An Improved Representation for Stroke-based Fonts

Elena Jakubiak & Sarah Frisken, Tufts University
Ronald Perry, Mitsubishi Electric Research Laboratories
Embedded Devices

• There is a proliferation of embedded devices
  – Cell phones, PDAs, car navigation systems, printers, cameras, kitchen appliances, etc.
  – Require the display of type
• Embedded devices are
  – Memory constrained
  – Processor constrained
East Asian Typefaces

- Can require large amounts of memory
  - Especially for high quality outline fonts, the de facto standard for scalable type

- Can be slow to process
  - Especially with hinting

The Chinese character for turtle
Stylized Stroke Fonts

• An improved representation that
  – Produces high quality fonts
  – Exploits the natural stroke construction of East Asian characters
  – Significantly reduces memory requirements
  – Does not increase processing requirements

The Japanese Hiragana character ‘wa’
Important Uses of Type

- Typefaces convey style and meaning

**Genre**  **Formality**  **Essence**  **IMPORTANCE**

- Serifs and subtle stroke variations increase legibility

In this presentation we describe an improved representation for stroke-based fonts, entitled Stylized Stroke Fonts (SSFs). SSFs overcome the shortcomings of existing scalable font representations by providing the detail and expressiveness of outline-based fonts with a memory footprint comparable to current stroke-based fonts. Each glyph of an SSF consists of three main elements: a stroke path which is composed of a set of line segments and Bezier curves and runs approximately along the centerline of the glyph; a set of stroke profiles which define the shape of the stroke by specifying the perpendicular distance from the stroke path to the edge of the glyph where the distances are...
Typefaces on Embedded Devices

- As memory and processing capacity increases on embedded devices, so does the demand for richer content
  - Richer content requires more varied and expressive type
  - Trend towards storing multiple Latin typefaces
East Asian Typefaces

- East Asian media uses a rich variety of type
**East Asian Typefaces**

- East Asian typefaces have significantly more characters than Latin typefaces
  - Difficult to store multiple East Asian typefaces on embedded devices

<table>
<thead>
<tr>
<th># Characters</th>
<th>Character Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>224</td>
<td>Latin-1</td>
</tr>
<tr>
<td>7,043</td>
<td>Japanese (Kanji)</td>
</tr>
<tr>
<td>7,580</td>
<td>Simplified Chinese</td>
</tr>
<tr>
<td>13,860</td>
<td>Traditional Chinese</td>
</tr>
<tr>
<td>17,142</td>
<td>Korean</td>
</tr>
<tr>
<td>18,267</td>
<td>Traditional Chinese with Hong Kong Character Set</td>
</tr>
</tbody>
</table>

Number of characters encoded in Unicode for Latin and East Asian Languages [Thomas]
Challenges for Embedded Devices

• Memory Requirements
  – Embedded devices have limited memory
  – East Asian typefaces typically require 2 – 25 mb per typeface
Challenges for Embedded Devices

• Rendering & processing speed
  – Embedded devices have limited computational resources
  – Rendering must be fast and simple
    • Cannot rely on significant cache sizes
    • Limited computation is available for hinting and anti-aliasing
Challenges for Embedded Devices

- Rendering Quality
  - Embedded devices have limited screen real estate
  - Characters must be rendered at small point sizes and legibility is crucial
  → problematic for complex characters
Traditional Font Representations

- Bitmaps – advantages
  - Bitmaps can be hand tuned to produce higher quality images at small point sizes

Bitmaps are typically monochromatic
Traditional Font Representations

- Bitmaps – disadvantages
  - Monochromatic bitmaps are not anti-aliased
  - Bitmaps are not scalable
  - Hand tuning bitmaps is labor intensive
  - Bitmaps for East Asian character sets require significant amounts of memory

- Typically store a full set of bitmaps for a limited set of point sizes

<table>
<thead>
<tr>
<th>28 pt</th>
<th>20 pt</th>
<th>12 pt</th>
<th>8 pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>纳 优 质 中 文 字 库 永…</td>
<td>纳 优 质 中 文 字 库 永…</td>
<td>纳 优 质 中 文 字 库 永…</td>
<td>纳 优 质 中 文 字 库 永…</td>
</tr>
</tbody>
</table>
Traditional Font Representations

- Outlines – advantages
  - Scalable
    - One set of outlines generates all point sizes
  - Require less memory than bitmaps

Outlines are scaled to the desired point size and then rasterized.
Traditional Font Representations

• Outlines – disadvantages
  – Requires hinting each character to optimize legibility
  – Hints require significant additional memory
  – Embedded bitmaps are often required for optimal resolution at small point sizes

Unhinted outlines lead to inconsistent stroke widths and dropout

Hinting aligns the outlines to the pixel grid for fewer rasterization artifacts
Compressing East Asian Typefaces

• Various approaches have been taken to compress East Asian typefaces
  – Component reuse
  – Pen-based representations
  – Uniform stroke representations
1. Component Reuse

- Characters are constructed from a set of basic strokes

Basic strokes

- Strokes are assembled into components (e.g., radicals or graphemes) which are composed into characters

Strokes + Radical → Character

1. Component Reuse

- **Advantages**
  - Store components and composition instructions rather than the outline for every character
  - Memory savings
    - Requires only 8 – 30 strokes to construct all Chinese characters
    - Only 330 components are required to construct 8,000 characters [Cun-Chang98]
1. Component Reuse

