I heard you like tiles...

Michal Migurski,
Geomeetup April 2013

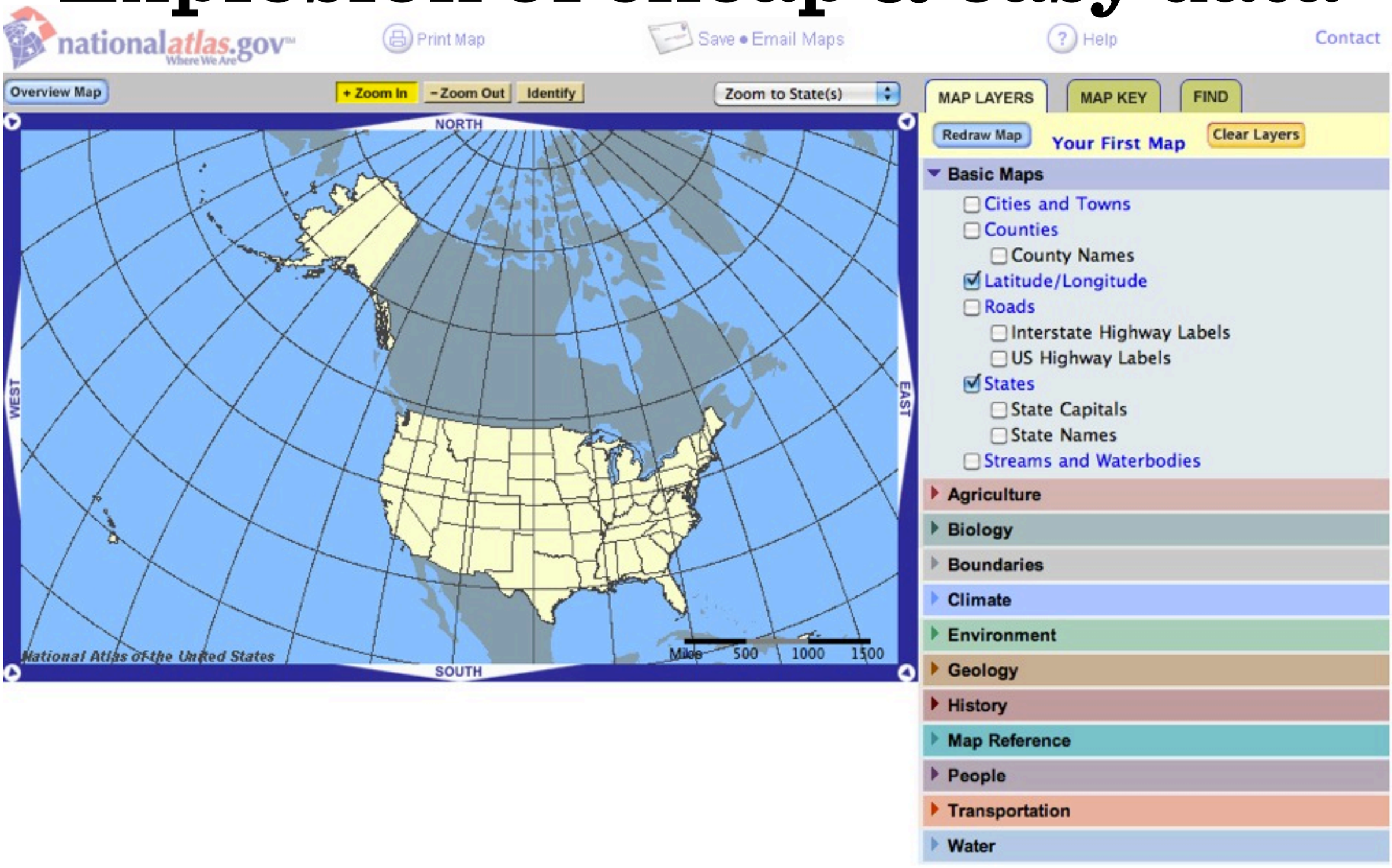
...so I put some vectors in your tiles so you could tile while you vector.

Why?

- Using OpenStreetMap should be as easy as pasting a URL.
- OSM is big, and computers & networks aren't getting much faster.
- Vector Tiles exist and work with worldwide data right now.

http://nationalatlas.gov

Explosion of cheap & easy data



We’re living in an explosion of cheap, easy-to-get geographic data on the internet.

http://naturalearthdata.com

Natural Earth Data: plain shapefiles, million+ downloads

Large scale data, 1:10m

Cultural | Physical | Raster

The most detailed. Suitable for making zoomed-in maps of countries and regions. Show the world on a large wall poster.

1:10,000,000
1" = 158 miles
1 cm = 100 km

Medium scale data, 1:50m

Cultural | Physical | Raster

Suitable for making zoomed-out maps of countries and regions. Show the world on a tabloid size page.

1:50,000,000
1" = 790 miles
1 cm = 500 km

Small scale data, 1:110m

Cultural | Physical

Suitable for schematic maps of the world on a postcard or as a small locator globe.

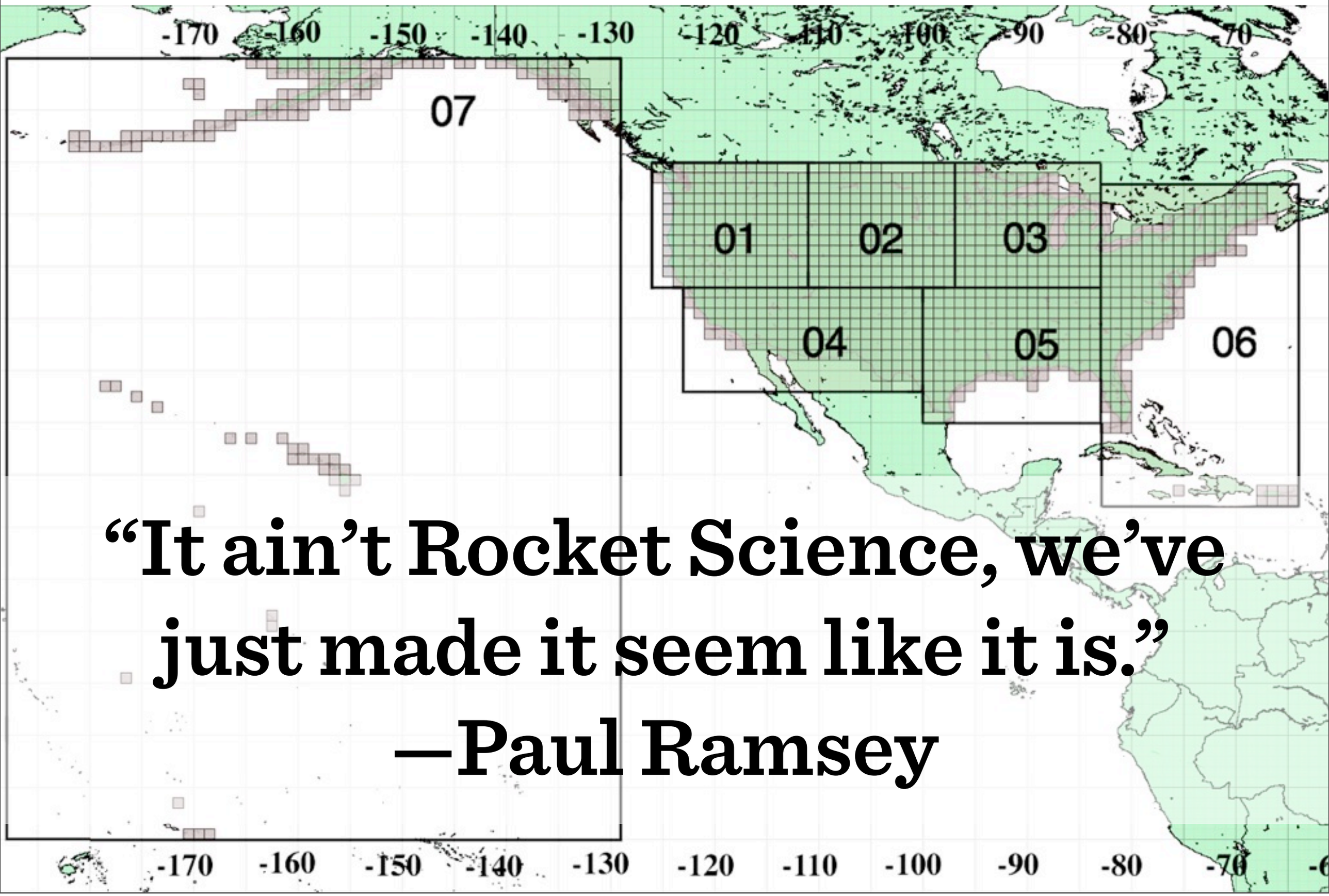
1:110,000,000
1" = 1,736 miles
1 cm = 1,100 km

Datasets like Natural Earth are widely available. Nathaniel publishes Natural Earth as a plain series of shapefiles, and has had hundreds of thousands of downloads from cartographers and mapmakers.

A simple file download or URL should never be far away

For those things that can’t be gotten with a simple download, a simple URL should suffice.

