

Mapbox



@kkaefer

Rendering Maps with OpenGL

Rendering maps

on the device

in the browser

CPU

Agg/Mapnik

HTML5 Canvas

SVG

GPU

OpenGL/WebGL

THE



PROJECT

Anti-Grain Geometry

HTML



Canvas

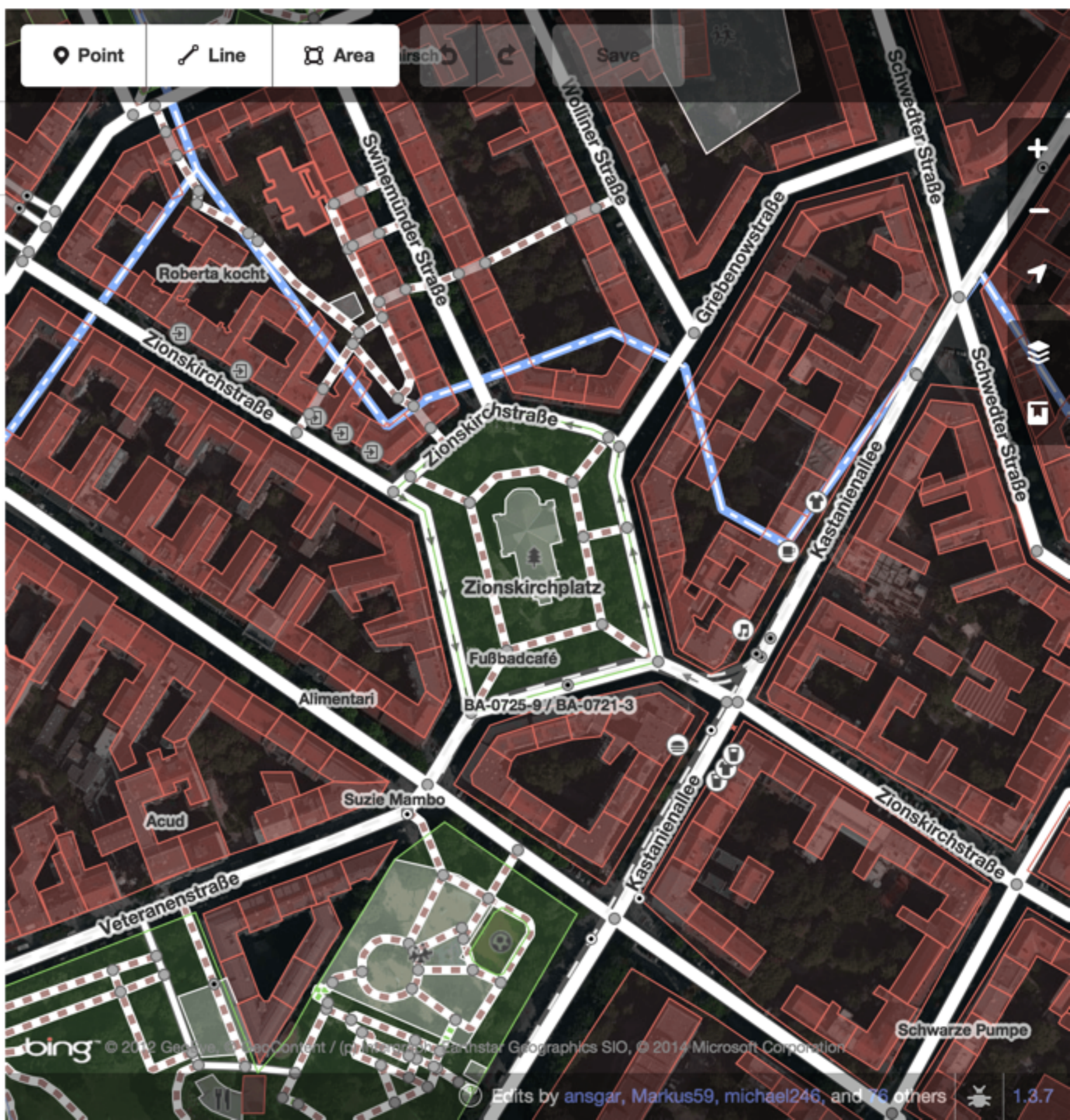
SVG

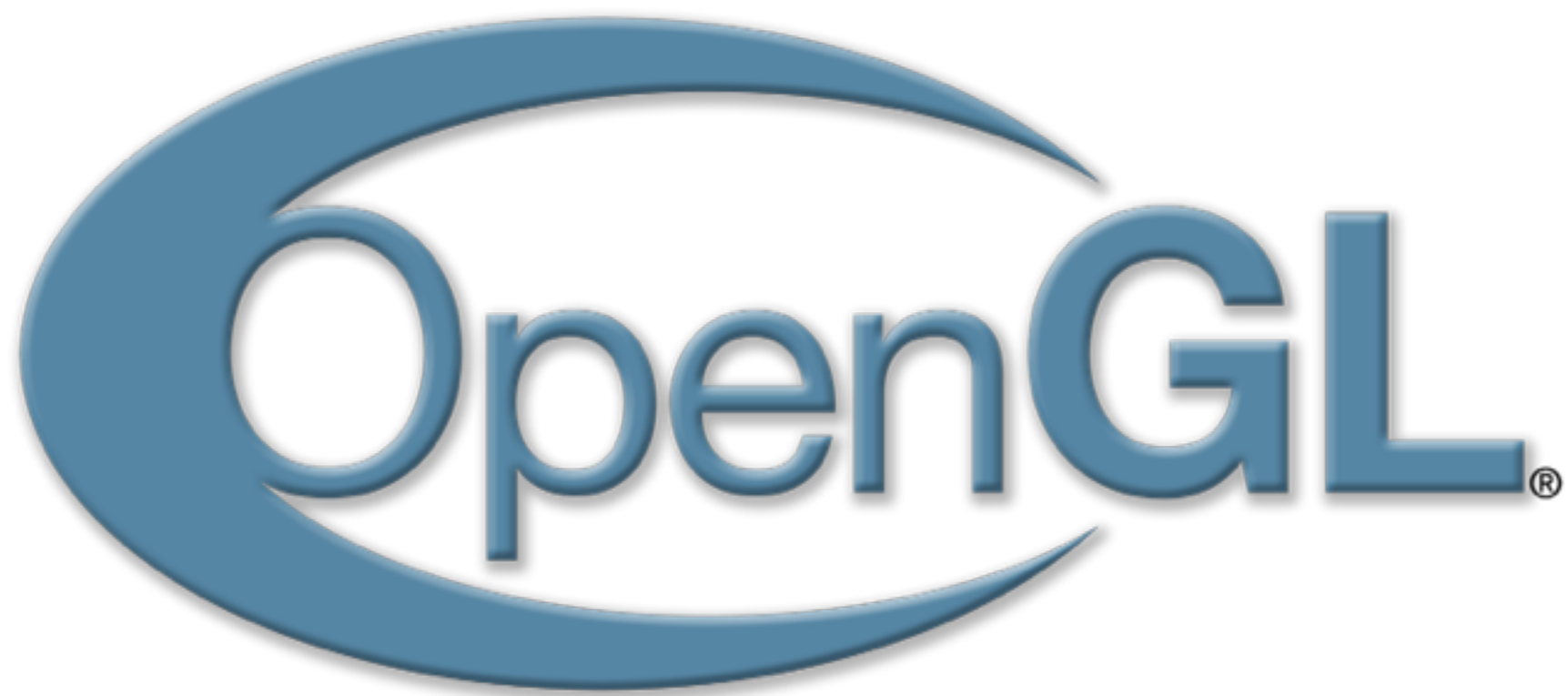
- SVG tiles (e.g. <https://github.com/springmeyer/svg-tiles>)
- Use as `` in Leaflet (Demo)
- **Advantages:**
 - Use CSS for map styling
 - Almost no client-side code needed
 - All platforms support SVG
- **Drawbacks:**
 - Rendering SVG is slow
 - SVG tiles are large (XML!), compression helps a little

Search features

Search

Point
Line
Area
Undo
Redo
Save



The logo for OpenGL, featuring the word "OpenGL" in a blue, sans-serif font. The "O" is stylized as a large, blue, curved shape that partially encloses the letters "pen". The "GL" is in a bold, sans-serif font. A registered trademark symbol (®) is located at the end of the "L".

OpenGL®

- Based on OpenGL ES 2.0
- Available in all browsers
- Often disabled on old or slow machines



