Algorithms for Geospatial Data Matching

An algorithm for error detection and correction of spatial data

Group Members

Surajnath Sidh	U101114FCS146
Shubham Singh	U101114FCS192
Mehak Bhatia	U101114FCS193
Kartik Bhadada	U101114FCS078

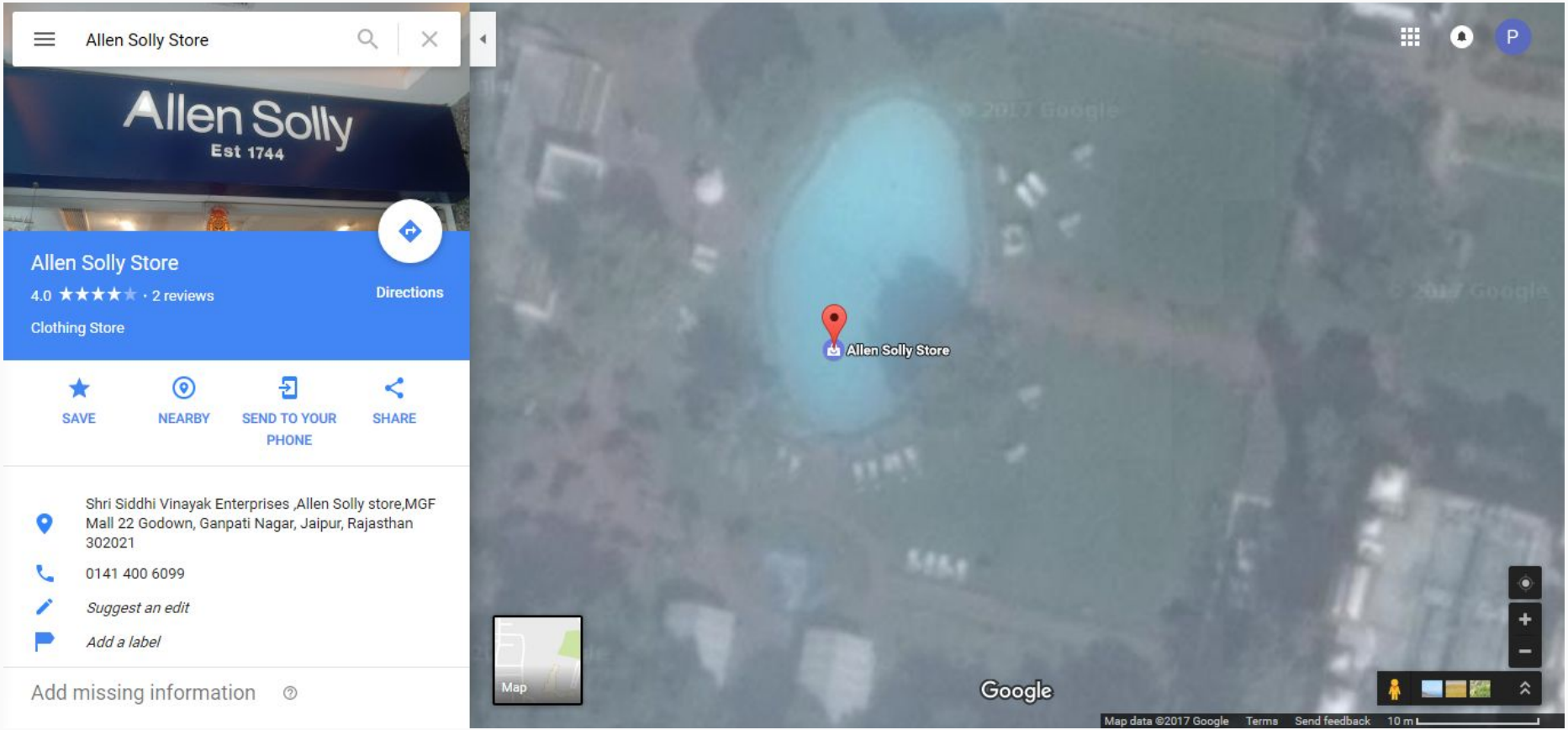
Project Mentor: Dr. Prosenjit Gupta

Rationale and Objectives of the work

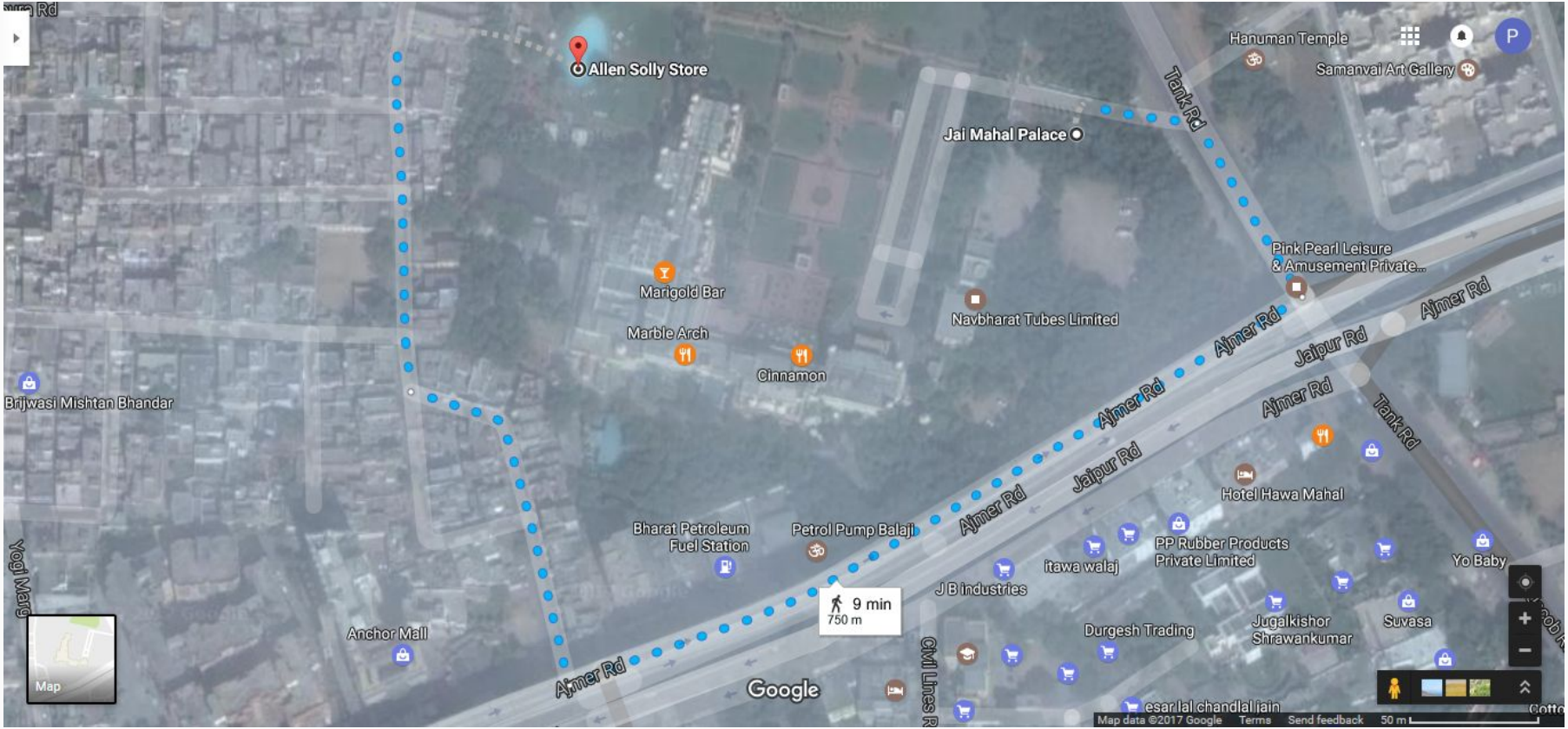
Problem Statement

Correcting Geospatial data using the VGI (Volunteered geographic information) data

