2, 4, 6, 8
Here's how we Interpolate

Julian Simioni
Mapzen
@juliansimioni

I’m Julian and I work at Mapzen. You don’t need to know anything else because a talk shouldn’t really be about the presenter :P

This background image though, IS really cool. Its a 1970s photo of two US Census employees presumably looking at something very important on a map. We'll see how the census relates to what we'll be talking about in a minute. Also lets take a moment to appreciate the 1970s styles going on here.

github.com/pelias/pelias/

I work on the open-source Pelias geocoder and today we're going to talk about a particularly cool feature we added not quite a year ago.

What's a geocoder? Basically, its the software that makes the search box on a map work. You type in the name of a place and it helps you find it. It does a lot behind the scenes of course :)

Pelias

addresses

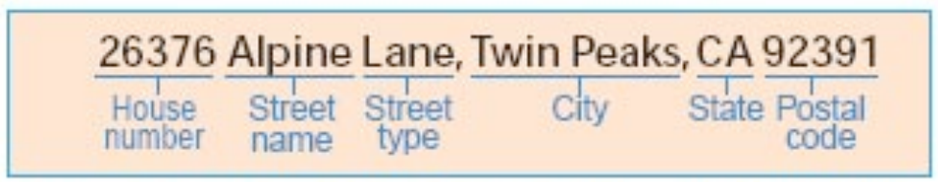
Here on the Pelias team, like all geocoders, we LOVE addresses. We want to collect them all and give each one special care so that people looking for it can find the place it represents.

Cool things about addresses

But why are addresses so cool? After all there are plenty of other ways to search on a map. You can enter the name of the place, a latitude and longitude, and other stuff.

Cool things about addresses

Addresses are structured!



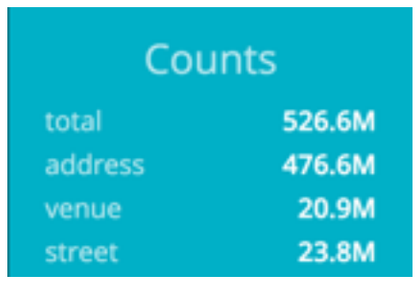
A common address format in the United States, consisting of the basic address elements

Addresses are structured! They don’t just tell you where a place is, but where it is in relation to other places like a street or a city or a country.

Cool things about addresses

Addresses are structured!

Addresses are numerous!



http://pelias-dashboard.mapzen.com

There are tons of addresses. This is a screenshot from our Pelias build dashboard, we index almost half a billion addresses all over the world from OpenStreetMap and OpenAddresses, which are both amazing and rapidly growing open data projects.

Not as Cool things about addresses

Okay, but there are some things that aren’t quite as cool

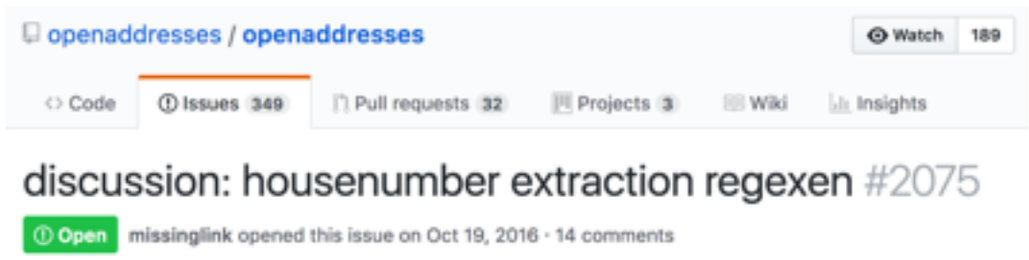
Not as Cool things about addresses

Addresses are structured…

Not as Cool things about addresses

Addresses are structured **differently everywhere**

Addresses everywhere in the world are different. Some have house number first, some have street name first. Some streets don’t have name. Some places don’t have housenumbers. Almost every place does things a little bit differently.



https://github.com/openaddresses/openaddresses/issues/2075

How do we solve this? Well. there’s never going to be an easy answer. Basically its going to take a lot of hard work and a lot of regular expressions.

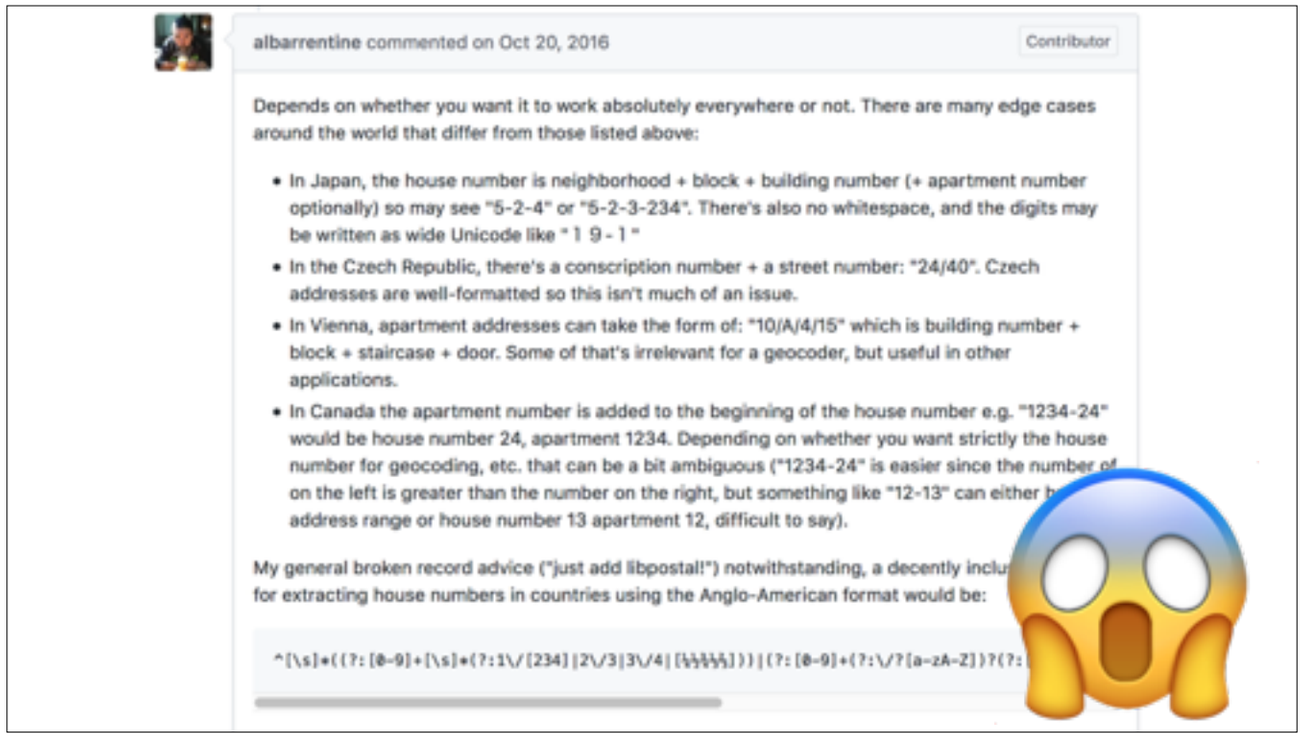
albarrentine commented on Oct 20, 2016 Contributor

