Meshes, Simplification, Refinement and Multiresolution

TU Berlin
2008

Dr Tamy Boubekeur
Meshes in CG

- CAD – Simulation
- Animation
- Medical imaging
- Cultural heritage
- Entertainment
Finite Element Methods (FEM) for physics simulation
Carefully designed control meshes for subdivision surfaces
Medical imaging

For visualization in medical imaging
Cultural heritage

Automatic generation by 3D scanning
Entertainment

Meshes:
- native language for
  
  *Graphics Processors Units*
- standard representation in
  
  video games
Meshes

• Surfaces meshes:
  – Discrete surfaces
    • arbitrary genus
    • borders
  – Piecewise linear approximations of a continuous surface
  – Sampling ratio: nb triangles, nb vertices
  – In practice: not always manifold
Polygonal meshes

- Polygons
  - Indexed face set
  - Polygon soup

- Arbitrary topology
- Hardware support
  - DirectX
  - OpenGL
- Wasteful storage
- Not smooth
Polygonal meshes

- Store vertices
  - \(v_1\) (x, y, z)
  - \(v_2\) (x, y, z)
  - \(v_3\) (x, y, z)
  - \(v_4\) (x, y, z)

- Store connectivity
  - \((v_1, v_2, v_3)\)
  - \((v_1, v_3, v_4)\)
Multiresolution and LODs

• Using meshes:
  – Analysis, rendering, interaction
  – Mesh resolution - tradeoff between:
    • Amount of power available
    • Desired precision of the surface approximation
Surface modeling

• Creating shapes
• Intuitive tools
• Computer as a brush
  – Designing base forms
  – Deformations
  – Smoothing, creasing, bending, twisting, extruding, lofting, beveling, splitting, merging, … and so on!
Meshes from Implicits

Contouring algorithms (e.g. Marcing Cubes)
Meshes from Implicits

Marching cubes
Meshes from point sets

• Context: **Automatic Modeling**

• Point sets: 3D scanning devices output

• Reconstruction: generate a mesh out of the point set:
  – Triangulation (e.g. Delaunay)
  – Via implicit (with marching cubes)
  – Direct (deformable models)
Problems

• Given a mesh M, how to generate (semi)automatically
  – A set of lower resolution level of details?
    • Simplification
  
  – A set of higher resolution level of details?
    • Refinement
  
  – A structure encoding all these LODs in a single representation
    • Multiresolution Meshes
Reducing mesh resolution

SIMPLIFICATION AND CLUSTERING
Context

360,000,000 triangles

500,000,000 triangles

TOO MUCH for most applications
Goal

Providing automatically a mesh consisting in less polygons, while preserving as much as possible the original shape of the object.

• Input :
  – A mesh

• Output :
  – A mesh

Note : similar techniques exists for other surface representations (e.g. point sets)
Properties

• **Efficiency**:
  – **Time complexity**: simplification speed
    • On-the-fly simplification
  – **Memory complexity**: input size
    • *Out-of-core algorithms*

• **Quality**:
  – Topology preservation
  – Geometry approximation
Classification

Surface Simplification

Cluster-based
1. Partition the input mesh
2. Replace each partition by a single representative element

Element-based
1. Compute the geometric importance of each element
2. Remove (decimation) or merge (contraction) the less important

Element = vertex or polygon
Few well-known simplification algorithms:

- **QEF (Quadric Error Function)**: fit quadrics to a vertex or a cluster [Garland 1997]
- **VSA (Variational Shape Approximation)**: normal-based Lloyd relaxation [Cohen-Steiner et al. 2004]
- **Mesh optimization**: Basis of Progressive Meshes (PM) [Hoppe et al., 1993, 1996]
Classification

Surface Simplification

Cluster-based

OOCS
VSA

Element-based

QEF
PM
Error Metrics

• Cost of the simplification.
• In terms of:
  – Geometry
  – Appearance
  – Perception
• Examples
  – L2 error: average distance of a set of samples from its Least-Square plane.
  – QEF: average distance of a set of samples from its Least-Square quadric.
  – L2.1 : normal-based error when elements have normal information
  – L∞ : distance between simplified and original mesh
  – Saliency: multiscale curvature estimation, proved to be perceptually critical
• Quadric Error Function: average (L2) distance of a set of samples from its Least-Square quadric.