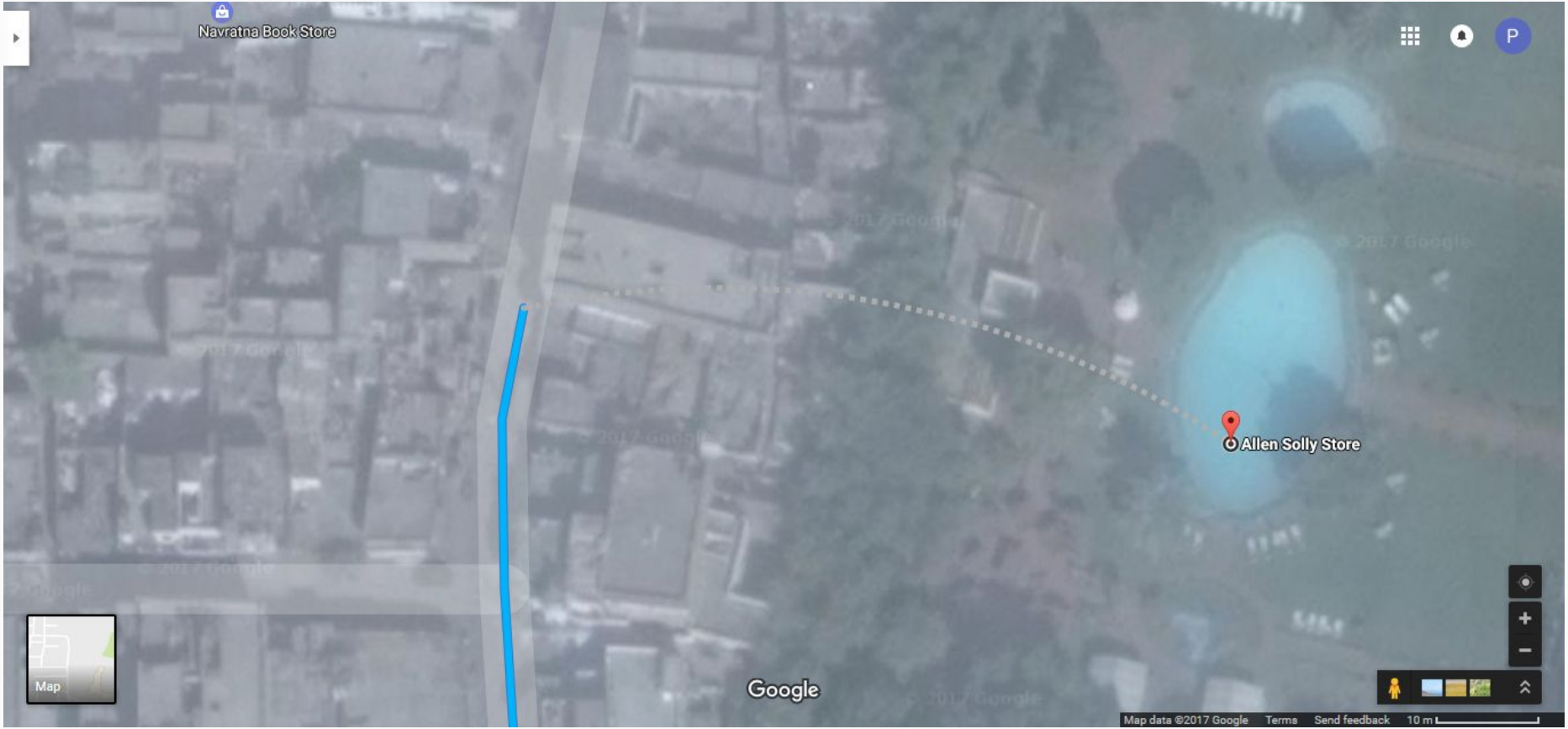
Real life Example of Problem



A clothing store in a swimming pool, *Allen Solly Store Google Maps accessed on 19 Mar 2017*



