Path

Éloi Perdereau

July 18, 2013

1 Behaviour of the local shortest path

The path from a to b through c is optimal (the shortest) on each axis, ie if we move c on any axis in any direction, $|\vec{ac}|$ and $|\vec{bc}|$ can be shorten by moving c in the other direction.
Let set the center of the orthonormal basis on c. Move a on (ac) and b on (bc) in direction of c so that $|\vec{ac}| = |\vec{bc}| = 1$.

1.1 Focus on z

Let P_a be the plane defined by a, c and the orthographic projection of a on the plane xy. Define P_b similarly. By construction, P_a and P_b respectively contains $[ac]$ and $[bc]$. We now unfold P_a and P_b in a single plane P. Due to the optimality of the path on the z axis, a, c and b consist on a straight segment.

Consider - on P - the rectangle with $[ab]$ as diagonal. The z axis intersect this rectangle verticaly in the middle and (ab) on c. Let a' and b' be the orthographic projections of a and b respectively on z - they belong to P and $z_a = z_{a'}$ and $z_b = z_{b'}$. We previously said that $|\vec{ac}| = |\vec{bc}|$ and that a, c and b are aligned. This implies that both triange $aa'c$ and $bb'c$ have the same dimensions, so $|\vec{a'c}| = |\vec{b'c}|$. Furthermore, a', c and b' are aligned in that order, thus

$$z_a = -z_b$$

2 Local shortest path

Given two points a and b and an edge e, all in the cubical complex, we want to compute the shortest path between a and b passing through e.

First, compute the visibility of e from a and b. It consist of two parts of e, let e_v be the intersection of theses partial edges. If e_v doesn't exists, the path from a to b has at least two intermediates vertices. Let p and q be the two end points of e_v. If the line segment $[ab]$ cross e_v, than the shortest path is direct. Else, the path pass through p or q (must be proved?).

```
LOCAL_OPT(a, b, e):
returns the path from a to b through e if it has one or less intermediate vertices

    e_vis := the part of e visible from both a and b
    if e_vis is empty
        then the path from a to b has more than one intermediate vertices
    else
        p, q := endpoints of e_vis
        if [ab] intersect e_vis:
            then return path (a, b)
        else if dist(ap) + dist(pb) < dist(aq) + dist(qb)
            then return path (a, p, b)
        else
            return path (a, q, b)
```