3D Convex Hulls (chapter 11)

Elisha Sacks
Convex Hull of 3D Points

- Smallest convex set that contains the points
- Convex polyhedron
- Used in shape approximation and collision detection
- 2D Voronoi diagram and Delaunay triangulation (next class)
Boundary Representation

- A vertex has coordinates and incident edges.
- An edge has a tail, a twin, a next edge, and a facet.
- Edge loops bound facets.
- A facet has one edge per boundary loop.
- A convex polyhedron is a convex set bounded by convex facets such that every edge is incident on one facet.
**Theorem 11.1** A convex polyhedron with $n$ vertices has at most $3n - 6$ edges and at most $2n - 4$ facets.

*Proof* Euler’s formula for a genus zero polyhedron with $e$ edges and $f$ facets is $n - e + f = 2$. Every facet has at least three edges and every edge is incident on two facets, so $2e \geq 3f$.

$n + f - 2 = e$ implies $n + f - 2 \geq 3f/2$ hence $f \leq 2n - 4$.

$e = n + f - 2$ implies $e \leq n + 2n - 4 - 2 = 3n - 6$. 
1. Randomize the points to $p_1, \ldots, p_n$.
2. Initialize the hull to $H = p_1p_2p_3$ in counterclockwise order.
3. For $r = 4$ to $n$:
   - If $p_r$ is outside of $H$
     - Remove the visible edges $a_1a_2, \ldots, a_{k-1}a_k$.
     - Create edges $a_1p_r$ and $p_ra_k$. 
The same idea works in 3D.

A facet is visible if $p_r$ is in its positive half-space.

The visible facets form a surface.

The boundary of this surface is the horizon curve.
Incremental 3D Algorithm

1. Randomize the points to $p_1, \ldots, p_n$.
2. Initialize the hull to $\mathcal{CH}(P_4) = p_1p_2p_3p_4$.
3. For $r = 5$ to $n$:
   If $p_r$ is outside of $\mathcal{CH}(P_{r-1})$
   Remove the visible facets.
   Create facets that link $p_r$ to the horizon edges.

Note: need to list horizon edges counterclockwise around $p_r$. 
Conflict Graph

- Each uninserted point is linked to its visible facets.
- Each facet is linked to its visible uninserted points.
- The graph is initialized with the facets of $\mathcal{CH}(P_4)$ and the uninserted points $p_5, \ldots, p_n$.
- It is updated during point insertion.
- The Delaunay triangulation algorithm uses the same idea.
Conflict Graph Update

1. Remove the $p_r$ node and its edges.
2. For each new horizon edge $e$ with new facet $f$:
   2.1 Let $S$ be the points that conflict with the old $e$ facets $f_1$ and $f_2$.
   2.2 Remove the facet nodes and their edges.
   2.3 Create the $f$ node.
   2.4 Add edges from the $f$ node to its visible points in $S$.

Correctness proof: If a point can see $f$ in $\mathcal{CH}(P_r)$, it can see $e$ in $\mathcal{CH}(P_r)$, so it can see $e$ in $\mathcal{CH}(P_{r-1}) \subset \mathcal{CH}(P_r)$, so it can see a facet incident on $e$ in $\mathcal{CH}(P_{r-1})$. 
Degenerate Cases

**Degeneracy:** A point \( p_r \) is coplanar with a facet \( f \). If \( f \) does not contain \( p_r \), it is visible. The new facet is not a triangle. The new facet has the same conflicts as the old one.

**Degeneracy:** The points \( p_1, p_2, p_3, p_4 \) are coplanar. Prevented by randomization. Or pick four other points.

**Degeneracy:** The points are coplanar. Use a 2D algorithm.

**Degeneracy:** The points are collinear. Return a line segment.
Complexity

- We prove that the expected number of facets created is $O(n)$.
- Let $s$ be the total number of points in the $S$ sets in step 2.1.
- The expected time complexity is $O(n + s)$.
- The book proves that $s$ is $O(n \log n)$.
- The proof is a complicated variant of earlier proofs.
Lemma 11.3 The expected number of facets created is at most $6n - 20$.

Proof
$\mathcal{CH}(P_4)$ has four facets.
The number of facets created by $p_r$ is the number of edges incident on $p_r$ in $\mathcal{CH}(P_r)$.
There are at most $3r - 6$ edges each incident on two vertices.
The expected number of $p_r$ facets is $(6r - 12)/r < 6$.
The sum over the $n$ points is at most $4 + 6(n - 4) = 6n - 20$. 