Meshes in computer graphics

Mesh $M$

- **$V$**
- **$F$**

Vertex 1: $x_1, y_1, z_1$
Vertex 2: $x_2, y_2, z_2$
...

Face 1: 2, 3
Face 2: 3, 4
Face 3: 2, 7
...

(appearance attributes: normals, colors, textures, ...)

Mesh $M$ consists of vertices and faces. Each vertex has coordinates $(x, y, z)$, and each face connects three vertices. Appearance attributes include normals, colors, and textures.
Complex meshes

43,000 faces

Challenges:
- rendering
- storage
- transmission

lots of faces!
Talk outline

SIGGRAPH ’96
- Progressive mesh (PM) representation
  - continuous resolution
  - efficient
  - progressive transmission

SIGGRAPH ’97
- View-dependent refinement of PM’s
- Progressive simplicial complex (PSC) repr.
Mesh simplification techniques

[Schroeder-etal92]

[Hoppe-etal93]

[Rossignac-Borrel93]

[Cohen-etal96]

[Garland-Heckbert97]

...
Traditional level-of-detail (LOD)

Concern: transitions may “pop” → would like smooth LOD
Progressive mesh representation

Basic idea:

- Simplify mesh through sequence of transformations.

- Record:

  simplified mesh

  + sequence of inverse transformations
Edge collapse $\rightarrow$ Simplification

ecol \left( V_s, V_t, V'_s \right)

$V_t$ $V_r$

$V'_s$

(optimization)

$M = \frac{M_0}{n}$

$\hat{e}_0$

$\hat{e}_1$

$\hat{e}_{n-1}$
Invertible! Vertex split transformation

\[ vspl(v_s, v_l, v_r, v'_s, v'_t, \ldots) \]
Reconstruction process

\[ M^0 \rightarrow M^1 \rightarrow \ldots \rightarrow M^{175} \rightarrow \ldots \rightarrow M^n = \hat{M} \]

progressive mesh (PM) representation
Application: Continuous-resolution LOD

From PM, extract $M^i$ of any desired complexity.

$M^0 \rightarrow vspl_0 \rightarrow vspl_1 \rightarrow vspl_{i-1} \rightarrow vspl_{i-n}$

$M^0$ vspl$_0$ vspl$_1$ vspl$_{i-1}$ vspl$_{i-n}$  $M^n = \hat{M}$

260K faces/sec!  180K faces/sec!

(200 MHz Pentium Pro)
VIDEO: PM construction and LOD

gameguy ecol's

gameguy vspl's

gameguy LOD

sandal LOD
Property: Vertex correspondence

- $M^f$: Vertices $V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8$
- $M^{f-1}$: Vertices $V_1, V_2, V_3, V_4, V_5, V_6, V_7$
- $M^{f-2}$: Vertices $V_1, V_2, V_3, V_4, V_5, V_6$
- $M^c$: Vertices $V_1, V_2, V_3$

Arrow labels: eco
Application: Smooth transitions

\[ M^f \leftarrow M^0 \] \[ M^c \rightarrow I \]

\[ \text{can form a smooth visual transition: geomorph} \]
VIDEO: PM geomorphs

gameguy 1 geomorph
gameguy 12 geomorphs
gameguy 6 rotating
Application: Mesh compression

Record:
- $v_s$ (log$_2 i$ bits)
- $v_l$ & $v_r$ (~5 bits)
- $v'_t$ - $v_s$ (delta)
- $v'_s$ - $v_s$ (delta)
- predict materials

Analysis:
- connectivity: $n(4 + \log_2 n)$ bits vs. $n(6\log_2 n)$ bits
- geometry: ~30$n$ bits vs. 96$n$ bits

[Deering95]
Application: Progressive transmission

Transmit records progressively:

Receiver displays:

\( M^0 \rightarrow vsp_{0} \rightarrow vsp_{1} \rightarrow \cdots \rightarrow vsp_{i-1} \rightarrow vsp_{i} \)

\( M^i \rightarrow M \)

(~ progressive GIF & JPEG)
Conversion

traditional mesh representation  progressive mesh representation

Optimization process

- Various metrics (speed vs. quality)
- Typically performed off-line
How to select edge collapses?

- Preserve *appearance*:
  - geometric shape
  - scalar fields (e.g. color, normals)
  - discontinuity curves

\[
E = \sum_{\text{points}} (e_{\text{shape}} + e_{\text{scalars}}) dA + \sum_{\text{points}} (e_{\text{disc}}) dL
\]
Error metric: point sampling

shape discontinuities

1600 faces

300 faces
Selecting edge collapses

- Greedy algorithm: always collapse edge resulting in smallest $\Delta E$.
- Optimize position and attributes of resulting vertex.