Let’s see how we can reach this store



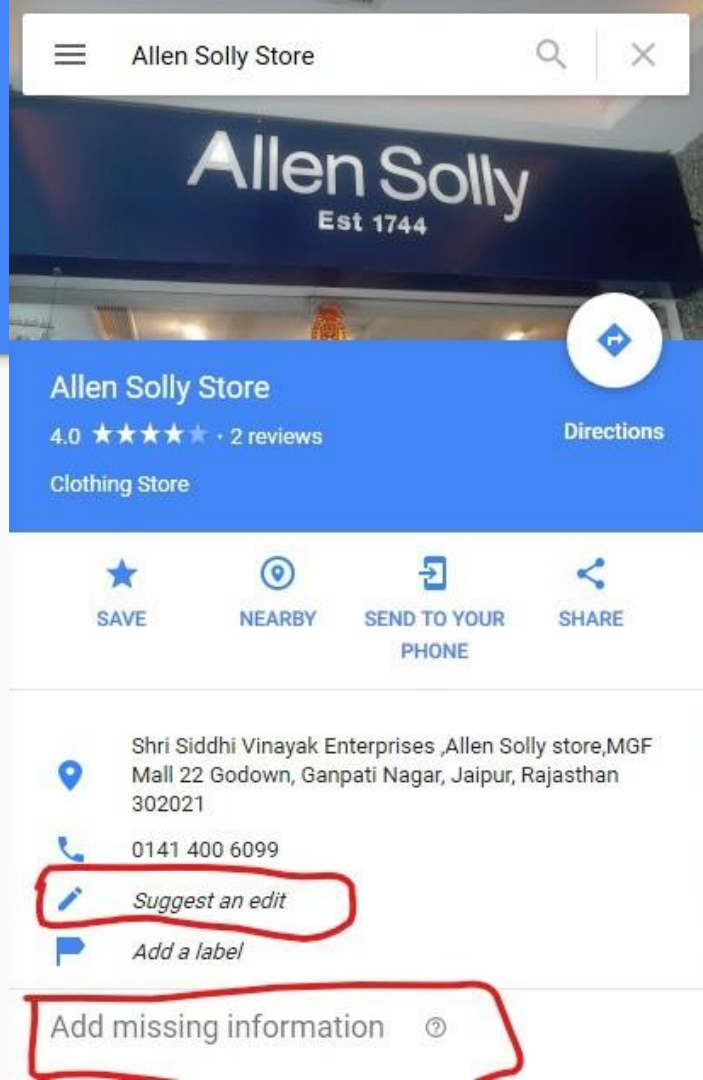
So we have to go through these buildings to reach our desired location

Why This Happened?

