Tight Cocone

DIEGO SALUME

SEPTEMBER 18, 2013
Summary

• Problem: The Cocone algorithm produces surfaces with holes and artifacts in undersampled areas in the data.

• Solution: Extend the Cocone algorithm to output water-tight surfaces by filling all holes without introducing extra points.

• Key insight: Mark and Peel algorithm introduced to seal reparable holes.
Summary

• Limitations:
  • Algorithm relies on the principle of locality.
  • Some noise handling.

• Guarantees a water-tight surface.

• $O(n^2)$ time and space complexity. Governed by the Voronoi diagram computation.
  • In practice, Delaunay triangulation is most time-consuming step.

• Method can be modified to handle internal voids.
Tight Cocone Algorithm Structure

1. Point set in 3D
2. Cocone Algorithm
3. Boundary
4. Mark/Peel
   - Modified Cocone
   - Surface without Boundary triangles
   - Water-tight surface
Modified Cocone

- Cocone algorithm:
  - Find pole vector $v_p$ (approx. surface normal at point $p$).

$$v_p = p^+ - p$$
Modified Cocone

- **Cocone algorithm:**
  - Find pole vector \( v_p \) (approx. surface normal at point \( p \)).
  - Compute Cocone on Voronoi cell associated with point.

\[
v_p = p^+ - p \quad \quad C_p = \{ y \in V_p : \angle((y - p), v_p) \geq \frac{3\pi}{8} \}\]
Modified Cocone

**Cocone algorithm:**
- Find pole vector $v_p$ (approx. surface normal at point $p$).
- Compute Cocone on Voronoi cell associated with point $p$.
- Consider Voronoi edges that intersect the cocone.
Modified Cocone

- **Cocone algorithm:**
  - Find pole vector $v_p$ (approx. surface normal at point $p$).
  - Compute Cocone on Voronoi cell associated with point.
  - Consider Voronoi edges that intersect the cocone.
  - Triangles made up by these edges are candidate triangles, which lie close to original surface.
Boundary Algorithm

- Boundary algorithm:
  - Normals are unreliable in undersampled areas.
  - Boundary algorithm detects these areas.
  - Ignore the candidate triangles for these points.
Tight Cocone

- Label Delaunay tetrahedral as “in” or “out” based on the initial surface approximation obtained with the Cocone algorithm.
- Peel off “out” tetrahedral.
- Compute boundary of union of “in” tetrahedra.

Requirement
- Principle of locality: undersampling is local. Cocone algorithm computes most of the surface except for reparable holes.
Tight Cocone: Marking

- Surface triangles incident to “good” points form a topological disk called umbrella.
- Marking is done by walking through Delaunay triangulation.
- For every good vertex, we know an “out” tetrahedron.
  - Start with a good point on the convex hull and label an infinite tetrahedron as “out”.
Tight Cocone: Marking

- Surface triangles incident to “good” points form a topological disk called umbrella.

- Marking is done by walking through Delaunay triangulation.

- For every good vertex, we know an “out” tetrahedron.
  - Start with a good point on the convex hull and label an infinite tetrahedron as “out”.
  - Compute its umbrella.
Tight Cocone: Marking

- Surface triangles incident to “good” points form a topological disk called umbrella.

- Marking is done by walking through Delaunay triangulation.

- For every good vertex, we know an “out” tetrahedron.
  - Start with a good point on the convex hull and label an infinite tetrahedron as “out”.
  - Compute its umbrella.
  - Every tetrahedron on the same side is also labeled “out”.
  - Otherwise, it’s labeled “in”.
Tight Cocone: Peeling

- Peel off tetrahedral marked as “out”.
- Walk along surface triangles that form boundary of peeled tetrahedral.
  - Initiate with all convex hull triangles.
  - Every triangle is incident to an “out” tetrahedron.
    - If the other incident tetrahedron is “out”, this is not a surface triangle.
    - If the other incident tetrahedron is “in”, then this is a surface triangle.
Tight Cocone: Peeling

- Peel off tetrahedral marked as “out”.
- Walk along surface triangles that form boundary of peeled tetrahedral.
  - Initiate with all convex hull triangles.
  - Every triangle is incident to an “out” tetrahedron.
    - If the other incident tetrahedron is “out”, this is not a surface triangle.
    - If the other incident tetrahedron is “in”, then this is a surface triangle.
Tight Cocone: Peeling

- Peel off tetrahedral marked as “out”.
- Walk along surface triangles that form boundary of peeled tetrahedral.
  - Initiate with all convex hull triangles.
  - Every triangle is incident to an “out” tetrahedron.
    - If the other incident tetrahedron is “out”, this is not a surface triangle.
    - If the other incident tetrahedron is “in”, then this is a surface triangle.
Tight Cocone: Peeling

- Peel off tetrahedral marked as “out”.
- Walk along surface triangles that form boundary of peeled tetrahedral.
  - Initiate with all convex hull triangles.
  - Every triangle is incident to an “out” tetrahedron.
    - If the other incident tetrahedron is “out”, this is not a surface triangle.
    - If the other incident tetrahedron is “in”, then this is a surface triangle.
Poor Tetrahedra

- Tetrahedra with all poor points are poor tetrahedra.
- Small poor tetrahedra are marked as “in”.
- Big poor tetrahedra could be “in” or “out”. Don’t label them.
  - By principle of locality, we have a small triangle (least circumradius).
  - Inner poor tetrahedra must be reached through small triangle.
- During peeling, we are only allowed to move into a poor tetrahedron if it’s not through the small triangle to avoid peeling inner tetrahedral.
  - If it’s an “out” tetrahedron, it’ll be reached through one of the big triangles.
  - If it’s an “in” tetrahedron, it will never be reached and won’t be peeled.