Depends on whether you want it to work absolutely everywhere or not. There are many edge cases around the world that differ from those listed above:

- In Japan, the house number is neighborhood + block + building number (+ apartment number optionally) so may see "5-2-4" or "5-2-3-234". There's also no whitespace, and the digits may be written as wide Unicode like "１９－１"
- In the Czech Republic, there's a conscription number + a street number: "24/40". Czech addresses are well-formatted so this isn't much of an issue.
- In Vienna, apartment addresses can take the form of: "10/A/4/15" which is building number + block + staircase + door. Some of that's irrelevant for a geocoder, but useful in other applications.
- In Canada the apartment number is added to the beginning of the house number e.g. "1234-24" would be house number 24, apartment 1234. Depending on whether you want strictly the house number for geocoding, etc. that can be a bit ambiguous ("1234-24" is easier since the number of on the left is greater than the number on the right, but something like "12-13" can either be an address range or house number 13 apartment 12, difficult to say).

My general broken record advice ("just add libpostal!") notwithstanding, a decently inclusive regex for extracting house numbers in countries using the Anglo-American format would be:

```
^[\s]+((?:[0-9]+[\s]+(?:1\/[234]|2\/3|3\/4|[½⅓¼⅕⅙⅐⅛⅑⅒]))|(?:[0-9]+(?:\/?[a-zA-Z])?(?:[\-\/][0-9
```



Oh god…

They say if you try to solve a problem with regular expressions, now you have two problems. Well, if we have a lot of regular expressions…

Not as Cool things about addresses

Addresses are structured differently everywhere

Okay, what else?

Not as Cool things about addresses

Addresses are structured differently everywhere

Addresses are numerous…

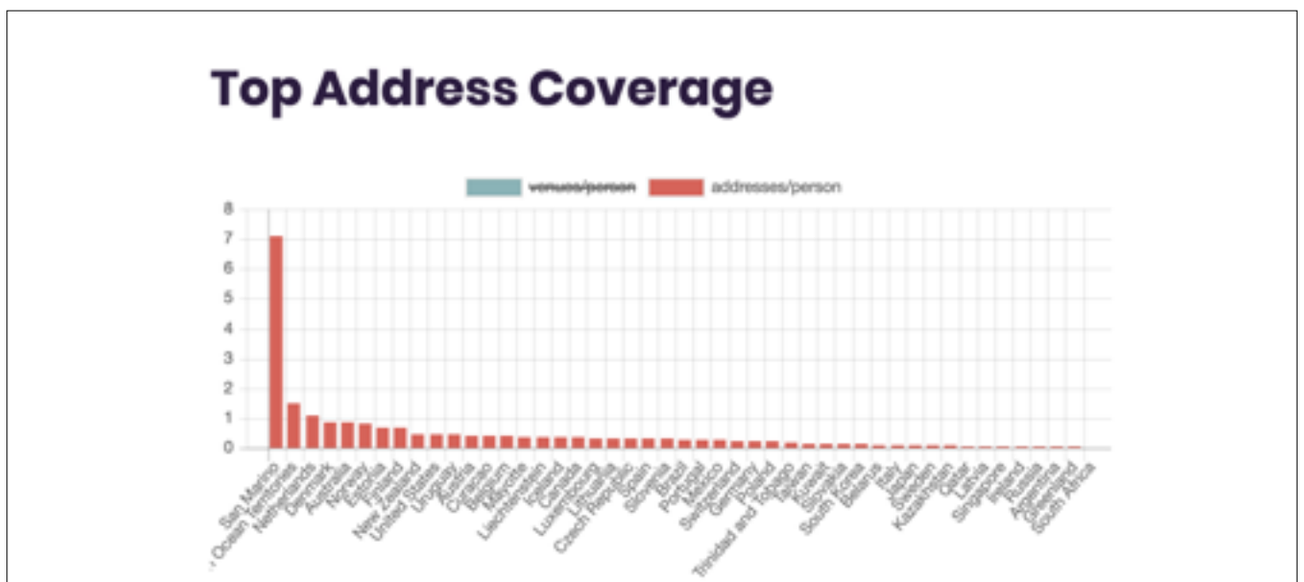
There _are_ tons of addresses…

Not as Cool things about addresses

Addresses are structured differently everywhere

Addresses are ~~numerous~~ **not numerous enough**

But not nearly enough.



https://pelias.github.io/scripts-geocoding-coverage/highlights.html

People who know about such things estimate that most parts of the world have an average of 2 people per address, or half an address per person.

