



**HAL**  
open science

## The Art of Text (rendering)

Nicolas P. Rougier

► **To cite this version:**

Nicolas P. Rougier. The Art of Text (rendering). 39C3 2025 - 39th Chaos Communication Congress, Chaos Computer Club, Dec 2025, Hambourg, Germany. <hal-05430837>

**HAL Id: hal-05430837**

**<https://inria.hal.science/hal-05430837v1>**

Submitted on 24 Dec 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

POWER CYCLES

3903

The Art of Text (rendering)  
Nicolas P. Rougier & Behdad Esfahbod



# NOPE, NOT ARABIC!

## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers

From “Nope, not Arabic” tumbler (<https://nopenotarabic.tumblr.com/>)



# I UNICODÉ

## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

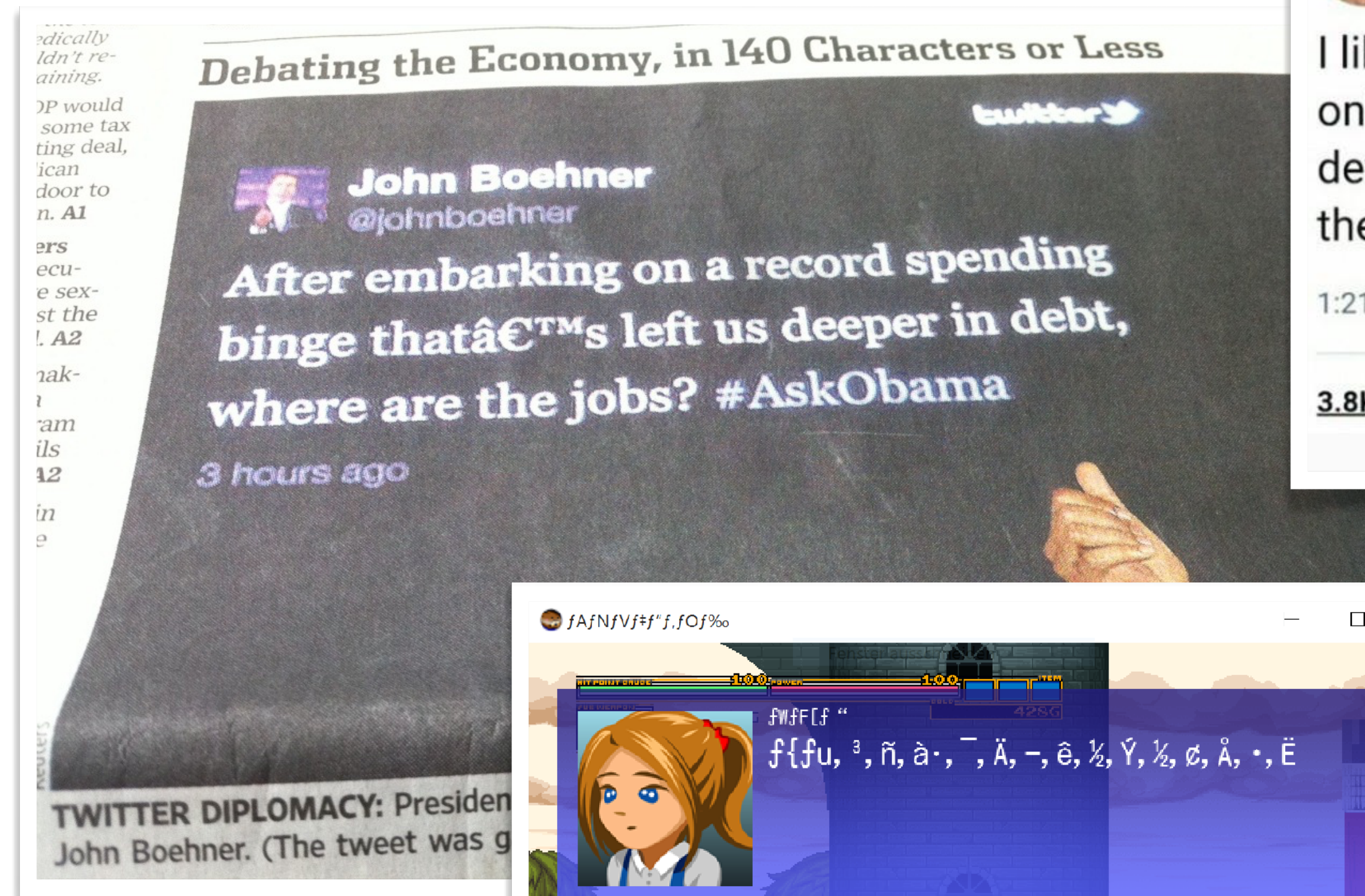
- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# BAD KERNING (keming)

Disaster is a space away... (FLICK, click, CLINT, FINAL)

Rockstar game removed the extra space ←



click versus dick



Apple added an extra space



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

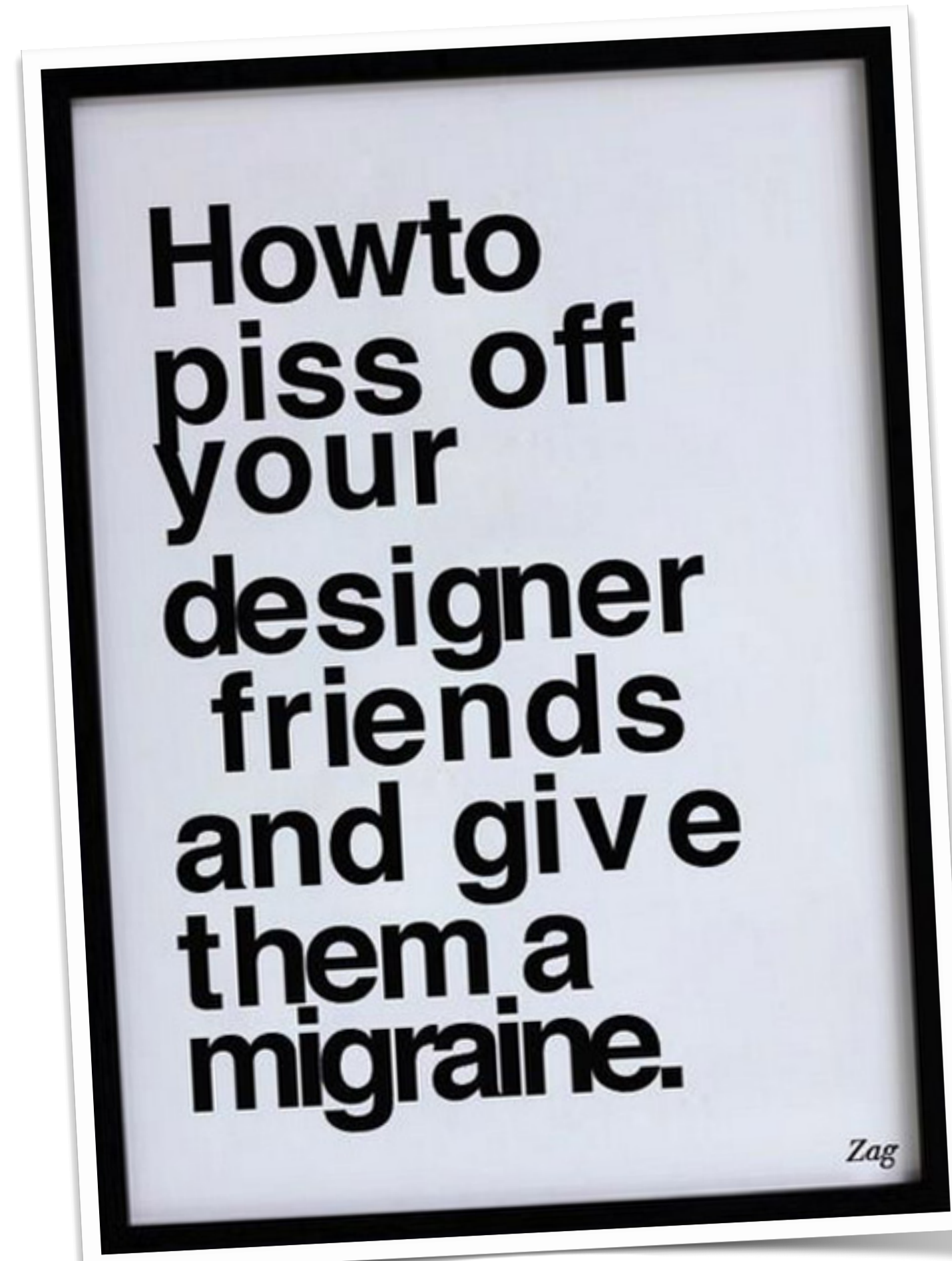
- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# DIGITAL TYPOGRAPHY

Typography is a vast and complex domain with many rules. You might consider to enforce some basic rules.



Canva

## The Anatomy of Typography

Stroke	Stem	Swash	Foot
<b>A</b> any linear element	<b>B</b> main vertical stroke	<b>C</b> decorative stroke	<b>D</b> bottom of stem, rests on baseline
<b>E</b> horizontal stroke not connected to stem on at least one end	<b>f</b> rises above the x-height	<b>g</b> extends below the baseline	<b>h</b> curved stroke attached to stem
<b>I</b> short line/stroke on open ends of letters	<b>j</b> curved stroke continuous with stem	<b>k</b> short, downward stroke	<b>L</b> where a stroke meets a stem
<b>M</b> the top and bottom points where two strokes meet	<b>n</b> Open: partially enclosed interior white space	<b>o</b> Closed: fully enclosed interior white space	<b>P</b> closed, round/oval curve
<b>Q</b> end of a stroke that's not a serif	<b>r</b> terminal that is circular in shape	<b>S</b> opening at the end of an open counter	<b>t</b> extends across a stem
<b>æ</b> enclosed horizontal stroke	<b>æ</b> two or more letters joined as one character	<b>ct</b> embellishment that connects a ligature	

Created for Canva by Janie Kliever



# FAMOUS TYPEFACES (FONT FAMILIES)

## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers

<p>Garamond</p> <p>Aa Qq Rr <i>á</i></p> <p><i>Aa Qq Rr</i> <i>á</i></p> <p>TRIANON</p> <p>abcdefghijklm nopqrstuvwxyz 0123456789</p> <p>1495</p>	<p>Caslon</p> <p>Aa Ee Rr <i>a</i></p> <p><i>Aa Ee Rr</i> <i>a</i></p> <p><i>Facquard</i></p> <p>abcdefghijklm nopqrstuvwxyz 1234567890  1234567890</p> <p>1734</p>	<p>Baskerville</p> <p>Aa Bb Cc <i>Q</i></p> <p>Xx Yy Zz <i>Q</i></p> <p><i>Nutgarden</i></p> <p>abcdefghijklm nopqrstuvwxyz 0123456789</p> <p>1757</p>	<p>Bodoni</p> <p>Aa Qq Rr <i>a</i></p> <p><i>Aa Qq Rr</i> <i>a</i></p> <p><i>HORATII</i></p> <p>abcdefghijklm nopqrstuvwxyz 0123456789</p> <p>1798</p>
<p><b>Gill Sans</b></p> <p>Aa Qq Rr <i>a</i></p> <p><i>Aa Qq Rr</i> <i>a</i></p> <p>COLLEGIUM</p> <p>abcdefghijklm nopqrstuvwxyz 0123456789</p> <p>1926</p>	<p>Times New Roman</p> <p>Aa Ee Rr <i>a</i></p> <p><i>Aa Ee Rr</i> <i>a</i></p> <p><b>Publisher</b></p> <p>abcdefghijklm nopqrstuvwxyz 0123456789</p> <p>1931</p>	<p>Helvetica</p> <p>Aa Ee Rr <i>a</i></p> <p><b>Aa Ee Rr</b> <i>a</i></p> <p><b>Kunsthalle</b></p> <p>abcdefghijklm nopqrstuvwxyz 0123456789</p> <p>1957</p>	<p>Arial</p> <p>Aa Ee Rr <i>a</i></p> <p><i>Aa Ee Rr</i> <i>a</i></p> <p><b>Deliverable</b></p> <p>abcdefghijklm nopqrstuvwxyz 0123456789</p> <p>1982</p>

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# ONE FONT TO RULE'EM ALL

When text is rendered by a computer, sometimes characters are displayed as “tofu”. They are little boxes to indicate your device doesn't have a font to display the text. Google has been developing a font family called Noto, which aims to support all languages with a harmonious look and feel. Noto(fu) is Google's answer to tofu.