- **Advantages:**

- Hardware acceleration
- Pixel level control

- **Drawbacks:**

- Very complex
- Fragmented devices

Maply

Google Maps

OpenScienceMap

Apple Maps

Mapbox GL

MapsWithMe

kMaps

Skobbler Mobile SDK

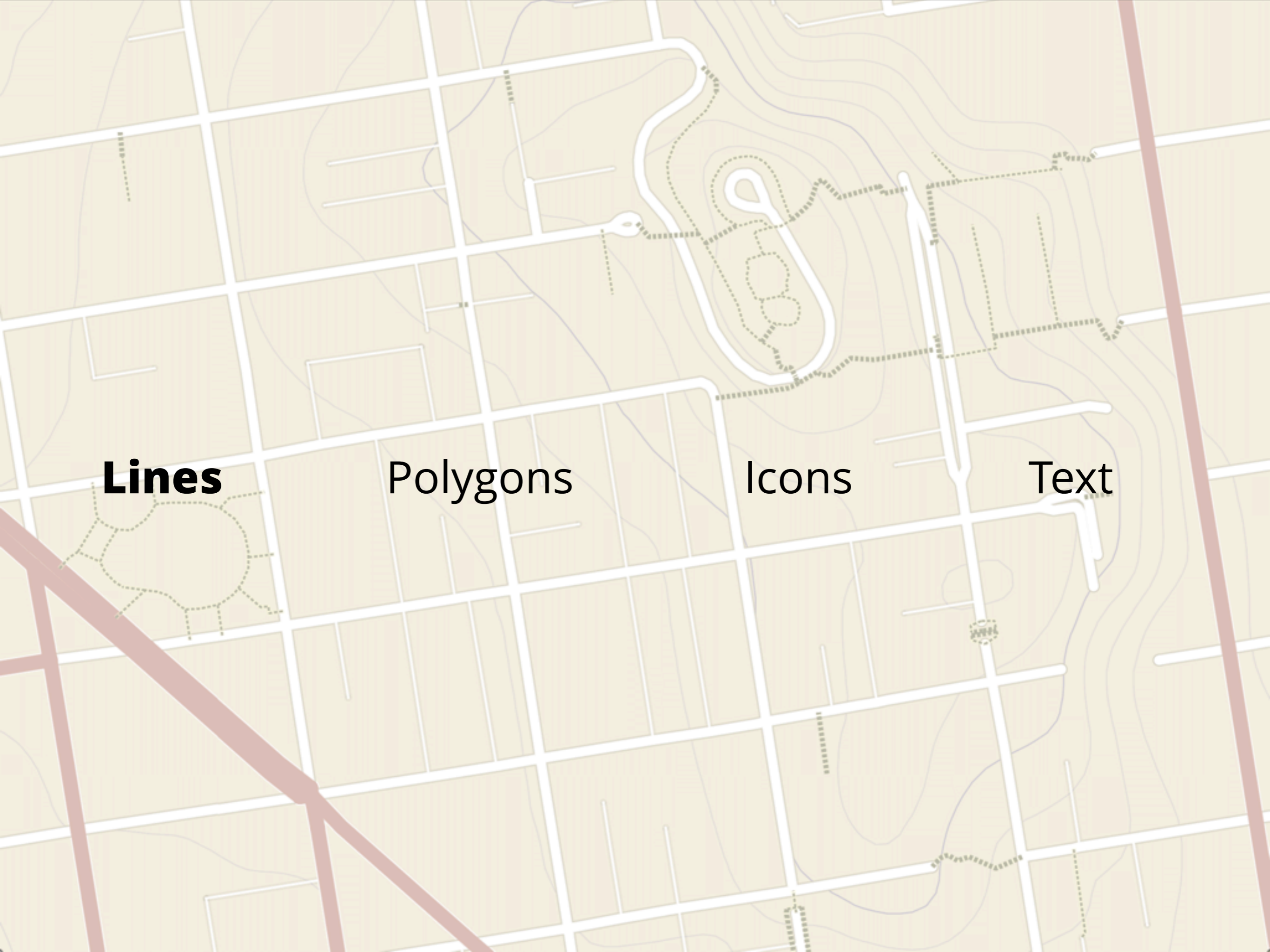
What is a map?

