Microfacet-Based Normal Mapping for Robust Monte Carlo Path Tracing

Non-Separable Masking-Shadowing

Vincent Schüßler[1] Eric Heitz[2] Johannes Hanika[1] Carsten Dachsbacher[1]
[1]Karlsruhe Institute of Technology [2]Unity Technologies

Contents

1 Introduction

In the main paper we made the assumption of separable-masking shadowing, i.e. $G_2(\omega_o, \omega_i) = G_1(\omega_o)\, G_1(\omega_i)$. This has the intuitive meaning of light being uniformly distributed on the facets. In fact, it is the only approximation of our model compared to ray tracing the microsurface. In order to visualize the impact of this assumption, we extend our algorithm to a true ray tracing simulation and thus incorporate effects of height correlation.

2 Importance Sampling

To importance sample our BRDF, we simply trace a ray on the microsurface that is not masked using Algorithm 1. Like before, we choose the facet on which the first intersection occurs based on the intersection probability. Additionally, we randomly choose a valid height for this intersection using the masking function. Starting from this initial point on the microsurface, we repeatedly sample a direction from the micro-BRDF and intersect the opposite facet to obtain the next path vertex. When a ray leaves the microsurface, we take its direction and accumulated energy throughput as the sampling result.

Algorithm 1 Importance Sampling Algorithm

```
ω_r ← −ω_i                                          ▷ ray direction
e ← 1                                               ▷ ray throughput
ω_m ← [U < λ_p(ω_i)  ?  ω_p  :  ω_t]                ▷ first facet
h ← 1 − U G_1(ω_r, ω_m)                             ▷ first height
while true do
    (ω'_r, w) ← sample f_m(−ω_r, ω'_r) ⟨ω'_r, ω_m⟩   ▷ sample
    e ← w e                                         ▷ update throughput
    h' ← intersect(other facet(ω_m), ω'_r, h)       ▷ intersect other facet for new height
    if h' > 1 then return (ω'_r, e)                 ▷ leave microsurface
    else
        ω_m ← other facet(ω_m)                      ▷ continue on other facet
        h ← h'                                      ▷ update height
        ω_r ← −ω'_r                                 ▷ update direction
    end if
end while
```

3 Evaluation

Evaluation of the BRDF could be performed like in the main paper by adding next-event estimation to Algorithm 1. We would have to introduce a binary shadowing test, because we can no longer use the masking function for this.

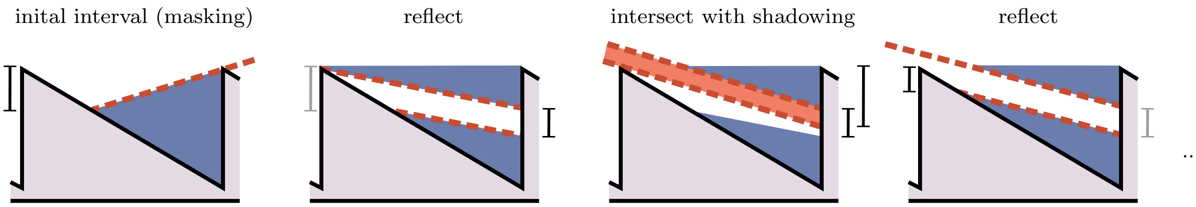

Figure 1: *Height interval tracking.* Instead of choosing a valid height for the first intersection, we track the interval of possible heights and use it for next-event estimation.

Instead, we employ a more efficient approach in Algorithm 2 (visualized in Figure 1) that results in less variance. The inefficiency of the naive approach stems from the early determination of an inital height. In this way, only one path on the microsurface is considered at once. Our algorithm is able to trace parallel paths simultaneously by tracking an interval of unmasked heights on the facets. The initial interval, which we used in our sampling algorithm for choosing the first height, is given by the masking function. Since we consider only parallel rays, the resulting distribution of heights on the opposite facet after one bounce is also a uniform interval. We update this interval by tracing rays from both the lowest and the highest point of the interval on the facet. For next event estimation, we compute a masking-shadowing value by intersecting the interval of unmasked heights with the interval of not-shadowed heights given by the masking function. Note that to simplify notation of our algorithm, we set the microfacet height to 1, which is in contrast to the convention of assuming an area of $1\mathrm{m}^2$ for the geometric surface.