Noto Serif

Noto Sans

Aa Gg Yy

Aa Gg Yy

a

Ankunft

思源黑体 思源黑體 源ノ角ゴシック 본고딕

Source Han Sans

	简体中文 Simplified Chinese	正體中文 Traditional Chinese	日本語 Japanese	한국어 Korean
ExtraLight	天地玄黄	律召調陽	色は匂へど	키스의고
Light	宇宙洪荒	雲騰致雨	散りぬるを	유조건은
Normal	日月盈昃	露結為霜	我が世誰ぞ	입술끼리
Regular	辰宿列张	金生麗水	常ならむ	만나야하
Medium	寒来暑往	玉出崑岡	有為の奥山	고특별한
Bold	秋收冬藏	劍號巨闕	今日越えて	기술은필
Heavy	闰余成岁	珠稱夜光	淺き夢見じ	요치않다

a b c d e f g h i j k l m

n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

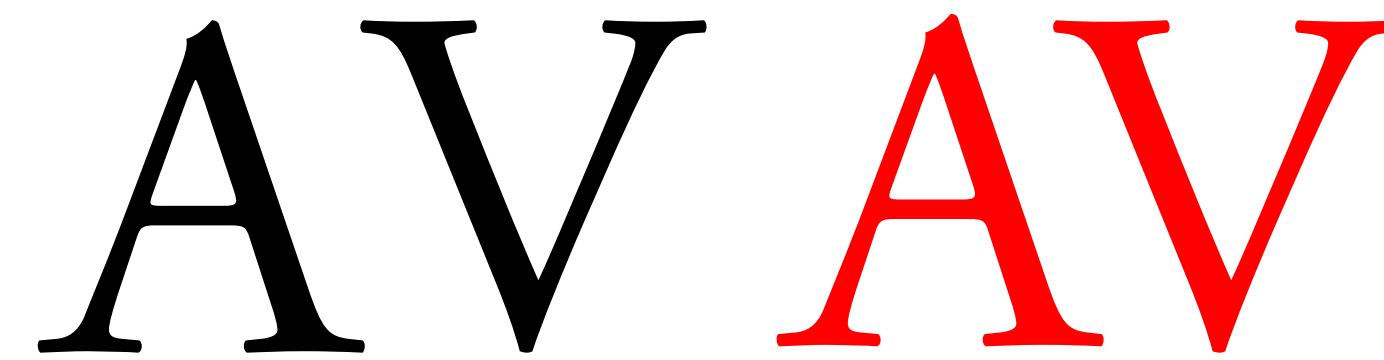
PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

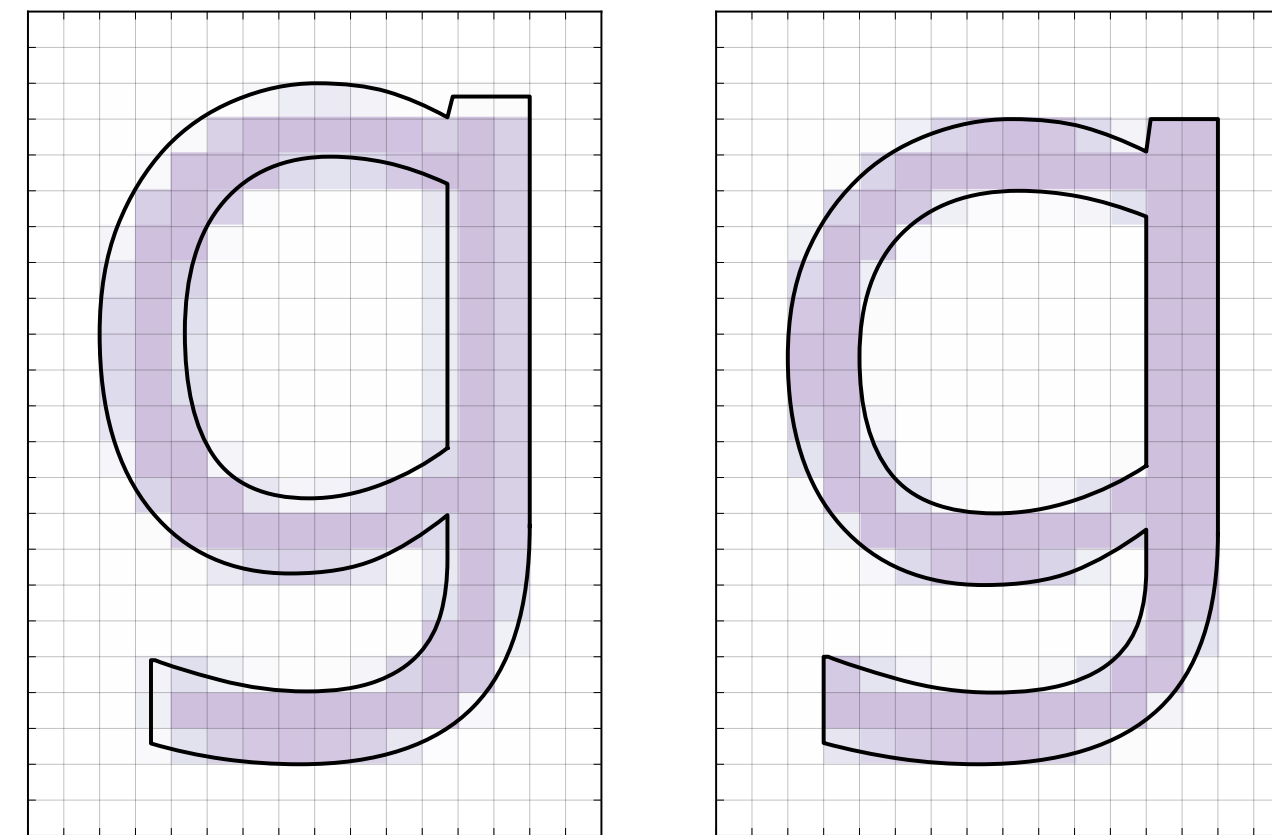
CONCLUSION

- Beyond this talk
- Questions & answers

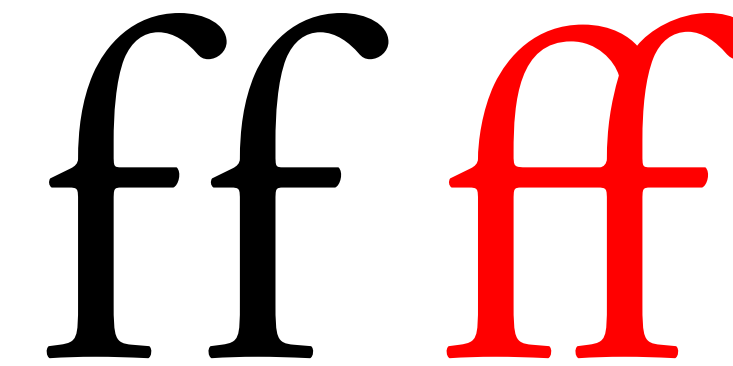
# BASIC TYPOGRAPHY



Kerning is the process of adjusting the spacing between characters in a proportional font.



Hinting is the use of instructions to adjust the display of a font so that it lines up with a rasterized grid.



A ligature occurs where two or more graphemes or letters are joined as a single glyph.



A diacritic mark is a glyph added to a letter, or basic glyph.



A various number of face metrics are defined for all glyphs in a given font.

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# ADVANCED TYPOGRAPHY

العربية العربية

Text shaping is the process of converting Unicode text to glyph indices and positions.

*Zapfino* *Zapfino*

Stylistic alternate allows to replace a glyph by some variant.

The title is **مفتاح معايير الويب** in Arabic.

The set of rules applied to produce the correct order at the time of display are described by the Unicode Bidirectional Algorithm.

द ् ध ्र ्य  
दध्र्य

Complex text layout (CTL) refers to the typesetting of writing systems in which the shape or positioning of a grapheme depends on its relation to other graphemes (wikipedia).

*Etc.*

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# FONT FORMATS

## Types

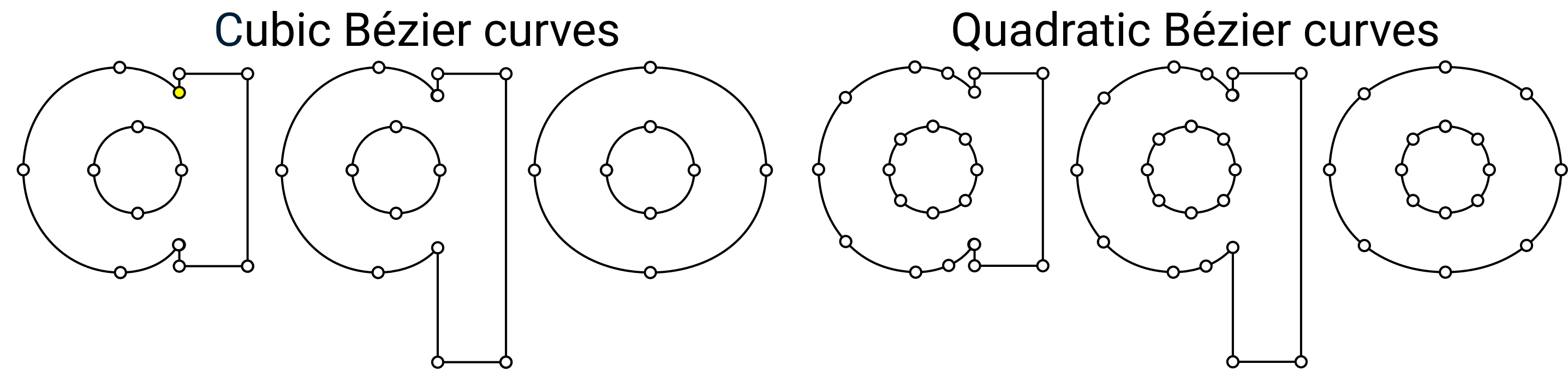
Type 1 (.pfm, .pfb): glyphs are described with cubic Bézier curves

True Type (.ttf): glyphs are described with quadratic Bézier curves

Open Type (.otf): glyphs are described with quadratic or cubic Bézier curves

Web Open Font (.woff): compressed TrueType or Open Type

And many more actually...



