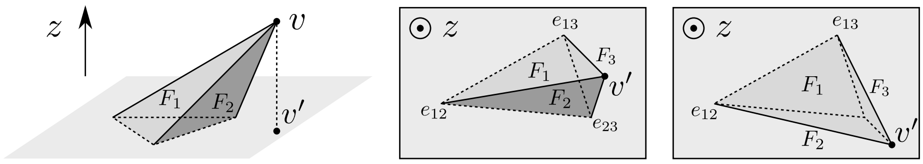

Figure 1: Finding a polyhedron's orientation in the positive z direction.

Splitting a Node

In the example of Figure 8 in the paper, the normal of the $\mathcal{P}_1$ polygons in child 1 are used to compute bit 1 of the mesh position of child 2. We need to find whether the splitting plane is inside or outside a mesh during the splitting pass. In this case it is applied to determine whether the face on the right of child 1 is inside of $\mathcal{P}_1$.

The orientation is given by the normal of one facet, the "extremal" facet. This facet is the one closest and most parallel to the splitting plane. For example, for the mesh in Figure 1, if v is the vertex with highest z coordinate, then the normal of its neighbors gives the mesh orientation. This is tricky because the mesh is possibly open, it may be cut during the bounding box splits. Ray shooting cannot be used because it will be run in parallel, and different threads will not select the same ray.

We want to find the extremal facet in one pass over the facets of the mesh. During this pass, we keep track of the vertex v with highest z seen so far, $z_{\max}$. The polygons this vertex belongs to contribute two half-edges each. We intersect the half-edges with plane $z = z_{\max} - 1$, and project the extremal point v on the plane as v'. Each facet intersects with this plane as a segment and each half-edge as a point. Henceforth, we work in this 2D plane.

To find the extremal facet, we consider the edge most remote from v'. In Figure 1, this is e_{12}, its distance is $d_{\max}$. This leaves two possibilities for the extremal facet: F_1 or F_2. In the first example, both have the same normal orientation, so it does not matter, but in the second, their orientations are different.

To choose between F_1 and F_2, we select the one whose slope is closest to orthogonal with (v', e_{12}). For example, the slope for F_1 is given by

$$s = \frac{| < (v' - e_{12})_\perp, e_{13} - e_{12} > |}{\|v' - e_{12}\|.\|e_{13} - e_{12}\|} \tag{1}$$

For the two half-edges contributed by each vertex in a facet, we compute (z, d, s). The triplets are compared lexicographically, ie. we compare d values only if the z values are the same. The maximum of the triplet gives the extremal facet. The current optimal triplet can be be maintained in one pass over the mesh facets, and triplets computed in different threads can be merged easily. Note that since the triplets for different facets are computed in the same way from the vertex coordinates, they are exactly the same: there are no roundoff errors to take care of when testing for equality.