Simplification rates: ~ 30 faces/sec

- off-line process
- could use faster, simpler metrics e.g. [Garland-Heckbert97]
VIDEO/DEMO: PM results

radiosity 6 geomorphs
mandrill image
PM Summary

- continuous-resolution
- smooth LOD
- space-efficient
- progressive
- single resolution

[Microsoft DirectX 5.0]
View-Dependent Refinement
of Progressive Meshes
[SIGGRAPH 97]
Adaptive refinement: motivation
Applications & Related work

- height fields
- parametric surfaces
- arbitrary meshes

[Cignoni et al. 95] [De Floriani et al. 96] [Lindstrom et al. 96] [Rockwood et al. 89] [Abi Ezzi et al. 93] [Kumar et al. 95] [Xia-Varshney 96]...
Using progressive meshes

(e.g. view frustum)
Contributions

- PM → vertex hierarchy → selective refinement
- Dependencies → consistent framework
- View-dependent refinement criteria
Parent-child vertex relations

\[ v_{s} \rightarrow v_{t} \rightarrow v_{u} \]

vsplit
Vertex hierarchy

PM:

$M^0$

vspl_0
vspl_1
vspl_2
vspl_3
vspl_4
vspl_5

v1

v2

v3

v4

v5

v6

v7

v8

v9

v10

v11

v12

v13

v14

v15

[PM: $M^0$]

$[Xia \& Varshney 96]$
Selective refinement

$M^0 \rightarrow vsp{l_0} \rightarrow vsp{l_1} \rightarrow vsp{l_2} \rightarrow vsp{l_3} \rightarrow vsp{l_4} \rightarrow vsp{l_5}$

selectively refined mesh

Restrictions?
New vspl/ecol parametrizations

\[ f_{n0} \quad f_{n1} \quad f_{n2} \quad f_{n3} \]

\[ V_s \quad V_t \quad V_u \quad V_r \]

\[ \text{vsplit} \quad \text{vspl} \quad \text{ecol} \]
• Algorithm:
  - incremental (frame coherence)
  - efficient (~15% of frame time)
  - amortizable
View-dependent refinement criteria

3 criteria:

- view frustum
- surface orientation
- screen-space geometric error
View frustum

- Too high: view is unchanged
- Too far right: view is unchanged
Surface orientation

view is unchanged

oriented away
Screen-space geometric error

tolerance = 0.5 pixels

refinement near silhouette

coarser in distance
All three criteria together

69,473 faces $\rightarrow$ 10,528 faces
1.9 frame/sec $\rightarrow$ 6.7 frame/sec
VIDEO: Selective Refinement

- view frustum (2-window terrain)
- screen-space error (2-window terrain)
- silhouette refinement (sphere)
- all 3 criteria including orientation (teapot)
- arbitrary mesh (gameguy)
- geomorph
- flythrough of gcanyon
Selective Refinement Summary

- View-dependent refinement
- Real-time algorithm
- Continuous-resolution
- Smooth LOD
- Space-efficient
- Progressive
Progressive Simplicial Complexes

[SIGGRAPH 97]

(Joint work with Jovan Popovic)
PM restrictions:

- Supports only “meshes” (orientable, 2-dimensional manifolds)
- Preserves topological type

<table>
<thead>
<tr>
<th>M^0</th>
<th>...</th>
<th>M^i</th>
<th>...</th>
<th>M^n</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,522</td>
<td>8,000</td>
<td>167,744</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Progressive Simplicial Complexes

- Represent arbitrary “triangulations”:
  - any dimension,
  - non-orientable,
  - non-manifold,
  - non-regular, …

- Progressively encode both geometry and topology.
Generalization

$PM$
- edge collapse ($ecol$)
- vertex split ($vsplit$)

$PSC$
- vertex unification ($vunify$)
- generalized vertex split ($gvspl$)
LOD sequence

$M^1 \rightarrow M^{22} \rightarrow M^{116} \rightarrow M^n = \hat{M}$

$gvspl_1 \ldots \ldots \ gvspl_i \ldots \ldots \ gvspl_{n-1}$

PSC representation
PSC Summary

\[ \hat{M} \]

arbitrary simplicial complex

\[ M^1 \rightarrow^\text{gvspl} \]

PSC

lossless

any triangulation

\[ V \]

single vertex

\[ K \]

progressive geometry and topology

\[ M \]

arbitrary simplicial complex
Future Work

- LOD on volumes
- Memory management for large models
- Refinement criteria for surface shading
- Animated models
- Editing using PM/PSC