Algorithm 2 Evaluation Algorithm

$L_o = 0$	▷ radiance collected from ω_o
$\omega_r \leftarrow -\omega_i$	▷ ray direction
$e \leftarrow 1$	▷ ray throughput
$\omega_m \leftarrow [\mathcal{U} < \lambda_p(\omega_i) \quad ? \quad \omega_p \ : \ \omega_t]$	▷ first facet
$H \leftarrow [1 - G_1(\omega_r, \omega_m), 1]$	▷ inital interval of heights
while true **do**	
$G_2 \leftarrow \lvert H \cap [1 - G_1(\omega_o, \omega_m), 1] \rvert$	▷ intersect intervals of valid heights on ω_m
$L_o \leftarrow L_o + e\, f_m(-\omega_r, \omega_o)\, \langle \omega_o, \omega_m \rangle\, \frac{G_2}{\lvert H \rvert}$	▷ eval
$(\omega_r', w) \leftarrow$ sample $f_m(-\omega_r, \omega_r')\, \langle \omega_r', \omega_m \rangle$	▷ sample
$e \leftarrow w\, e$	▷ update throughput
$H' \leftarrow$ intersect(other facet$(\omega_m), \omega_r', H)$	▷ intersect other facet for new heights
if $\mathcal{U} < \frac{\lvert H' \cap [1, \infty) \rvert}{\lvert H' \rvert}$ **then return** L_o	▷ leave microsurface
else	
$\omega_m \leftarrow$ other facet(ω_m)	▷ continue on other facet
$H \leftarrow H' \cap [0, 1]$	▷ update heights
$\omega_r \leftarrow -\omega_r'$	▷ update direction
end if	
end while	

4 Evaluation with Specular Tangent Facet

We modify our algorithm in the same way as with the separable model, resulting in Algorithm 3. To avoid the singularity at the specular tangent facet ω_t, we evaluate the BRDF on ω_p in direction $\omega_o' = \text{reflect}(\omega_o, \omega_t)$ instead. The height of rays starting on ω_p in direction ω_o' that are reflected on ω_t in direction ω_o is contained in an interval H_r. We use this interval to determine shadowing and compute it by intersecting rays from ω_t in direction $-\omega_o'$.

Algorithm 3 Evaluation Algorithm with Specular Tangent Facet

```
L_o = 0                                              ▷ radiance collected from ω_o
ω_r ← −ω_i                                           ▷ ray direction
e ← 1                                                ▷ ray throughput
ω_m ← [U < λ_p(ω_i)  ?  ω_p  :  ω_t]                 ▷ first facet
H ← [1 − G_1(ω_r, ω_m), 1]                           ▷ inital interval of heights
ω'_o ← reflect(ω_o, ω_t)
H_r ← intersect(ω_p, −ω'_o, [1 − G_1(ω_o, ω_t), 1])  ▷ heights on ω_p reflected towards ω_o
while true do
    if ω_m = ω_p then
        G_2 ← |H ∩ [1 − G_1(ω_o, ω_p), 1]|           ▷ intersect intervals of valid heights on ω_p
        L_o ← L_o + e f_m(−ω_r, ω_o) ⟨ω_o, ω_p⟩ G_2/|H|           ▷ eval
        G'_2 ← |H ∩ H_r|                             ▷ intersect intervals of valid heights on ω_p
        L_o ← L_o + e f_m(−ω_r, ω'_o) ⟨ω'_o, ω_p⟩ G'_2/|H|        ▷ eval
        (ω'_r, w) ← sample f_m(−ω_r, ω'_r) ⟨ω'_r, ω_m⟩            ▷ sample
        e ← w e                                      ▷ update throughput
    else
        ω'_r ← reflect(−ω_r, ω_t)                    ▷ reflect
    end if
    H' ← intersect(other facet(ω_m), ω'_r, H)        ▷ intersect other facet for new heights
    if U < |H' ∩ [1,∞)| / |H'| then return L_o       ▷ leave microsurface
    else
        ω_m ← other facet(ω_m)                       ▷ continue on other facet
        H ← H' ∩ [0, 1]                              ▷ update heights
        ω_r ← −ω'_r                                  ▷ update direction
    end if
end while
```

5 Results

We compare the appearance of the separable model to the non-separable model with specular material tangent facet (Figure 2 and Figure 3) and with same material tangent facet (Figure 4 and Figure 5). We also provide a Mitsuba plugin (**normalmap_microfacet_nonseparable**) that implements our non-separable model.