This is a graph of our data analysis for Pelias data coverage. Only a few countries in the world have even close to half an address per person. Except, oddly, San Marino, where they're doing great. Good job San Marino!

We on the Pelias team are constantly looking for ways not just to increase our coverage with new data, but for ways to make our existing data go further. One of the datasets we've been inspired by is…

Enter…TIGER. Its a dataset published by…The US Census Bureau. Its a dataset of well…a lot of things. Besides an amazing retro logo, that dataset contains tons of stuff useful for mapmaking.

Anyone remember what TIGER stands for?

Its Topologically Integrated Geographic Encoding and Referencing

The TIGER dataset has a long history. Its been around in some form or another for decades. Back in the 70s it took 1300 people hand drawing maps to manage the dataset. Now they probably have someone with QGIS.

This is a photo from their facility in Jeffersonville, Indiana

History lover distraction links:

http://census.maps.arcgis.com/apps/MapJournal/index.html?appid=2b9a7b6923a940db84172d6de138eb7e

https://www.pinterest.com/uscensusbureau/census-history/

There’s tons of cool history here if your interested. They have some cool ESRI map journal things and, obviously, a… Pinterest page?

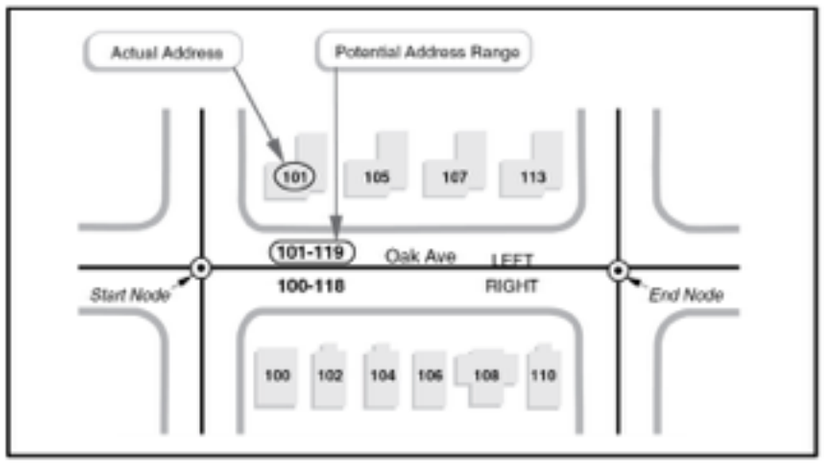
What’s cool about TIGER?

So what’s so cool about this TIGER dataset for us anyway?

What's GRRREAT about TIGER?

Obligatory joke… sorry

Address Ranges

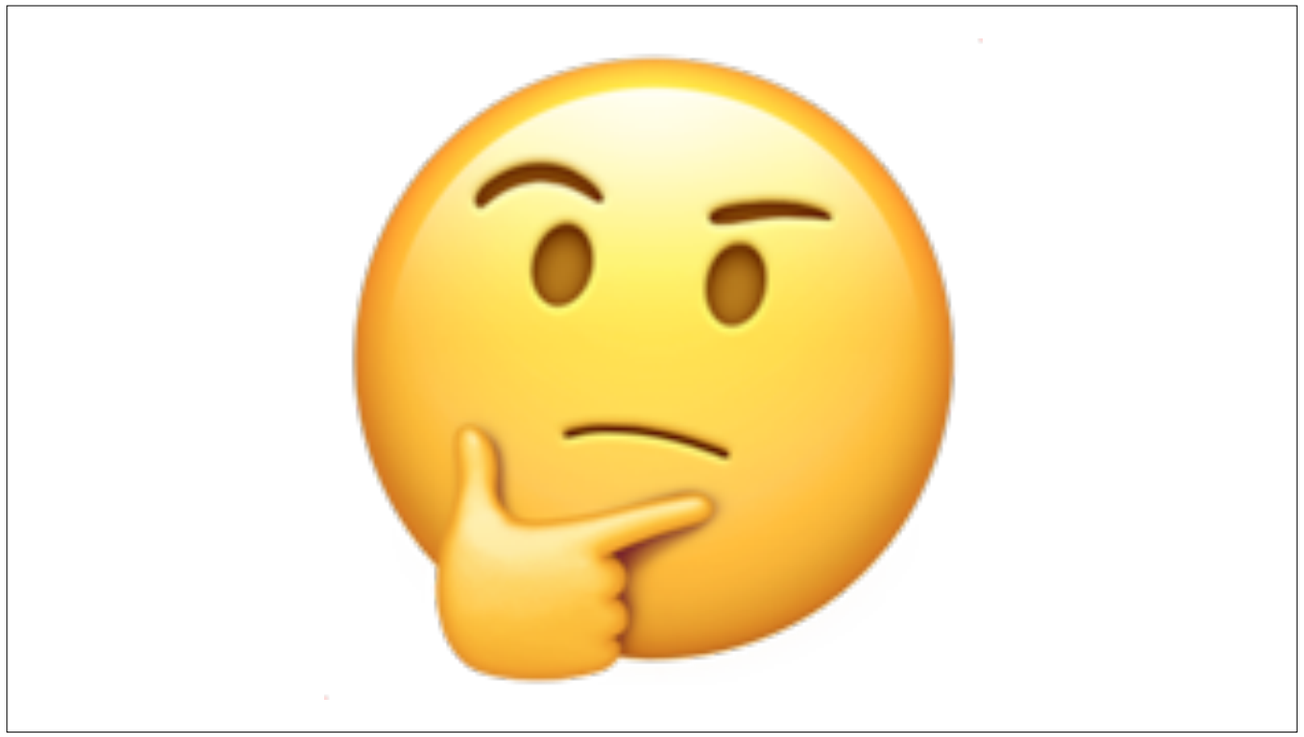


https://www2.census.gov/geo/pdfs/maps-data/data/tiger/tgrshp2015/TGRSHP2015_TechDoc.pdf

Address ranges! Address ranges are an amazing way of fairly accurately and very comprehensively storing lots of addresses.

