Computational Geometry

Convex hulls in 3D

September 29, 2022

0 Announcement

The homeworks will be graded with 5 points per exercise. Where 5 points mean *perfectly solved*, 4 mean *solved, but there are inaccuracies*, ...

1 Euler's Formula

Let V be the number of vertices (0-faces), E of edges (1-faces) and F of facets (2-faces) (in a polyhedron of genus 0). Then the following assertion holds true:

$$V - E + F = 2.$$

At the example of a simple cube, we see that $8 - 12 + 6 = 2$.

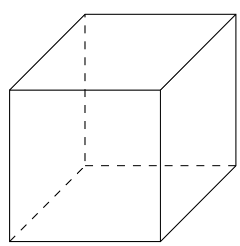

Figure 1: A simple cube with its eight vertices, twelve edges and six faces.

David Eppstein has published 21 proofs for this. Here is one of them:

Proof.

1. Flatten polyhedron to get a planar graph G (By taking a face and stretch it to infinity)
2. Draw the dual graph $G*$ (red) of G. Now are the faces of G the vertices of $G*$ and vice versa, while a edge of G intersects exactly one edge of $G*$.
3. A cycle in G disconnects $G*$ and any acyclic subgraph F in G (a forest) doesn't disconnect $G*$

4. Choose a spanning tree T (blue) of G
5. The dual of its compliment $(G \setminus T)*$ is a spanning tree (green).
6. $G \setminus T$ is a acyclic connected subgraph of G.
7. The two trees have $V - 1$ (blue) and $F - 1$ (green) edges.
8. So in total they have $V + F - 2 = E$ edges.

