Teddy: A Sketching Interface for 3D Freeform Design

SIGGRAPH 99 paper, by Takeo Igrashi, Satoshi Matsuoka, and Hidehiko Tanaka
Motivation

- Easy to use and intuitive toolkits that animators can create a model quickly.
- E.g., SKETCH by Robert Zeleznik et al. & this paper
Goals

• Construction of approximate polygonal surface models
• Users can draw gestural strokes to construct objects
• Check out their video online:
  http://www-ui.is.s.u-tokyo.ac.jp/~takeo/video/teddy.avi
Overview: Creating A New Object

- User draws silhouette – cannot self intersect
- Thick parts of silhouette are fat and narrow areas are skinny
Algorithm for Creating a New Object

• The Silhouette
  – User drawn silhouette converted to line segments
  – If open, end points are joined
  – Silhouette cannot itself intersect

- Inflate the polygon
  – Construct constrained Delaunay Triangulation
  – Find chordial Axis
  – Prune insignificant branches
    • Merge triangles
    • Compute pruned spine
    • Retriangulate
Algorithm for Creating a New Object

- Inflate the polygon (details next slide)
- Find chordial Axis
- Prune insignificant branches
  - Merge triangles
  - Compute pruned spine
  - Re-triangulate

a) initial 2D polygon  
b) result of CDT  
c) chordal axis

d) fan triangles  
e) resulting spine  
f) final triangulation
Algorithm for Creating a New Object

- Inflate the polygon (details next slide)
  Construct constrained Delaunay Triangulation
  - Delaunay triangulation: A triangulation such that the circumcircle of every triangle contains no other points
  - Constrained delaunay triangulation: a delaunay triangulation forced to contain edges
    - here the edges of the input silhouette

![Diagram showing Terminal, Sleeve, and Junction with external edges labeled and CDT result.](http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=477)
Algorithm for Creating a New Object

• Finding the Chordial Axis (Spine)
  – If open, end points are joined
  – Silhouette cannot itself intersect
  – Find the chordial axis by connecting the midpoints of the internal edges
Algorithm for Creating a New Object

- Prune insignificant branches
  - Merge triangles
  - Compute pruned spine
  - Re-triangulate

![c) chordal axis](image)
![e) resulting spine](image)
Algorithm for Creating a New Object

- Prune insignificant branches
  - **Merge triangles**
    
    For each terminal triangle \( X \),
    
    1. \( C \): The semicircle on \( X \)’s interior edge
        
        \( T \): the triangle sharing \( X \)’s internal edge
    
    2. If all vertices of \( X \) are within \( C \), merge \( X \) and \( T \):
        
        \( X = X + T \)
    
    3. Else if \( X \) contains vertices not in \( C \), make a fan of triangles from interior edge midpoint. STOP.
    
    4. If \( T \) is a Junction triangle, make a fan of triangles from midpoint of \( T \). STOP.
    
    5. Goto step 1.

- Compute pruned spine
- Re-triangulate

---

**CMSC 635**  January 15, 2013  Quadric Error Metrics <#>/20

---

![Diagram](attachment:diagram.png)

a) start from T-triangle  b) advance  c) stop  d) fan triangles  e) advance to J-triangle
Algorithm for Creating a New Object

- Prune insignificant branches
  - Merge triangles
  - Compute pruned spine
    - Pruned spine is obtained by connected midpoints of sleeve and junction triangles’ internal edges
  - Re-triangulate
    - Divide remaining sleeve triangles at spine & re-triangulate resulting polygons

![Diagram showing steps d) fan triangles, e) resulting spine, f) final triangulation]
Algorithm for Creating a New Object

- Elevating the Spine
  - Elevate each spine vertex by the average distance between it and its connected external vertices
  - Convert all internal edges to quarter ovals
  - Sew neighboring elevated edges
Teddy: Painting on the Surface

- Convert input stroke to line segments
- For each line segment
  - Compute bounded plane containing segment and camera
  - Intersect plane with each polygon of surface (use closest)
  - Connect line segments on surface
- If line segments cannot be connected (i.e., painting across a fold), painting fails.

Figure 16: Construction of surface lines.
Interaction: Extrusion

- Base ring: closed polyline on mesh surface
- Normal: Best fit plane to the base ring
- Projective plane: plane through base ring center of gravity and parallel to the normal
- Project 2D extruding stroke onto plane
- Sweep base ring along extruding stroke such that
  - Base rings are almost perpendicular to the direction of the extruding stroke
  - Base rings are resized to fit extruding stroke
- Delete polygons underlying the base of the extrusion and sew extrusion to surface
- Same algorithm used for digging cavities

![Diagram](a) projection of the stroke   b) sweep along the projected stroke

Figure 17: Extrusion algorithm.
Interaction: Cutting

- Based on painting algorithm
- For each line segment of cutting stroke
  - Project onto front and back facing polygons intersected by bounded plane
  - End points of projected segment are connected to create a planar polygon
- Splice planar polygons together
- Triangulate planar polygons
- Remove all polygons on surface to left of cutting stroke

Figure 20: Cutting.
Smoothing

- Change coordinate system so that the normal of base ring is parallel to the Z-axis.
- Project base ring into XY-plane and triangulate it.
- Determine Z-values for vertices of triangulated base ring.
  - For each vertex
    - For each edge opposite the vertex
      1. Consider plane parallel to the Z-axis through the vertex and the mid-point of the edge
      2. Choose Z-value so that the point lies on the Bezier curve that smoothly interpolates both ends of the ring on the plane
  - The final Z-value is the average of the Z-values across all edges.

![Smoothing Algorithm Diagram](image.png)

Figure 21: Smoothing algorithm.