PostScript Outlines TrueType Outlines

TTF

OTF

WOFF

The Missing Guide to Font Formats  
SVG

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

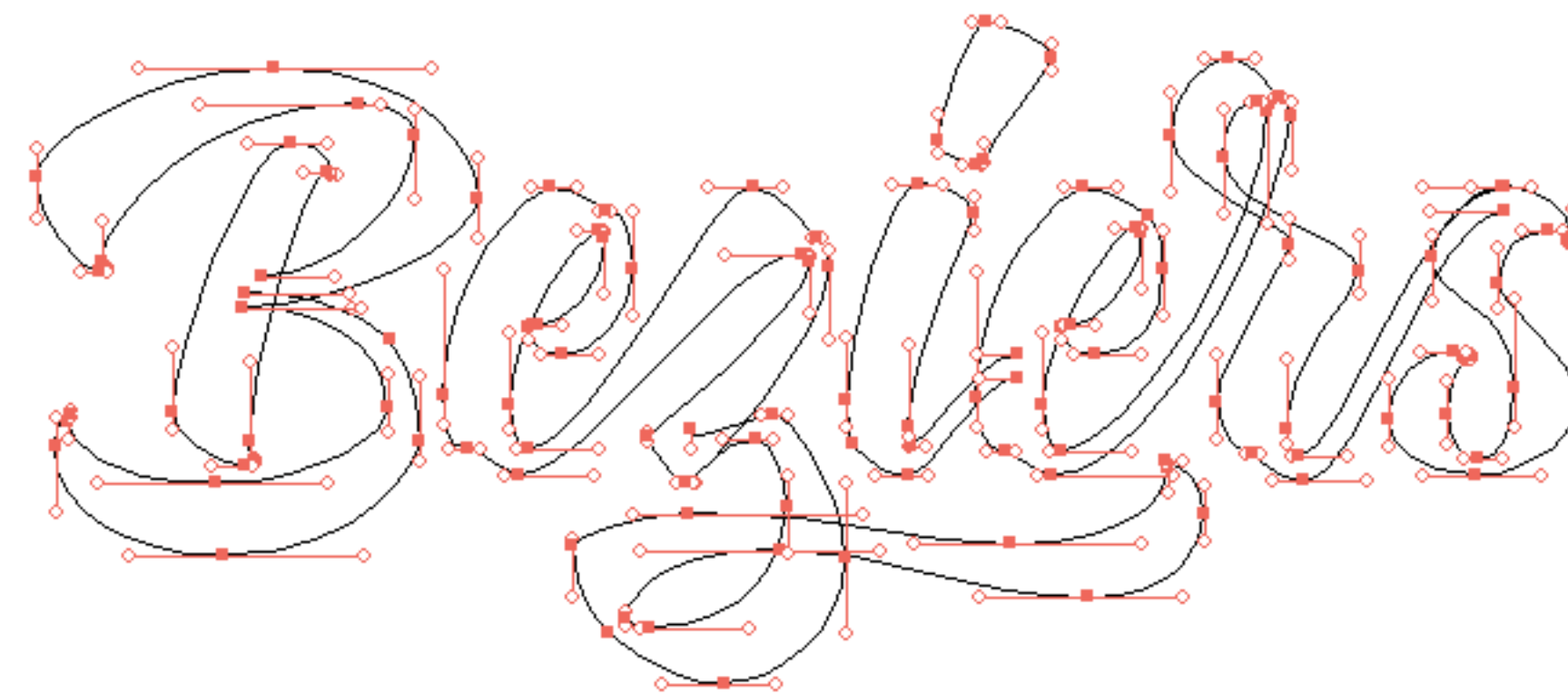
CONCLUSION

- Beyond this talk
- Questions & answers

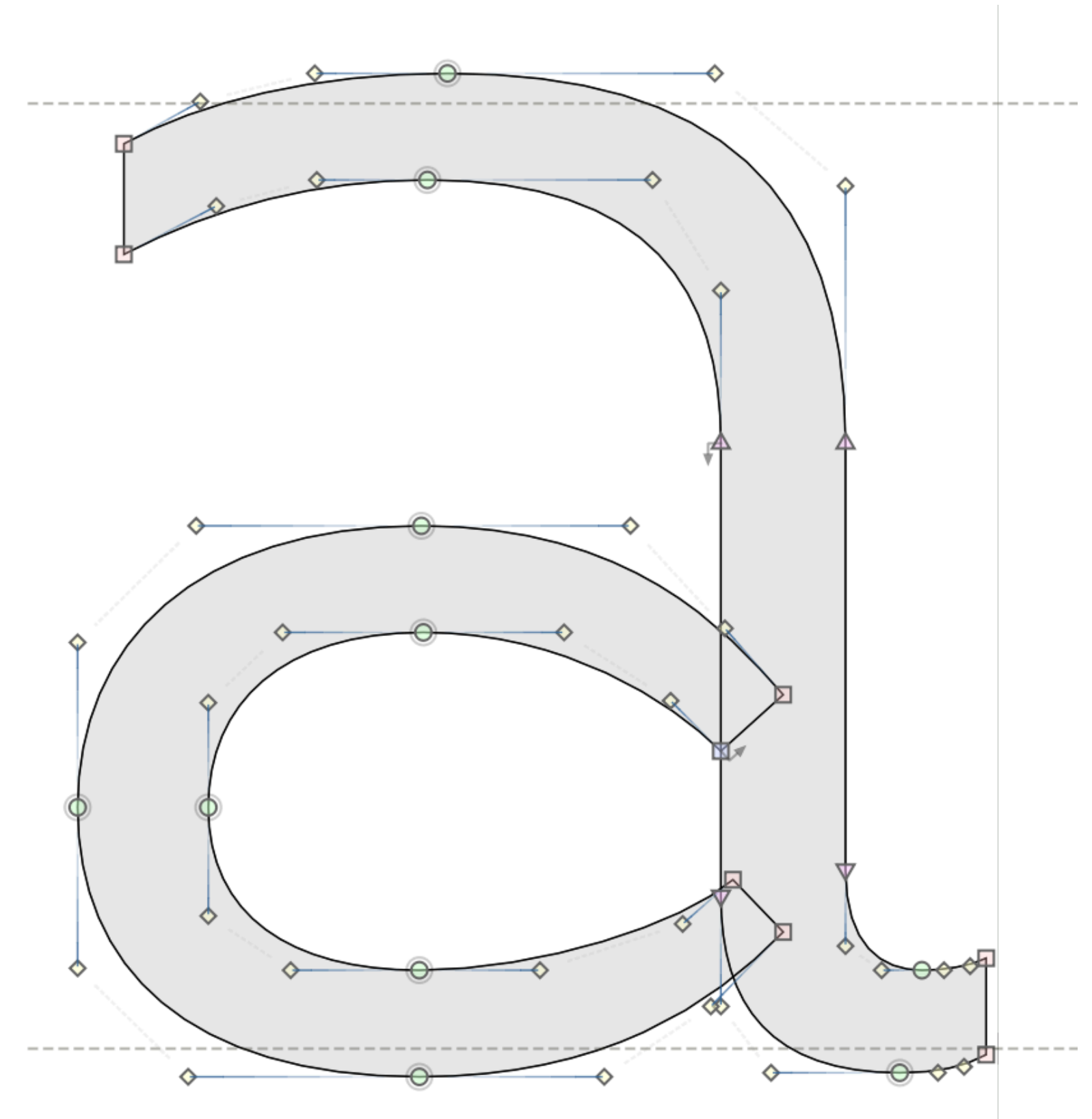
# ANATOMY OF A GLYPH

Vector fonts are collections of vector images, consisting of lines and curves defining the boundary of glyphs (wikipedia). Type 1 and Type 3 Postscript fonts are described with cubic Bézier curves. Truetype fonts are described with quadratic Bézier curves.

Beziérs



Images by The Australian Graphic Supply Co



Images by FontLab

## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers

# FONT TABLES

## Open Type Tables

BASE: Baseline data

CMAP: Character to glyph mapping

GDEF: Glyph definition data

GSUB: Glyph substitution data

Single: Replaces one glyph with one glyph.

Multiple: Replaces one glyph with more than one glyph.

Alternate: Replaces one glyph with one of many glyphs.

Ligature: Replaces multiple glyphs with one glyph.

Context: Replaces one or more glyphs in context.

Chaining: Replaces one or more glyphs in chained context.

GPOS: Glyph positioning data

JSTF: Justification data

and many more...

## Libraries

**FreeType** (standard support without text shaping)

→ <https://www.freetype.org>

**HarfBuzz** (advanced support with text shaping)

→ <https://harfbuzz.github.io>



# TEXT RENDERING PIPELINE

## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

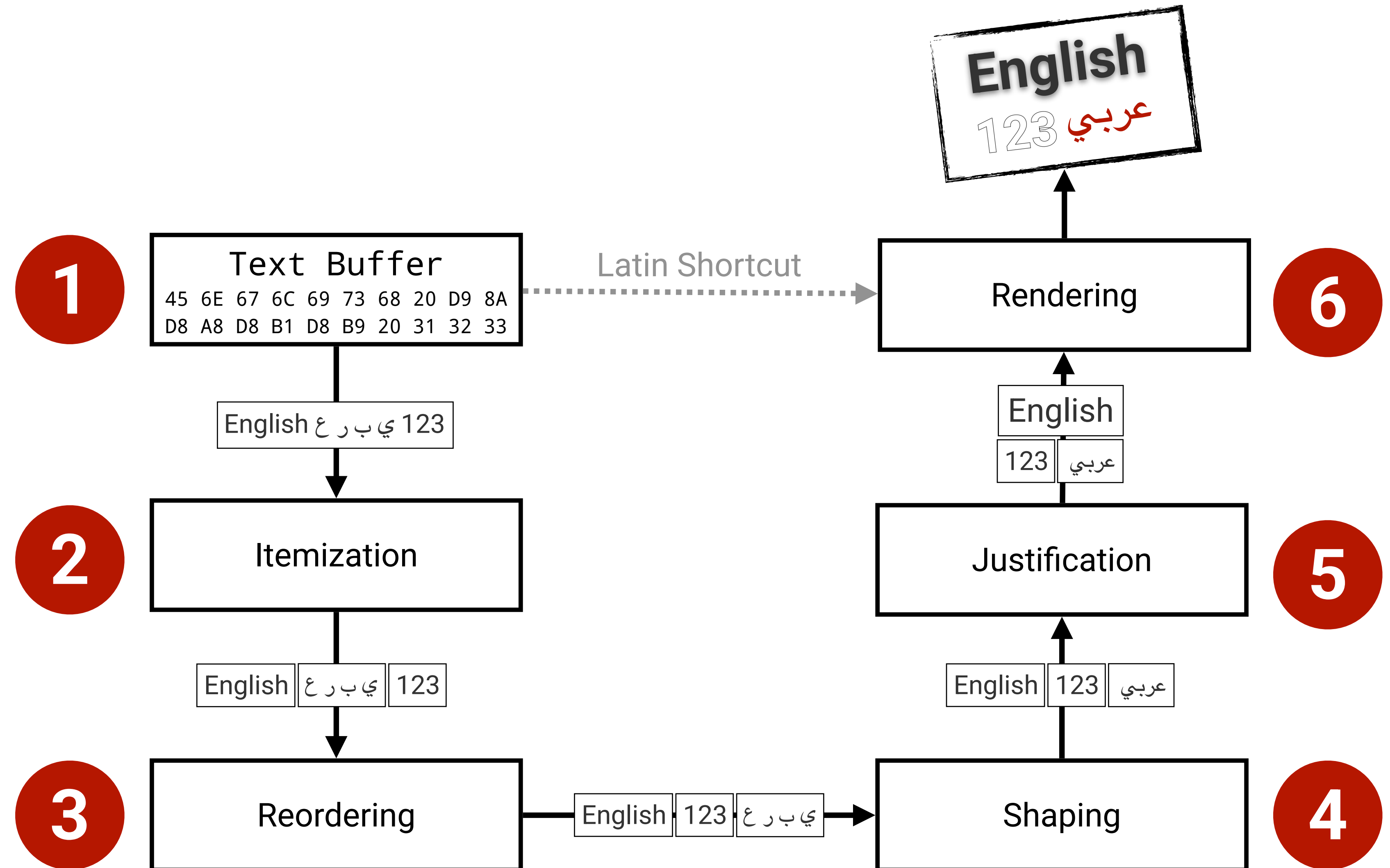
- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers



## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers

# PART I

Near the end of the last century (1997), Mark Kilgard introduced a simple OpenGL-based API for texture mapped text.

The method packed many rasterized glyphs into a single alpha-only texture map and used a lookup table to assign texture coordinates to a quadrilateral to extract a glyph when rendering.



## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers

# 2D RASTERIZATION

## Same input, different outputs

コンピュータによる CJKV (中国語、日本語、韓国語、ベトナム語) テキスト処理は、多くの謎や複雑な仕組み (intrigue) に包まれている。そこにはたしかにたくらみ (intrigue) があることは認めるが、それよりも謎の多いことのほうが問題である。

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sit amet dui. Nam sapien. Fusce vestibulum ornare metus. Maecenas ligula orci, consequat vitae, dictum nec, lacinia non, elit. Aliquam iaculis molestie neque. Maecenas suscipit felis ut pede convallis malesuada. Aliquam erat volutpat. Nunc pulvinar condimentum nunc. Donec ac sem vel leo bibendum aliquam. Pellentesque habitant

### FreeType Native CFF Rasterizer

コンピュータによる CJKV (中国語、日本語、韓国語、ベトナム語) テキスト処理は、多くの謎や複雑な仕組み (intrigue) に包まれている。そこにはたしかにたくらみ (intrigue) があることは認めるが、それよりも謎の多いことのほうが問題である。

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sit amet dui. Nam sapien. Fusce vestibulum ornare metus. Maecenas ligula orci, consequat vitae, dictum nec, lacinia non, elit. Aliquam iaculis molestie neque. Maecenas suscipit felis ut pede convallis malesuada. Aliquam erat volutpat. Nunc pulvinar condimentum nunc. Donec ac sem vel leo bibendum aliquam. Pellentesque habitant

### FreeType Light Auto Hint Rasterizer

コンピュータによる CJKV (中国語、日本語、韓国語、ベトナム語) テキスト処理は、多くの謎や複雑な仕組み (intrigue) に包まれている。そこにはたしかにたくらみ (intrigue) があることは認めるが、それよりも謎の多いことのほうが問題である。

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sit amet dui. Nam sapien. Fusce vestibulum ornare metus. Maecenas ligula orci, consequat vitae, dictum nec, lacinia non, elit. Aliquam iaculis molestie neque. Maecenas suscipit felis ut pede convallis malesuada. Aliquam erat volutpat. Nunc pulvinar condimentum nunc. Donec ac sem vel leo bibendum aliquam. Pellentesque habitant

### FreeType using the New Adobe CFF Rasterizer (2013)

## Quartz Rendering

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis facilisis varius elementum. Proin cursus posuere mollis. Vivamus id justo nec massa convallis dapibus. Praesent a arcu augue, pretium commodo elit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

## ClearType Rendering

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis facilisis varius elementum. Proin cursus posuere mollis. Vivamus id justo nec massa convallis dapibus. Praesent a arcu augue, pretium commodo elit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

# SUBPIXEL ZOO (GEOMETRIAN.COM)

## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

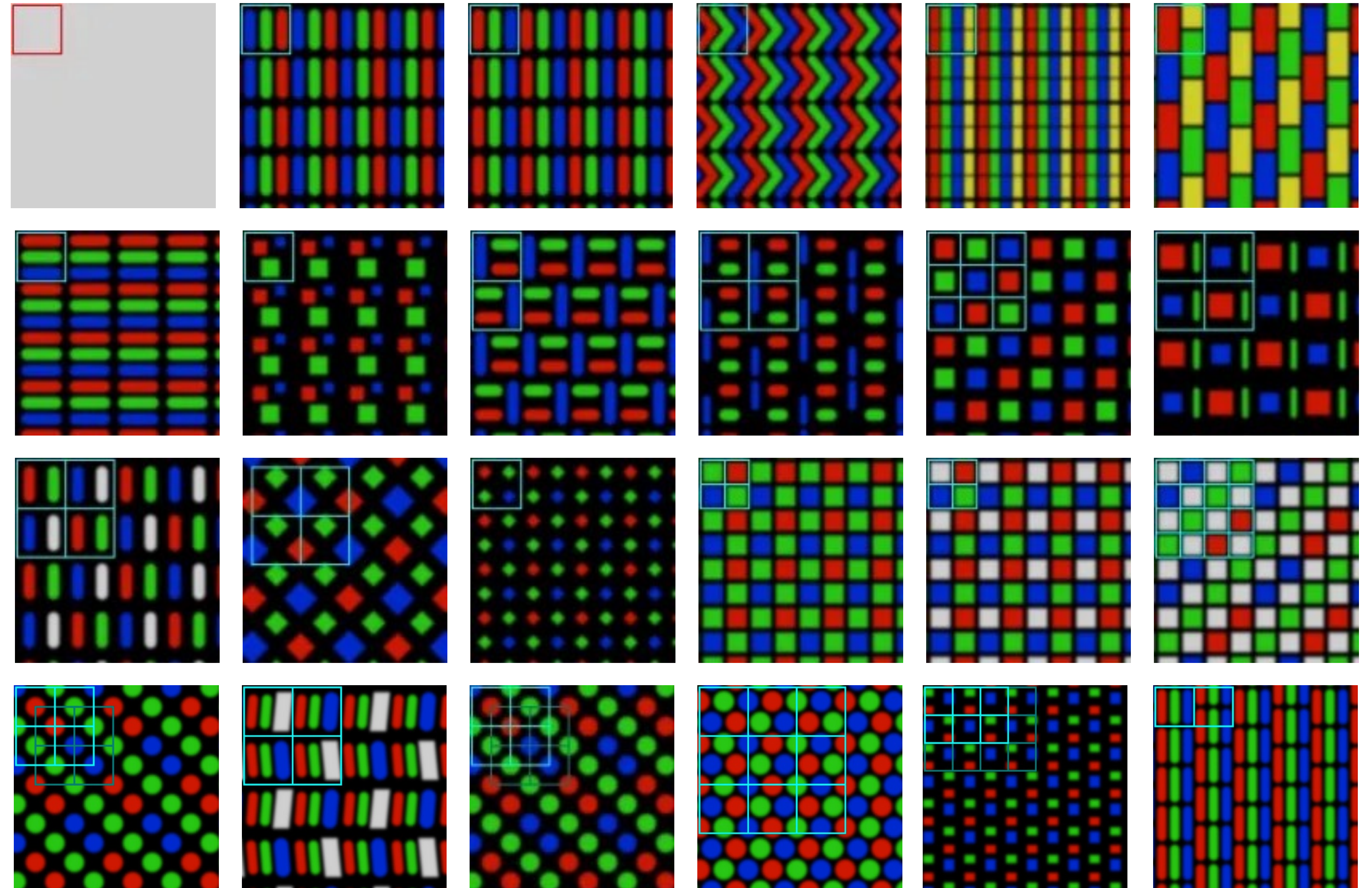
- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers





INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# TEXTURE FONT I

## Kilgard (1997) A Simple OpenGL-based API for Texture Mapped Text

*Texture mapping is well suited for rendering text because textures can be rendered quickly with current 3D hardware and even via clever programming of today's fast CPUs. Textures can be stretched, rotated, scaled, and even projected (assuming the texture mapping is perspective correct) so that texture mapped text looks reasonable in 3D scenes.*



<https://github.com/rougier/freetype-gl>

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

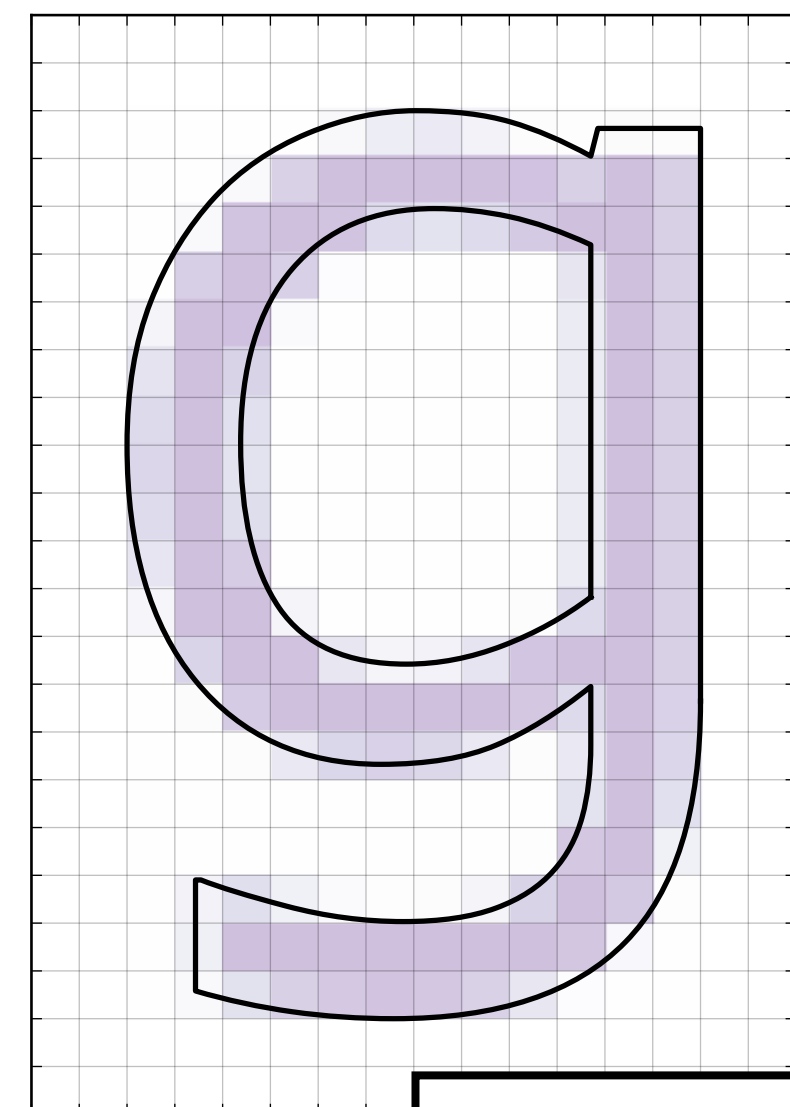
CONCLUSION

- Beyond this talk
- Questions & answers

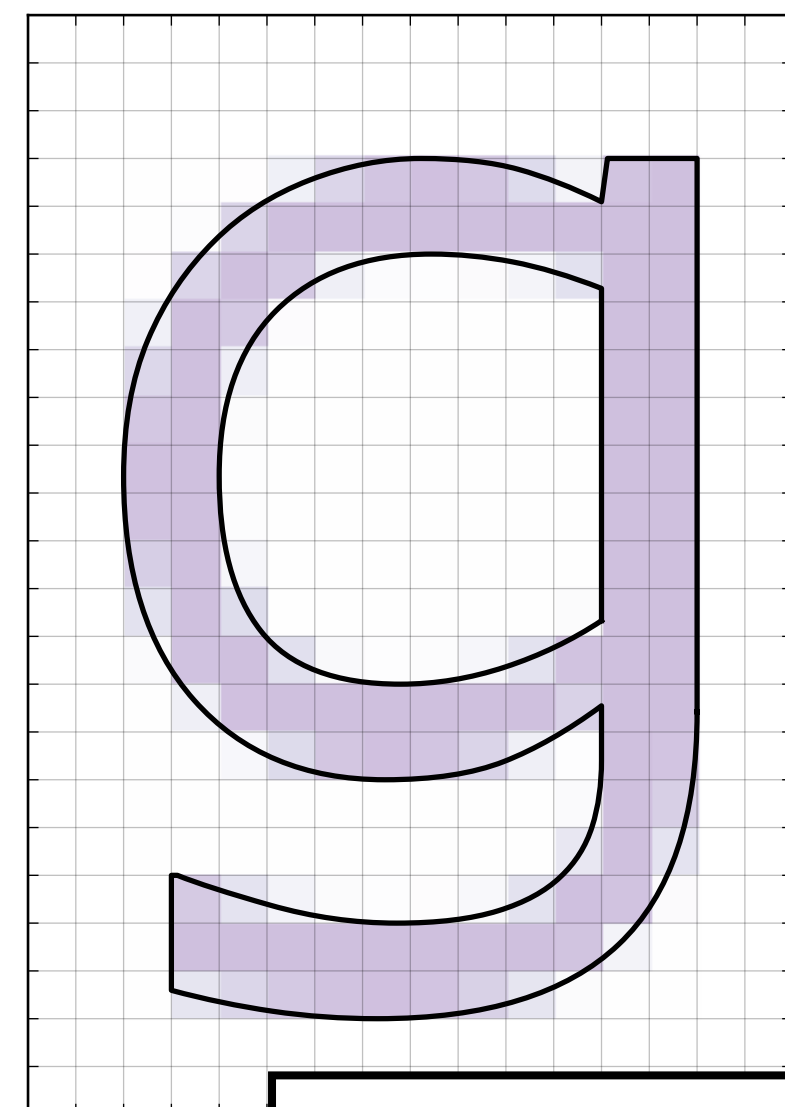
# TEXTURE FONT II

## Rougier (2013) Higher Quality 2D Text Rendering

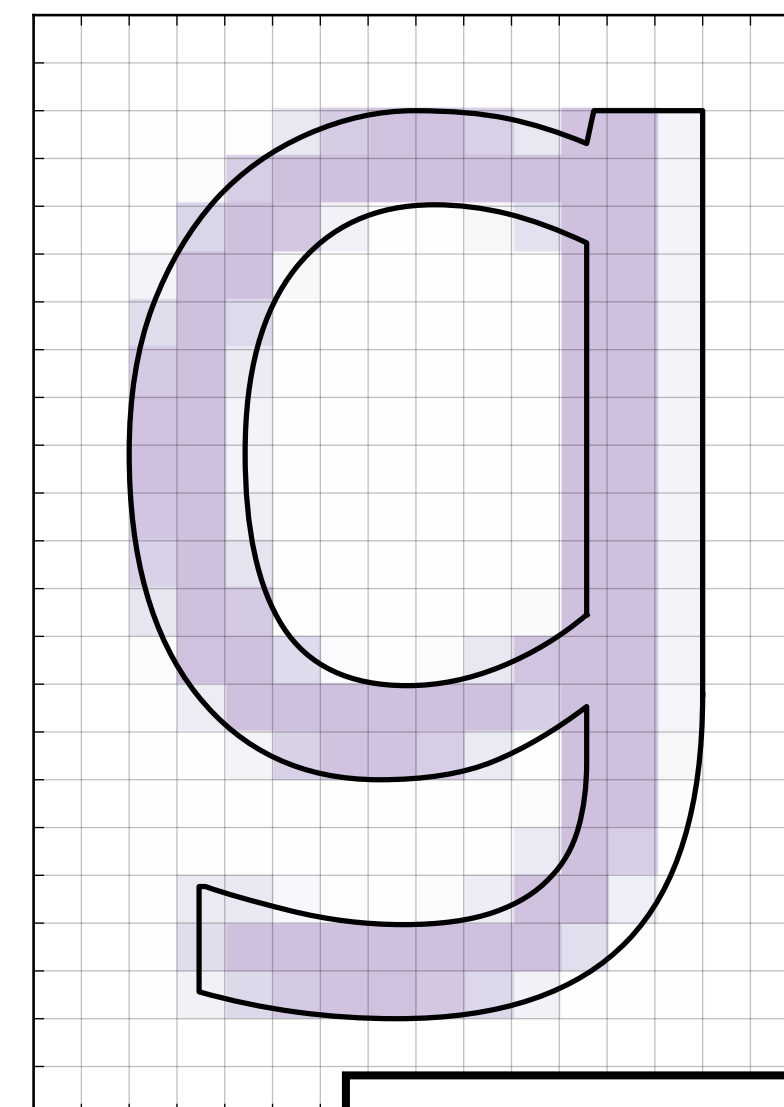
When displaying text on low-resolution devices ( $DPI < 150$ ), one typically has to decide if one wants to respect the pixel grid (e.g., ClearType technology / Microsoft / native hinting) for crisp rendering or, to privilege glyph shapes (Quartz technology / Apple / no hinting) at the cost of blurring. There is, however, a third way that may combine the best of the two technologies (vertical hinting).