• Allows to better locate the representative

• Computed over
  – A cluster for cluster-based methods
  – A topological neighborhood (e.g. 1-ring neighbor) for element-based methods

*Linear L2 approximation (plane)*

*QEF approximation (quadric)*
OOCS

Idea : design an output-sensitive (input-insensitive) simplification methods for very large models

• Cluster the mesh in a grid by streaming each element independently

• For each cluster/grid cell, keep only the average quadric

• For each triangles, 3 cases :
  – 3 vertices in the same cluster > collapse to a point > ignore it
  – 2 vertices in the same cluster > collapse to an edge > ignore it
  – 3 vertices in three different clusters > reindex and output it
OOCS

Mesh

Gird clustering

Representative elements, from accumulated QEF in cells

Non-collapsing triangles

« Reindex » over representative elements

Simplified mesh
Variational Shape Approximation

2 steps:

1. Cluster the mesh with a normal-driven Lloyd relaxation procedure
2. Extract the border of the cluster as output polygons for the simplified model

High-quality output
Anisotropy and surface structure better preserved

Slow, moderate input only
Simplification methods can be combined at different scales:
Increasing mesh resolution

TESELLATION, SUBDIVISION AND REFINEMENT
General Mesh Refinement

- Increasing both topological and geometrical information
  - Topology: **Tessellation**
  - Geometry: **Displacement**

  \[\text{Refinement} = \text{Tessellation} + \text{Displacement}\]

- Input Domain as a **coarse mesh**
  - Better abstraction for animation, physics, analysis, editing, ...

- **Tessellation**
  - Regular or adaptive, dual or primal

- **Displacement** function
  - An efficient description of « smaller » geometric features
  - Displacement maps, procedural functions, « smooth » functions (subdivision surfaces)...

- **Synthesizing the Hi-res shape:**
  1. **Tessellate the polygons**
  2. **Displace the vertices**
Tessellation

Inserting new vertices in the mesh structure
Tilling to mesh domain

Uniform

Adaptive

Primal

Dual
Displacement

- Moving each vertices
  - **Scalar displacement**: along the normal direction
  - **Vector displacement**: arbitrary offset in 3D

Procedural Displacement

Height field for terrains

Scalar Displacement Mapping
Subdivision Surfaces

- A special case of refinement
  - Various schemes
  - Displacement function: a smoothing operator
    - Often derived from spline functions
    - Converges to a limit smooth surface (C1 at least)

2 passes of the Loop subdivision scheme
Subdivision Surfaces

• Quick overview

See other slides....
Multiresolution Modeling with Subdivision Surfaces

Control at coarse level

Deformation propagation to finer levels
Subdivision Surface - Box Modeling
Subdivision Surface – Refinement and fitting

1. Starting from a dense mesh
2. Simplify it (cf previous section)
3. Subdividing the simple mesh “towards” the dense one
   - Explicit mapping encoded in a tree or
   - Displacement mapping
     • 2D (requires parameterization)
     • 3D (volumetric)
Displaced Subdivision Surfaces

- Refinement combining subdivision (smoothness) and displacement maps (details)
- The most successful refinement technique
- Most of today’s soft shapes in SFX, animation, design, …
- Success of Zbrush (Pixologic)
Encoding the different levels of details

MULTIRESOLUTION MESHES
Multiresolution
Multiresolution modeling
Multiresolution modeling

User → Control → Simple Object → LOD 1 → LOD 2 → LOD N-1 → Complex Object

Refinement
Multiresolution modeling
Multiresolution modeling

Real World

Acquisition

Complex Object

Simple Object

LOD 1

LOD 2

LOD N-1

Simplification
Multiresolution modeling

User → Control → Simple Object → LOD 1 → LOD 2 → LOD N-1 → Complex Object → Real World → Acquisition

Refinement

Simplification
Why multiresolution

- Visualization
- Modeling/Editing
- Physically-based modeling
- Progressive transmission
- Compression
Multiresolution modeling

- Control
- User
- Real World
- Acquisition
- Simple Object
- LOD 1
- LOD 2
- LOD N-1
- Complex Object
- Refinement
- Simplification
- Multiresolution model
- Visualization
- Physics
- Transmission
- Compression
Hierarchical representation

- Wavelets

- Analysis of arbitrary meshes
Progressive Meshes

- Order the vertices according to their importance
- Per-vertex resolution selection
- Two operator to switch from a level of detail to another
- Adaptive and \textit{Geomorph}
Progressive Meshes

1. Sort the vertices according to their importance.
   – Simplification error metrics
2. « **Collapse an edge** » incident to the less important vertex
   (remove 2 triangles)
   – Dual operation: **split a vertex**
3. Restart 1

Note: topology preservation can be checked at each step

---

**Multiresolution encoding in a tree structure**

- **Leaves:** vertices
- **Internal nodes:** collapse/split relationship
Example

Image Progressive Mesh (Hoppe 1996)
Adaptive Refinement

- Use more vertices in critical regions, less in others
- Define an *importance metric*, computed for each vertex
- View-dependent in many cases
- Few examples:
  - Viewpoint distance
  - Curvature
  - Silhoutteness
Refinement metric

Distance
Far Near

Curvature
Min Max

Silhouette
Conclusion

• Efficient tools to set the resolution of a mesh and define its levels of details:
  • Lower resolution > simplification
  • Higher resolution > refinement
    • procedural (subdivision)
    • user-driven (displacement painting)
    • data-driven (3D scanners)

• Elegant encoding of the so defined LODs in a multiresolution structure