An Algorithm for Distance Measurement on a Polygonal Mesh

Zsolt Terék

1 Mesh Distance

Given points p_0, p_1, …, p_n on a mesh such that $p_0 = p_n$, polylines l_i between p_i and p_{i+1}, $0 \leq i < n$ and a nonempty set $S \subseteq \{l_i\}$, determine an estimated distance of every vertex of the mesh from one of the closest polylines in S.

The following algorithm approximates the distances of vertices from (the possibly) closest polyline:

Data structures:

- $d_i[v]$ holds the (currenty known, estimated) shortest distance from vertex v to l_i.
- Set *READY* holds the vertices v for which $d_i[v]$ contains the length of the shortest path.
- Set *CURRENT* holds the triangles for which exactly two out of the 3 vertices are in the set *READY*.

Algorithm: (for a fixed i)

1. For every vertex[1] v and $l_i \in S_i$ initialize

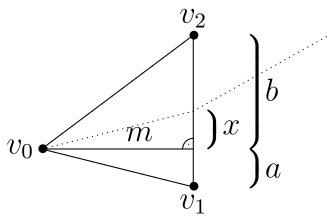
   $$d_i[v] = \begin{cases} 0, & \text{if } v \text{ is on the polyline } l_i. \\ \infty, & \text{otherwise.} \end{cases}$$

   Put vertices of the polylines in S into *READY*. Put the triangles having an edge on any $l_i \in S$ into *CURRENT*.

[1] Suppose the polylines consist of mesh vertices.

2. For every triangle $T \in CURRENT$ estimate the $d_i^T[v_0]$ value for the third vertex of the triangle as follows: (the estimated length of the dotted polyline)

$$d_i^T[v_0] = \min_x \left\{ \sqrt{m^2 + x^2} + \frac{x - a}{b - a} \cdot (d_i[v_2] - d_i[v_1]) + d_i[v_1] \right\}$$



3. For every vertex v that is a third of vertex of a triangle in $CURRENT$ update $d_i[v] = \min_{T \ni v} d_i^T[v]$.

4. Let V be the set of vertices not in $READY$ for which $d_i[v]$ is minimal. Put v into $READY$ for $v \in V$.

5. Remove some triangles from $CURRENT$ and add some others as needed.

6. Goto step 2. if $CURRENT$ is not empty.