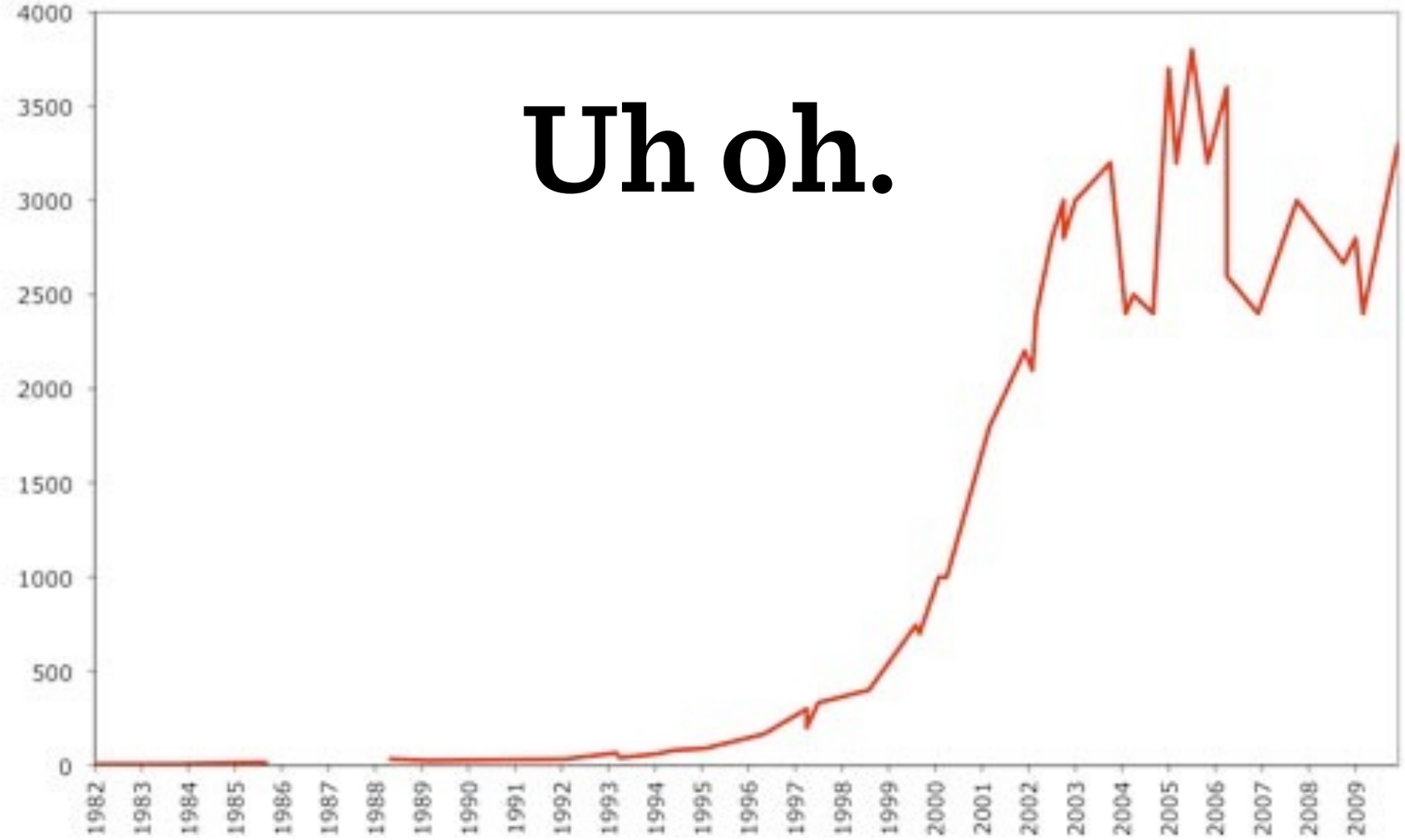
http://teczno.com/s/9b8



“It ain’t Rocket Science, we’ve just made it seem like it is.”

—Paul Ramsey

Most things should be gettable with dumb files in directories. *Paul Ramsey rant about FTP* I spent weeks locating non-seamless, non-UI zip files for NED and SRTM data before finally finding them.



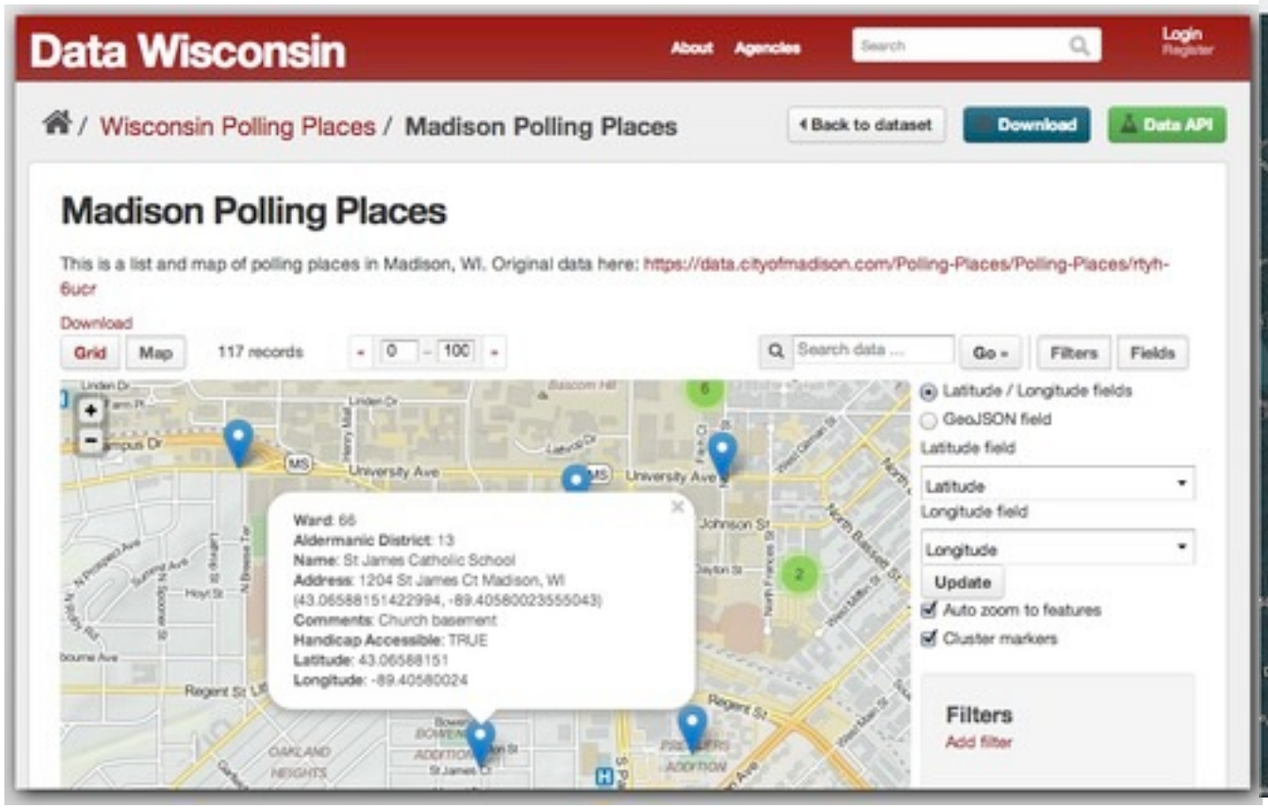
"...we hit the era of what I'm calling Peak MHz in about 2004. That's the point when processor speed effectively peaked as chip manufacturers began competing along other dimensions."

