
Algorithm 1 Algorithm to get the normal for a line by intersection with base surfels.

SurfelLineIntersec(Line l , Surfel s_1 , Surfel s_2 , Normal \mathbf{n})

```

 $d := ||\mathbf{P}_{s_1} - \mathbf{P}_{s_2}||;$ 
if ( $d < r_{s_1} + r_{s_2}$ )
    // bounding spheres of  $s_1$  and  $s_2$  intersect
    if ( $s_2 \in \mathcal{S}_{base}$ )
         $\mathbf{P} := \text{IntersectionPoint}(l, s_2);$ 
         $d := ||\mathbf{P} - \mathbf{P}_{s_2}||;$ 
        if ( $d < r_{s_2}$ )
            //  $l$  intersects  $s_2$ 
             $\mathbf{n} := \mathbf{n} + \mathcal{G}_{s_2}(d) \cdot \mathbf{n}_{s_2};$ 
        end if
    else
        for each child surfel  $c$  of  $s_2$  do
            SurfelLineIntersec( $l, s_1, c, \mathbf{n}$ );
        end for
    end if
end if
else
    for each child surfel  $c$  of  $s_2$  do
        SurfelLineIntersec( $l, s_1, c, \mathbf{n}$ );
    end for
end if
end if
end
```

NormalForLine(Line l , Surfel s , Normal \mathbf{n})

```

 $\mathbf{n} := (0, 0, 0)^T;$ 
SurfelLineIntersec( $l, s, rootSurfel, \mathbf{n}$ );
 $\mathbf{n} := \mathbf{n} / ||\mathbf{n}||;$ 
end
```

sured (see d_4 and d_7 in fig. 7(a)). Then the weights $w_{s'}$ are given by:

$$w_{s'} = \mathcal{G}_s(d_{s'}). \quad (14)$$

The final procedure to get the normal $\mathbf{n}_{x,y}$ of a raster point $\mathbf{P}_{x,y}$ that forms together with the surfel normal \mathbf{n} a line is summarized in algorithm 1. Finally the normal $\mathbf{n}_{x,y}$ is coded to RGB values and stored in the normal map.

4.2 Normal Map for Hybrid Hierarchies

If the highest available LOD in the hierarchy is a triangular mesh, then an algorithm similar to that for pure point hierarchies will be used. Since triangles do not overlap in well-formed triangular meshes only the triangle that intersects the line l through raster point $\mathbf{P}_{x,y}$ have to be found, instead of a set of base surfels. If this triangle $t = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ was found, the normal at the intersection point is interpolated using barycentric coordinates $\mathbf{c} = (u, v, w)^T$ at this point:

$$\mathbf{n}_{x,y} = u\mathbf{n}_{\mathbf{A}} + v\mathbf{n}_{\mathbf{B}} + w\mathbf{n}_{\mathbf{C}} \quad (15)$$

The pseudo-code for this procedure is shown in algorithm 2.

4.3 Normal Map Size

To create a normal map for a point hierarchies a raster size (w_s, h_s) has to be selected for every surfel s in addition to the algorithms described before. Since surfel

Algorithm 2 Algorithm to get the normal for a line by intersection with a triangle that is a surfel child node.

PrimitiveIntersec(Line l , Surfel s , Primitive p , Normal \mathbf{n})

```

 $B := \text{BoundingSphere}(p);$ 
 $d := ||\mathbf{P}_s - \mathbf{M}_B||;$ 
if ( $d < r_s + r_B$ )
    // bounding spheres of  $s$  and  $p$  intersect
    if ( $p$  is triangle)
        if ( $l$  intersects  $p$ )
             $\mathbf{P} := \text{IntersectionPoint}(l, p);$ 
             $\mathbf{C} := \text{BarycentricCoordinates}(\mathbf{P}, p);$ 
             $\mathbf{n} := u_{\mathbf{C}}\mathbf{n}_{p_A} + v_{\mathbf{C}}\mathbf{n}_{p_B} + w_{\mathbf{C}}\mathbf{n}_{p_C};$ 
        end if
    else
        for each child surfel  $c$  of  $p$  do
            PrimitiveIntersec( $l, s, c, \mathbf{n}$ );
        end for
    end if
end if
end
```

NormalForLine(Line l , Surfel s , Normal \mathbf{n})

```

PrimitiveIntersec( $l, s, rootSurfel, \mathbf{n}$ );
end
```

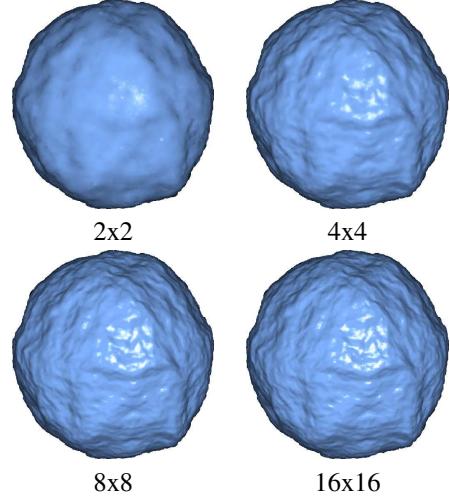


Figure 8: Comparisons of images of a meteoroid model allowing surfels up to a radius of 16 pixel in the interior and using different normal map sizes.

disks are circular it is natural to choose $h_s = w_s$. For base surfels of a pure point hierarchy we only need one pixel ($w_s = 1$), to store the surfel normal itself. For every inner surfel of the point hierarchy the same w_s can be chosen, because surfel size in image space is limited by a quality threshold (see section 5). As it can be seen in fig. 8 this w_s should exceed at least the half of this threshold to preserve features.