$\square$

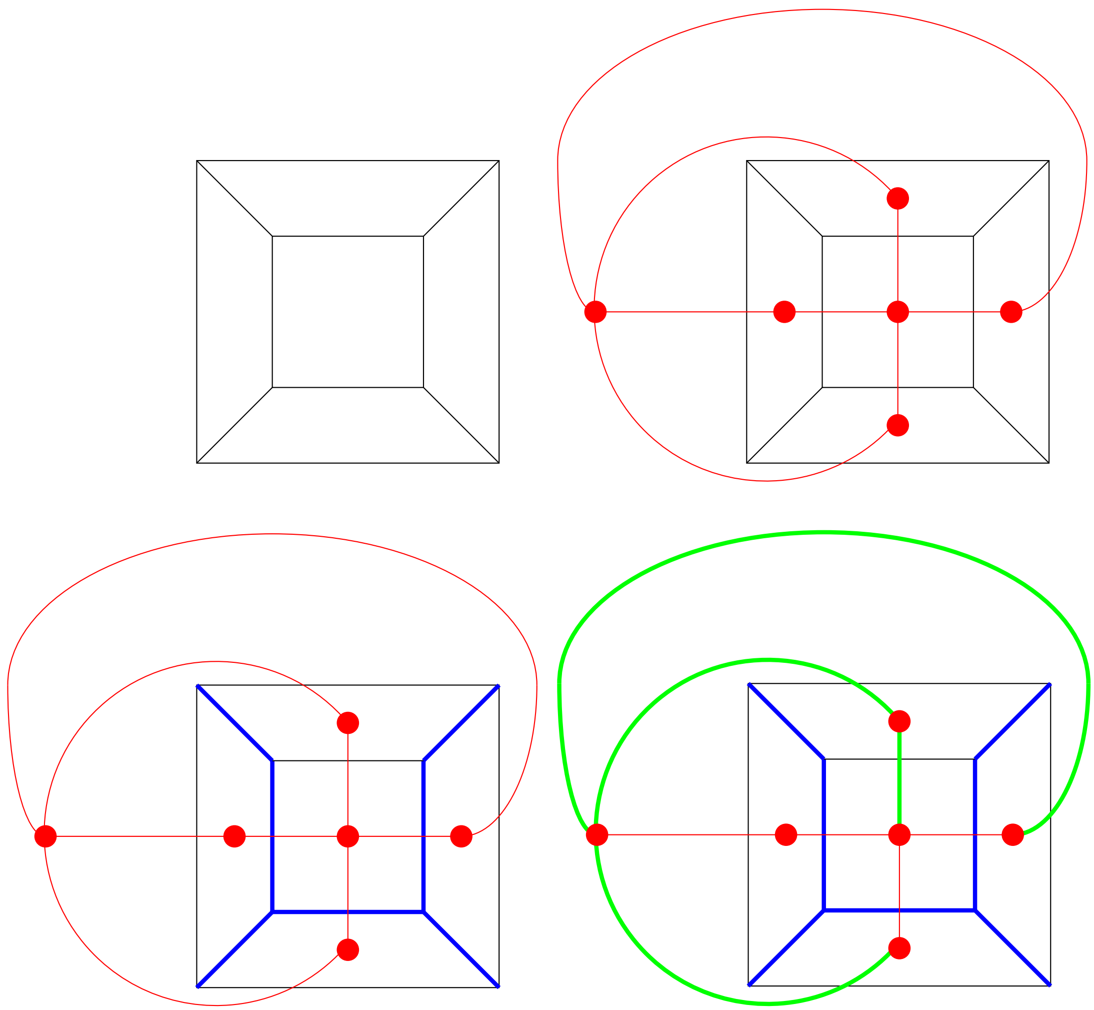

Figure 2: Algorithm from the proof

2 Convex Hulls in 3D

We could use Gift wrapping [Chand + Kapur 1970], but the running time is too bad. We better use Divide and Conquer. The dividing is not difficult, but the conquering is. Assume you have two correct calculated convex hulls. How to put them together, that you get a convex hull for the sumgraph. And how to do that in a short running time.

2.1 Chan's Kinetic Divide and Conquer CH Algorithm

The idea is to represent a 3D (lower) hull as a 2D movie.

- an initial 2d hull
- a sequence of points added to/subtracted from the inital 2D hull over time

Time determines a projection of 3D points to 2D. So that extreme points in 2D are extreme in 3D.

Projection: $P_i(t) = (x_i, z_i - ty_i)$ (in 2D projection points move vertically at different speeds determined by original y-coordinates)

- P_i is on 3D (lower) hull iff P_i lies on same plane $z = sx + ty + b$
- All other points are above iff $P_i(t)$ lies on same line $y' - sx + b$ and all other points lie above for some t.

Algorithm for merging the convex hulls:

- Sort points by x-coordinate.
- Split points into L (left) and R (right) halves.
- To calculate movie:
  - Recursively find `movie(L)` and `movie(R)`
  - Merge movies (frame by frame) to get hull H.
  - For the next time steps, the following events may happen:
    1. L sees an insertion/deletion event $\Rightarrow$ H gets the same event if point is to left of u
    2. R sees an insertion/deletion event $\Rightarrow$ H gets the same event if point is to right of u
    3. $u^- - u - v$ turns ccw ($u - v$ becomes bridge) $\Rightarrow$ H gets delete u event
    4. $u - u^+ - v$ turns ccw ($u^+ - v$ becomes bridge) $\Rightarrow$ H gets insert u^+ event
    5. Same as 3. for R
    6. Same as 4. for R

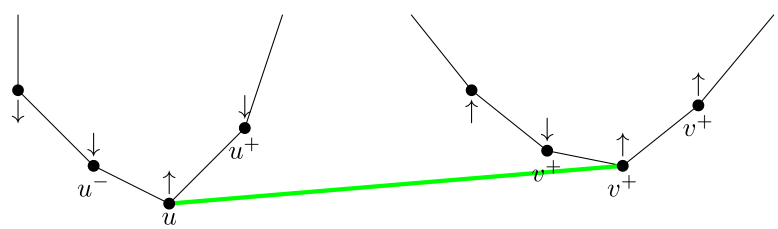

Figure 3: The merging of Chan's Kinetic D&C