Due to incorrect location data of that place

Current Solution

Manually edit these places by crowdsourcing this editing process



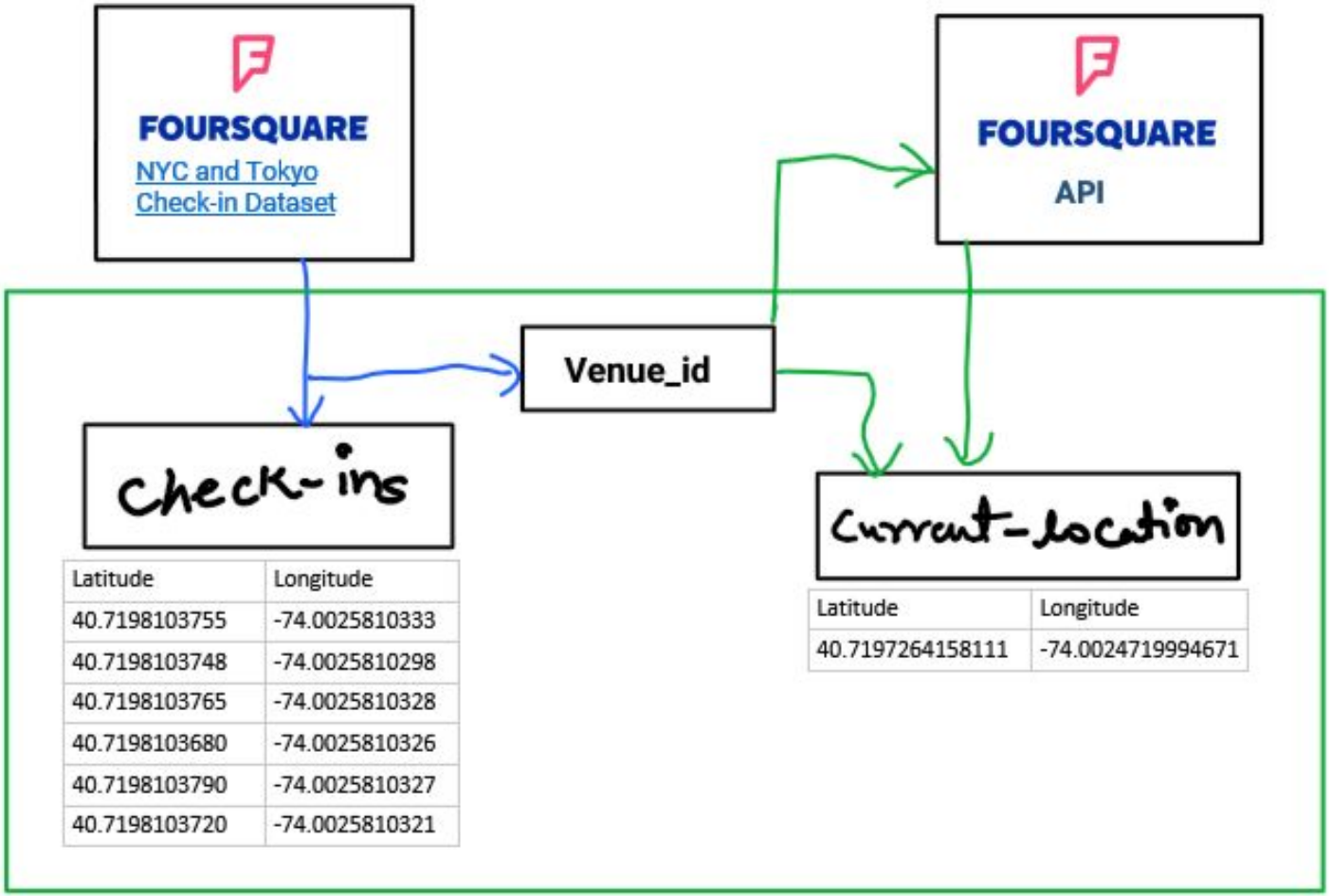
Current Solution

Problem with Current approach

- Too much data to edit by hand
- By the rate maps changes with time it’s hard to keep up with it
- Not so reliable