Basically, you take a list of streets, their shapes, and their names, and annotate them with what the ranges of the house numbers are for each part of the street. Its not perfect, but it lets you estimate to a pretty decent level where any address on that street would be.

TIGER has address ranges for every single part of the United States. Yes. All 50 states, all the territories. Find the tiniest town in the middle of nowhere, and as long as it can get mail delivered via the postal service, it will be in the TIGER address range dataset.



Okay, so that covers the United States. But we said we wanted a geocoder thats useful all over the globe. A few other countries have detests comparable to TIGER, but most don’t.

There must be something we can do with existing open data that can help us.

Lets see…first we’ll need some street geometry

Oh hey. OpenStreetMap has great street geometry all over the world. Okay, next we need address ranges. Well…we just said we don’t have those. BUT! We also just said we have lots of addresses!

Yeah, both OpenAddresses and OpenStreetMap have tons of addresses. What if we could try to estimate what the address ranges might be using that data?

It might not be perfect, but it would be…

Good enough. And we could keep making it better over time.

https://github.com/pelias/interpolation

Okay, so we did that. And it turned out to be not too bad. This is the Github repo, because it’s open source. Our interpolation engine takes streets from OpenStreetMap, addresses from OpenStreetMap and OpenAddresses, and even throws in those address ranges from TIGER just for good measure.

So lets take a look at the results. This is our interpolation demo and debugging interface, pointed at a street you should probably recognize, since its centered on the building we're all in.

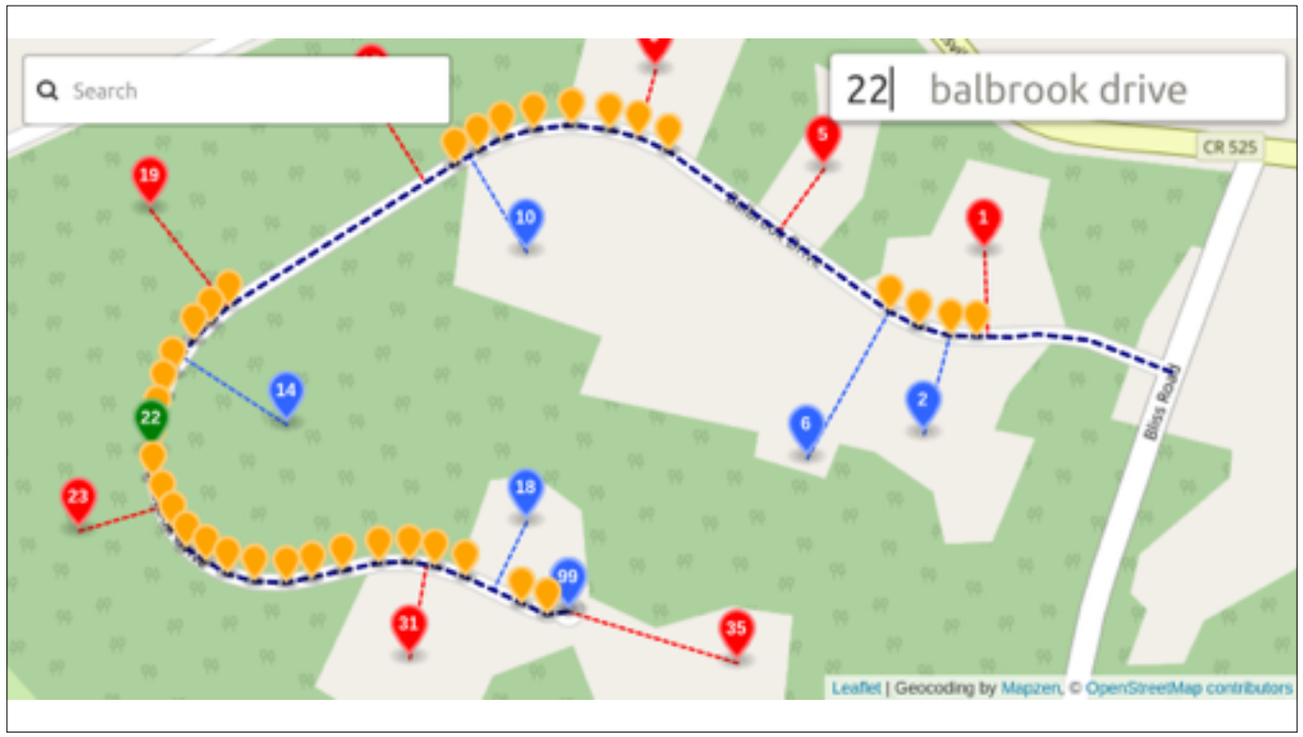
What have we got here? Well, we've got a blue dashed line for the geometry of the street. We've got markers of the different known addresses.

If you click on a marker, you can see what we know about it. This house number came from OpenAddresses. We also determine street parity, which is just what side of the street the address is on. This one's on the right side, like all the other red markers. The blue markers are on the left side. Left versus right is obviously a little arbitrary, but it's useful to keep track of

