Direct3D 11 Tessellation

Tianyun Ni (倪天芸), Nvidia
Pixels are meticulously shaded, but geometric detail is modest
Introduction(1)

• Provides cinematic visual details for real-time rendering

• Achieves more details with less storage
  – Input: a coarse mesh + displacement map
  – Geometry expansion on GPU

• Performs expensive computations at lower frequency
Introduction (2)

- Flexible Level of Detail (LOD) control
  - Adjusts LOD at run time
  - Creates geometry only where needed
  - Generates continuous level of detail
- Can be applied to almost everything that needs more details in a game scene
- Focus on character tessellation in this talk
Demos
Content Creation Pipeline

- Existing DCC tools: 3dsMax, Maya, Blender, Zbrush, Mudbox, Modo, Grazy Bump
  - Modeling Tools
  - Sculpting Tools
  - Baker Tools (generates displacement map, normal map, occlusion map, cavity map, etc)
Content Creation Pipeline

Parameterized coarse model (3DS MAX, Maya, ...)

Subdivide and Sculpture (ZBrush, ...)

Displacement & occlusion maps

Baker Tool

Modeling Tool

Sculpturing Tool

Engine preprocessor

gameworks.nvidia.com
Overview of Various Tessellation Schemes
Tessellation Schemes

• Choose appropriate schemes for your art assets
  – Mesh connection type
  – Implementation complexity
  – Tradeoff between performance and visual quality
• Linear interpolation
• Local construction schemes
  – PN, Phong Tessellation
• Approximating Catmull-Clark Schemes
PN Triangles

• “Curved PN Triangles”, by Alex Vlachos, Jörg Peters, Chas Boyd, and Jason Mitchell, I3D 2001.

• “PN Quads”, by Jörg Peters, 2008.
  http://www.cise.ufl.edu/submit/files/file_020f70fe71888f602530143e2e326be2.pdf
  – The same formulae except for computing interior control points
Phong Tessellation

Figure 3: Phong Tessellation principle. Instead of interpolating normals as in Phong Shading, we interpolate projection onto vertices tangent plane to define a curve geometry for each triangle.

- From Boubekeur and Alexa, Siggraph Asia 2008
Phong Tessellation

• Simple scheme
• Does not handle inflections well
• Acceptable if initial subdivision is rather dense

Boubekeur and Alexa, 2008
Approximating Catmull-Clark Subdivision Surfaces Schemes

• Provides movie-quality surfaces
  – Catmull-Clark subdivision surfaces are extensively used in movie production and modeling & sculpting tools
  – Suitable for quadrilateral meshes with few triangles in it

• Approximation rather than interpolation

• Requires the mesh info of a facet and its 1-ring neighborhood
Approximating Catmull-Clark Subdivision Surfaces Schemes

• Convert each face of an input mesh to a gregory patch
  – Regular quad ➔ Bicubic Bézier Patch
  – Irregular quad ➔ Tensor-product
  – Triangle ➔ Triangular gregory patch
**Tessellation Schemes Comparison**

<table>
<thead>
<tr>
<th>Scheme</th>
<th># of vertices in a patch primitive</th>
<th># of control points</th>
<th>Base mesh</th>
<th>Surface fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phong</td>
<td>3 Or 4</td>
<td>6 Or 9</td>
<td>▲ □</td>
<td>Artifacts at inflection points and high curvature area</td>
</tr>
<tr>
<td>PN</td>
<td>3 Or 4</td>
<td>10+6 Or 16+9</td>
<td>▲ □</td>
<td>Artifacts at high curvature area</td>
</tr>
<tr>
<td>Gregory ACC</td>
<td>16 to 32</td>
<td>15 Or 20</td>
<td>▲ □</td>
<td>Similar to CC surfaces</td>
</tr>
</tbody>
</table>

- Various tessellation schemes differ at
  - Patch primitive
  - Control points computations (in Hull Shader)
  - Surface evaluation (in Domain Shade)

[gameworks.nvidia.com](http://gameworks.nvidia.com)
DirectX 11
**Direc3D11 Tessellation Pipeline**

- **Patch Primitive**
  - Input Assembler
  - Vertex Shader
  - Hull Shader
  - Tessellator
  - Domain Shader
  - Setup/Raster
  - Pixel Shader

- **Skinning**

```cpp
struct VERTEX {
    float3 vPosition : POSITION;
    float2 vUV : TEXCOORD0;
    float3 vTangent : TANGENT;
    uint4 vBones : BONES;
    float4 vWeights : WEIGHTS;
};
```

- Input Mesh
  (a collection of patch primitives)
  Displacement Map
  Normal Map (optional)
Direc3D11 Tessellation Pipeline