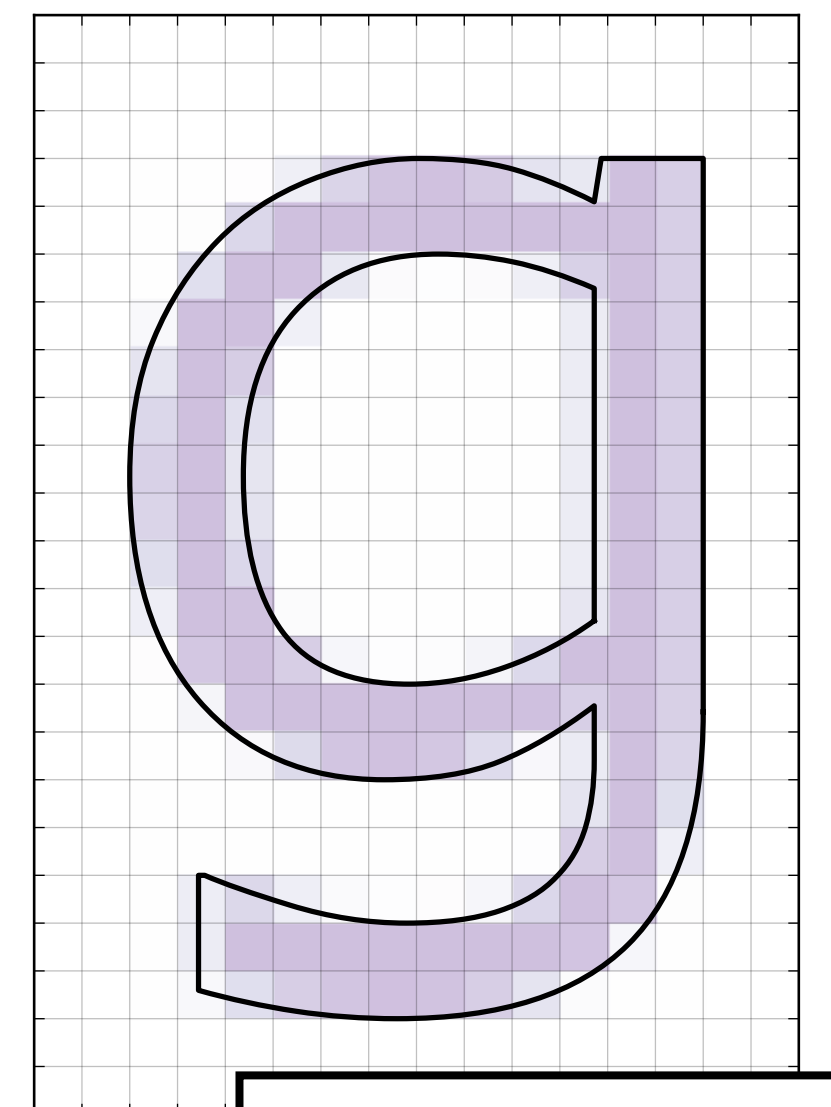
No hinting



Native hinting



Auto hinting



Vertical hinting

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

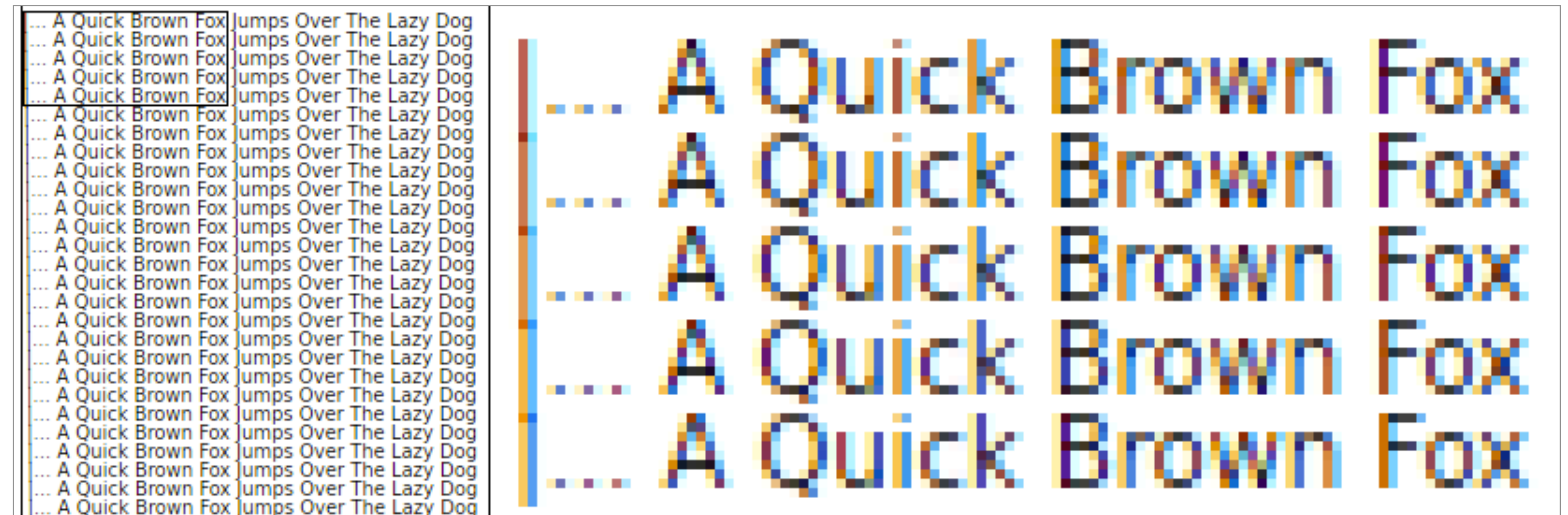
CONCLUSION

- Beyond this talk
- Questions & answers

# TEXTURE FONT II

## Rougie (2013) Higher Quality 2D Text Rendering

- Use horizontal RGB sub-pixel anti-aliasing for LCD flat panels.
- Use vertical hinting only and completely discard the horizontal one.
- Use accurate glyph advance values from unhinted glyph
- Use accurate, high resolution values from the kerning table.



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

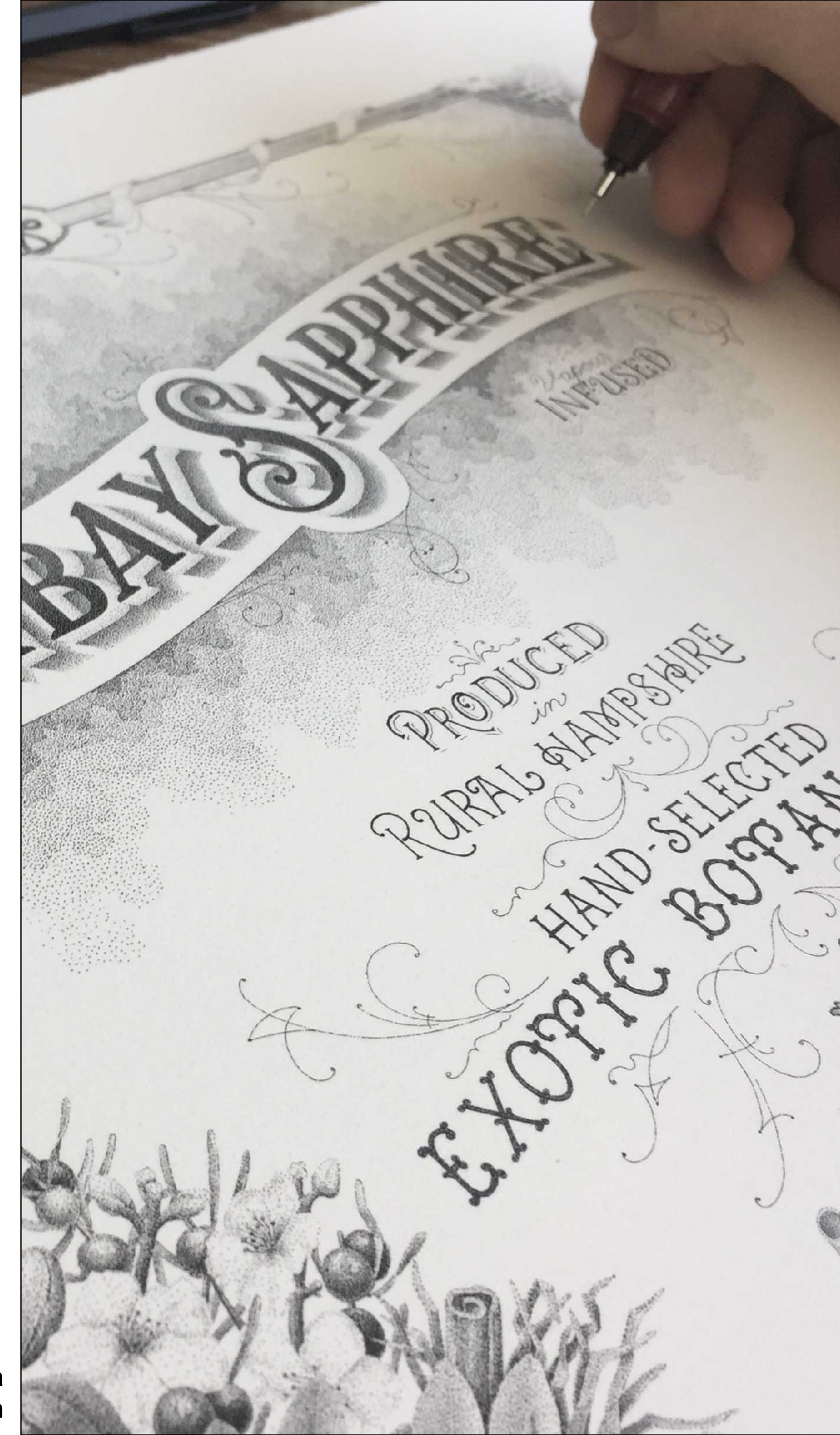
CONCLUSION

- Beyond this talk
- Questions & answers

# PART II

Distance based rendering takes advantage of a signed distance function (that can be approximated) to compute the individual coverage for each pixel.

The coverage can be selected to implement various effects (stroke, thinner, thicker, shadow, etc)



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

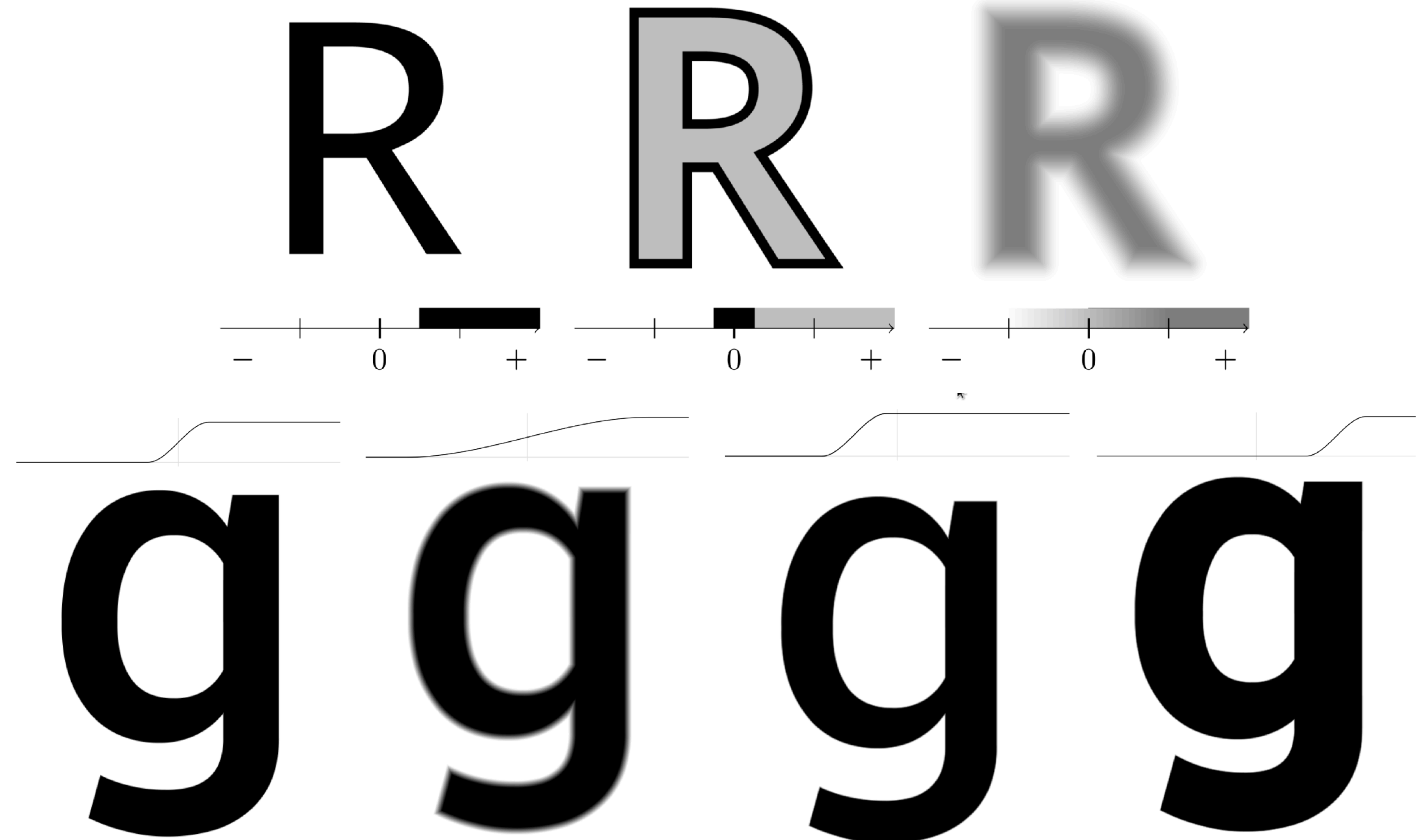
- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# DISTANCE BASED RENDERING

Distance based rendering takes advantage of a signed distance function (that can be approximated) to compute the individual coverage for each pixel. The coverage can be selected to implement various effects (stroke, thinner, thicker, shadow, etc)



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# COMPUTING SDF

## Gustavson & Strand (2011) Anti-Aliased Euclidean distance transform

*We present a modified distance measure for use with distance transforms of anti-aliased, area sampled grayscale images of arbitrary binary contours. The modified measure can be used in any vector-propagation Euclidean distance transform. Our test implementation in the traditional SSE8 algorithm shows a considerable improvement in accuracy and homogeneity of the distance field compared to a traditional binary image transform.*

See also <http://contourtextures.wikidot.com>



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

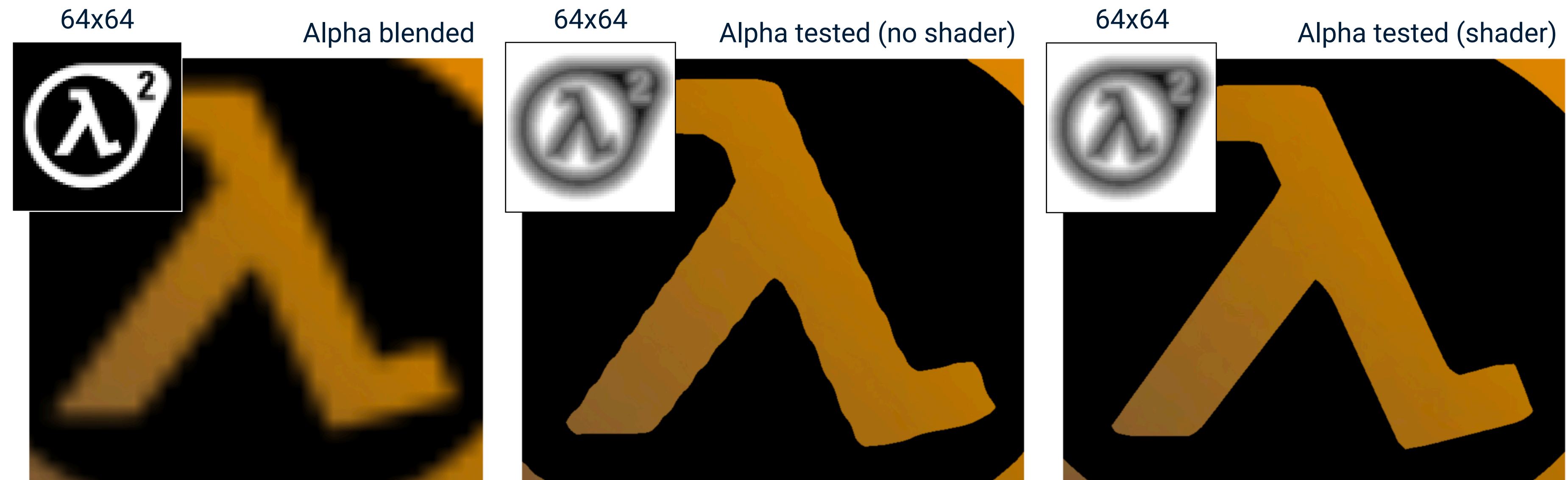
CONCLUSION

- Beyond this talk
- Questions & answers

# SINGLE CHANNEL SDF

## Green (2007) Improved Alpha-Tested Magnification for Vector Textures

*A distance field is generated from a high resolution image, and then stored into a channel of a lower-resolution texture. In the simplest case, this texture can then be rendered simply by using the alpha-testing and alpha-thresholding feature of modern GPUs, without a custom shader. This allows the technique to be used on even the lowest-end 3D graphics hardware. With the use of programmable shading, the technique is extended to perform various special effect renderings, including soft edges, outlining, drop shadows, multi-colored images, and sharp corners.*



