Supplementary: A Realistic Multi-scale Surface-based Cloth Appearance Model

Junqiu Zhu
University of California, Santa Barbara
USA
junqizhu@ucsb.edu

Carlos Aliaga
Meta Reality Labs Research
USA
caliaga@meta.com

Christophe Hery
Meta Reality Labs Research
USA
chery@meta.com

Lukas Bode
Meta Reality Labs Research
Switzerland
lbode@meta.com

Adrian Jarabo
Meta Reality Labs Research
Spain
ajarabo@meta.com

Ling-Qi Yan
University of California, Santa Barbara
USA
lingqi@cs.ucsb.edu

Matt Jen-Yuan Chiang
Meta Reality Labs Research
USA
mattchiang@meta.com

1 TERMS OF OUR BSDF MODEL
Our complete model is defined as:
\[ f(x, i, o) = f^{\text{s\text{heen}}}(x, i, o) + T^{\text{s\text{heen}}}(x) \left( f^{r,s}(x, i, o) + f^{r,d}(x, i, o) \right) + T^{\text{s\text{heen}}}(1 - l(x)) \delta(i, o). \] (1)

Surface Reflection. The specular reflection term \( f^{r,s} \) is defined following the SpongeCake model, as
\[ f^{r,s}(x, i, o) = \frac{k^{r,s}(x)D(h, x)G^r(i, o, x)}{4(i, n_s)(o, n_s)}. \] (2)
Where \( D \) is the SGGX distribution [Heitz et al. 2015] centered in the fiber tangent \( t(x) \) distribution, \( k^{r,s} \) is the albedo, and \( G^r \) is the attenuation defined as
\[ G^r(i, o, x) = \frac{1 - e^{-T(A(l(x)))}}{\Lambda(i, x) + \Lambda(o, x)}, \] (3)
with \( \Lambda \) the Smith shadowing/masking function.

The diffuse reflection term \( f^{r,d} \), on the other hand, approximates multiple scattering inside yarns, and is defined as
\[ f^{r,d}(x, i, o) = \frac{k^{r,d}(x)}{\pi}, \] (4)
with \( k^{r,d}(x) \) the diffuse albedo, \( n_s \) is the surface normal.

Surface Transmission. The specular transmission term \( f^{t,s} \) is defined following the SpongeCake model, as
\[ f^{t,s}(x, i, o) = \frac{k^{t,s}(x)D(h, x)G^t(i, o, x)}{4(i, n_s)(o, n_s)}, \] (5)
where \( G^t \) is the transmission attenuation defined as
\[ G^t(i, o, x) = \frac{1 - e^{-T(A(l(x)))}}{\Lambda(i, x) + \Lambda(o, x)} e^{T\Lambda(o, x)}. \] (6)

The diffuse transmission \( f^{t,d} \) term is derived analogously to its reflection counterpart as
\[ f^{t,d}(i, o, x) = \frac{k^{t,d}(x)}{\pi}. \] (7)

The delta transmission term \( f^\delta \) is defined as:
\[ f^\delta(x, i, o) = \frac{\delta(i + o)}{(i \cdot n_s)}. \] (8)

2 MULTI-SCALE SHADING DETAILS
Here we provide detailed formula for the base and residual term for our specular reflection and diffuse reflection lobes. The rest of the lobes can be derived in a similar fashion. Note that for brevity we omit the directional dependence on each term.

2.1 Specular Reflection
The multi-scaled integration formula for specular reflection:
\[ f^{r,s} = \int_P T^{\text{s\text{heen}}}(i(p)) f^{r,s}(p) V(p)(n_s \cdot i) dp \] (9)
\[ = T^{\text{s\text{heen}}}(G r \bar{I} \cdot \bar{k} \int_P D(p) V(p) dp), \] (10)
with \( \bar{I} = \int_P I(p) dp \) and \( \bar{k} = \int_P k^{r,s}(p) dp \). As stated in the paper, here we assume that albedo \( k^{r,s} \) is smoothly varying and has little correlation with the other terms and can be integrated separately via mip-mapping. Inspired by the observation made by Zhu et al. [2022], we applied the same assumption to the gap indicator function \( I \). This means that our focus is on understanding the statistical likelihood of light passing through the gap, rather than the specific location where it passes through. \( \bar{I} \) is calculated with similar SAT-based approach introduced in Sec. 4.3 of the paper for constant time lookup. The remainder of the term \( \int_P D(p) V(p) dp \) is then solved with our multi-scale shading scheme. While we already introduced
the equations in the paper, for completeness, fitting into the multi-scale scheme via Clustered Control Variates in Eq.8 of the paper:

\[
\int_{V} D(p)V(p)dp = \sum_{c \in 1,...,N_C} \tilde{c}_c F_c + \int_{V} f \cdot (v - \tilde{v}_c) dp \tag{11}
\]

\[
= \sum_{c \in 1,...,N_C} \tilde{c}_c \int_{V} D(p)dp + \int_{V} D(p) \cdot (V(p) - \tilde{v}_c)dp \tag{12}
\]

2.2 Diffuse Reflection
The multi-scaled integration formula for diffuse reflection:

\[
P_r d = \int_{V} T^{\text{sheen}}(p) f_r d (p)V(p) \langle n_\pi \cdot i \rangle dp \tag{13}
\]

\[
= T^{\text{sheen}} \frac{1}{\pi} \langle n_\pi \cdot i \rangle \hat{k} \cdot \hat{I} \cdot \hat{V} \cdot dp. \tag{14}
\]

Following the assumptions we made for the specular reflection, we assume albedo, gap indicator function and the visibility can all be integrated separately. Here \( \hat{V} = \int_{V} V(p)dp \) is again calculated with the SAT-based approach. By leveraging the visibility clusters that we have, we construct an SAT table where each texel indicates which clusters it belongs to. This allows us to query for the histogram and compute a weighted sum of our total 8 clusters to obtain \( \hat{V} \).

3 PSEUDOCODE FOR CORE ALGORITHMS
In this section, we provide the pseudocode for our core algorithm to enhance understanding and facilitate reimplementation.

3.1 Summed Area Table Creation and Query

**Algorithm 1 Create SAT**

**Input:** \( t m \) as tangent map, \( dr \) as directional resolution, \( D \) as the cloth bsdf.

**Output:** summed area table \( sat \)

1: function \( cDmSAT(t m, dr, D) \)
2: \( \text{height} \leftarrow tm.\text{height} \)
3: \( \text{width} \leftarrow tm.\text{width} \)
4: declare \( sat[dr][|dr|][\text{height}][\text{width}] \)
5: for \( i \leftarrow 0 \) to \( \text{dr} - 1 \) do
6:   for \( j \leftarrow 0 \) to \( \text{dr} - 1 \) do
7:     for \( p \leftarrow 0 \) to \( \text{height} - 1 \) do
8:       for \( q \leftarrow 0 \) to \( \text{width} - 1 \) do
9:         \( \text{qd} \leftarrow \text{Vectorize}(i, j) \)
10:        \( \text{pv} \leftarrow D(tm[p][q]) \)
11:       \( \text{left} \leftarrow sat[i][j][p][q - 1] \) if \( q > 0 \) else \( 0 \)
12:       \( \text{above} \leftarrow sat[i][j][p - 1][q] \) if \( p > 0 \) else \( 0 \)
13:       \( \text{aboveLeft} \leftarrow sat[i][j][p - 1][q - 1] \) if \( p > 0 \) and \( q > 0 \) else \( 0 \)
14:       \( \text{sat}[i][j][p][q] \leftarrow \text{pv} + \text{left} + \text{above} - \text{aboveLeft} \)
15:     end for
16:   end for
17: end for
18: return \( sat \)
19: end function

**Algorithm 2 Query SAT**

**Input:** \( sat \) as precomputed summed area tables, \( h \) as half vector, \( x_1, y_1, x_2, y_2 \) as the four corner of the query pixel footprint.