Lines

Polygons

Icons

Text



Lines

Polygons

Icons

Text

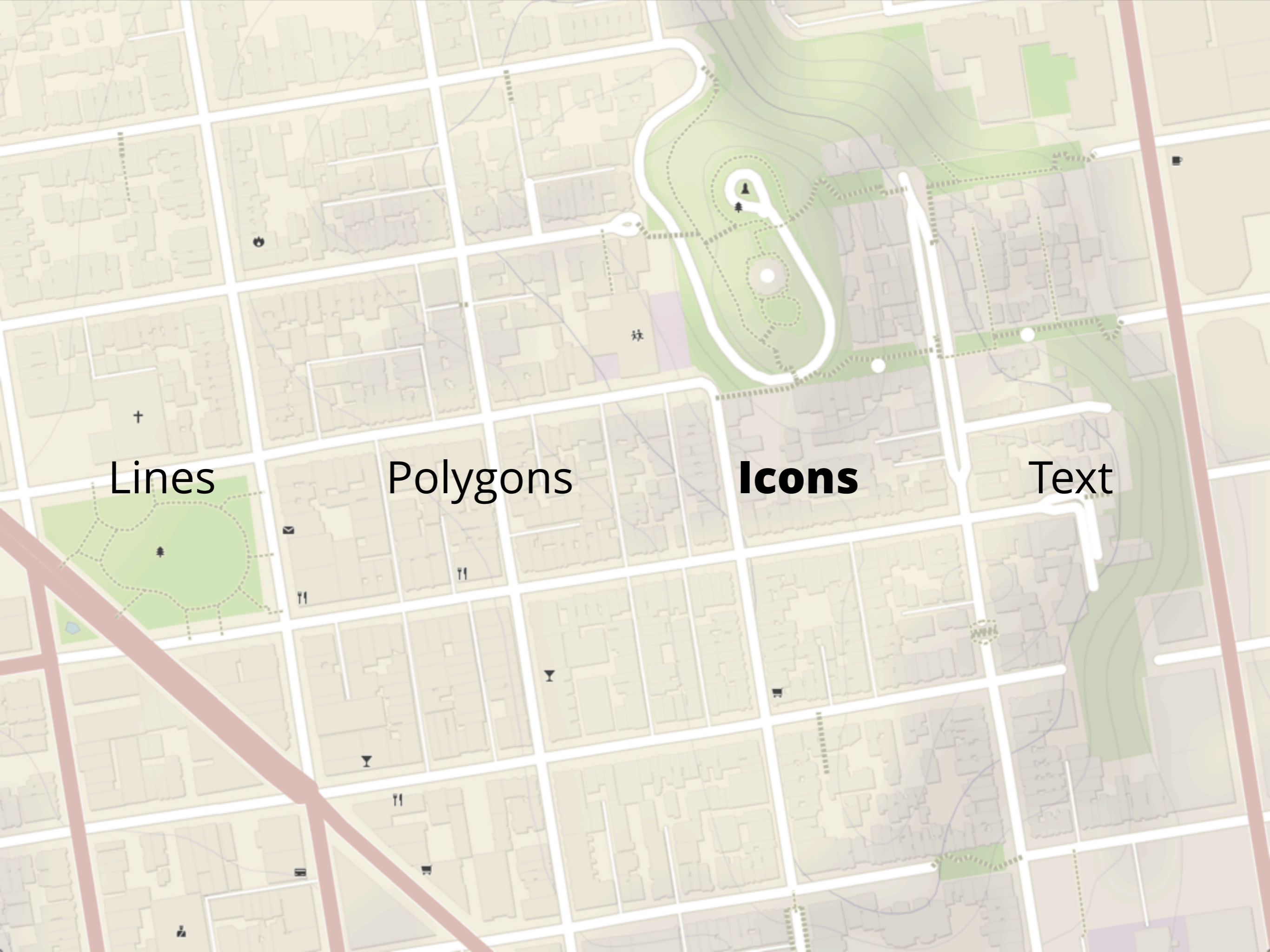


Lines

Polygons

Icons

Text



Lines

Polygons

Icons

Text



Lines

Polygons

Icons

Text

Greenwich St

Sansome St

Sansome St

Sansome St

Montgomery St

Pioneer Park

Garfield Elementary School

Saint Peter and Paul Church

North Beach

Kearny St

Telegraph Hill

Fire Fighter's Memorial

Washington Square Park

Union St

Tuk-Tuk Thai Cafe

La Spiaggia Delicatessen