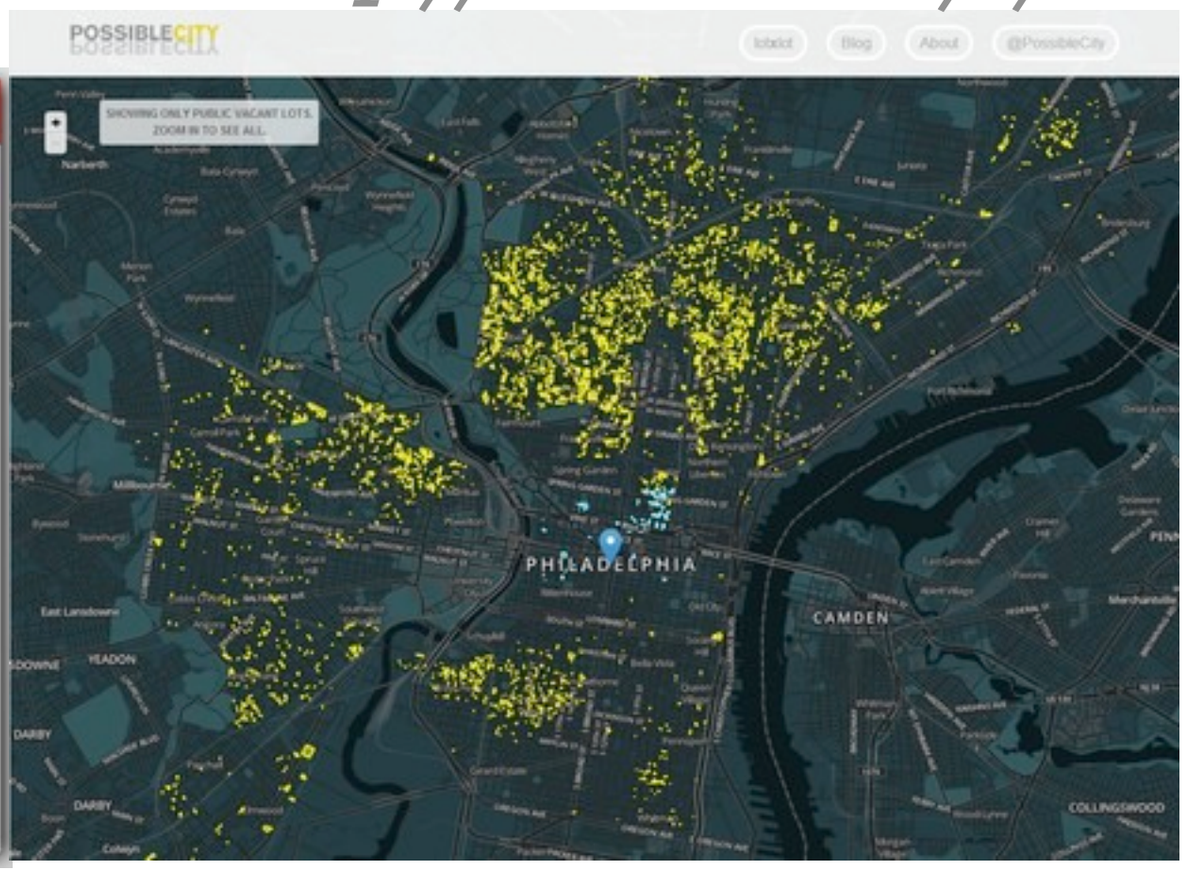
—Mike Kuniavsky, http://sta.mn/5cm

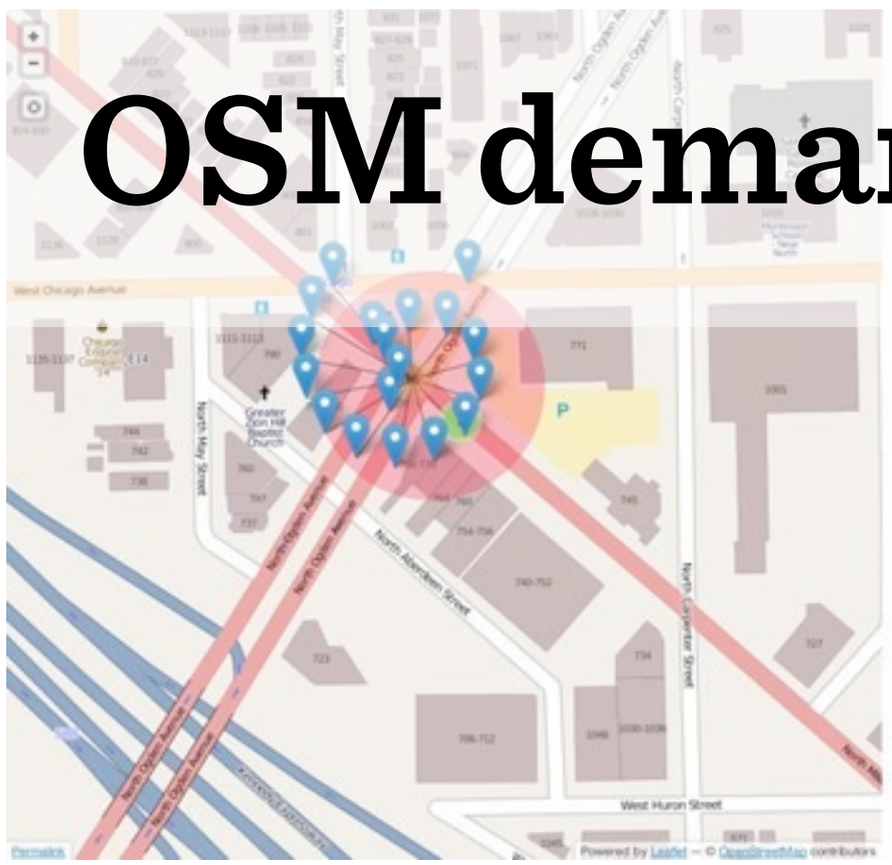
Peak MHz: uh oh. Our computers haven't actually gotten significantly faster in almost ten years. They've gotten smaller, and able to do more things at once, but not faster. Our typical internet has also not gotten faster in ten years. It's everywhere now including the BART tunnel, but for a lot of typical users it's actually smaller.

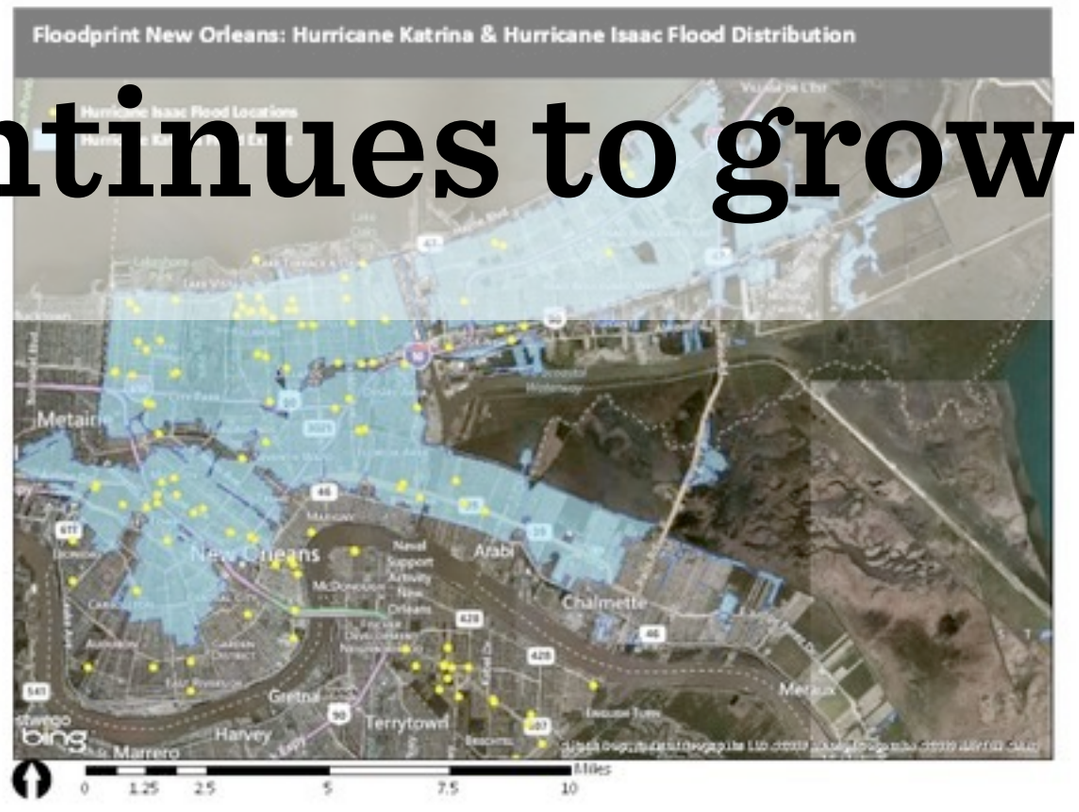
http://teczno.com/s/cbz

OSM demand continues to grow







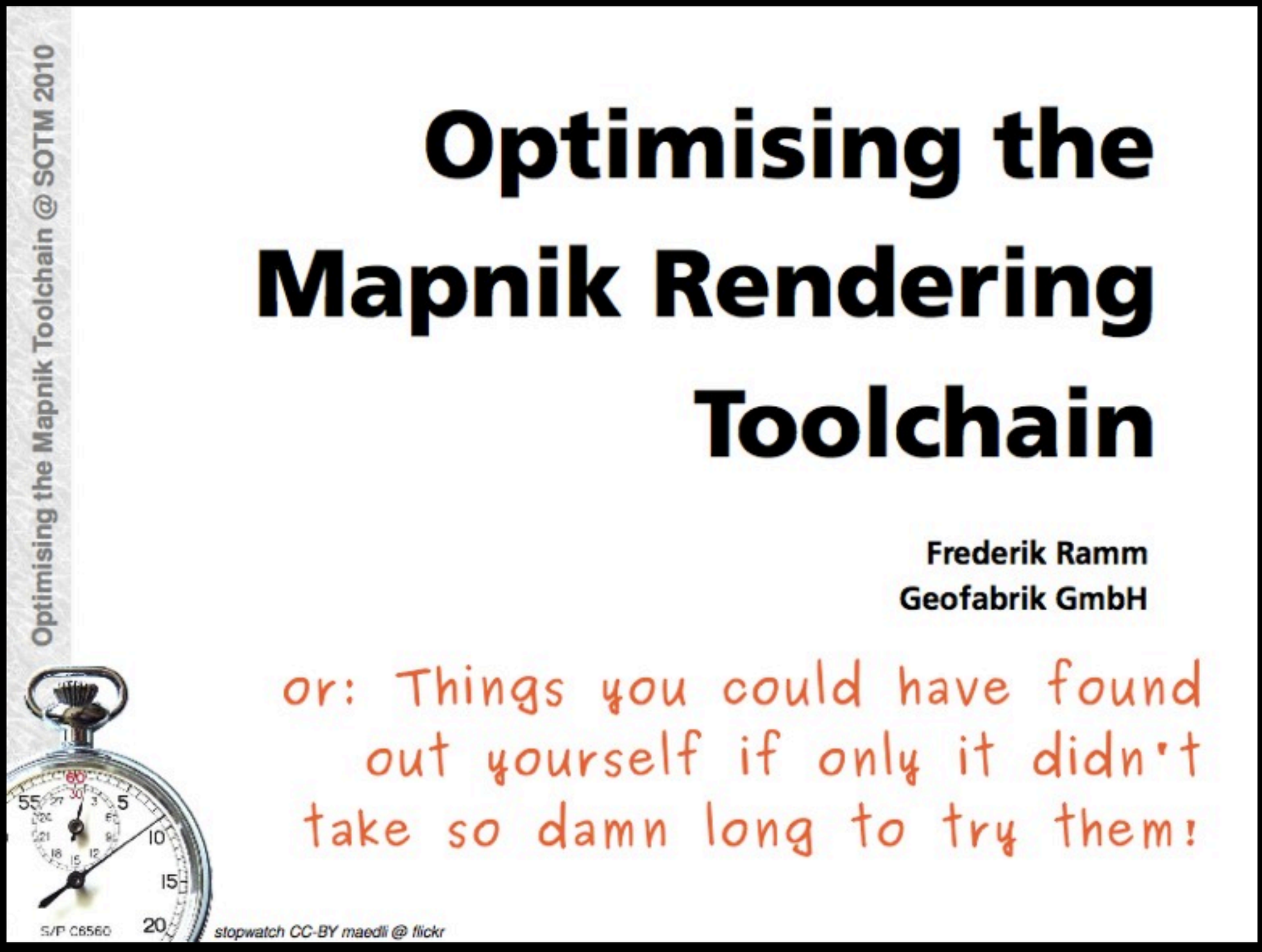


OSM is getting hugely popular, 5 of the 6 map projects in Atlantic Cities's "12 Fresh Ideas" choose OSM. How do they use them? Dumb tile URLs. Which one doesn't? The one with satellite imagery and fancy choropleths.