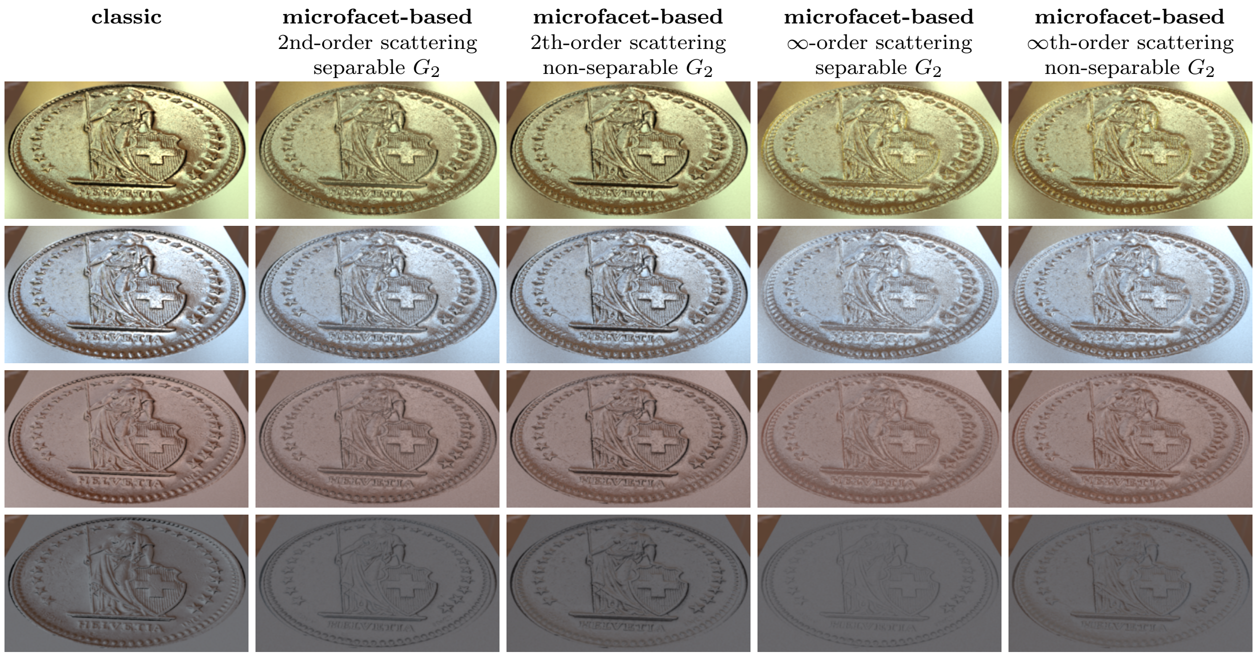

Figure 2: *Appearances of our model with specular material tangent facet.*

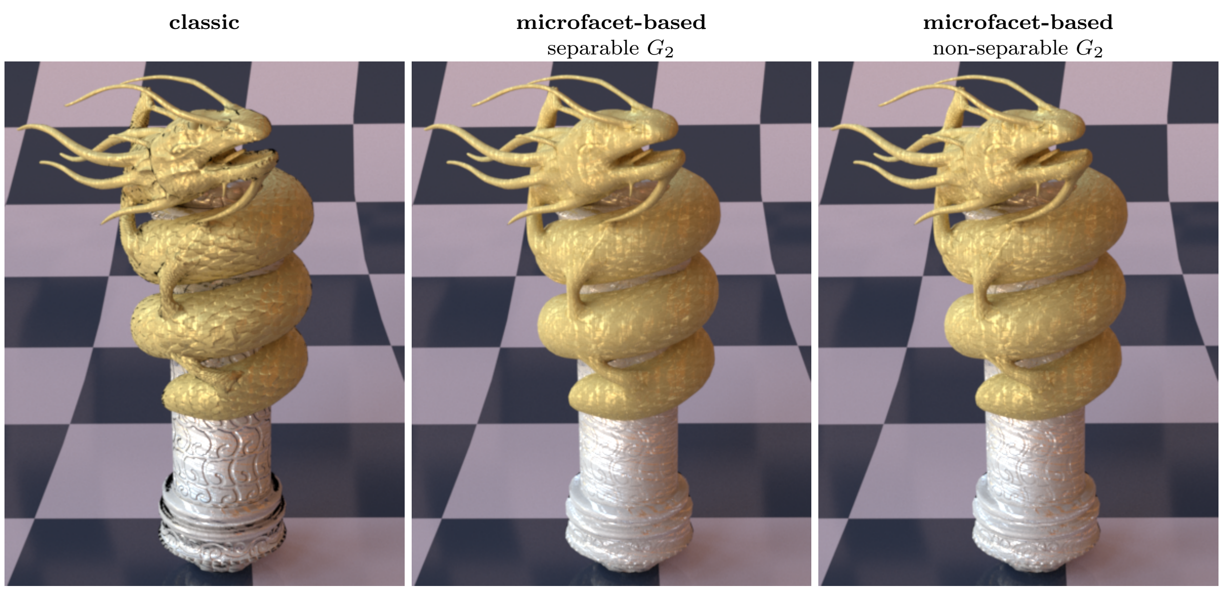

Figure 3: *Our specular material tangent facet model applied on a more complex geometric model (20000 triangles) with textured GGX roughness.*