Green St

Il Pollaio

Fugazi Hall

Green St

Grant Ave

Nico's Tacos

Stockton St

Columbus Ave

Green St

Montgomery St

Vallejo St

50 m

50 m

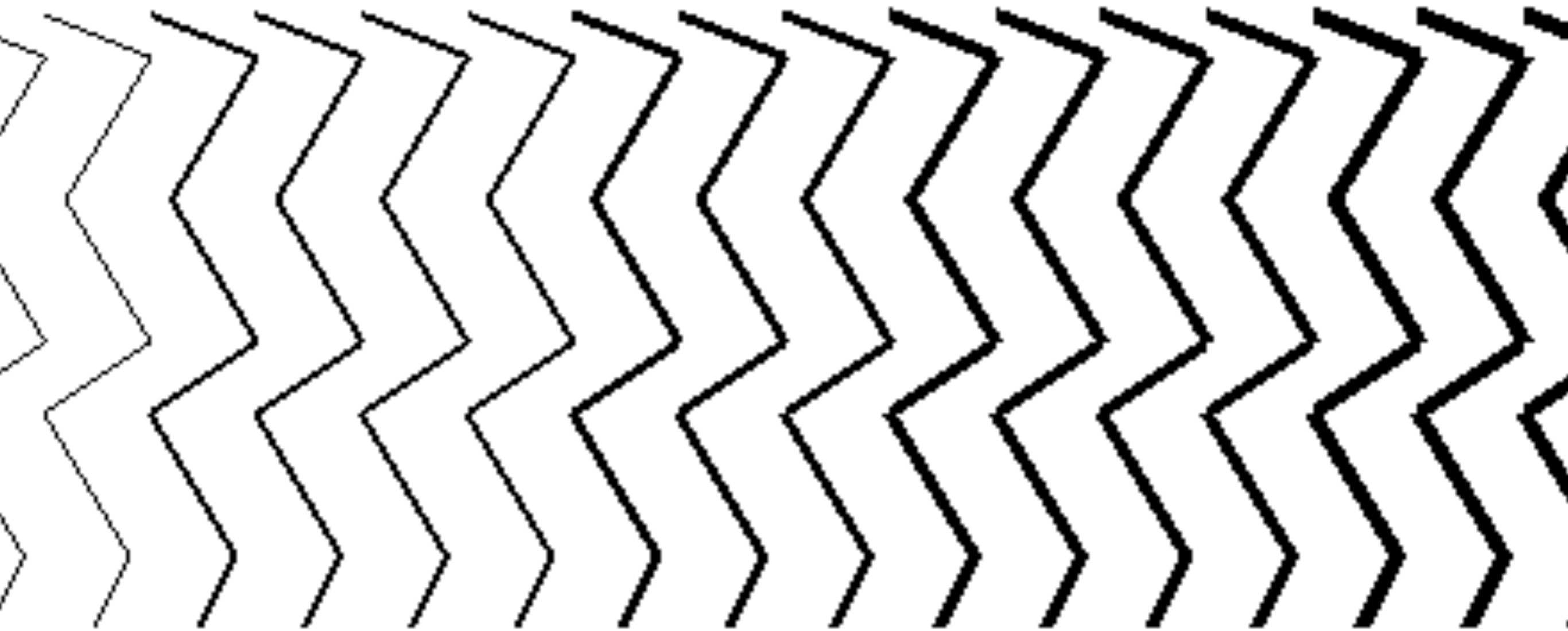
50 m

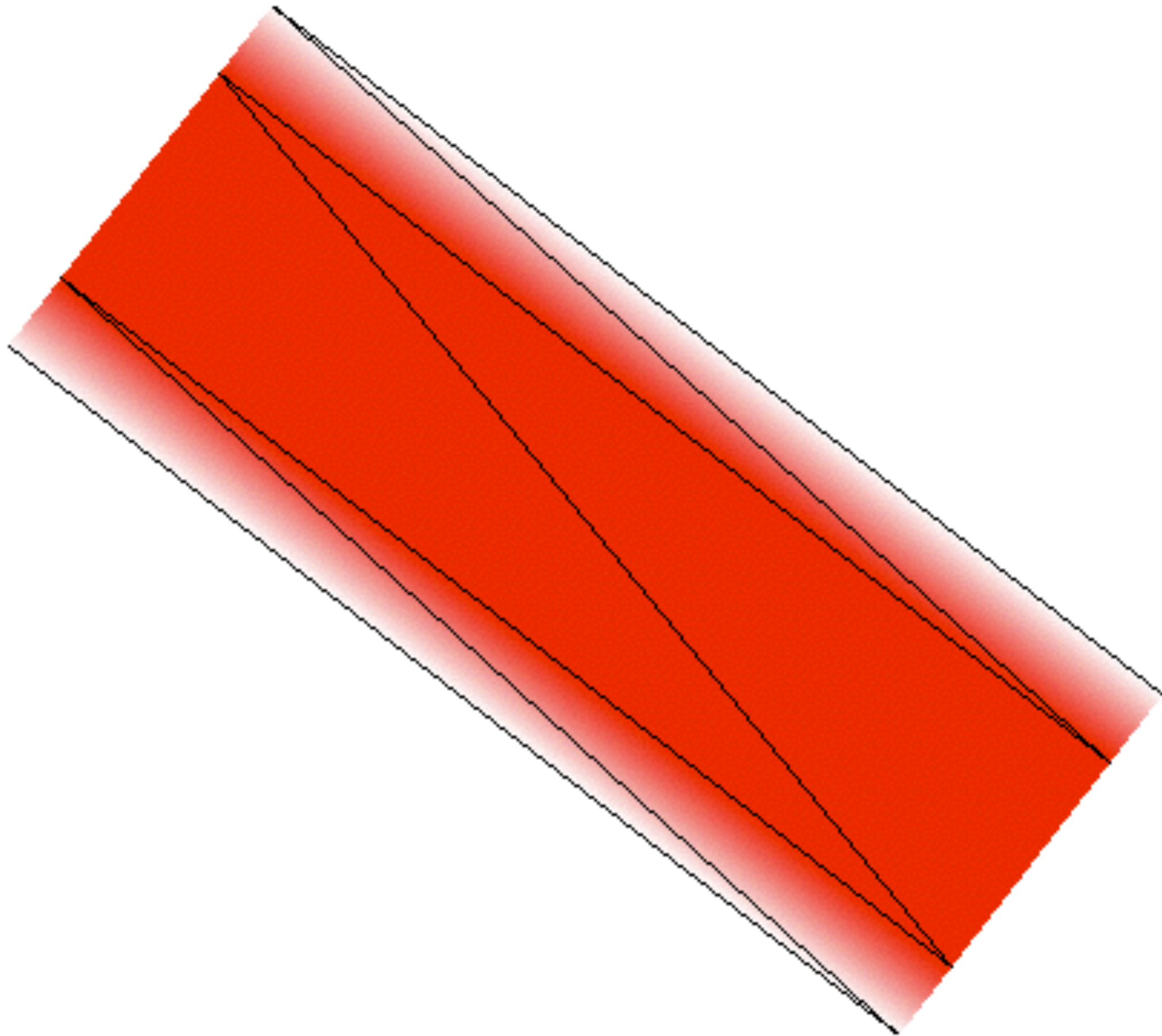
50 m

50 m

Lines



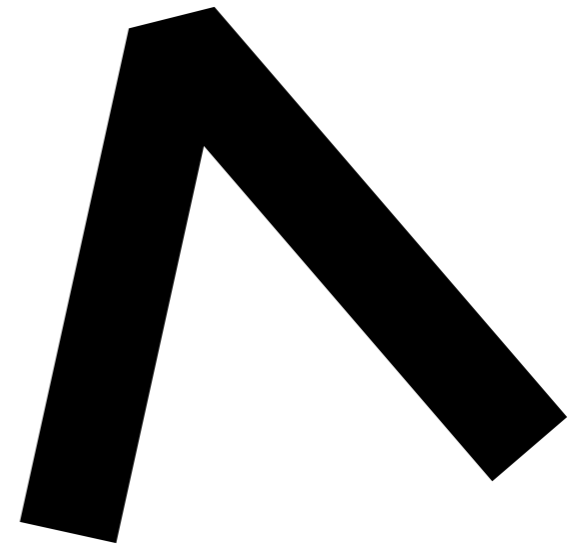
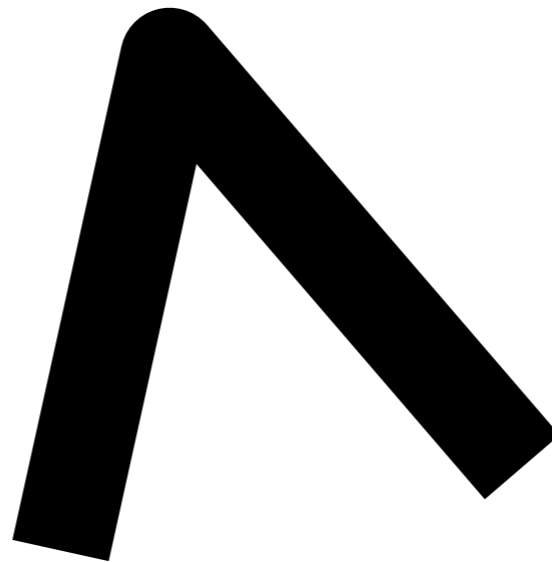
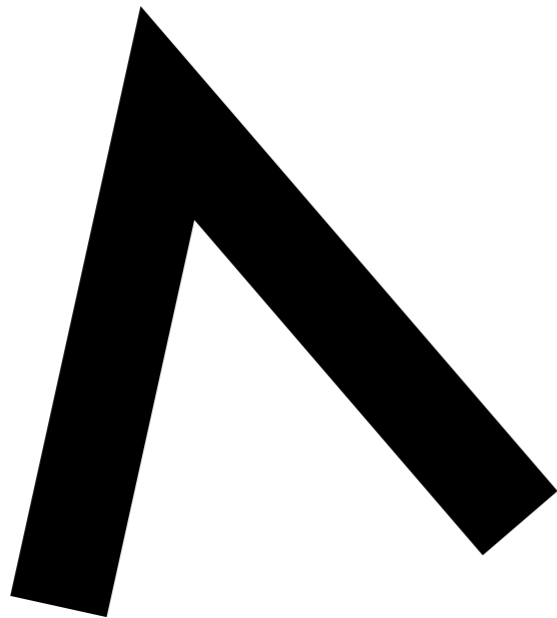




Miter

Round

Bevel



Read more on

mapbox.com/blog/drawing-antialiased-lines

Text Rendering



Berlin

Berlin

Berlin

Berlin

Berlin

Berlin

Berlin

Berlin

Berlin

The background features a large, faint, light blue watermark of the letters 'ft' on the left and a large, faint, light blue lowercase 'f' on the right, both in a serif font. The main text is centered and reads 'the Free Type Project' in a white serif font. The word 'the' is smaller and positioned to the left of 'Free Type', and 'Project' is smaller and positioned to the right of 'Free Type'.