Planet file is now 27GB

OSM data is getting bigger, though. When I first publicly talked about this a year and a half ago, the planet file was a paltry 19GB. Now it's 27GB and growing all the time.

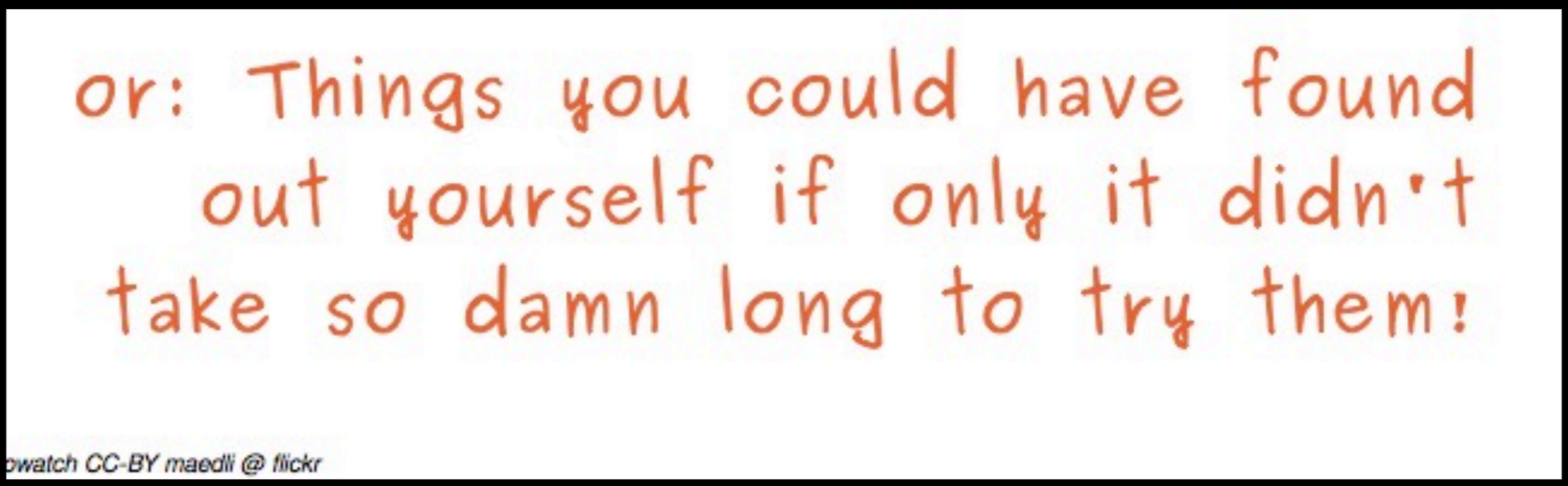
http://sta.mn/gjh



PostGIS is not getting any easier to administer, especially if you’re a newspaper or a civic data gadfly.

http://sta.mn/gjh

or: Things you could have found out yourself if only it didn't take so damn long to try them!

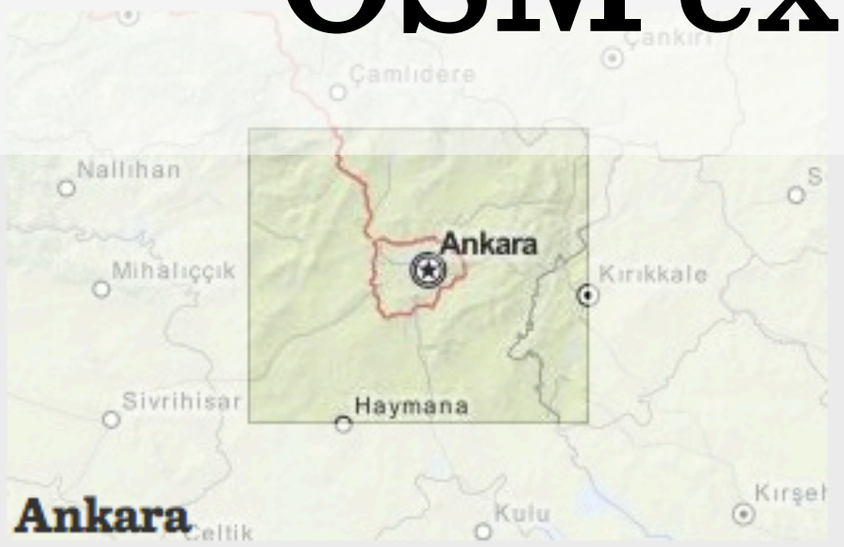


owatch CC-BY maedli @ flickr

OSM data is getting bigger, more difficult, and slower to handle.

OSM extracts help, sort–of.

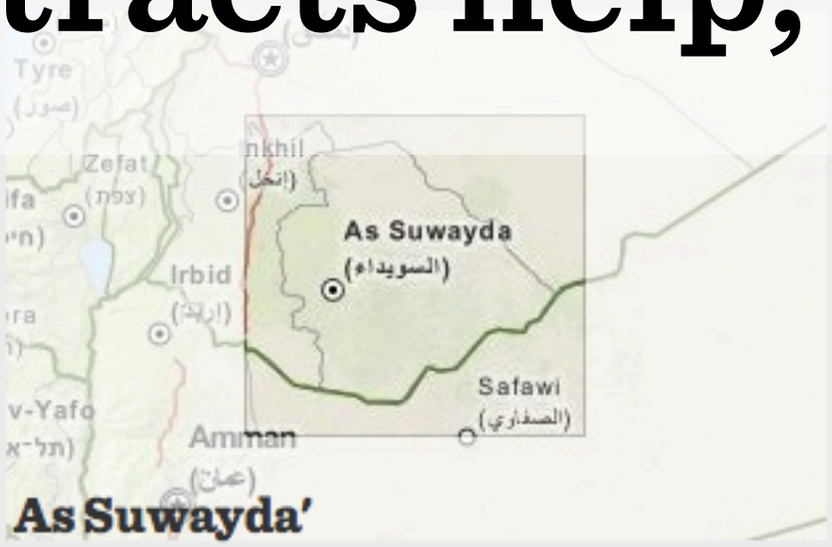
1.1 KB coastline shapefile
654 KB osm2pgsql shapefiles
1.3 MB imposm shapefiles



Ankara

3.0 MB bzip'ed XML OSM data
1.8 MB binary PBF OSM data
1.3 KB coastline shapefile
3.6 MB osm2pgsql shapefiles
7.0 MB imposm shapefiles

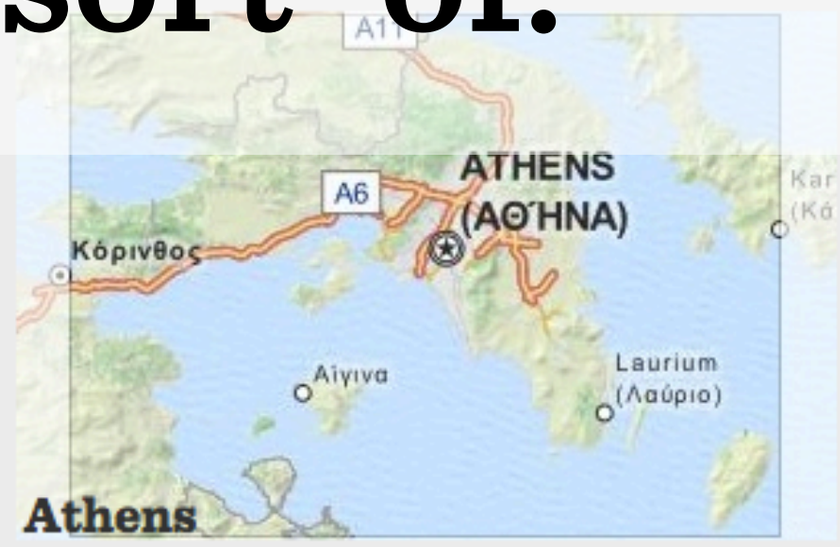
1.0 KB coastline shapefile
2.0 MB osm2pgsql shapefiles
3.2 MB imposm shapefiles



As Suwayda'

