHI T., NAITO S.: Principles and applications of pencil tracing. In *Computer Graphics (Proc. SIGGRAPH)* (New York, NY, USA, 1987), SIGGRAPH '87, Association for Computing Machinery, pp. 45–54. 3

[SW01] SUYKENS F., WILLEMS Y. D.: Path differentials and applications. In *Rendering Techniques 2001* (Vienna, 2001), Gortler S. J., Myszkowski K., (Eds.), Springer Vienna, pp. 257–268. 3

[TK19] TOKUYOSHI Y., KAPLANYAN A. S.: Improved geometric specular antialiasing. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2019), I3D, pp. 18:1–18:9. 3, 7, 8, 9, 10, 11

[Tur19] TURQUIN E.: *Practical multiple scattering compensation for microfacet models.* Tech. rep., Industrial Light & Magic, 2019. 4

[WHD17] WEBER P., HANIKA J., DACHSBACHER C.: Multiple vertex next event estimation for lighting in dense, forward-scattering media. *Computer Graphics Forum (Proc. of Eurographics)* (Apr. 2017). 10

[WMLT07]   WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.:
Microfacet models for refraction through rough surfaces. In *Proc. Euro-graphics Symposium on Rendering* (Aire-la-Ville, Switzerland, Switzer-land, 2007), EGSR'07, Eurographics Association, pp. 195–206. 5, 6

[WZHB09]   WALTER B., ZHAO S., HOLZSCHUCH N., BALA K.: Sin-gle scattering in refractive media with triangle mesh boundaries. *ACM Transactions on Graphics 28*, 3 (July 2009). 3

[YHJ*14]   YAN L.-Q., HASAN M., JAKOB W., LAWRENCE J.,
MARSCHNER S., RAMAMOORTHI R.:   Rendering glints on high-resolution normal-mapped specular surfaces. *ACM Transactions on Graphics 33*, 4 (July 2014). 2

[ZJL*15]   ZWICKER M., JAROSZ W., LEHTINEN J., MOON B., RA-MAMOORTHI R., ROUSSELLE F., SEN P., SOLER C., YOON S.-E.: Re-cent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum (Proc. of Eurographics) 34*, 2 (May 2015), 667–681. 2

Appendix A: Normal derivatives on implicit surfaces

We provide a small toolbox for computing normal derivatives that are needed for temporal and spatial ray differentials and as well as to obtain $\partial_t \mathbf{N}(t)$ in Eq. (21).

Definition 1 (Projection) For a function $\mathbf{v} : \Omega \to \mathbb{R}^n$, we define

$$\mathcal{P}[\mathbf{v}] := \left(\mathbb{1} - \frac{\mathbf{v}}{\|\mathbf{v}\|} \otimes \frac{\mathbf{v}}{\|\mathbf{v}\|} \right) \quad \text{with } a \otimes b := ab^T. \qquad (25)$$

Remark 1

(a) We have

$$\partial_t \left(\frac{\mathbf{v}(t)}{\|\mathbf{v}(t)\|} \right) = \frac{1}{\|\mathbf{v}(t)\|} \left(\mathbb{1} - \frac{\mathbf{v}(t)}{\|\mathbf{v}(t)\|} \otimes \frac{\mathbf{v}(t)}{\|\mathbf{v}(t)\|} \right) \cdot \partial_t \mathbf{v}(t)$$

(b) More general, for a differential operator D,

$$D \left(\frac{\mathbf{v}}{\|\mathbf{v}\|} \right) = \frac{1}{\|\mathbf{v}\|} P[\mathbf{v}] D\mathbf{v}. \qquad (26)$$

(c) Therefore, temporal derivative of the normal is

$$\partial_t \mathbf{N}_\phi(t,x) = \partial_t \left(\frac{\nabla_x \phi(t,x)}{\|\nabla_x \phi(t,x)\|} \right) = \frac{1}{\|\nabla_x \phi\|} \mathcal{P}[\nabla_x \phi] \partial_t \nabla_x \phi(t,x). \qquad (27)$$

(d) Similarly, the gradient of the normal is given by

$$\nabla_x \mathbf{N}_\phi(t,x) = \frac{1}{\|\nabla_x \phi\|} \mathcal{P}[\nabla_x \phi] D^2 \phi(t,x). \qquad (28)$$

See also [SS11] where similar equations are used to estimate tan-gential velocities from a time-dependent level set function.

Definition 2 (Tangential gradient) The tangential gradient of a function $u \in \mathcal{C}^1(\Omega)$ is given by $\nabla_\mathcal{M} u := \mathcal{P}[\mathcal{M}] \nabla_x u$.

Remark 2 (Shape Operator) The matrix representation of the Weingarten map on the tangent space of $\mathcal{M}_c[\phi]$ is given by

$$\mathcal{S} := \frac{1}{\|\nabla_x \phi\|} \mathcal{P}[\nabla_x \phi] D^2 \phi \mathcal{P}[\nabla_x \phi] = D\mathbf{N}_\phi \mathcal{P}. \qquad (29)$$

This is used to compute the derivative of the normal for classic ray-differentials, i.e.,

$$\frac{\partial N}{\partial u} = \mathcal{S} \frac{\partial x}{\partial u}. \qquad (30)$$