Text rendering in 3D scenes

THIERRY DELISLE
Classic text rendering

Formatter is run on input to set characters.

Formatting with knowledge of page size, resolution, etc.

Character glyphs are converted to bitmaps.
Challenges of 3D

Viewing Angle:
  ◦ Text is parallel to screen? Maybe not!
  ◦ Text is right side up? Maybe not!

Animations:
  ◦ Angle, Distance, Size, etc. can change dynamically.

Interactivity:
  ◦ Users can affect all of the above and expect real-time corrections.

Goal: Transformation resistant rendering algorithms.
A well-known example

During the battle, Rebel spies managed to steal secret plans to the Empire’s ultimate weapon, the DEATH STAR, an armored space

Star Wars is a registered trademark and © 1977 Lucasfilm Ltd.
A well-known example

The viewing angle is not perpendicular.

Perspective projections mean the text has depth.

Words will appear to vary in size as they recede from the screen.
Scope

Aspects outside the scope of this discussion:

- Formatting
- Ligatures
- Text direction
Rendering Basics

RENDERING PIPELINE AND HARDWARE
Pipeline use by most PCs, Phones, Game Consoles, etc.

A rectangular stage is programmable: code dictates its behaviour.

A rounded stage is fixed: configurations dictate its behaviour.

* Several stages have been collapsed together for simplicity.
Input Stage

First step: Send some geometry to the GPU.

Rasterizer requires triangles/lines.

GPU allows significant transformation on the geometry before rasterization.

- Example: transforming quads into triangles.

Common transformations:

- Perspective projection
- Animation
- Tessellation
Rasterization

Converts 3D primitives to pixel information.

Requires lines or triangles.

Linearly interpolates data on vertices.

Handle depth testing.
Fragment Shading

Programmable stage ran on every pixel.

- Potentially more than once of covered by several primitives.

Commonly where textures are used.

Hardware can do Bilinear Interpolation automatically when sampling textures.
Blending

Colors the desired pixel.

Overlapping means a given pixel can be colored several times.

Several rules are possible when recoloring:

◦ Replace old color.

◦ Interpolate new and old colors based on transparency.

◦ Etc.
Texture-Based Rendering

THE SIMPLE APPROACH
What’s wrong with bitmaps?

Use texture sampling + transparency to render text.

1. Make texture with desired text + transparent background.
2. Render a triangle or a quad (2 triangle forming a flat rectangle).
3. In fragment shader, paste texture onto triangle(s).
4. Done!
Primitive Submission

SINGLE PRIMITIVE
Submit one big quad.
Write the text in one big texture.
Formatting done in texture.

PRIMITIVE PER CHARACTER
One quad per letter.
Texture contains alphabet.
Formatting done on primitives.
Direct bitmap

Simplest approach.

Works well for printers.

Very easy to use since:

◦ Hardware rasterization

◦ Hardware blending

◦ Hardware sampling

Means texture hardware can be reused with little or no programming.
Result: Blur

A NEW HOPE

It is a period of civil war.


disabled starships, striking
drew a death star, and 

the evil Galactic Empire.

During the battle, Rebel
spies managed to steal secret
plans to the Empire’s
ultimate weapon, the DEATH
STAR, an armored space
station with enough power to
destroy an entire planet.

Pursued by the Empire’s
sinister agents, Princess
Leia races home aboard her
starship, custodian of the
stolen plans that can save
her people and restore
freedom to the galaxy.....
Alpha-Testing

Eliminates the blur using threshold instead.

- Opacity > 0.5 → Full Color
- Opacity < 0.5 → No Color

Still done in hardware so no additional requirements compared to previous approach.
During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet.

Pursued by the Empire's sinister agents, Princess Leia races home aboard her starship, custodian of the stolen plans that can save her people and restore freedom to the galaxy.....
Bitmap problem

Magnification is handled by hardware using bilinear interpolation.

Interpolation is defined on Scalars not on Booleans.

Interpolating bitmaps results in... Something.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inside</td>
</tr>
<tr>
<td>0.6</td>
<td>Somewhere... Maybe in a galaxy far far away</td>
</tr>
<tr>
<td>0</td>
<td>Outside</td>
</tr>
</tbody>
</table>
Signed Distance Fields

Use full width of color channel.
Interpolation has proper meaning.
Still using hardware!
Well-known example:

Improved Alpha-Tested Magnification for Vector Textures and Special Effects

Chris Green
Valve

Using SDFs

Submit a signed distance field instead of the bitmap.

... 

That’s all, the hardware does the rest.
Result: Not bad

During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet.

Pursued by the Empire's sinister agents, Princess Leia races home aboard her starship, custodian of the stolen plans that can save her people and restore freedom to the galaxy.....
Anisotropy: the problem

During the battle, Rebel spies managed to steal secret plans to the Empire’s ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet.

