Supplemental Material: A Stationary SVBRDF Material Modeling Method Based on Discrete Microsurface

We provide the pseudo code of $P$-NDF evaluation for re-implementation in alg.1. A challenge part of the evaluation is how to relate the $K$-clustering method to the scale of rendering. To solve this problem, we change the searching size of clustered $K$-lobes when ray footprint changes (alg.2). Searching the lobes around boundaries is simple, we also provide the pseudo code in alg.3.

Algorithm 1 $P$-NDF Evaluation

1: function EVAL$P$-NDF($u, s, P, \sigma_r, BoundaryDealt, NH_I, NH_B$)
2: contribution $\leftarrow$ 0.0
3: aabb $\leftarrow$ BOUNDINGBOX($u, P, s, \sigma_r$)
4: $L \leftarrow$ NULL
5: if $BoundaryDealt =$TRUE then
6: SEARCHBOUNDARYLOBES($NH_B, aabb, L$)
7: end if
8: aabbi $\leftarrow$ m(aabb)
9: maxsize $\leftarrow$ MAXCLUSTERSIZE($P$)
10: SEARCHKLOBES($NH_I, aabbi, L, maxsize$)
11: for $j = 0$ to $L.size()$ do
12: contribution $\leftarrow$ LOBECOMTRIBUTION($L[j], u, s, P, \sigma_r$)
13: end for
14: return contribution
15: end function

Algorithm 2 $K$-lobes Searching

1: function SEARCHKLOBES($node, aabb, L, maxsize$)
2: if INTERSECT(aabb, node.aabb)$=FALSE or node=NULL then
3: return
4: end if
5: if node.isleaf$node =$TRUE then
6: for $i = 0$ to $node.size( )$ do
7: if INTERSECT(aabb, node.lobe[i].aabb) then
8: L.push(node.lobe[i])
9: end if
10: end for
11: else
12: searchchildren $\leftarrow$ TRUE
13: if node.size( )$\leq maxsize$ and node.clustered $=$TRUE then
14: for $i = 0$ to node.clusterednumber( ) do
15: if INTERSECT(aabb, node.klobe[i].aabb) $=$TRUE then
16: L.push(node.klobe[i])
17: searchchildren $\leftarrow$ FALSE
18: end if
19: end for
20: end if
21: if searchchildren $=$ TRUE then
22: SEARCHKLOBES(node.leftchild, aabb, L, maxsize)
23: SEARCHKLOBES(node.rightchild, aabb, L, maxsize)
24: end if
Algorithm 3 Boundary Lobes Searching

1: function SEARCHBOUNDARYLOBES(node, aabb, L)
2: if intersect(aabb, node.aabb) = FALSE or node = NULL then
3:    return
4: end if
5: if node.isleafnode = TRUE then
6:   for i = 0 to node.size() do
7:      if intersect(aabb, node.lobe[i].aabb) then
8:         L.push(node.lobe[i])
9:      end if
10:   end for
11: else
12:   SEARCHKLOBES(node.leftchild, aabb, L)
13:   SEARCHKLOBES(node.rightchild, aabb, L)
14: end if
15: end function

If the texture shows continuity around the boundaries such as leather and brushed metal, we deal with the lobes around the boundaries. The parameter BoundaryDealt is TRUE.

If the texture shows a separately features such as the structured material, the parameter BoundaryDealt is FALSE and NH_B is NULL.