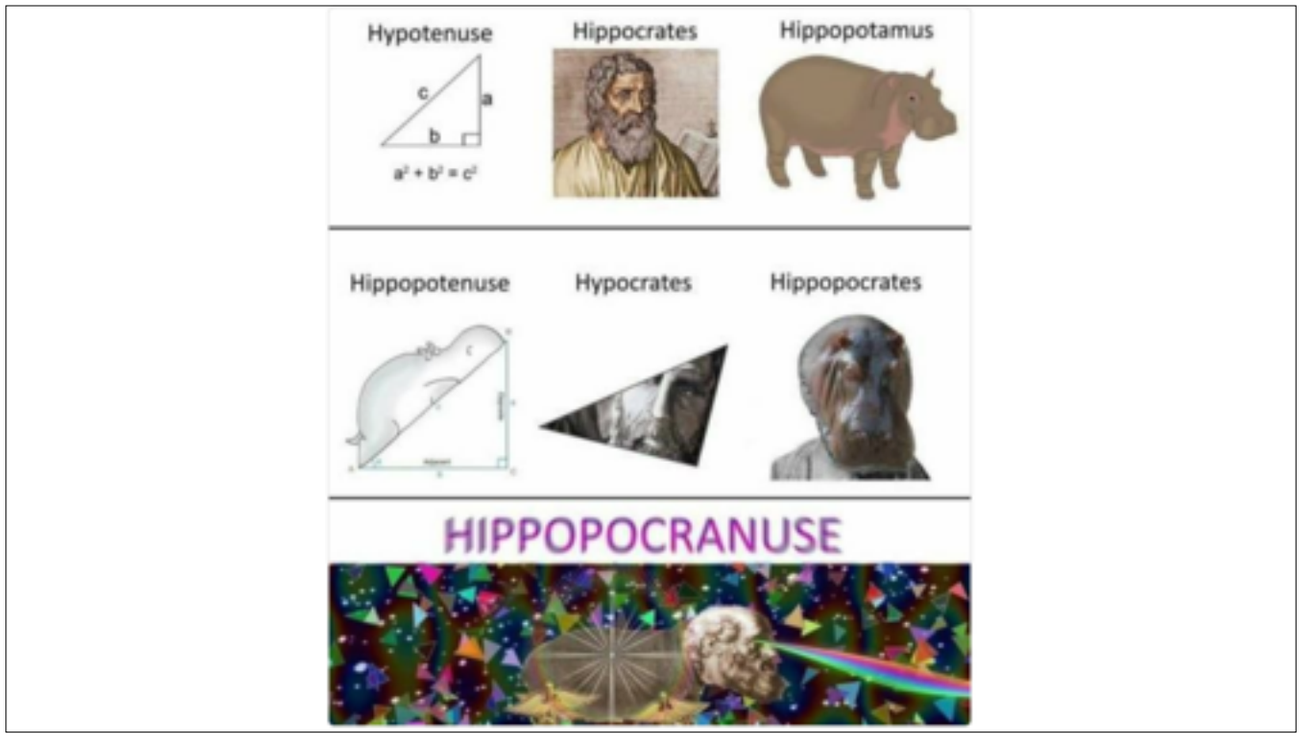
Now, if you enter an address that we _don't_ know about, we'll tell you where we think it is. So the green maker for house number 14, is put somewhere between 16 and 10. On this street we have pretty good coverage, so there probably isn't a 14, but that's ok. There are plenty of places where we won't have everything.

Anyways, as New Yorkers, we may forget sometimes that not every street is a perfectly straight line. But the interpolation engine can't forget.

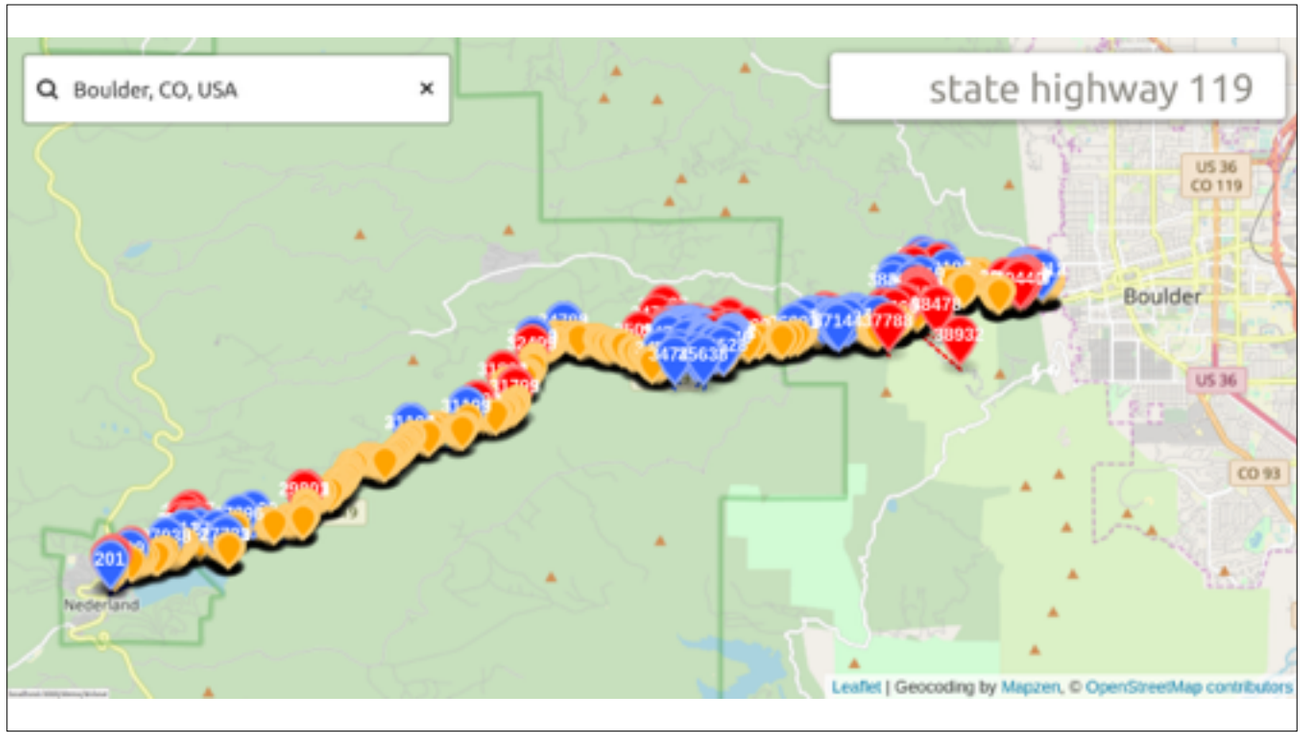
This is a street out in New Jersey somewhere. It curves all the way back on itself.