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

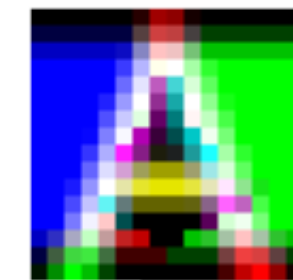
- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# SDF: MULTIPLE CHANNELS

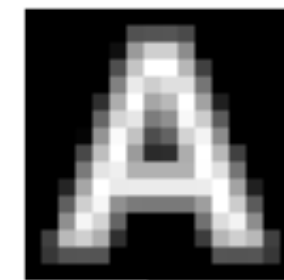
## Chlumsky (2015) Shape Decomposition for Multi-channel Distance Fields



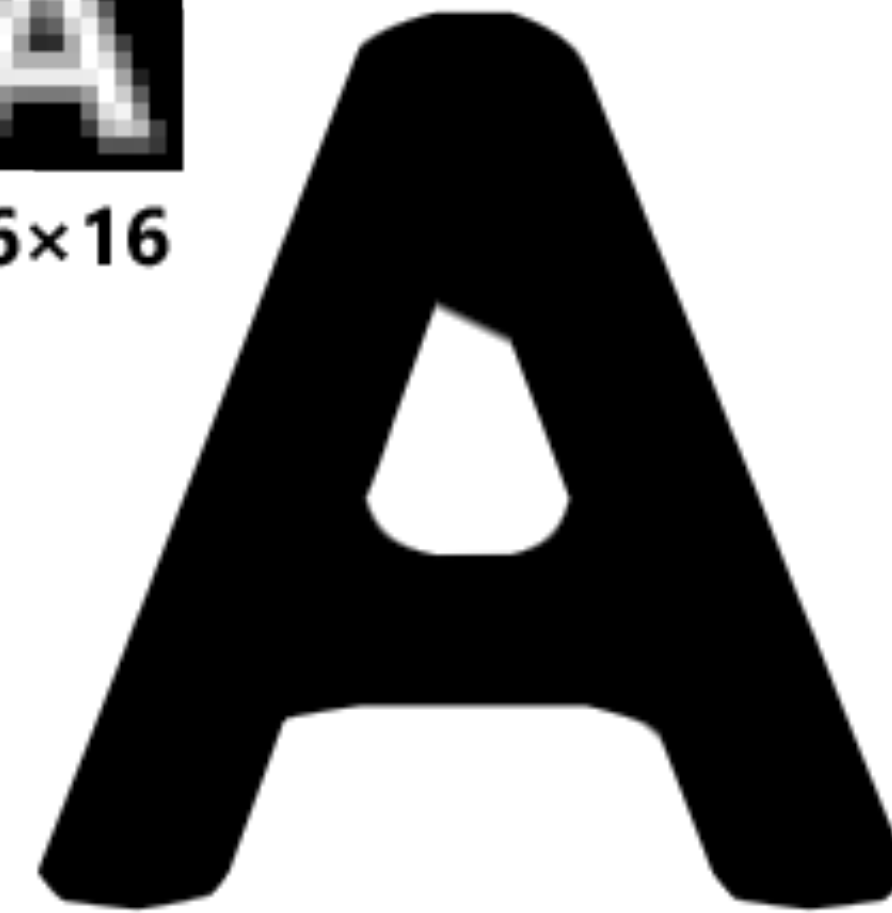
16x16



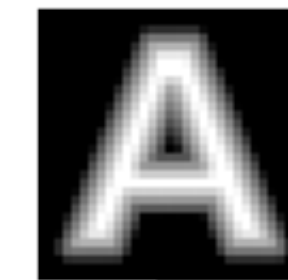
Edge Coloring



16x16



Plane partitioning

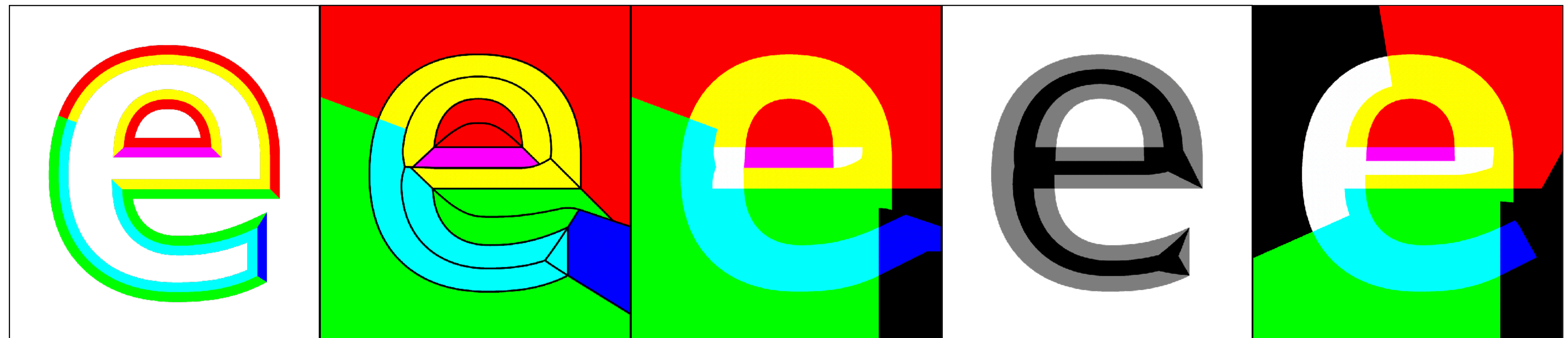


32x32



Inner padding

Padded Decomposition



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

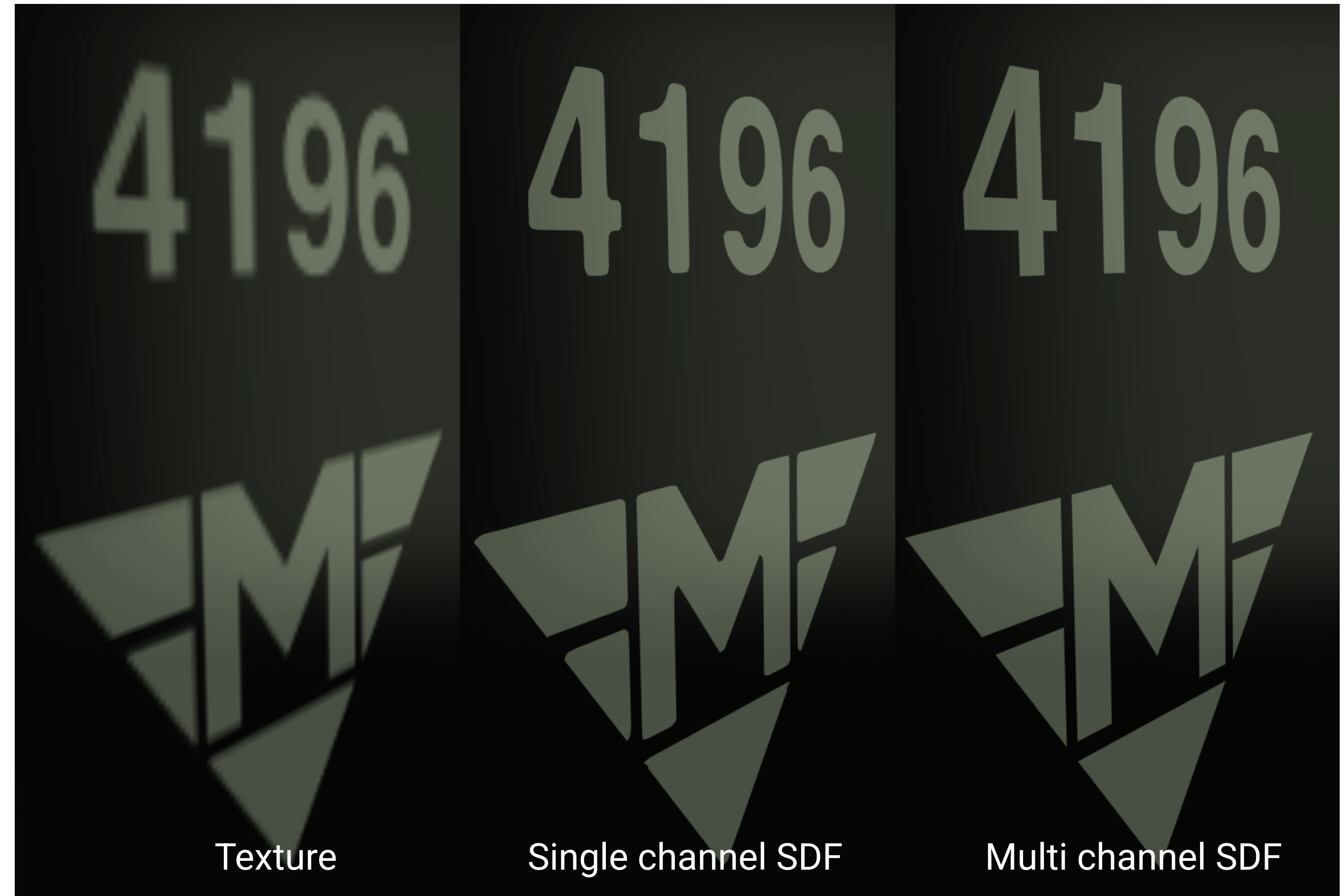
- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# SDF: MULTIPLE CHANNELS

Chlumsky (2015) Shape Decomposition for Multi-channel Distance Fields



Texture

Single channel SDF

Multi channel SDF

## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers

# PART III

Charles Loop and Jim Blinn introduced in 2005 a new approach for resolution-independent rendering of quadratic and cubic spline curves.

By tessellating a glyph the proper way, they offered de facto a method for resolution independent rendering of a glyph with good rendering quality.



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

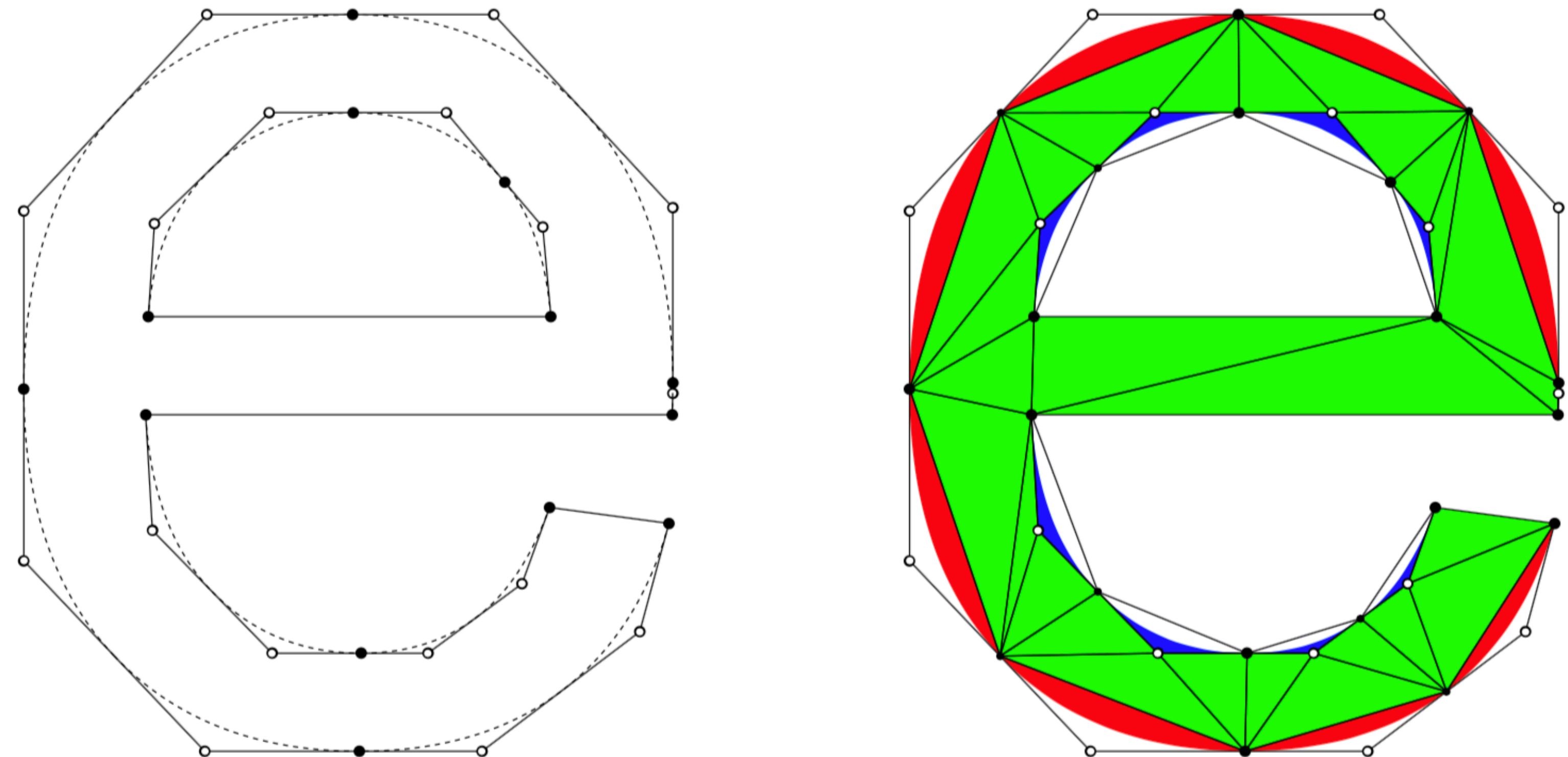
CONCLUSION

- Beyond this talk
- Questions & answers

# GPU FRIENDLY

## Loop & Blinn (2005) Resolution Independent Curve Rendering

*We present a method for resolution independent rendering of paths and bounded regions, defined by quadratic and cubic spline curves, that leverages the parallelism of programmable graphics hardware to achieve high performance. Our result is a mechanism for rendering vector geometry that has the following properties: resolution independence, compact geometric representation, high performance.*



