Resample Hardware for 3D Graphics

Koen Meinds
Bart Barenbrug

Outline

- High image quality, yet low cost
- Texture anti-aliasing
- Edge anti-aliasing
- Closing remarks

3D graphics hardware with

- **HIGH IMAGE QUALITY:** requires proper anti-aliasing use of “video” resample technology
- **LOW COST:** reduce off-chip data traffic

Quality comparison: movie

<table>
<thead>
<tr>
<th>Philips</th>
<th>anisotropic 2x2</th>
<th>anisotropic 4x4</th>
</tr>
</thead>
</table>

Our quality > 4x4, yet
Bandwidth cost < 2x2

Outline

- High image quality, yet low cost
  - Texture anti-aliasing
  - Edge anti-aliasing
  - Closing remarks

Texturing mapping = resampling

**Theoretical resample process:**

1. **reconstruct continuous signal from texels**
2. transform the continuous signal to screen
3. filter out high frequencies, in screen
4. sample continuous signal at screen pixels

Outline

- High image quality, yet low cost
- Texture anti-aliasing
- Edge anti-aliasing
- Closing remarks

Outline

- High image quality, yet low cost
- Texture anti-aliasing
- Edge anti-aliasing
- Closing remarks

Outline

- High image quality, yet low cost
- Texture anti-aliasing
- Edge anti-aliasing
- Closing remarks

Outline

- High image quality, yet low cost
- Texture anti-aliasing
- Edge anti-aliasing
- Closing remarks
Texture mapping = resampling

Theoretical resample process:
1. reconstruct continuous signal from texels
2. **transform the continuous signal to screen**
3. filter out high frequencies, in screen
4. sample continuous signal at screen pixels

Screen position

---

Proper anti-aliasing

Theoretical resample process:
1. reconstruct continuous signal from texels
2. transform the continuous signal to screen
3. **filter out high frequencies, in screen**
4. sample continuous signal at screen pixels

For proper anti-aliasing step 3 is most important.

---

Approximation of resample process:
**Inverse texture mapping**

Proper anti-aliasing filtering = difficult

---

Approximation of resample process:
**Forward texture mapping**

Proper anti-aliasing filtering = easy
FTM: Forward texture mapping

FTM requires:
Multiple contributions to same pixel in non-sequential order.

- Problem: contributions to off-chip pixels: high memory data traffic
- Solution:
  - expensive: write cache.
  - better: “two-pass” FTM

Avoid buffer between two-passes

- New problem:
  Off-chip data traffic needed for intermediate image
- Solution:
  Interleave horizontal and vertical pass.
  Only a few on-chip line memories needed

MIP-map

We have included MIP-map technique for further data traffic reduction

- Problem:
  Combining MIP-maps and 2-pass forward texture mapping
- Solution:
  Generate, on-the-fly, 4D MIP-maps from standard 3D MIP-maps

Scalability

We reuse existing video resampler technology:
- scalable: quality-cost trade-off

Outline

- High image quality, yet low cost
- Texture anti-aliasing
  - Edge anti-aliasing
- Closing remarks
Texture filtering across edges
Forward texture mapping enables high quality edge anti-aliasing

Pixel fragment buffer
Fragment storage:
- depth sorted
- ≤ 4 fragments per pixel

Comparison: Required bandwidth (1)
2x2 super sampling:

Comparison: Required bandwidth (2)
Measured off-chip data traffic:

Outline
- High image quality, yet low cost
- Texture anti-aliasing
- Edge anti-aliasing
- Closing remarks
Closing remarks (1)

Implemented in OpenGL SW pipeline:
• suited to hardware implementation
• two-pass FTM
• extended to edge anti-aliasing

Efficient forward texture mapping, due to:
• use of video resample structures
• Interleaving two passes: reduces data traffic
• integration of MIP-maps
• Uniform solution: texture and edge anti-aliasing

Closing remarks (2)

Compared to anisotropic inv. texture mapping:
• Quality > 4x4 super sampling, yet
• Bandwidth cost < 2x2 super sampling

Work is “in progress”
• pixel shading
• dependent multi texturing (e.g. bumpmapping)

movie can be found at:
www.extra.research.philips.com/graphics