**Output:** radiance integral over the query area \( areaSum \)

1: function \( qAreaSum(sat, h, x_1, y_1, x_2, y_2) \)
2: \( (dt, dp) \leftarrow \text{Spherize}(h) \)
3: \( \text{total} \leftarrow sat[dt][dp][y_2][x_2] \)
4: \( \text{left} \leftarrow sat[dt][dp][y_2][x_1 - 1] \) if \( x_1 > 0 \) else \( 0 \)
5: \( \text{above} \leftarrow sat[dt][dp][y_1 - 1][x_2] \) if \( y_1 > 0 \) else \( 0 \)
6: \( \text{aboveLeft} \leftarrow sat[dt][dp][y_1 - 1][x_1 - 1] \) if \( x_1 > 0 \) and \( y_1 > 0 \) else \( 0 \)
7: \( areaSum \leftarrow \text{total} - \text{left} - \text{above} + \text{aboveLeft} \)
8: return \( areaSum \)
9: end function

3.2 Offset Precomputation

**Algorithm 3 Precompute offset based on position map and viewing angles**

**Input:** \( posMap \) is the 2D map of 3D positions, \( \omega_o \) is the viewing direction, \( x, y \) are the current intersected positions on the surface corresponding to \( \text{pos} \).

**Output:** The offset length to the primary intersected position \( \text{offset} \)

1: function \( \text{OFFSET}(posMap, \omega_o, x, y) \)
2: \( \text{maxH} \leftarrow -\infty \)
3: for each \( \text{pos} \) in \( posMap \) do
4:   if \( \text{posMap}[\text{pos}] > \text{maxH} \) then
5:     \( \text{maxH} \leftarrow \text{posMap}[\text{pos}] \)
6: end if
7: end for
8: \( \theta, \phi \leftarrow \text{Spherize}(\omega_o) \)
9: \( \text{offset} \leftarrow 0 \)
10: for \( t \) from 0 to \( texCount \) do
11:   \( \text{currH} \leftarrow t \cdot \text{cos}(\theta) \)
12:   \( \text{nextH} \leftarrow (t + 1) \cdot \text{cos}(\theta) \)
13:   \( \text{currX} \leftarrow x + t \cdot \text{sin}(\theta) \cdot \text{cos}(\phi) \)
14:   \( \text{currY} \leftarrow y + t \cdot \text{sin}(\theta) \cdot \text{sin}(\phi) \)
15:   \( \text{nextX} \leftarrow x + (t + 1) \cdot \text{sin}(\theta) \cdot \text{cos}(\phi) \)
16:   \( \text{nextY} \leftarrow y + (t + 1) \cdot \text{sin}(\theta) \cdot \text{sin}(\phi) \)
17: if \( \text{posMap}[\text{currX}], \text{currY} < \text{maxH} - \text{currH} \) and \( \text{posMap}[\text{nextX}], \text{nextY}] > \text{maxH} - \text{nextH} \) then
18:   \( \text{offset} \leftarrow t \)
19: break
20: end if
21: end for
22: return \( \text{offset} \)
23: end function
3.3 Clustered Control Variates

**Algorithm 4 Clustered Control Variates**

**Input:** \( \omega_i, \omega_o \), are the incident and outgoing direction, \( A \) is the pixel footprint covering area, \( D \) is the cloth bsdf, \( p \) is the sampling position

**Output:** corrected estimate of radiance over the pixel footprint \( Val \)

1: function CCV(\( \omega_i, \omega_o, A, D, p \))
2: \( C_{ID} \leftarrow \text{FINDCLUSTER}(p) \)
3: \( h \leftarrow (\omega_i + \omega_o)/2 \)
4: \( x_1, y_1, x_2, y_2 \leftarrow \text{QUERY}(A) \)
5: \( BASE \leftarrow \text{QUERYSAT}(sat, h, x_1, y_1, x_2, y_2) \)
6: \( \text{Residual} \leftarrow D(h) \)
7: \( \bar{o} \leftarrow \text{CLUSTERVIS}(C_{ID}) \)
8: \( o \leftarrow \text{Vis}(p) \)
9: \( Val \leftarrow BASE \cdot \bar{o} + \text{Residual} \cdot (o - \bar{o}) \)
10: return \( Val \)
11: end function

**REFERENCES**