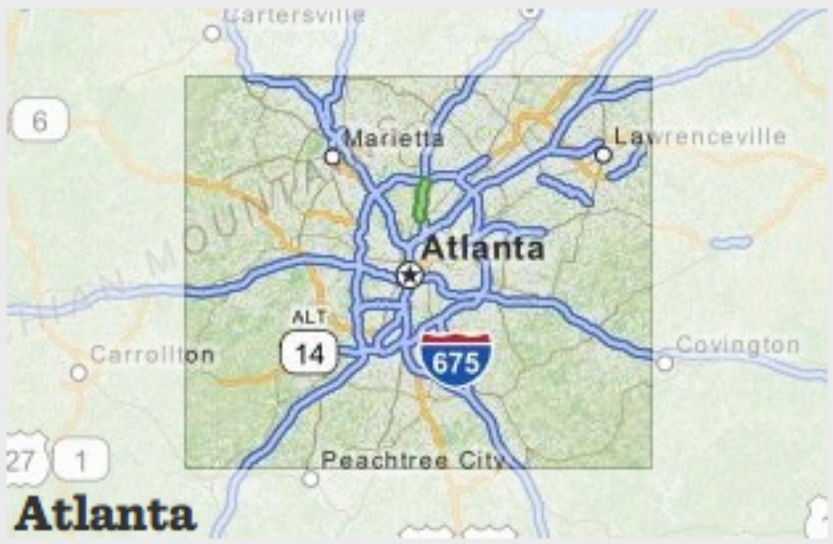
1.3 MB bzip'ed XML OSM data
810 KB binary PBF OSM data
1.3 KB coastline shapefile
1.4 MB osm2pgsql shapefiles
2.8 MB imposm shapefiles

193 KB coastline shapefile
98 MB osm2pgsql shapefiles
115 MB imposm shapefiles



Athens

16 MB bzip'ed XML OSM data
9.6 MB binary PBF OSM data
635 KB coastline shapefile
18 MB osm2pgsql shapefiles
32 MB imposm shapefiles

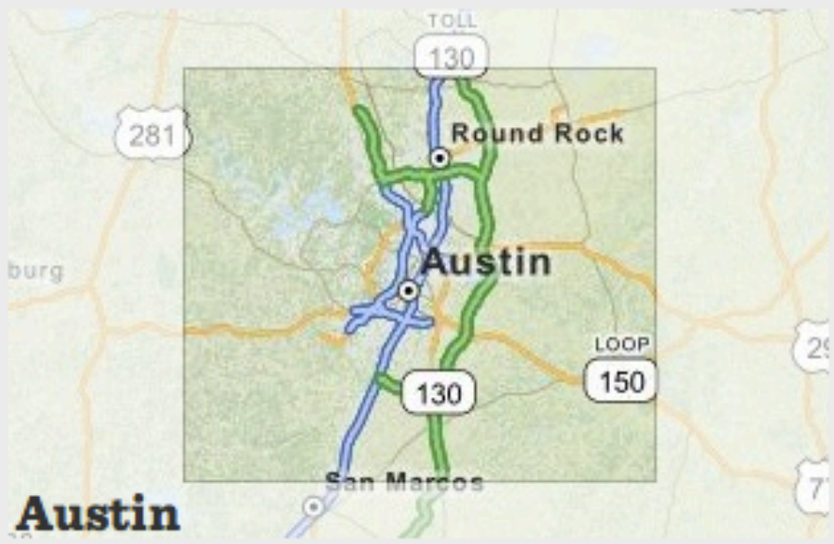


Atlanta

33 MB bzip'ed XML OSM data
18 MB binary PBF OSM data
1.2 KB coastline shapefile

Auckland

10.0 MB bzip'ed XML OSM data
6.0 MB binary PBF OSM data
864 KB coastline shapefile

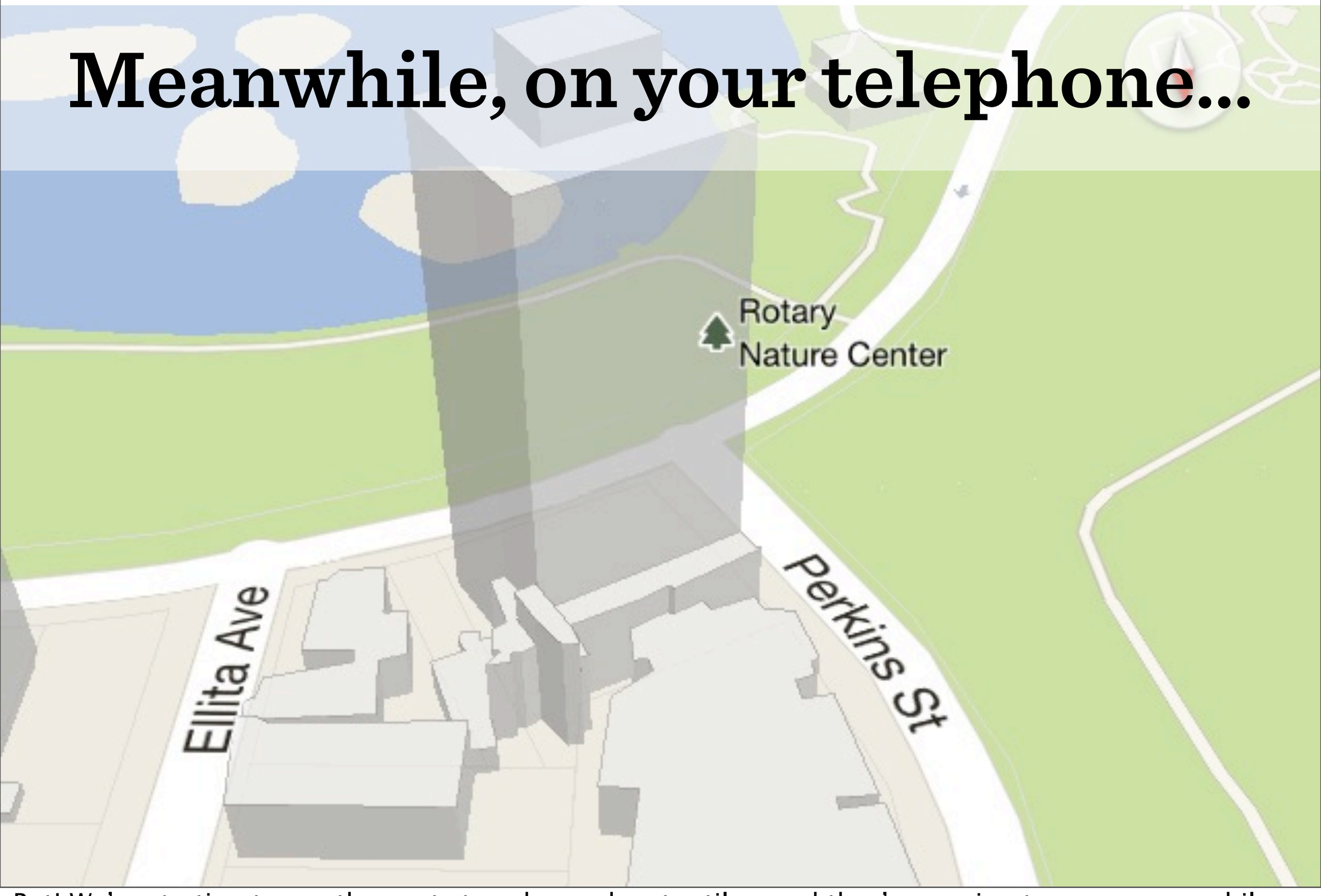


Austin

10 MB bzip'ed XML OSM data
6.4 MB binary PBF OSM data
1.2 KB coastline shapefile

I've helped this situation some on the past by creating metro extracts, which helps on just one dimension of ease by making shapefiles for small urban areas available. If you're not looking for one of these almost cities and can't wait for me to accept a pull request, you are out of luck.

Meanwhile, on your telephone…



But! We're starting to see the next steps beyond raster tiles, and they're coming to us on our mobile phones in iOS, Google and Nokia maps.

Bottom line: this stuff should be easy.

It’s time for OSM to get (more) creative about distribution.

It’s time for OpenStreetMap to get creative (again) about how its data is distributed.

Time for vector tiles!
(tl;dr: teczno.com/s/r3p)

Why are vector tiles right for now?

http://sta.mn/xq

Tiles over HTTP are easy: edge caches, REST, etc.