You can see when we ask for an address in the middle of some that we know about, the interpolation engine places that interpolated address _on the street_. If we did something else, like put it midway between the two nearest addresses, or the two ends of the street, we would get things very wrong. To do this requires….



Lots of math! Okay this slide is just put in here because it’s wonderful. We do have code to deal with hypotenuses, unfortunately we don’t yet have any code concerning Hippocrates or Hippopotamuses.

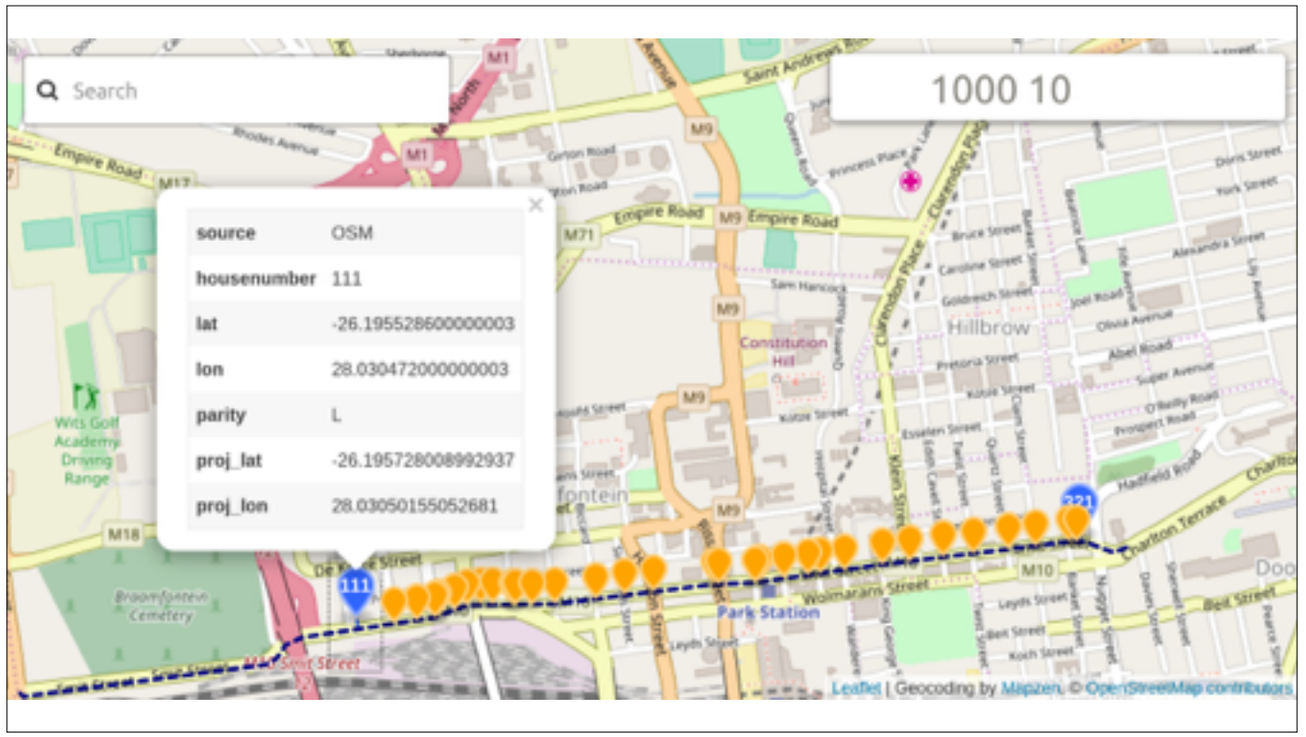


But, for the most part, we've gotten the math right, and this works even for very long, very curvy streets. This is Boulder Canyon Drive out near Boulder, where State of the Map US was just held. Was anyone else there? It was a ton of fun.

Anyways, this street is something like 16 miles long, but it all still works and you can interpolate along it as much as you want.

Again in this case though, you don't need to, because we've got pretty much perfect address coverage in the US. So what about somewhere with not so perfect coverage?

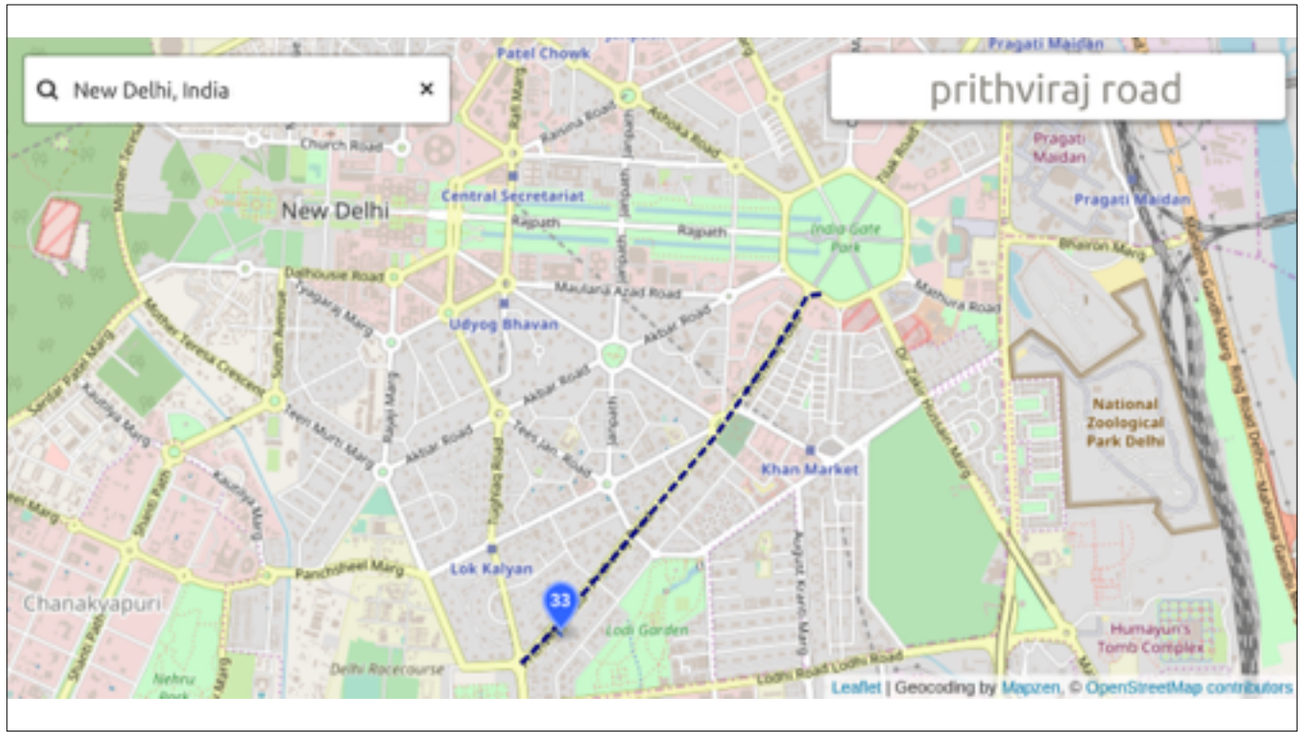
This is the Braamfontein neighborhood in Johannesburg, South Africa. Now Johannesburg, or Joberg if you want to be cool and refer to it how all the locals do, is a pretty big city, something like 8 million people. But as you can see the OpenStreetMap data is pretty sparse. There's obviously buildings and stores and whatnot everywhere in this city, but we don't know about most of them. There's no OpenAddresses data either.