the Free Type Project

Improved Alpha-Tested Magnification for Vector Textures and Special Effects

Chris Green*
Valve



(a) 64x64 texture, alpha-blended



(b) 64x64 texture, alpha tested



(c) 64x64 texture using our technique

Figure 1: Vector art encoded in a 64x64 texture using (a) simple bilinear filtering (b) alpha testing and (c) our distance field technique

Abstract

A simple and efficient method is presented which allows improved rendering of glyphs composed of curved and linear elements. 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, outlines, drop shadows, multi-colored images, and chamfered corners.

3D renderings and also user-interface elements. Our technique is capable of generating high quality vector art renderings as shown in Figure 1c.

2 Related work

Many techniques have been developed to accurately render vector graphics using texture-mapping graphics hardware. In [Friskin et al. 2000], distance fields were used to represent both 2-dimensional glyphs and 3-dimensional solid geometry. Quadtrees and octrees were used to adaptively control the resolution of the distance field based upon local variations. While GPU rendering

R

Demo

Text Shaping

unicode character



font glyph


sigma → σοφός → sigma

context-dependent shaping

Unshaped: مِ يَحْرُلْ اِنْ مَحْرُلْ اِهْلْ اَمْ سِبْ

Shaped: يَسْمُ اللهُ الرَّحْمَنُ الرَّحِيمُ

ابش 1234 دَفق



bidirectional text

ରସ ଧ୍ୱଜା ଧ୍ୱଜା
ରସ ଧ୍ୱଜା ଧ୍ୱଜା

Stacked diacritic symbols

ରସ ଧ୍ୱଜା ଧ୍ୱଜା
ରସ ଧ୍ୱଜା ଧ୍ୱଜା

Label placement

No pops

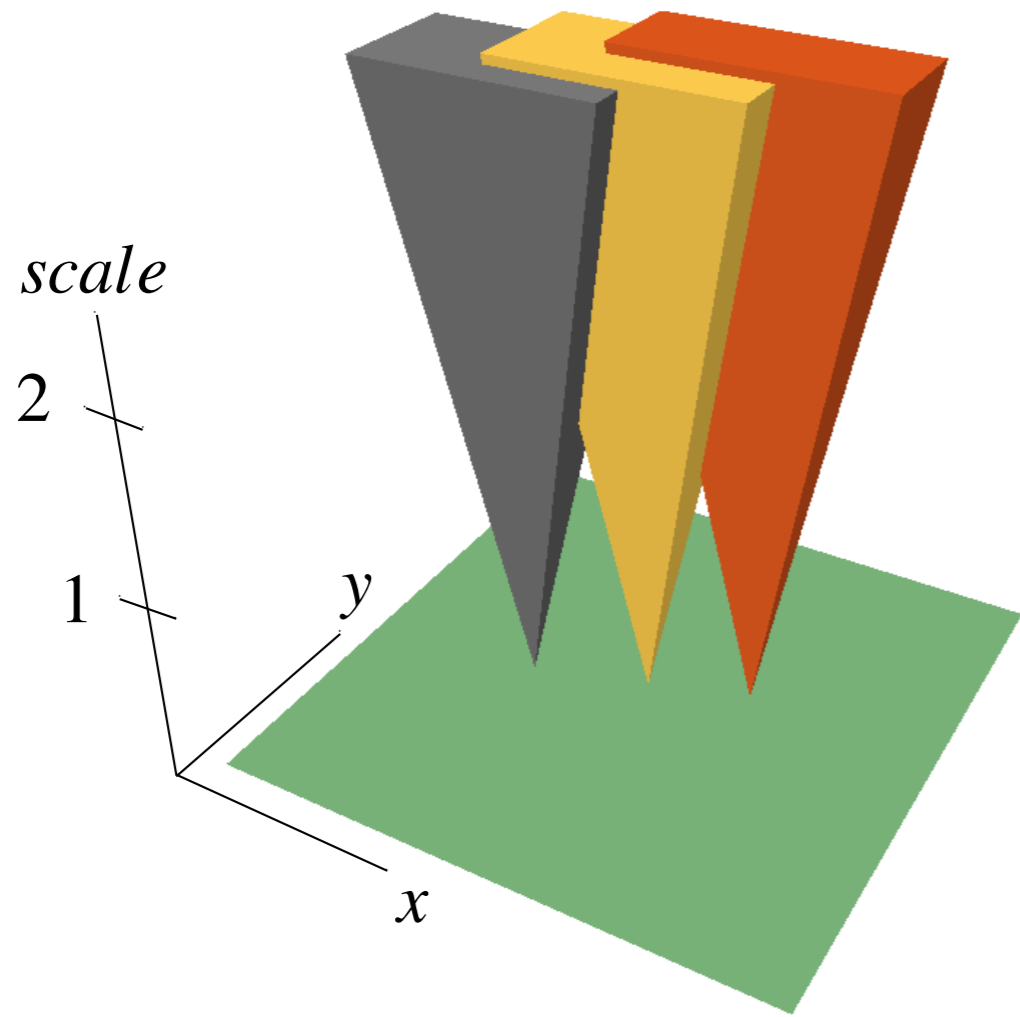
If a label appears, it must never vanish when zooming in