They're easy to bake and distribute. Vector tiles are cheaper to render than image tiles, because there's no server-side rasterization step beyond the packaging of the data. Distribution is easier thanks to all the HTTP infrastructure that we're now able to take for granted: edge caches like Fastly, etc.

http://sta.mn/v7f

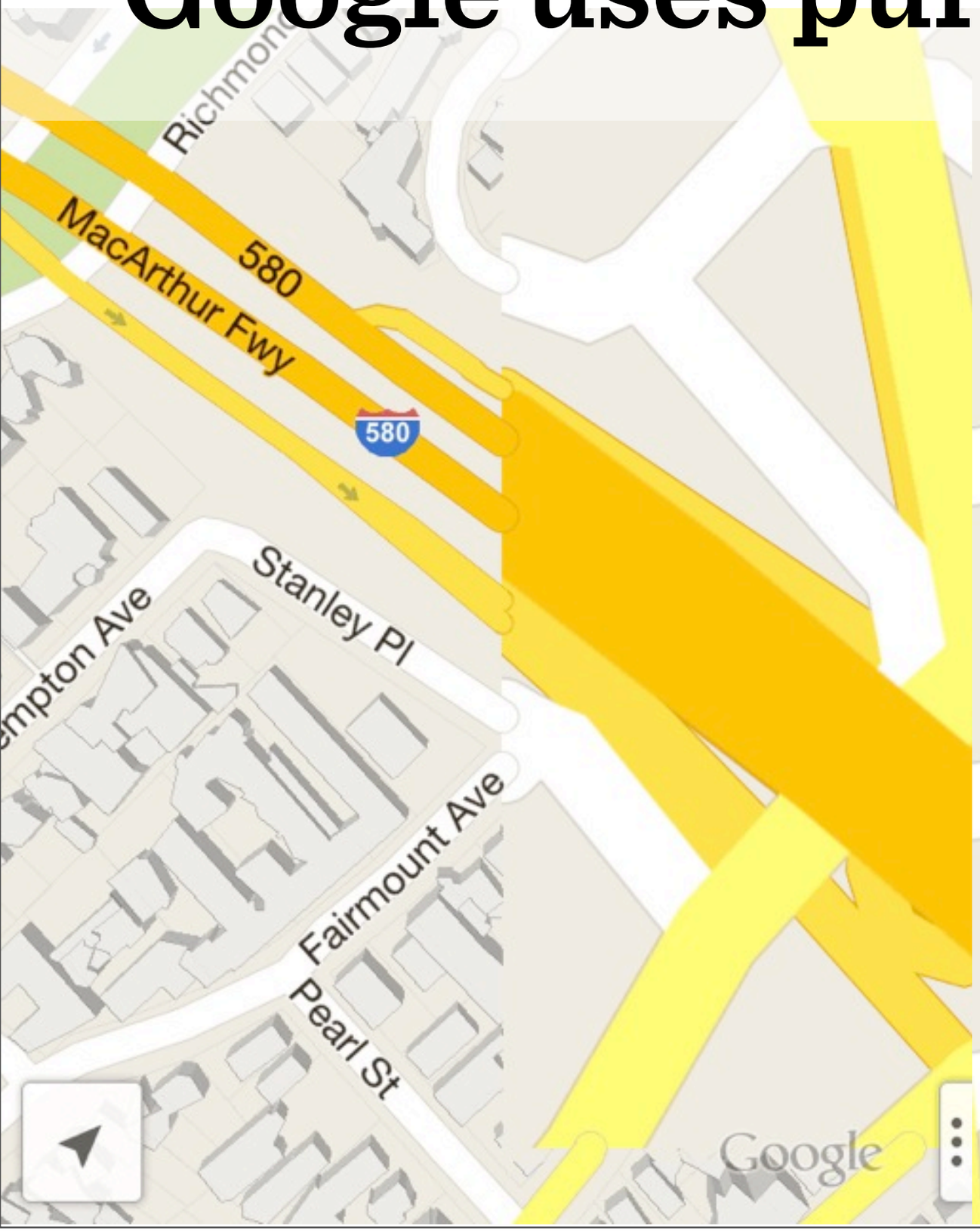
They've worked well before; we used BSON & PVR for iOS

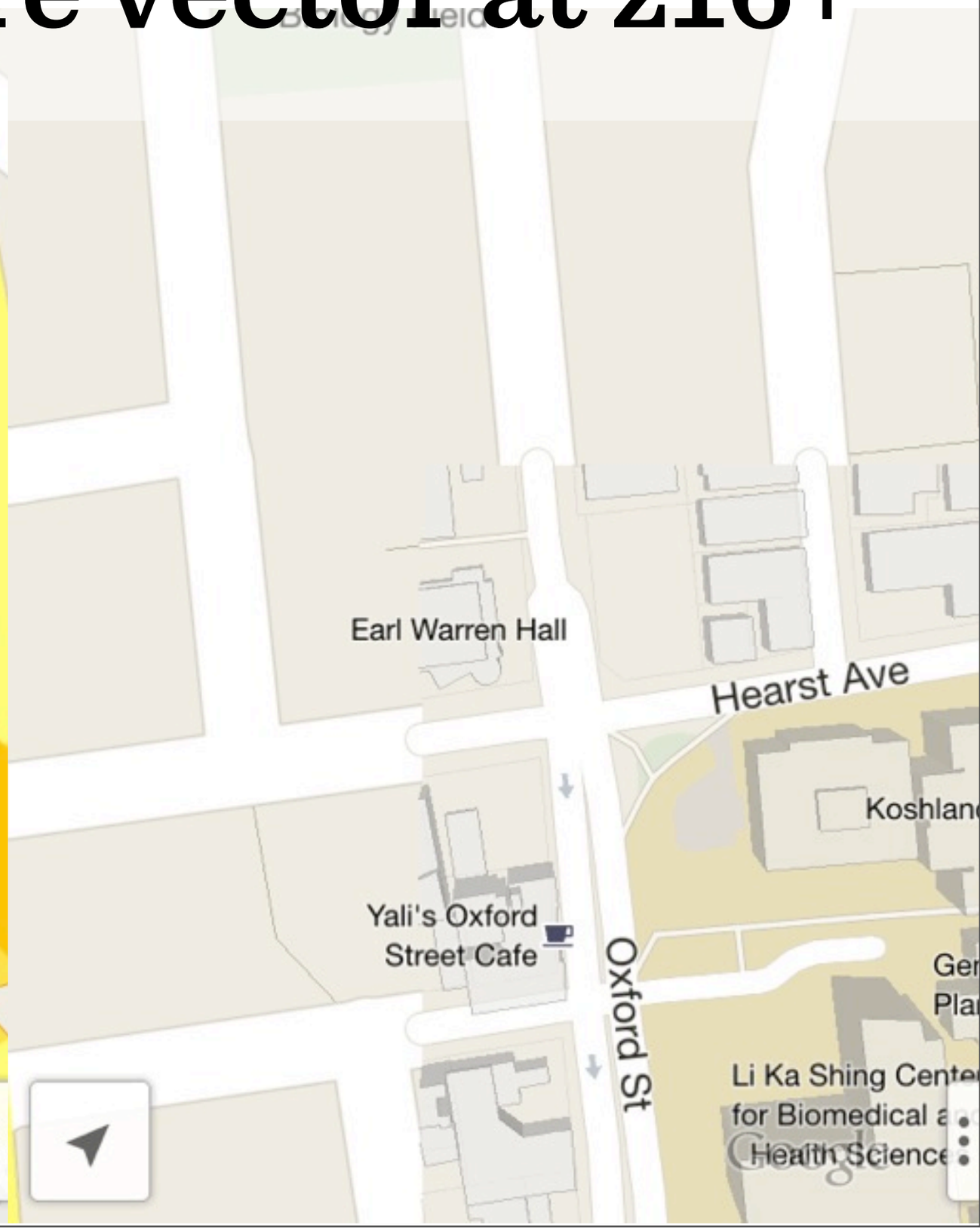
(talk GeoJSON vs. BSON + National Geographic here? backend infrastructure of that project)

Storage is cheap, rendering is last-mile

Storage for these things is cheap. You can imagine putting an entire world of render-ready tile onto a single small hard drive, the size wouldn't be much larger than planet itself. Most of that size is buildings anyway. Rendering can be a last-mile problem, with fast, high-quality GPU in everyone's pocket.

Google uses pure vector at z16+





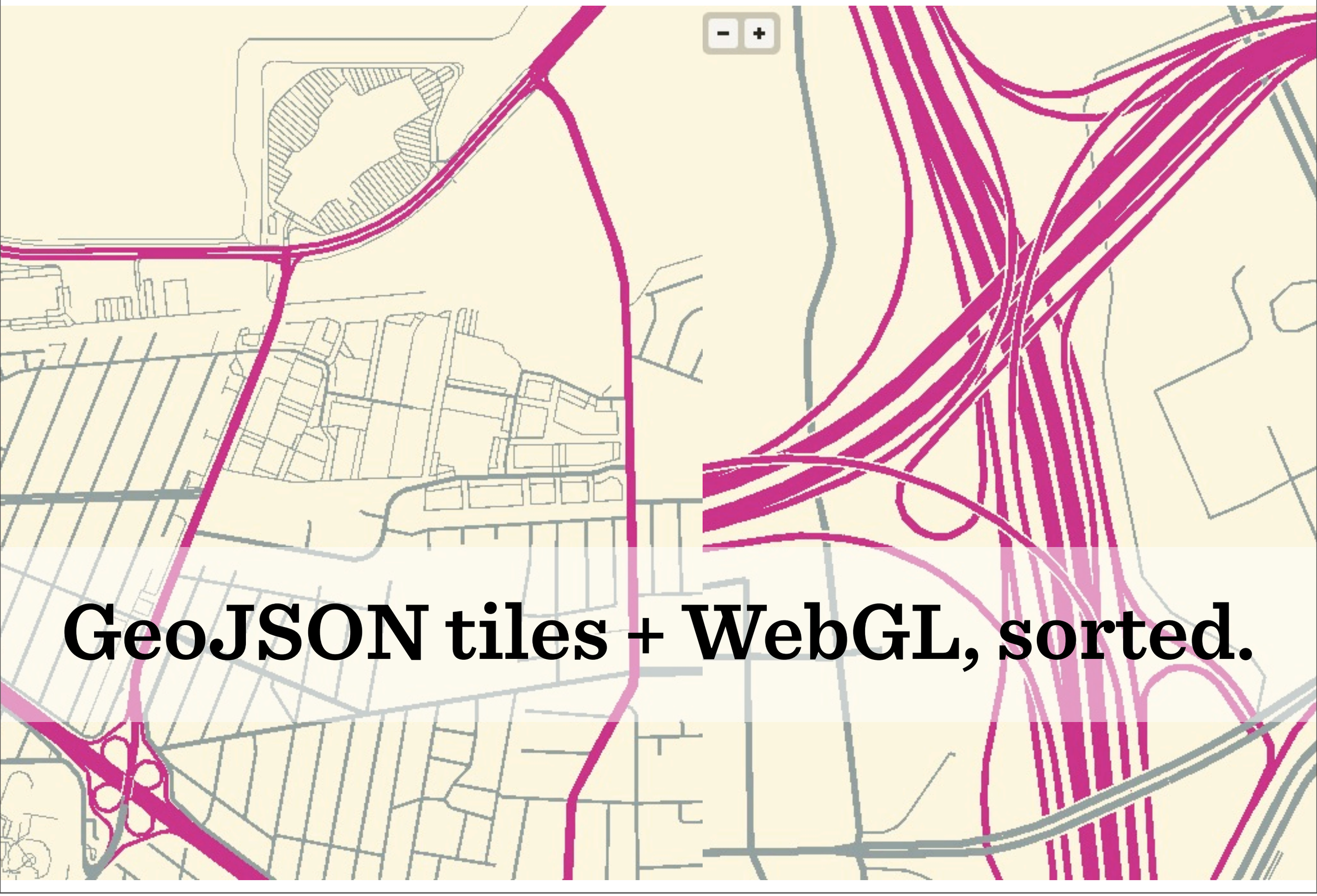
(Show some screen grabs of Google iOS maps with tile boundaries visible)