But hey look, on this big street here, there happen to be two addresses defined in OSM. That means…

We can interpolate estimated address positions anywhere in between them! So if we want to guess where 175 might be, it's probably about there. Maybe not exactly, but it should be close enough. This is just so cool. Two little addresses in OSM are giving us a _lot_ of extra coverage.

Here’s another example near New Delhi, India. Two addresses from OSM, pretty long street, interpolation between them. But its worth noting, we can only interpolate in between the two addresses we know about. So that even longer section of the street on the left gets nothing. Oh well.



Here’s another street in New Delhi. Here we only have one address, so we can’t interpolate at all. Bummer! Maybe someone can add an address somewhere else on the street, maybe near that park to the north. Just one, that’s all it takes!

Okay so just in case you’re thinking all the code we’ve written is perfect and we can do no wrong, I have plenty of evidence to the contrary.

So this will be the “1950s rockets exploding montage” part of the talk.

https://github.com/pelias/interpolation/issues/1

Remember when I said address formats are different everywhere. Well we didn't have to go very far to find some differences. Many of you New Yorkers will know about this, but Queens addresses are…special. And of course at first we got it wrong.

This is literally issue number one in our Github repo. And we probably fixed it with a regular expression

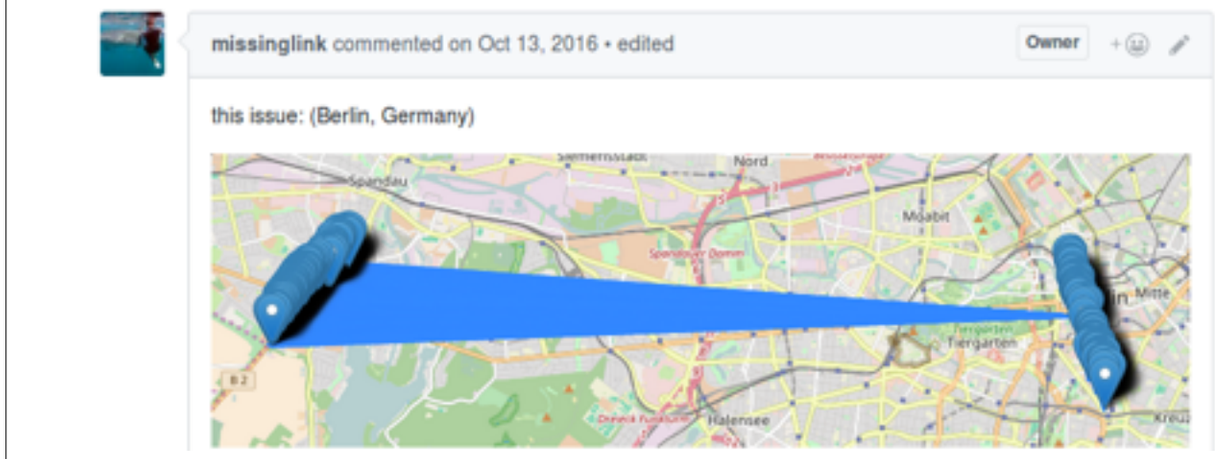
https://github.com/pelias/interpolation/issues/19

Sometimes our code is a little wrong and the data is a little wrong. Here we have one street but we aren't detecting the parity correctly. You can see how the "left" and "right" side of this street swap back and forth. It shouldn't do that.

Here it turned out the roundabouts you see on the left weren't connected to the rest of the street in OSM in a way that we could understand. So we fixed it

