Efficient Pixel-Accurate Rendering of Curved Surfaces

Young In Yeo  
University of Florida

Lihan Bin  
University of Florida

Jörg Peters  
University of Florida

Optimal Tessellation?

Tessellation too low  
> 2000 fps

Tessellation too high  
< 50 fps
Outline

• What is Efficiency?
  – Triangle size
  – Non recursive

• What is Accuracy?

• Test & Predict (SLEFE)

• Implementation & Demo
**Efficiency**

Low Tessellation

- > 2000 fps

High Tessellation

- < 50 fps

- ▲ size < 5 pixels
- △ size < 10 pixels
- ▲ size < 20 pixels
- △ size >= 20 pixels

**Efficiency**

Efficiency:
- Triangle size

- size < 5 pixels
- size < 10 pixels
- size < 20 pixels
- size >= 20 pixels
Micropolygonization

Naïve approach

REYES

Our approach

Recursive !!

Not Recursive !!

Introduction
efficiency
pixel-accuracy
test & predict
implementation
Efficient Pixel-Accurate Rendering of Curved Surfaces

Our Approach

> 2000 fps
~ 1500 fps
< 50 fps

- size < 5 pixels
- size < 10 pixels
- size < 20 pixels
- size >= 20 pixels
Outline

• What is Efficiency?

• **What is Accuracy?**
  – Covering accuracy (depth)
  – Parametric accuracy (distortion)

• Test & Predict (SLEFE)

• Implementation & Demo
Covering accuracy (depth)

- Correct surface piece determines the pixel

Low Tessellation

High Tessellation

Inaccurate covering pixels
Parametric accuracy
(no distortion & wrong normals)

Our Approach

Patch

Linear Approx.

Rasterization

Low Tessellation

dist < 0.1 pixel
accurate

dist < 0.5 pixel

dist < 1.0 pixel

dist ≥ 1.0 pixel
inaccurate
Our Approach

dist < 0.1 pixel
dist < 0.5 pixel
dist < 1.0 pixel
dist ≥ 1.0 pixel

accurate

inaccurate

> 2000 fps
~ 1500 fps
< 50 fps

Low Tessellation

High Tessellation

efficiency
dist < 0.1 pixel
dist < 0.5 pixel
dist < 1.0 pixel
dist ≥ 1.0 pixel

accurate

inaccurate

Efficient Pixel-Accurate Rendering of Curved Surfaces

SurfLab
University of Florida
Outline

• What is Efficiency?
• What is Accuracy?
• Test & Predict (SLEFE)
• Implementation & Demo
Key insight

• Triangulations are rendered accurately.

• Need to only control the variance of (the projection of) the triangulated surface to the exact surface.
What is a SLEFE?

Curve and Control Polygon

AABB  Bounding Disk  Convex Hull  OBB  m=3-piece slefe
SLEFE: Subdividable Linear Efficient Function Enclosure
(tightly sandwich non-linear functions)

Non-linear functions

\[ \underline{p} \leq p \leq \overline{p} \]

Lower piecewise approximation

\[
p(t) \leq \overline{p}(t) := \ell(t) + \sum_{j=1}^{d-1} \max\{0, \nabla_j^2 p\} \overline{a}_j^m(t) \\
+ \sum_{j=1}^{d-1} \min\{0, \nabla_j^2 p\} \underline{a}_j^m(t).
\]

Width(\(W\))

Slefe table
Why SLEFE?

3 segments

2 * 3 segments

$W = 0.2767$

$\bar{w_2} = 0.0579$

$W \geq 4\bar{w_2}$

$W \geq h^2\bar{w_h}$
Test & Predict Accuracy in Pixel

\[ \frac{1}{h^2} \leq 1 \text{ pixel} \iff h > \sqrt{W} \]

Tessellation Factor \[ 3 \times \sqrt{W} \]
Outline

• What is Efficiency?
• What is Accuracy?
• Test & Predict (SLEFE)
• Implementation & Demo
Implementation

$\text{patch } p \text{ (control points } c_{ij} \text{), slefeTable}$

$\overline{p}$

$\text{TF}_p = 3 \times \sqrt{W}$

Pixel-Accurate Rendering

- Hull Shader
- Tessellator
- Domain Shader
- Rasterizer
- Pixel Shader

Standard DX11 Rendering

$\text{TF per patch & edge} \rightarrow \text{water-tight}$
Demo

Efficient Pixel-Accurate Rendering of Curved Surfaces

Captured @ 1440 x 900 resolution on NVIDIA GTX 480
Summary

Accurate
- Distortion < 1/2 pixel (= error not visible).

Efficient
- Triangles of maximal size & no recursion.

Automatic
- No need for manual LoD.

Fast
- > 300 fps for 130k patches.

Thank You