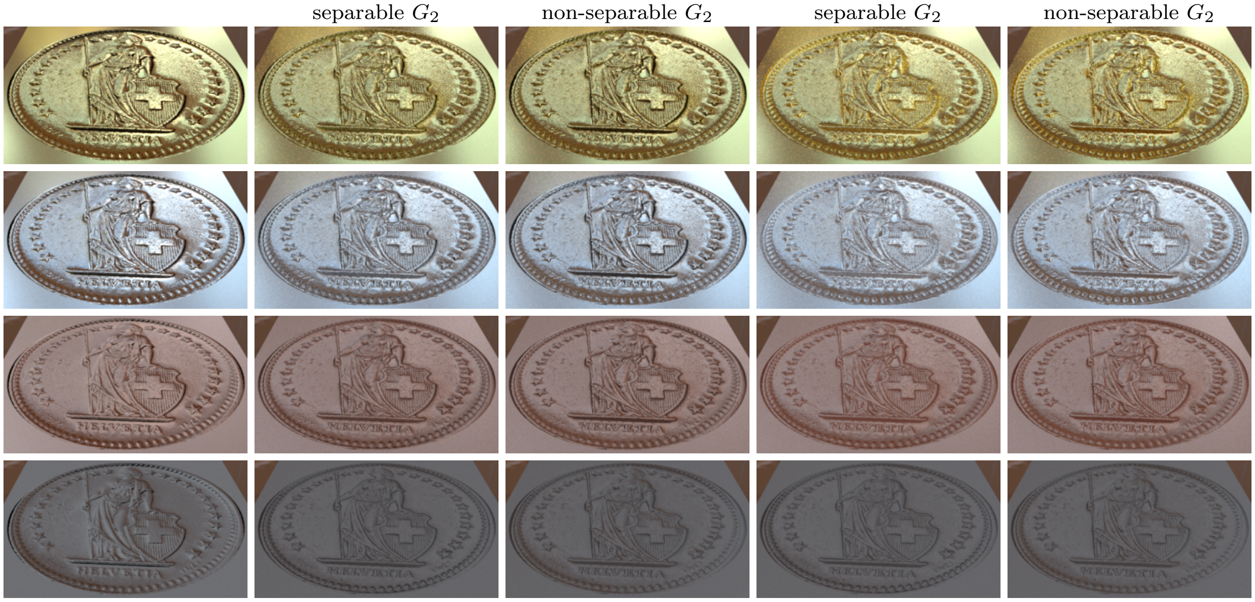

Figure 4: *Appearances of our model with same material tangent facet.*

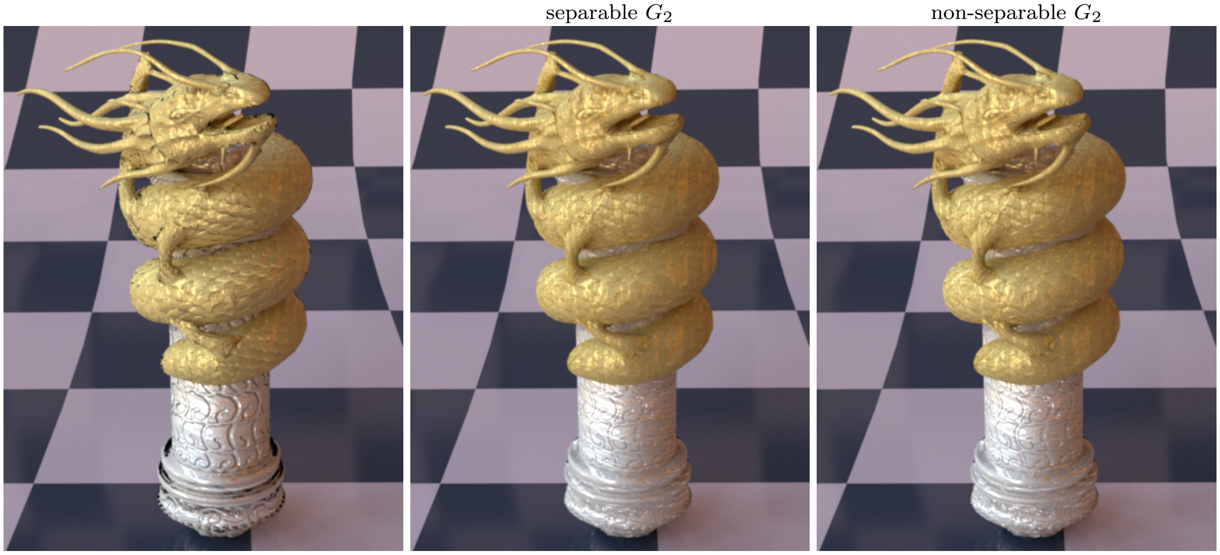

Figure 5: *Our same material tangent facet model applied on a more complex geometric model (20000 triangles) with textured GGX roughness.*