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.
We’re living in an explosion of cheap, easy-to-get geographic data on the internet.
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.
“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.
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.
PostGIS is not getting any easier to administer, especially if you’re a newspaper or a civic data gadfly.
or: Things you could have found out yourself if only it didn’t take so damn long to try them!
OSM data is getting bigger, more difficult, and slower to handle.
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.
Time for vector tiles!
(tl;dr: teczno.com/s/r3p)

Why are vector tiles right for now?
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.
They’ve worked well before; we used BSON & PVR for iOS

(talk GeoJSON vs. BSON + National Geographic here? backend infrastructure of that project)
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)
GeoJSON tiles + WebGL, sorted.
Debugging Boston’s z-index

Demo of [Squares](http://teczno.com/s/hjk) showing OpenStreetMap centerlines loading via tiled GeoJSON (sample), and rendered with WebGL and Canvas. Pan, scroll and mousewheel zooming are all enabled, and you can right-click on maps to edit layer properties at any zoom level. Chevron line controls allow you to view other edges as well. Double-click to go to my blog on OpenLayers and GeoJSON.
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*
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
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)
Render your own hills

http://mapy.cz
Combine streets and data
Go nuts.
Thank you.

@michalmigurski
mike.teczno.com