Simple formats like GeoJSON work well, with clipped precision

E.g., "37.123456" instead of "37.1234567890..."

(Talk relative efficiency of GeoJSON with clipped precision; show Rainbow Road map)

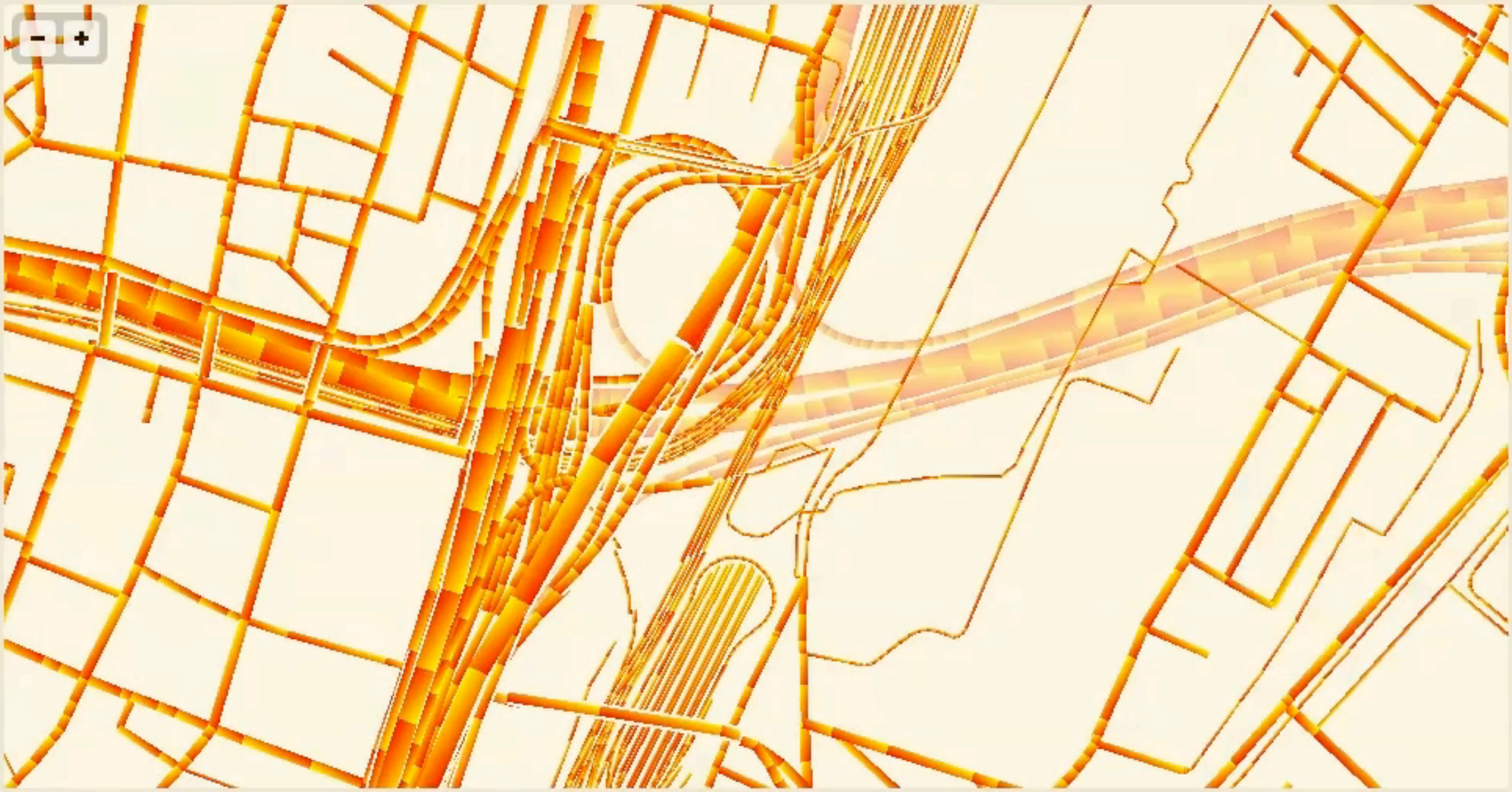
http://teczno.com/s/m0x



GeoJSON tiles + WebGL, sorted.

http://teczno.com/s/hjk

Debugging Boston's z-index



Michal Migurski, Feb 2013.

http://teczno.com/s/djb

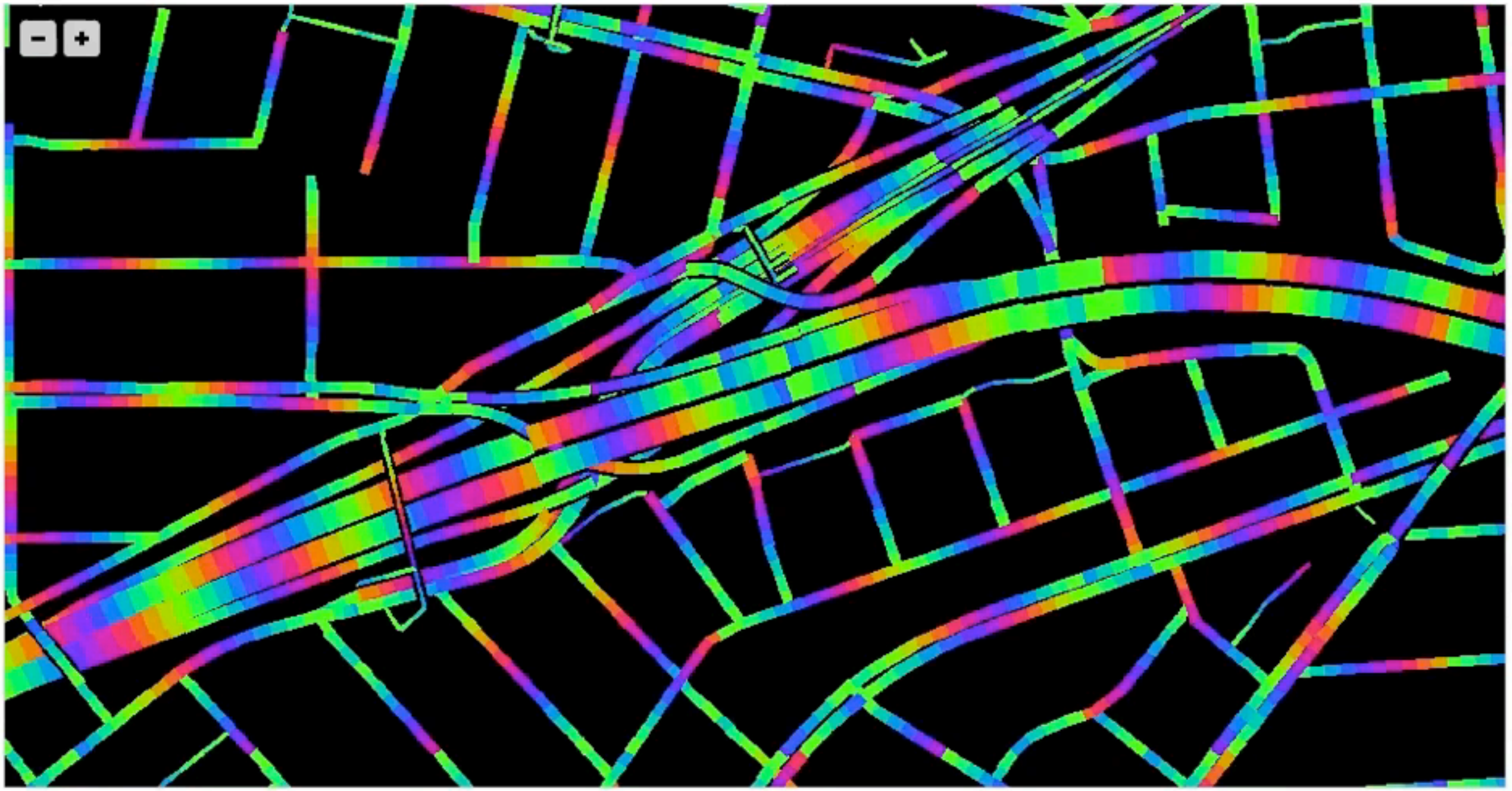
bl.ocks.org/migurski/5130639

Google Console | AWS Console | AWS Chicago

migurski's block #5130639

GL-Solar, Rainbow Road edition

March 15, 2013



A thing made with Squares, WebGL demo.