• Disadvantages

  – Stroke and component shapes vary given their position within the character

  The rén radical: a pictograph of a person
  Straight rén: A regular variation
  Slanted rén: An irregular variation

  – Parameters are needed to quantify differences
    • Parameters require memory
    • Requires type designers to think like programmers
2. Pen-based Representations

- A stroke is represented by a stroke path and a pen tip

2. Pen-based Representations

- Advantages
  - Stroke variation is expressive
  - Can mimic calligraphy
  - A single stroke path can be used with multiple pen tips to achieve different styles
2. Pen-based Representations

• Disadvantages – complicated to define
  – Defining how the pen should move is more a matter of mathematics than art
  – “asking an artist to become enough of a mathematician to understand how to write a font with 60 parameters is too much” – Knuth on METAFONT
2. Pen-based Representations

- Disadvantages – slow to render
  - Stamping
    - Move the pen tip incrementally along the stroke path
    - Sometimes must fill the same pixel many times
2. Pen-based Representations

- Disadvantages – slow to render
  - Skinning – create outlines from the pen-based representations
    - Elliptical pens swept along cubic curves produce high order outline curves
    - Slow, complex, and prone to errors
    - Difficult to do on the fly on embedded devices

Some rendering methods require that outlines do not overlap
3. Uniform Stroke Representations

- Control points are placed along a stroke center line
- A uniform stroke weight is specified
- Render using thick line drawing or as conventional outlines after skinning
3. Uniform Stroke Representations

- Advantages
  - Scalable
  - Can specify different uniform stroke weights (e.g., light, medium, bold)
  - Less memory
    - Less than half the points of outline fonts (e.g. Monotype’s Simplified Chinese stroke font which uses both component reuse and uniform strokes is 10x smaller than its outline equivalent)
3. Uniform Stroke Representations

- Disadvantages
  - Bland & unexpressive
  - Lacking cultural acceptance

<table>
<thead>
<tr>
<th>Monotype Stroke (SansMT2312)</th>
<th>我不是非有趣</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS Mincho</td>
<td>我不是非有趣</td>
</tr>
<tr>
<td>Epson Futo Gyoshu</td>
<td>我不是非有趣</td>
</tr>
</tbody>
</table>
Stylized Stroke Fonts

- Expressive fonts that require minimal memory
  - Pen-based representation
  - Variable width strokes
  - Specialized stroke end caps
Components of Stylized Stroke Fonts

• Stroke path
  – Composed of line segments and Bezier curves
  – Typically runs along the centerline of the stroke
Components of Stylized Stroke Fonts

- **Stroke profiles**
  - Define the shape of the stroke
  - Specify the perpendicular distance from the stroke path to an edge of the stroke

Stroke profiles can be one-sided or two-sided
Components of Stylized Stroke Fonts

- **Stroke end**
  - Represented as an outline
  - Determines the shape at each end of a stroke
Compression of Stylized Stroke Fonts

- Reuse end caps throughout the typeface
  - Translate, rotate, and scale end caps automatically to match a given stroke body
Compression of Stylized Stroke Fonts

- Reuse profile sets throughout the typeface
Compression of Stylized Stroke Fonts

• Reuse stroke paths across multiple typefaces

The same stroke path can be used for multiple typefaces
Cost of Storing Stylized Stroke Fonts

- Memory costs for storing a simplified Chinese typeface with 7,663 characters

<table>
<thead>
<tr>
<th>Representation</th>
<th>Size</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlines</td>
<td>2.5 mb</td>
<td><img src="image1" alt="Example" /></td>
</tr>
<tr>
<td>Uniform Stroke Fonts</td>
<td>250 kb</td>
<td><img src="image2" alt="Example" /></td>
</tr>
<tr>
<td>Stylized Stroke Fonts</td>
<td>338 kb</td>
<td><img src="image3" alt="Example" /></td>
</tr>
</tbody>
</table>

Stylized Stroke Fonts add ~25% to uniform stroke fonts for end caps and profile indices and 25kb for storing end caps and profile representations.
Rendering Stylized Stroke Fonts

• Leverage Saffron Type System for rendering Stylized Stroke Fonts [Perry2006]
  – Developed at Mitsubishi Electric Research Laboratories
  – Main type engine in Macromedia Flash 8
Rendering Stylized Stroke Fonts

- The Saffron Type System represents glyphs using various instantiations of Adaptively Sampled Distance Fields [Frisken2000]
  - Fast to render
  - High quality anti-aliasing
  - Low memory requirements

Adaptively Sampled Distance Field  Rendered glyph
Rendering Stylized Stroke Fonts

- Distance fields are implicit representations of shape

2D shape with sampled distances to the surface

2D distance field

Regularly sampled distance values
Rendering Stylized Stroke Fonts

- Advantages of rendering using distance fields
  - Simple, fast, and high quality anti-aliasing
  - Use constructive solid geometry operations (e.g., union) to combine strokes → **No skinning**
Stylized Stroke Fonts

• Remaining work
  – Incorporate Stylized Stroke Fonts into the Saffron Type System
  – Designing Stylized Stroke Fonts
    • Develop a semi-automatic approach for converting existing outline representations to Stylized Stroke Fonts
    • Create design tools for composing and editing Stylized Stroke Fonts
• Many thanks to
  – Mitsubishi Electric Research Laboratories for financial and technical support
  – Monotype Imaging for useful insight, discussion, and feedback
Some Relevant Papers


