Interactive Multiresolution Mesh Editing

Denis Zorin
Caltech

Peter Schröder
Caltech

Wim Sweldens
Bell Labs
Multiresolution Editing

• The desired output of mesh editing is a high resolution mesh
• Not all changes can be made easily at such a high resolution
• Considering making these changes to a car:
  – Change the shape of the front fender
  – Change the shape of the door handle
  – Which resolution works best?
Surface Representations

• Patches / Hierarchical B-Splines (H-Splines)
  – Work is needed to keep continuity at seams
  – Many patches/control points needed for small details
  – Good at representing large, smooth areas
  – Small internal representation

• Polygonal Meshes
  – Good at representing small details
  – Bad at representing large, smooth areas,
  – Large internal representation
Subdivision Surfaces

- Represents a smooth surface as the iterative refinement of a polygonal mesh
- The control mesh contains large polygons to represent large, smooth areas
- The control mesh can contain small polygons to represent small, intricate details
Contribution

- Design highly adaptive and dynamic data structures and algorithms, allowing the system to function interactively on a wide range of platforms
Loop Subdivision Surfaces

Algorithms – Analysis

• Analysis propagates changes from finer levels to coarser levels

\[
\text{Analysis}(i) \\
\forall v \in V^{i-1} : v.s[i - 1] := \text{smooth}(v, i) \\
\forall v \in V^i : v.d[i] := v.F(i)^t \ast (v.s[i] - \text{subd}(v, i - 1))
\]

The vertex’s location at the next coarser level

The vertex’s detail vector

The point’s local coordinate system

The location you’d get by subdividing the coarser mesh
Algorithms – Synthesis

• Synthesis uses the detail vector to compute the location of a vertex at a higher resolution

\[
\forall v \in V^i : s.v[i] := v.F(i) \ast v.d[i] + \text{subd}(v, i - 1)
\]

- The vertex’s detail vector
- The point’s local coordinate system
- The location you’d get by subdividing the coarser mesh
Algorithms – Analysis Example

The edits made at level 3 have an effect on the mesh at level 2.
Why Adaptive Algorithms?

- The storage cost grows exponentially with the levels of subdivision!
Algorithms – Adaptive Synthesis

- Refine triangles and recompute points until a local flatness threshold is reached
- Parent-level triangles may need to be temporarily refined:
Algorithms – Local Synthesis/Analysis

• Utilize the locality of an edit to propagate changes only where they are necessary
• Applies to synthesis and analysis

\[ V_{i-1}^* = \bigcup_{v \in V_i^*} R_{i-1}^{-1}(v) \]

The affected vertices at level \( i-1 \)

The vertices affected by moving \( v \)

The set of vertices moved at level \( i \)
Algorithms – Adaptive Rendering

• Similar techniques can be applied to adapt rendering performance to a desired framerate
• This also adapts to the platform’s hardware and resources
• The triangles that get drawn are adaptively selected to capture as much detail as possible
Algorithms – Adaptive Rendering
Algorithms – Adaptive Rendering
Results

• Two hardware platforms:
  – Indigo R10000 @ 175MHz with Solid Impact graphics
  – PentiumPro @ 220Mhz with Intergraph Intense 3D board

• 172,000 triangle model after 6 levels of subdivision

• Target performance was 5 frames per second
  – SGI rendered 113,000 triangles
  – PC rendered 35,000 triangles