Pursued by the Empire’s sinister agents, Princess Leia races home aboard her starship, custodian of the stolen plans that can save her people and restore freedom to the galaxy....
Anisotropy: the solution

Hardware to the rescue yet again!

Anisotropic filtering exists for regular textures.

Instead of bilinear sampling 2x2:
  - Sample 4x2, 8x2, 16x2, etc.

Hardware caches result in textures (right).

Better filtering exist by Qin, Zheng from the University of Waterloo.

Anisotropy: 4x2

During the battle, Rebel spies managed to steal secret plans to the Empire’s ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet.

Pursued by the Empire’s sinister agents, Princess Leia races home aboard her starship, custodian of the stolen plans that can save her people and restore freedom to the galaxy......
Anisotropy: 32x2

During the battle, Rebel spies managed to steal secret plans to the Empire’s ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet.

Pursued by the Empire’s sinister agents, Princess Leia races home aboard her starship, custodian of the stolen plans that can save her people and restore freedom to the galaxy.....
New problem: Corners

Text rendered using SDFs lose sharp corners.

Fundamental problem of SDF.

SDFs naturally round-off shapes.
Solutions

2-channel SDF from Green07

3-channel SDF from Chlumský

4-channel SDF from Lambda cube

Quasi-SDF from Adam Simmons

It all starts to look like Outline-Based Rendering...
Outline-Based Rendering

THE PRETTY APPROACH
What’s wrong with textures?

Double rasterization:
1. Text rasterized to texture.
2. Texture ‘rasterized’ again for rendering.

Solution: Go directly from text to final render.

Direct rasterization

Idea:

*Convert the Glyph outline directly by using geometry + shader code to differentiate inside and outside.*

Many techniques exist:

- Loop-Blinn uses quadratic Béziers.
- Slug uses winding order.
- Nvidia uses ‘Stencil, then Cover’.
- Other techniques exist.
Loop-Blinn

From Glyph outline create triangulated geometry.

Create implicit curve for each triangle:

\[ f(u, v) = u^2 - v \]

Where \( uv \) coordinates are \([0,0], \left[\frac{1}{2}, 0\right], [1,1]\) for all triangles.

Source: Loop, Charles, and Jim Blinn. ‘Resolution independent curve rendering using programmable graphics hardware.’
Use single large primitive.

Cast rays in shape and find intersections.

Add or subtract for each intersection:
- Total > 0 → means inside.
- Total = 0 → means open space.

Source: Lengyel, Eric. ‘GPU-Centered Font Rendering Directly from Glyph Outlines.’
Stencil, then Cover

Starting from Glyph outline:

1. Create *Line-edged triangle fans (b).*
2. Render to stencil using invert operation.
3. Render curves using Loop-Blinn and same operation 
   *(c).*
4. Add both stencils and fill, i.e., \((d) \oplus (c) = (f)\).

Uses triangle fans instead of regular triangles to avoid 
triangulating on the CPU.

Source: Kokojima, Yoshiyuki, et al. ‘Resolution independent rendering of 
deformable vector objects using graphics hardware.’

Nvidia version: Kilgard, Mark J., and Jeff Bolz. ‘GPU-accelerated path 
rendering.’
Non-Flat Text

ADAPTING TO THE SCENE
Many cases

Several techniques depending on the need:

◦ Decals

◦ Curved text

◦ Advanced texture mapping

◦ Innumerable custom or one-off solutions.

Decals: Stick Text/Textures onto objects

Project texture on top of objects.

Used both for text and regular textures.

Can be done in advance or dynamically.

Creator: Rob Schluntz
Decals: Techniques

GEOMETRY TO PIXEL

1. Create projector prism.
2. Create duplicate of geometry clipped to a prism.
3. Render clipped geometry.

Disadvantage:
- Step 2 is hard to get right and slow to process.

PIXEL TO PIXEL

1. Render first pass.
2. Render projector prism.
3. Find affected pixels.
4. Find position of pixels in geometry.
5. Re-shade relevant pixels.
6. Render second pass.

Disadvantage:
- Requires Deferred Rendering (Multi-Pass)
Curved Text

Using geometry means text can be transformed.

The finer grain the triangulation is, the more transformations can be applied.

Advanced Texture Mapping

Example:

Ptex: Per-Face Texture Mapping for Production Rendering

Brent Burley
Walt Disney Animation Studios

Dylan Lacewell
University of Utah
Full Geometry

THE EXPENSIVE APPROACH
3D meshes of text

When text needs to be part of the world rather than pasted onto it:

- Use 3D meshes.

Appearance limited only by Renderer.

Amount of work required not limited.
Questions