Patch Primitive

Input Assembler

Vertex Shader

Hull Shader

Tessellator

Domain Shader

Setup/Raster

Pixel Shader

Input Mesh
(a collection of patch primitives)

Skinning

• Compute Control Points
• Compute LOD

Geometry expansion
Direc3D11 Tessellation Pipeline

**Patch Primitive**
- Input Assembler
- Vertex Shader
- Hull Shader
- Tessellator
- Domain Shader
- Setup/Raster
- Pixel Shader

**Skinning**
- Compute Control Points
- Compute LOD

**Geometry expansion**
- Surface Evaluation
- Displacement mapping
- Normal mapping (move to DS stage?)
- Shading calculation

Input Mesh (a collection of patch primitives)

Patch Surface

High-detailed Mesh
Practical Concerns
Tessellation Factors

• Domains: Triangle / Quad / Isolines
• Spacing:
  – Discrete / Continuous / Pow2
• Computed in the constant phase stage of the hull shader
  – Edge tessellation factors
  – Interior tessellation factors
    • ProcessTriTessFactorsMax
    • ProcessTriTessFactorsMin
    • ProcessTriTessFactorsAvg
Avoid Unnecessary Geometry

• Camera distance
• Screen space
  – Be careful that edges are curved
  – Triangles under 8 pixels are not efficient
• Silhouette
  – Patches might be curved after tessellation
  – Displaced mapping makes silhouette detection harder
• Height values in Displacement Map
Avoid Unnecessary Geometry

- Frustum culling and back face culling in the hull shader
  - Set all tessellation factors to 0
  - Culled patches will not send down to the pipeline

- Issues when back faces are detected
  - Patches might be curved after tessellation
  - Displaced mapping adds more variations for face orientations
Water-tight Tesselllation

• Non-watertight position evaluation causes small cracks
• Non-watertight normal and tangent space evaluation causes shading discontinuity
• Causes
  – Limited floating point precision issue
  – Texture seams are discontinuities in the parameterization
Floating Point Consistency

- FP addition is non commutative:
  - \( A + B + C + D \neq D + C + B + A \)
  - \( (A + B) + (C + D) = (D + C) + (B + A) \)

- FMA is not equivalent to MUL+ADD
  - \( A*a + B*b \rightarrow FMA(A*a, B, b) \neq FMA(B*b, A, a) \)

- Use consistent or symmetric evaluation order along shared edges
  - Corner and edge control points computation
  - Surface evaluation
Cracks: Discontinuity in parametrization
Solution for cracks

• Artist work
  – Hide texture seams
  – Kill displacement to zero on texture atlas seams
  – Remapping to unit length and orientation

• We suggested automatic ownership-based solution
Cracks: Ownership-Based Solution

- Boundary triangles share the same texture coordinates data
Watertight Displacement

- Store 4 texture coordinates per vertex (0: interior, 1,2: edges, 3: corner)
  - 16 per quad patch, 12 per triangle patch
  - The corresponding texture coordinate can be selected using the value of the parametric coordinate UV
  - Compute Barycentric interpolation of texture coordinates
Creases and Corners

• Indicated in the input mesh where vertices that share the same position but have different normals
  – Solution 1: insert tiny polygons (triangles or quads) on creases/corners
    • Insertion by artists
    • Automatic detection and insertion
  – Solution 2: tag the vertices on creases/corners
    • Preprocessing step to add tags
    • Modify correspondent shader code

[KMDZ09]
Tessellation in Metro 2033
**Metro 2033: the game**

- A combination of horror, survival, RPG and shooting
- Based on a novel by Dmitry Glukhovsky
Tessellated Monster in Metro 2033

Use Phong tessellation

From Metro 2033, © THQ and 4A Games
More Metro Monsters...

From Metro 2033, © THQ and 4A Games
... and Characters

From *Metro 2033*, © THQ and 4A Games
... more Characters

From *Metro 2033*, © THQ and 4A Games
Start with a Coarse Base Mesh
HW-based Subdivision
Apply Displacement
Computing Displacement

- Input: coarse and detailed model

Detailed surface

Coarse surface
LOD Computation

• TESS_FACT = LEN * NP * Q / DIST
• Where LEN is edge length in world space
• NP is number of pixels on the screen
• Q is quality constant
• DIST is distance from observer to edge center
Constant Triangle Size
Artifacts on Hard Edges
Transitional Polygons
Summary

• Direct3D11 Tessellation enriches visual details with flexible LOD control
• Choose a tessellation scheme that fits your needs
• Implement it efficiently
• It’s time to bring games to the next level
Thank you!

Q&A: tni@nvidia.com