Open in a new window.

Browser technologies like WebGL are an obvious way forward for showing this stuff. Google has had GL in its maps for a year now, though it's a useless gimmick from our point of view without a data API.

WKB data with clipped precision also compresses well

E.g. replace three bytes in each double-precision floating point value with 0x00, then zlib that

(Talk MVT specifically)

Small size, easy to parse, easy to describe

Three kinds of efficiency for file formats

Two kinds of efficiency for a file format:

- small size
- easy to parse

A third kind of efficiency:

- easy for a programmer to understand

And so: Mapnik Vector Tiles

teczno.com/s/r3p

With the introduction of Mapnik Python Datasource last year, we can use vector tiles right in our favorite raster renderer.

Currently, Four Layers

- Streets
- Street Labels
- Urban Areas (parks, schools, etc.)
- Water

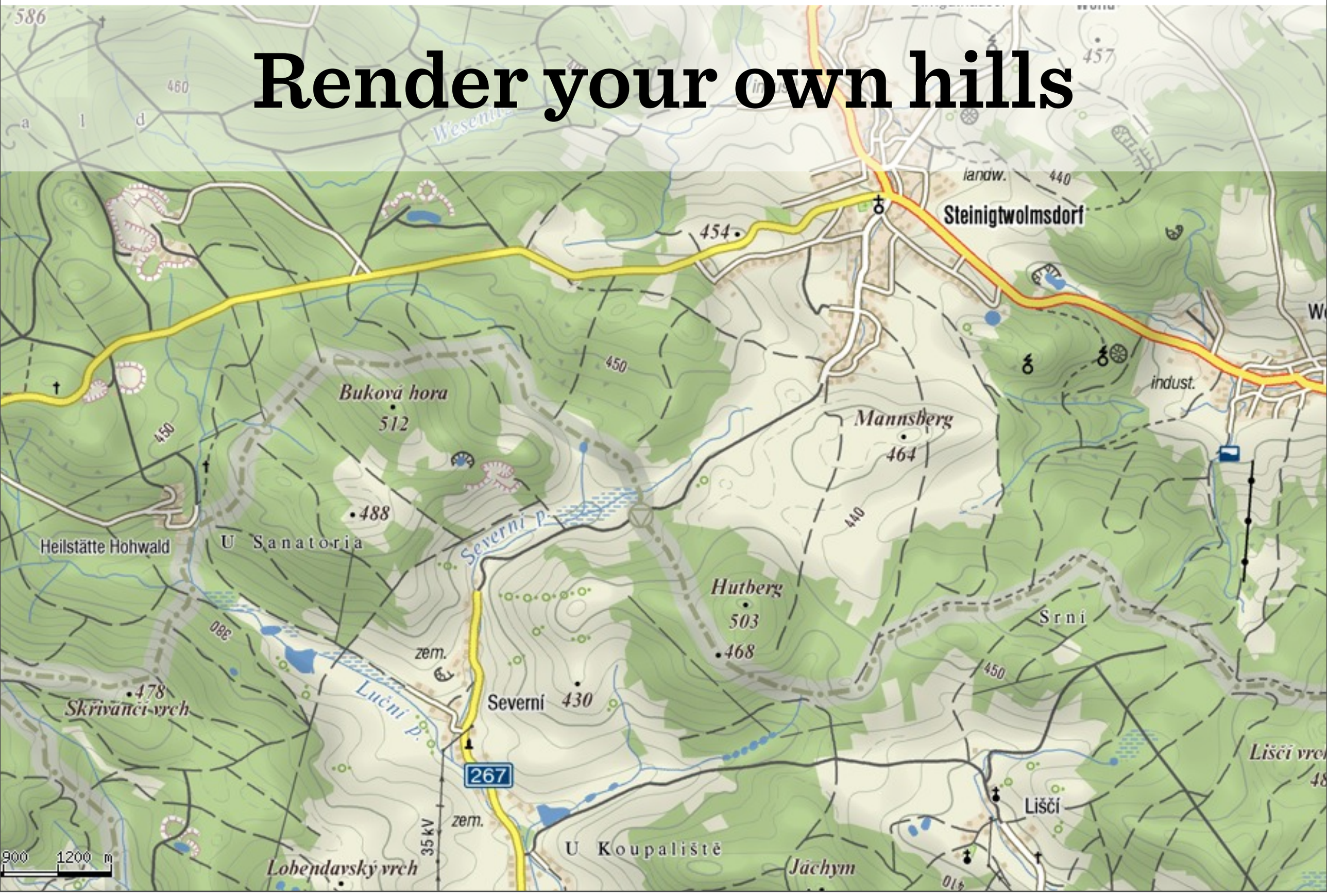
Use with Carto & Mapnik

```
{
 "type": "python",

 "factory":
 "TileStache.Goodies.VecTiles:Datasource",

  "template":
  "http://tile.openstreetmap.us/vectiles-highroad/{z}/{x}/{y}.mvt",
}
```

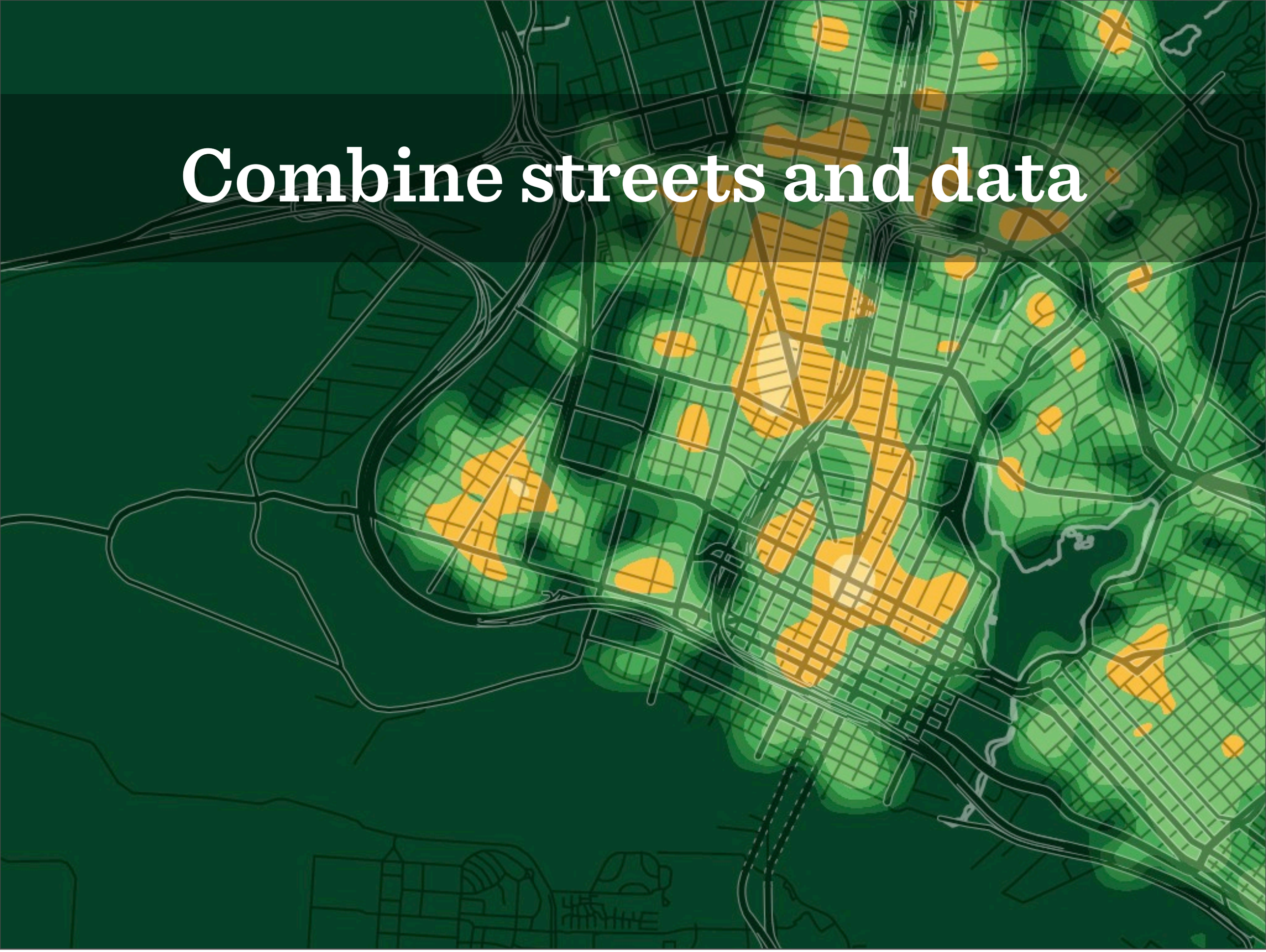
(not Tile Mill, yet)

http://mapy.cz

Render your own hills



Combine streets and data



http://sta.mn/pm

Go nuts.

teczno.com/s/r3p

Thank you.

@michalmigurski

mike.teczno.com