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

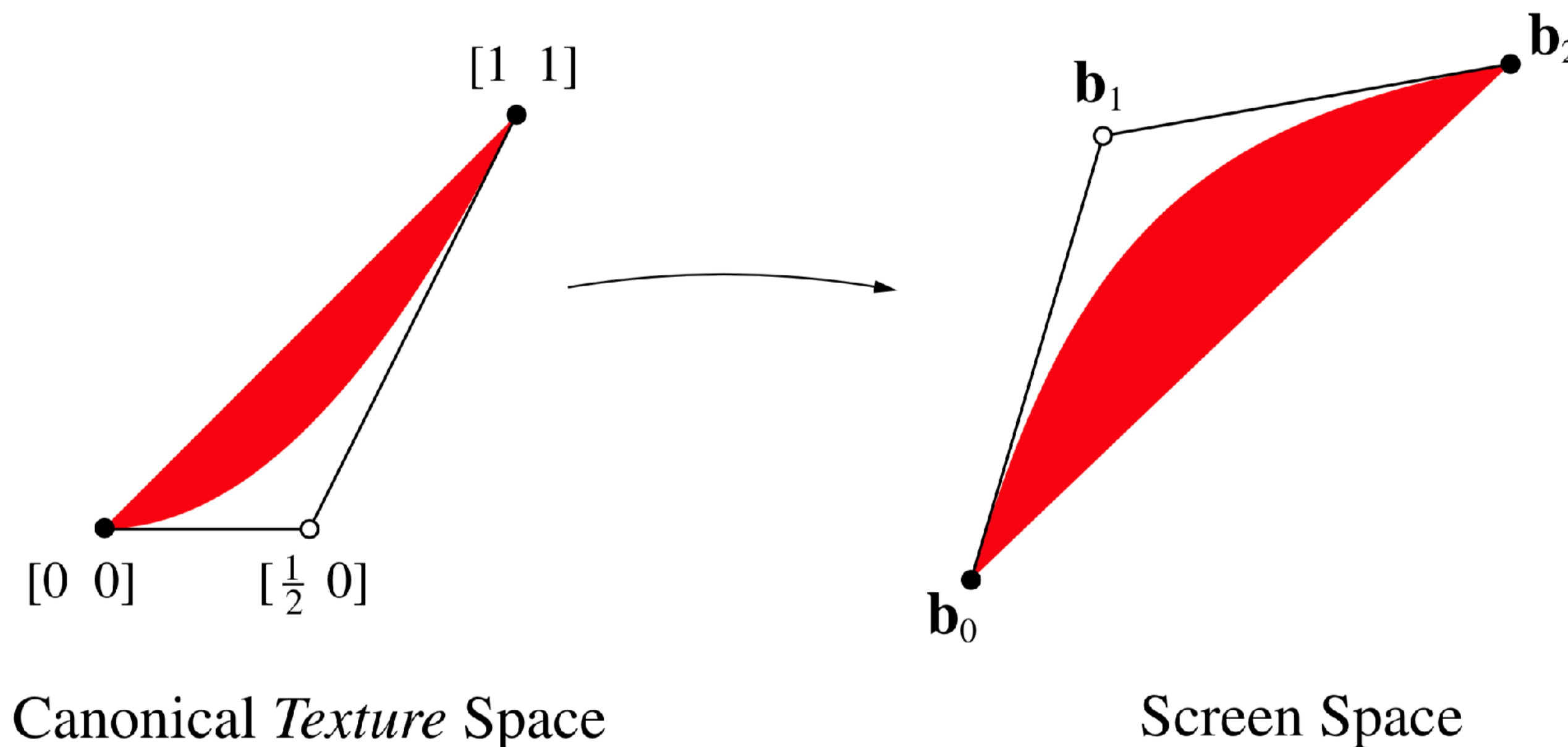
CONCLUSION

- Beyond this talk
- Questions & answers

# GPU FRIENDLY

## Loop & Blinn (2005) Resolution Independent Curve Rendering

We determine if the pixel is inside or outside the curve by evaluating  $f(u, v) = u^2 - v$  in a pixel shader program. If  $f(u, v) < 0$  then the pixel is inside the curve, otherwise it is outside.



See also

- ["Rendering Vector Art on the GPU"](#) (Charles Loop & Jim Blinn, GPU Gems 3, Chapter 25)
- ["Easy Scalable Text Rendering on the GPU"](#) (Evan Wallace Medium)

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

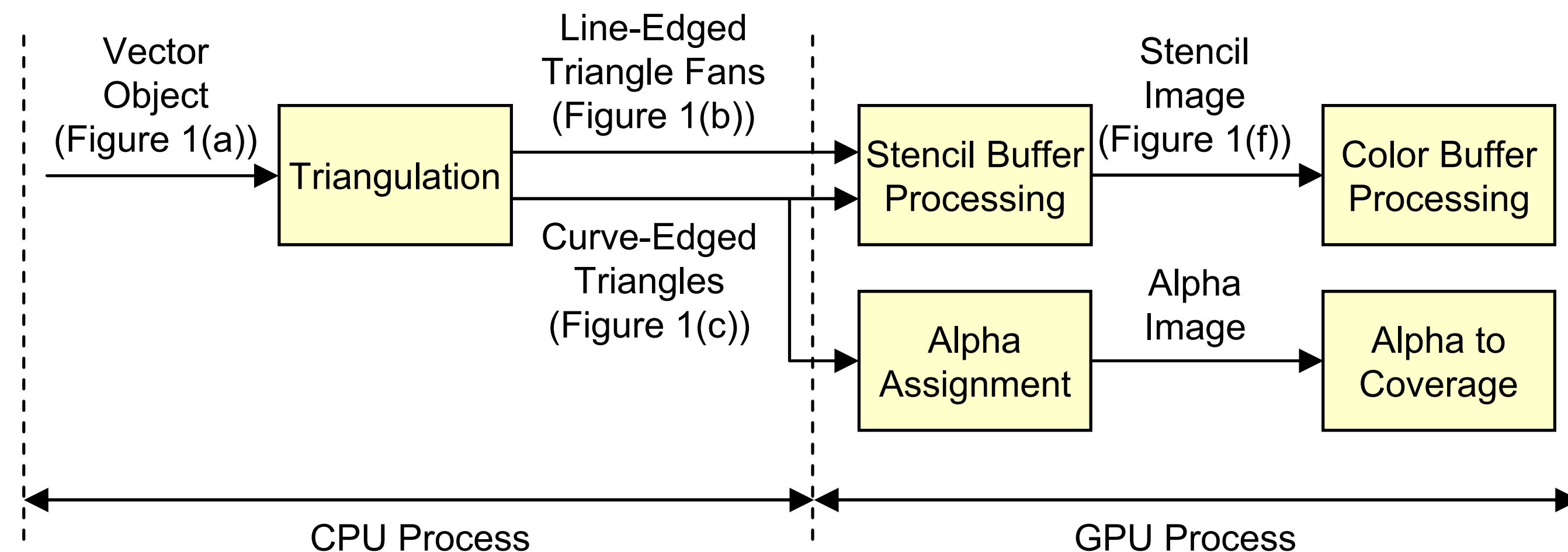
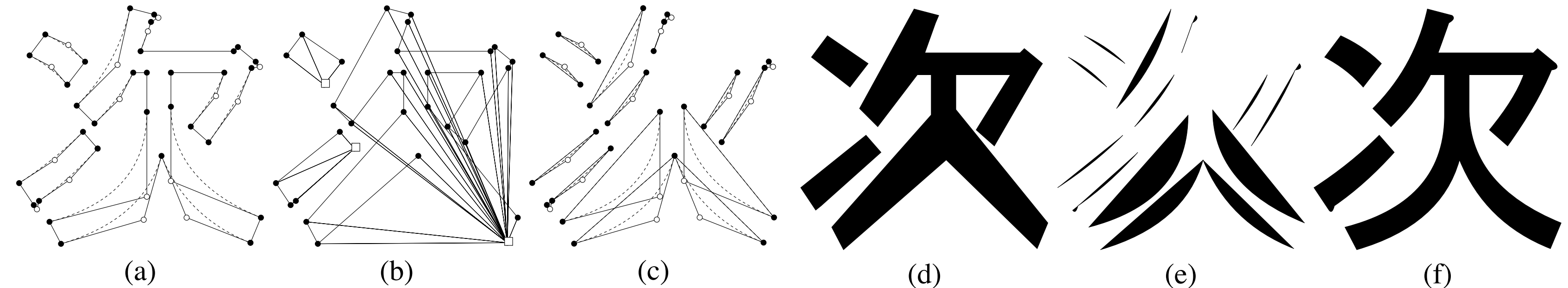
CONCLUSION

- Beyond this talk
- Questions & answers

# GPU FRIENDLY

## Kokojima et al (2006) Resolution Independent Rendering of Deformable Vector Objects

*This sketch presents a new method for resolution independent rendering of vector images suitable for programmable graphics hardware. We have enhanced a previous method [Loop and Blinn 2005] by using a stencil buffer and transparency multisampling.*



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

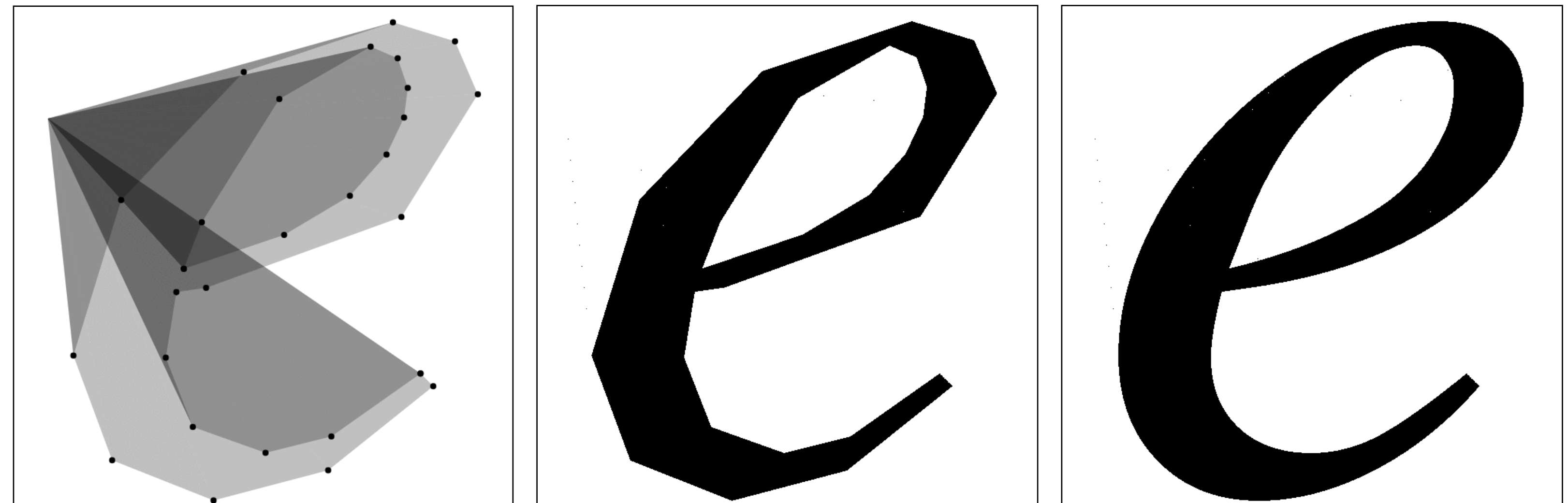
CONCLUSION

- Beyond this talk
- Questions & answers

# GPU FRIENDLY

## Kokojima et al (2006) Resolution Independent Rendering of Deformable Vector Objects

*This sketch presents a new method for resolution independent rendering of vector images suitable for programmable graphics hardware. We have enhanced a previous method [Loop and Blinn 2005] by using a stencil buffer and transparency multisampling.*



Animations by Evan Wallace (Easy Scalable Text Rendering on the GPU)