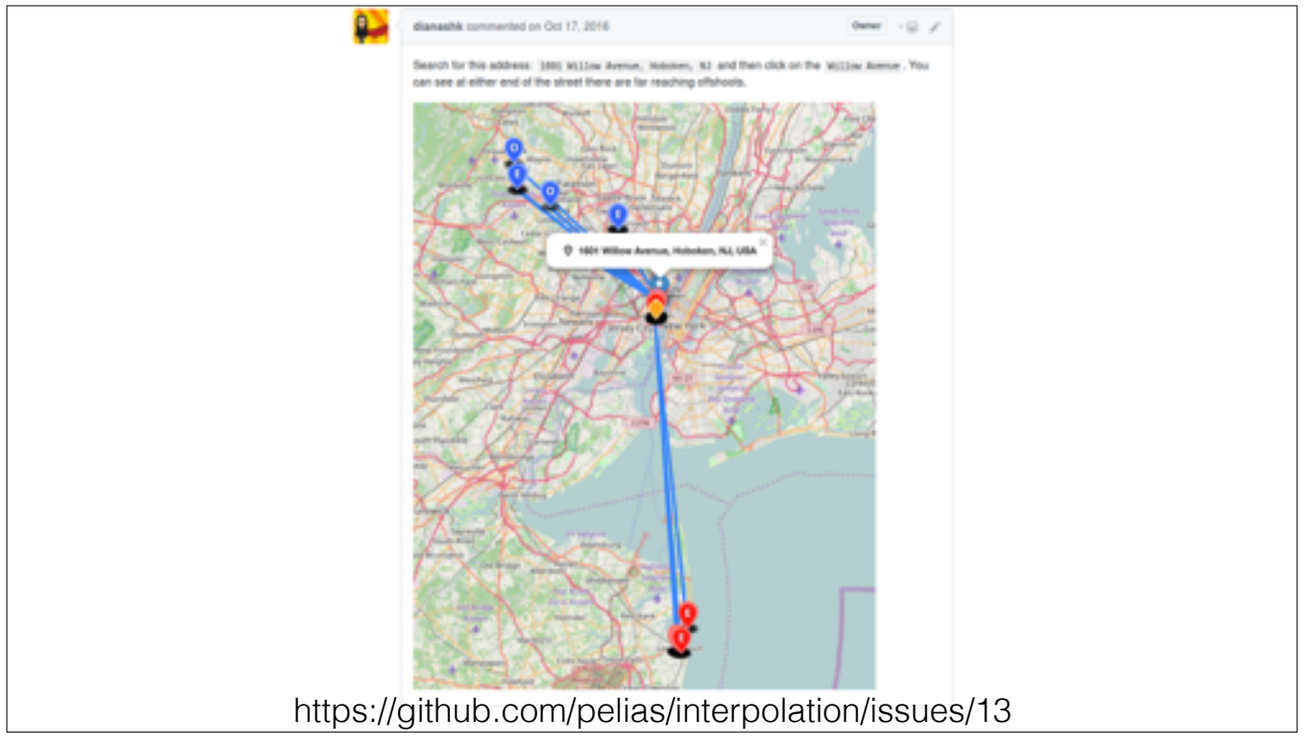
https://github.com/pelias/interpolation/issues/19

Now its all good :)



https://github.com/pelias/interpolation/issues/8

Another thing you learn really quickly when you make a geocoder is that street names are NOT unique. Not even a little bit. Here's a really early screenshot of our interpolation engine thinking that one street in Berlin is half on one side of the city, and half on the other. It's definitely not. Fortunately, in this case the two streets have different postal codes, so we can now tell them apart.



https://github.com/pelias/interpolation/issues/13

Of course, any time there’s an issue somewhere, New Jersey will do it better than anyone else. This is the same problem, but here not all those streets have postal code information. Here I think we work around things by assuming data from different OpenAddresses files is always a different street. Its a little messy but it works.

Sometimes streets are just weird, and we have to account for that weirdness. Here's a street in Berlin called Potsdamer Plaz that intersects with itself. Four. Times. This confused us and we wouldn't interpolate correctly. I don't even remember how we fixed this one.

Okay, so we've done some not so bad stuff, what do we want to do going forward?

Also wow, super cheesy stock photo

Autocomplete

This is a big one. You saw the autocomplete demo at the start of the talk (this is the same one again), and probably entering the names of places letter by letter like a human does is the only way you think to search, because Google has trained us all.

Well, right now you can search for _regular_ addresses with autocomplete in Pelias, but not interpolated addresses.

If you want to search for an interpolated address, you have to enter the entire address. This is okay in some cases, like if you're writing a program to search through a huge list of addresses, but not when there's a human at the keyboard.

This will be a huge endeavor but it's really important, so we're going to do it…eventually.

SPEED

Right now building the interpolation dataset takes 16 days :(

Good software is fast. Right now our interpolation engine is pretty fast when searching, but it takes a long time to go through all the data initially. A long time…

So long that it feels like its running on THIS computer.

16 days is a long time to wait, and if we don’t do anything it will only get longer, because there’s more data coming in all the time.

But again, we’ll make it faster.

By the way this is another census employee doing her awesome job.

https://www.pinterest.com/pin/533535887087899041

What YOU Can Do

- Vote, obviously. Polls close at 9
- Add streets to OSM
- Add street **names** to existing streets
- Add addresses to existing OSM venues
- Extra fancy: add address ranges to streets

So that’s what we’ll be doing, but what can you do?

Vote today, if you can and haven’t, you’ve still got time probably.

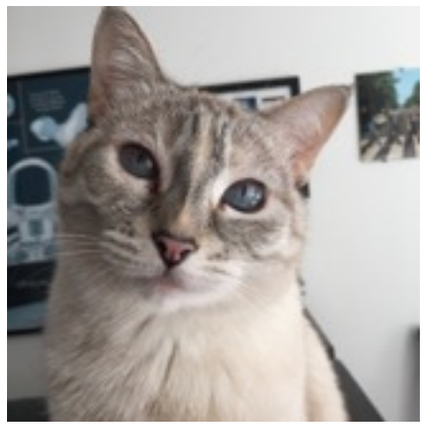
Add more streets to OSM. There are already projects to help with this like Humanitarian OpenStreetMap Team.

Also important is to make sure streets have NAMES. Without names we can’t combine them with address data to make address ranges.

And like I already said, add addresses anywhere you can. Add them as new data, add them to existing data. As we saw just a few can go a long way.

If you want to get really fancy OSM has tag formats for adding address ranges directly, and we support most of them. The OSM Wiki describes them.

Thank You!



twitter.com/juliansimioni

Thank you, and of course, here’s a picture of my cat.