No jumps

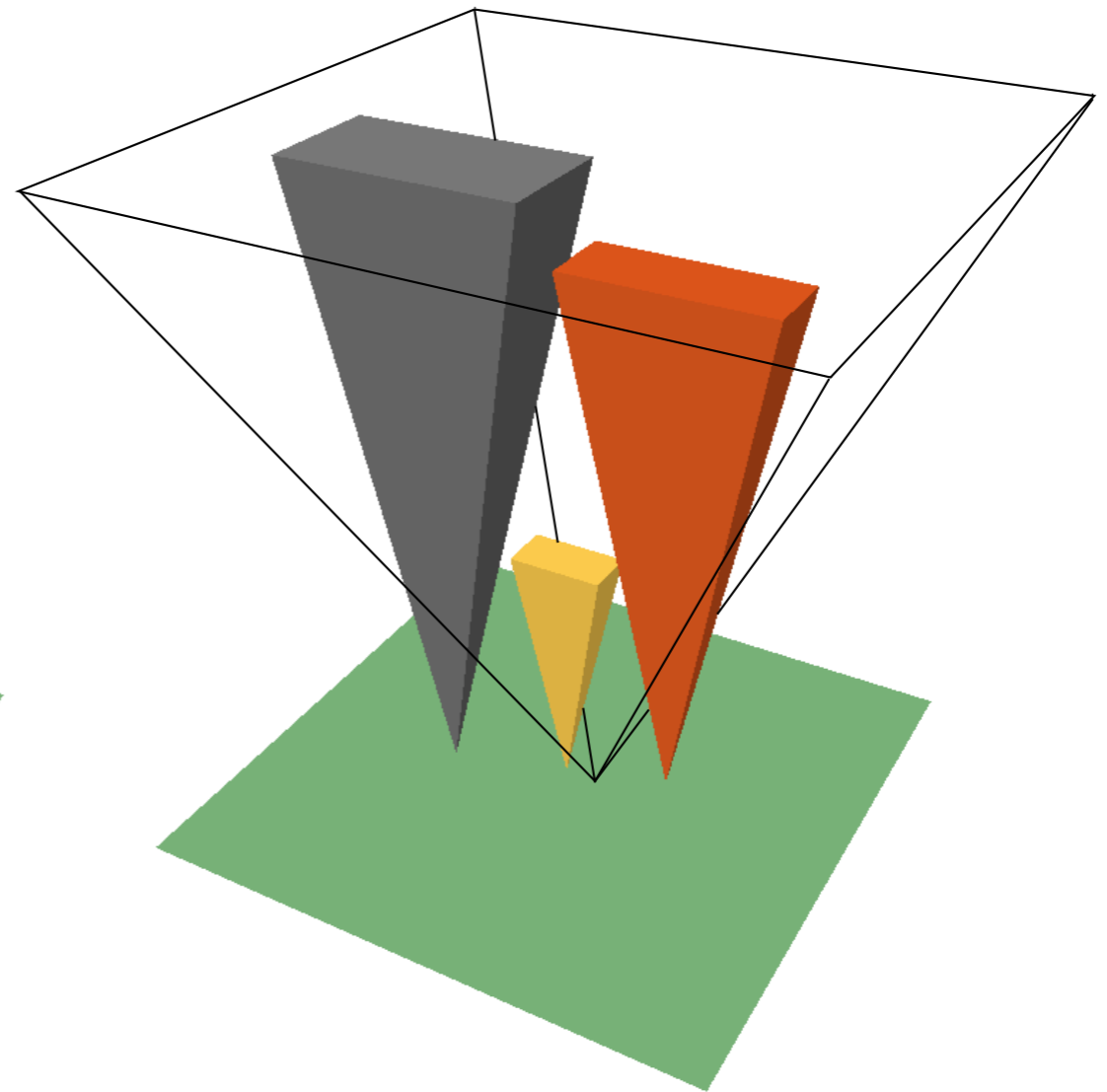
Labels may not change position when panning or zooming

No history

The labelling may not be dependent on navigation history



(a)



(b)

Thanks. **Questions?**



Mapbox