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

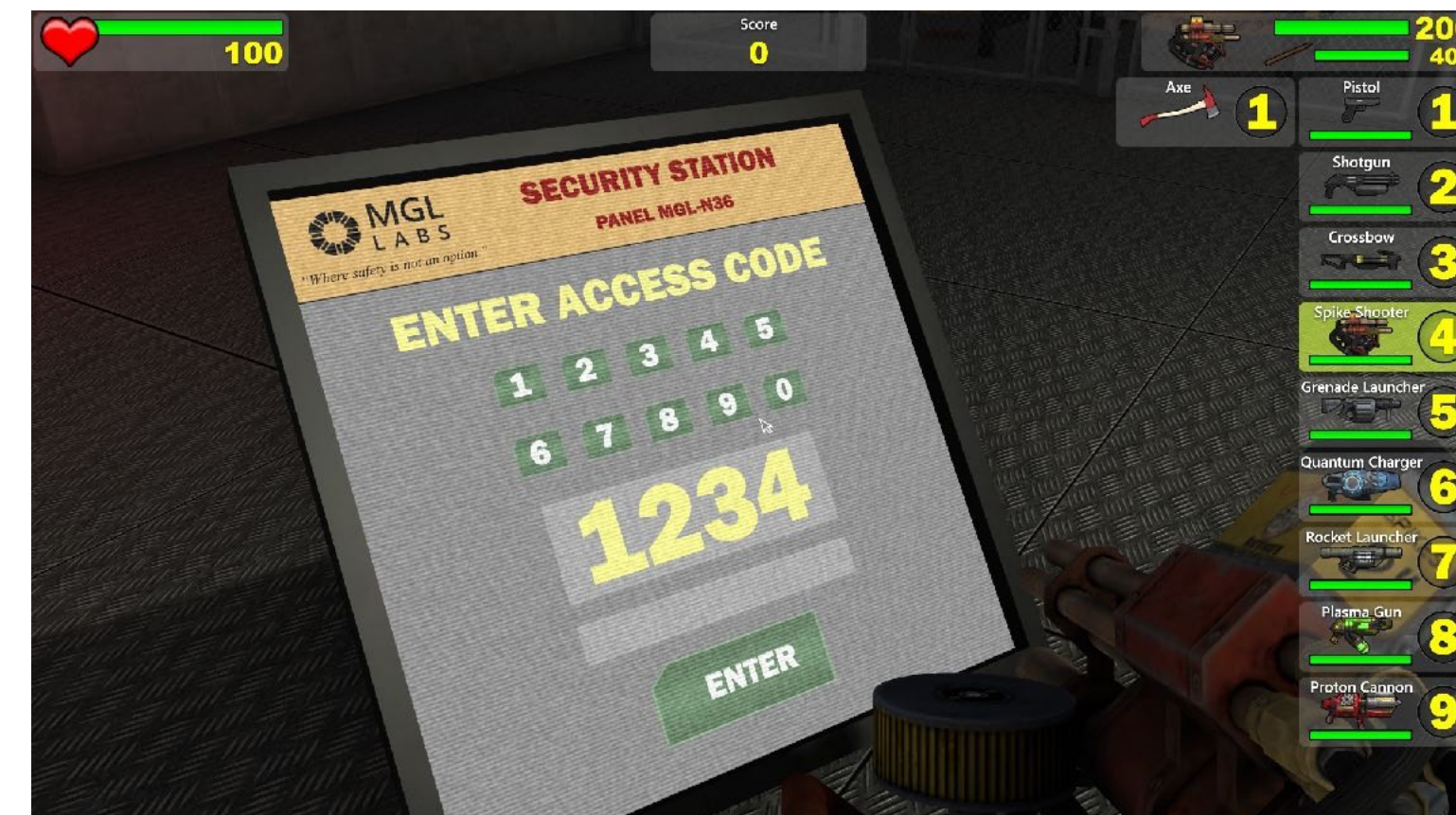
# GPU ONLY

## Lengyel (2017) GPU-Centered Font Rendering Directly from Glyph Outlines

*This paper describes a method for rendering antialiased text directly from glyph outline data on the GPU without the use of any precomputed texture images or distance fields. This capability is valuable for text displayed inside a 3D scene because, in addition to a perspective projection, the transform applied to the text is constantly changing with a dynamic camera view. Our method overcomes numerical precision problems that produced artefacts in previously published techniques and promotes high GPU utilization with an implementation that naturally avoids divergent branching.*

See demo at [sluglibrary.com](http://sluglibrary.com)

See also [GPU-Centered font rendering](#) & [GPU font rendering](#)



Font	Sample	Complexity	Time
Arial	ABCDEFGFG	20	1.1 ms
Centaur	ABCDEFGFG	48	1.3 ms
Halloween	ABCDEFGFG	72	3.8 ms
Wildwood	ABCDEFGFG	546	13.3 ms

INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

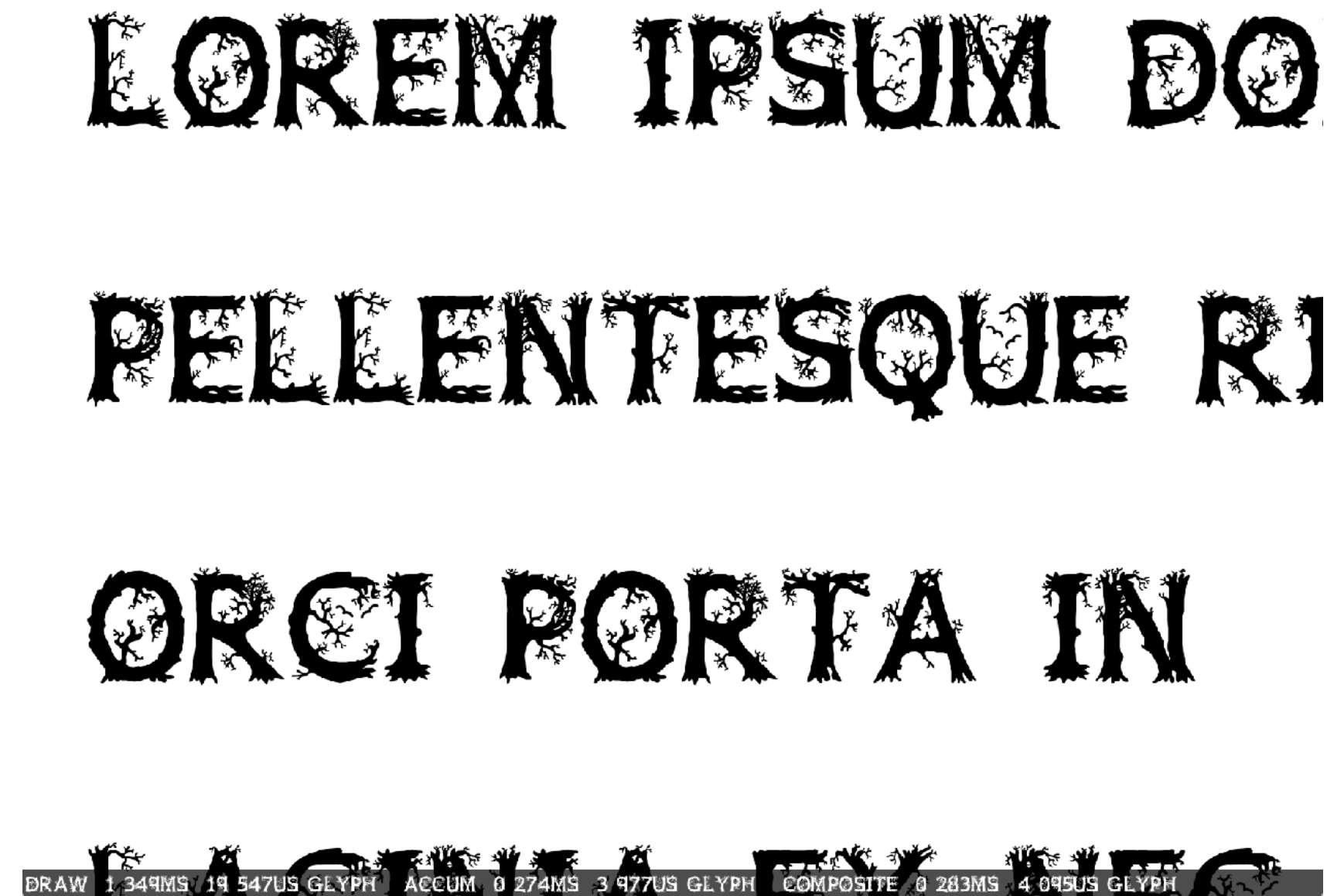
CONCLUSION

- Beyond this talk
- Questions & answers

# GPU ONLY

## Walton (2017) Pathfinder, a fast GPU-based font rasterizer in Rust

*“Today I’m pleased to announce Pathfinder, a Rust library for OpenType font rendering. The goal is nothing less than to be the fastest vector graphics renderer in existence, and the results so far are extremely encouraging. Not only is it very fast according to the traditional metric of raw rasterization performance, it’s practical, featuring very low setup time (end-to-end time superior to the best CPU rasterizers), best-in-class rasterization performance even at small glyph sizes, minimal memory consumption (both on CPU and GPU), compatibility with existing font formats, portability to most graphics hardware manufactured in the past five years (DirectX 10 level), and security/safety.”*



## INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

## PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

## PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

## PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

## CONCLUSION

- Beyond this talk
- Questions & answers

# CONCLUSION

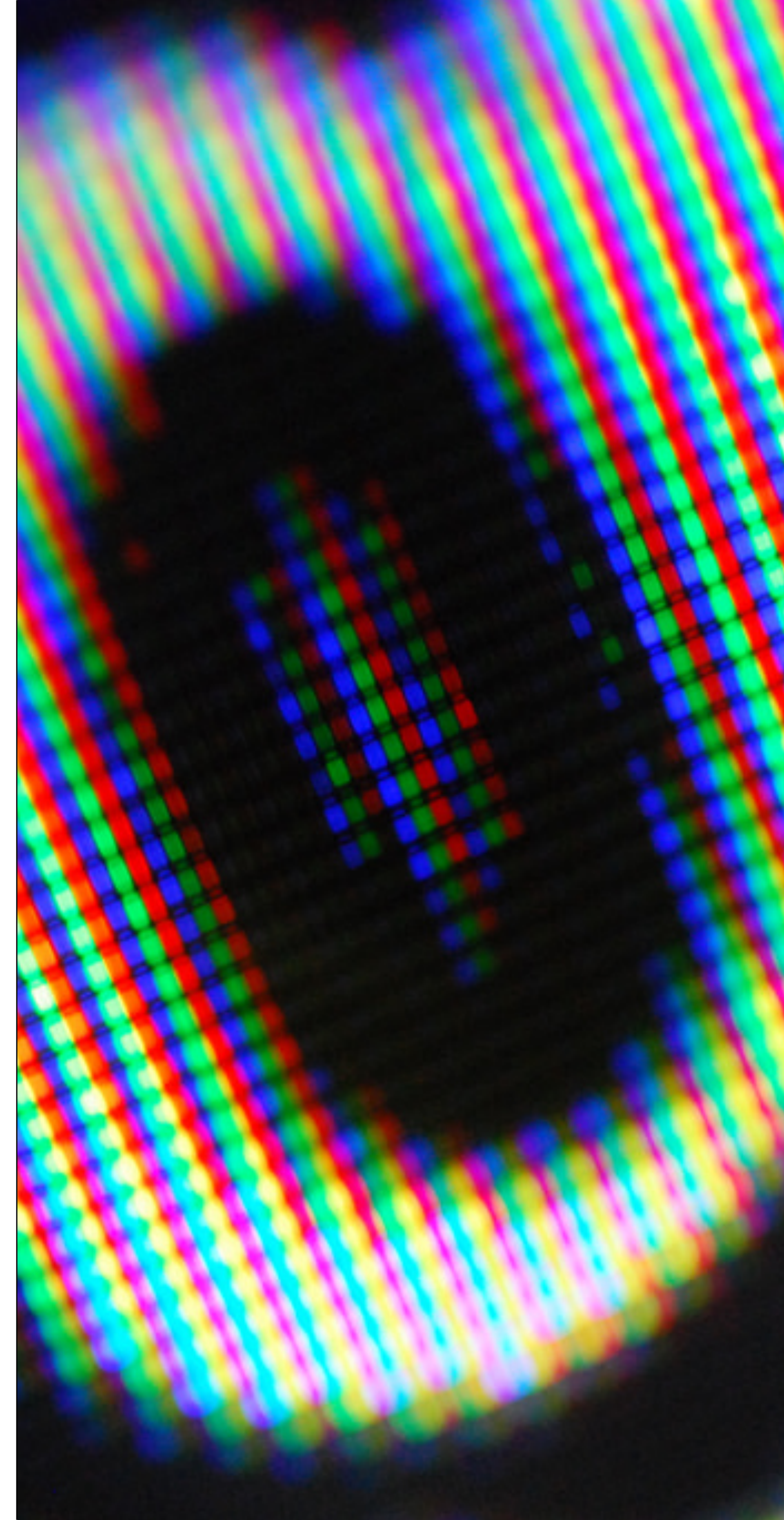
Many available techniques. Choice is dependent of usage (2D or 3D, dynamic or static, memory vs speed vs accuracy, etc).

PathFinder 3, Slug library and MSDF seem to be the main players in 2025. Loop & Blinn still competitive.

Hinting and anti-aliasing still needed until we (all) get screen with dpi > 600.

Complex text layout is (really, really) difficult, you don't want to do it yourself: [harfbuzz.github.io](https://harfbuzz.github.io)

Lot of patents around !



INTRODUCTION

- Digital Typography
- Font Types & Formats
- Text Rendering Pipeline

PART I : TEXTURE BASED

- Rasterization
- Fast & Versatile (but ugly)
- Fast & Beautiful (but only 2D)

PART II : DISTANCE BASED

- Signed Distance Fields
- Single Channel
- Arc approximation
- Multiple Channels

PART III : GEOMETRY BASED

- Bézier curves & glyphs
- GPU friendly
- GPU only

CONCLUSION

- Beyond this talk
- Questions & answers

# QUESTIONS?

- Where to find good fonts? [dafont.com](http://dafont.com), [fontquirrel.com](http://fontquirrel.com)
- When does cleartype patent end ? Not clear (ha ha)
- Are hinting and aliasing still needed? [Yes](#)
- What are the most complete font families? [Noto](#), [DejaVu](#)
- How many languages in the world? Around 7000
- How many scripts in latest unicode (17.0)? 172
- How many characters in latest unicode (17.0)? 159,801
- Is Calibri woke ? Trump thinks so yes (really).
- **What are variable fonts? See Bernd's talk tomorrow!**
- Are there any open source tools to design fonts? [FontForge](#)
- Why "ff" & "fi" disappear when I copy text ? [Ligatures](#)
- Should I apply AA before or after gamma correction? [Before](#)
- Is there really a movie about Helvetica ? Yes.
- Should I stay or should I go? stay (a few more minutes)

Any other questions?

# should i use comic sans?

## will your document be viewed by the public?

yes

no



# don't use comic sans!