Fast and memory efficient view-dependent trimmed NURBS rendering

Michael Guthe, Jan Meseth, Reinhard Klein

Bonn University
Computer Graphics Group

Trimmed NURBS

Screenshot from CATIA from Dassault Systemes

Models generated with Discreet's 3D Studio Max
Trimmed NURBS Models

- Large number of NURBS patches
- Non-manifold
- No connectivity information

VW Golf:
- 8036 trimmed NURBS patches
- 3.6 million triangles for visualization with required, typical accuracy (0.2 mm)

Model courtesy of Volkswagen AG

Previous Work

Early approaches:
- Rockwood et al., Real-time rendering of trimmed surfaces, SIGGRAPH 1989
- Kumar et al., Interactive Display of large scale NURBS models, I3D 1995

- Individual trimmed NURBS patches
- Cracks for collections of trimmed NURBS patches
- Triangle count
Previous Work

Kumar et. al
Accelerated Walkthrough of Large Spline Models, I3D 1997

- First to create superpatches
- Good results on multiprocessor machine
- Require online sewing of superpatches

Previous Work

Baxter et. al
Giga Walk: Interactive Walkthrough of Complex Environments, EGRW 2002

- Triangulate model
- Partition/Cluster triangles
- Generate HLODs
- High frame rates on multiprocessor system
- High memory requirements
- Popping artefacts
Algorithm Features

- High quality rendering
- Very complex, non-manifold models
- Close to real-time
- Memory efficient (<50 Bytes/Vertex)
- Maintains surface features
- Simplification down to a single point
- Avoid cracks and popping artefacts

Algorithm Overview

Input: soup of trimmed NURBS patches

Preprocessing:
- Convert trimming curves into polylines
- Sew polylines in 3D
- Generate Seam Graph
- Generate LOD on Seam Graph

Interactive Rendering:
- Select view-dependent LOD from Seam Graph
- Culling of invisible NURBS patches
- On-the-fly tessellation of modified patches
Seam Graph

- Non-manifold models
- Stores individual surface parts and their connectivity
- Patchwise, efficient, standard operations
- Consistent changes

Algorithm Overview

Input: soup of trimmed NURBS patches

Preprocessing:
- Convert trimming curves into polylines
- Sew polylines in 3D
- Generate Seam Graph
- Generate LOD on Seam Graph

Interactive Rendering:
- Select view-dependent LOD from Seam Graph
- Culling of invisible NURBS patches
- On-the-fly tesselation of modified patches
LOD on Seam Graph

- Apply standard LOD techniques
- Whole patches are removed!

Model courtesy of Volkswagen AG
LOD on Seam Graph

Model courtesy of Volkswagen AG
LOD on Seam Graph

Model courtesy of Volkswagen AG

University of Bonn • Computer Graphics Group
Michael Guthe, Jan Meseth, Reinhard Klein
Algorithm Overview

Input: soup of trimmed NURBS patches

Preprocessing:
- Convert trimming curves into polylines
- Sew polylines in 3D
- Generate Seam Graph
- Generate LOD on Seam Graph

Interactive Rendering:
- Select view-dependent LOD from Seam Graph
- Culling of invisible NURBS patches
- On-the-fly tessellation of modified patches

Patchwise Culling

- Patches represent natural subdivisions
- Culling becomes efficient
- View-Frustum Culling
  - Test Bounding Box of whole patch
- Backface Culling
  - Compute minimal Normal Cone per patch

⇒ Culled patches need not be triangulated
Patchwise Culling

No Culling

Model courtesy of Volkswagen AG

Patchwise Culling

No Culling

View Frustum Culling

Model courtesy of Volkswagen AG

University of Bonn • Computer Graphics Group
Michael Guthe, Jan Meseth, Reinhard Klein
Patchwise Culling

No Culling | View Frustum Culling | View Frustum and Backface Culling
--- | --- | ---

Model courtesy of Volkswagen AG

Load Balancing

Number of patches to retriangulate differs strongly from frame to frame

⇒ Guarantee stable frame rates by balancing load between frames
  ⇒ Can triangulate about 25000 triangles per second
  ⇒ To achieve 25 fps, restrict number of triangles to be generated to 1000 per frame
  ⇒ Allow LOD modifications only until the estimated sum of triangles resulting from retriangulation of affected patches exceeds 1000

⇒ Screen-projection error not always below one pixel
Results

Model courtesy of Volkswagen AG

Effects of Culling

PC Configuration:
Intel P4 1.8GHz
512 MB RAM
Geforce 3, 64 MB
1024 x 768 Pixel

Screen-Space Error

Frame Rate
Conclusions

- High-quality trimmed NURBS rendering
- Close to real-time
- Seam Graph
- Load-balancing strategies

Future Work

- Parallelization
- Occlusion Culling
- Normal Maps (submitted)
- Textures
- Different triangulation schemes
- Integration into OpenSG
  (www.opensg.org)
Acknowledgements

- The models used in this talk were kindly provided by Volkswagen AG
- This work was funded by the BMBF (German Federal Ministry of Education and Research) under the project OpenSG PLUS