Methodology

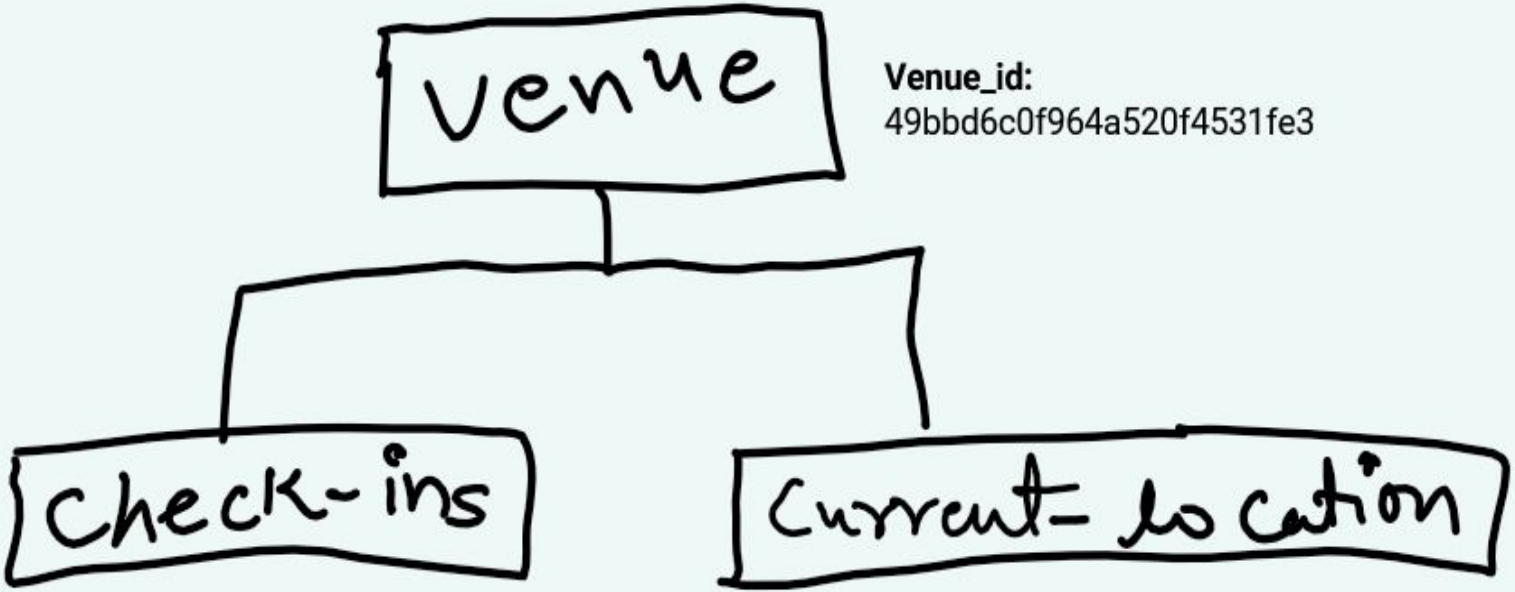
Dataset



Latitude	Longitude
40.7198103755	-74.0025810333
40.7198103748	-74.0025810298
40.7198103765	-74.0025810328
40.7198103680	-74.0025810326
40.7198103790	-74.0025810327
40.7198103720	-74.0025810321

Latitude	Longitude
40.7197264158111	-74.0024719994671

Fig: creation of dataset using NYC and Tokyo Check-in dataset and Foursquare API



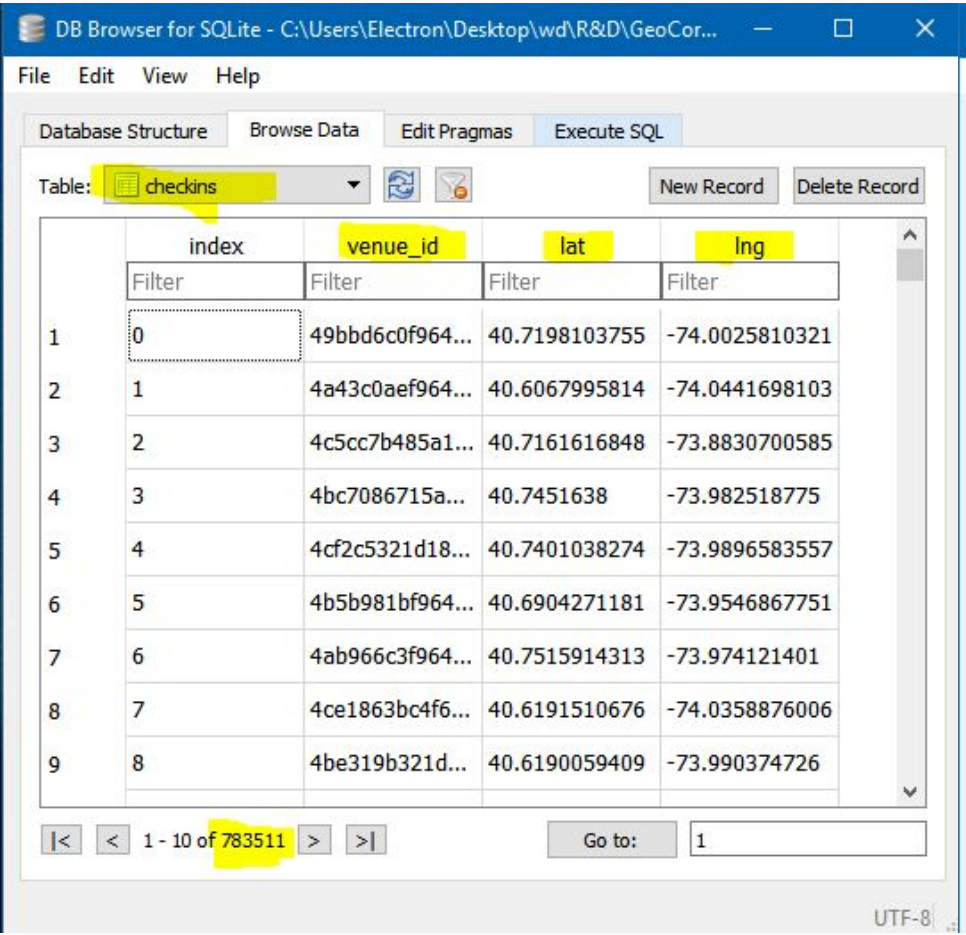
Latitude	Longitude
40.7198103755	-74.0025810333
40.7198103748	-74.0025810298
40.7198103765	-74.0025810328
40.7198103680	-74.0025810326
40.7198103790	-74.0025810327
40.7198103720	-74.0025810321

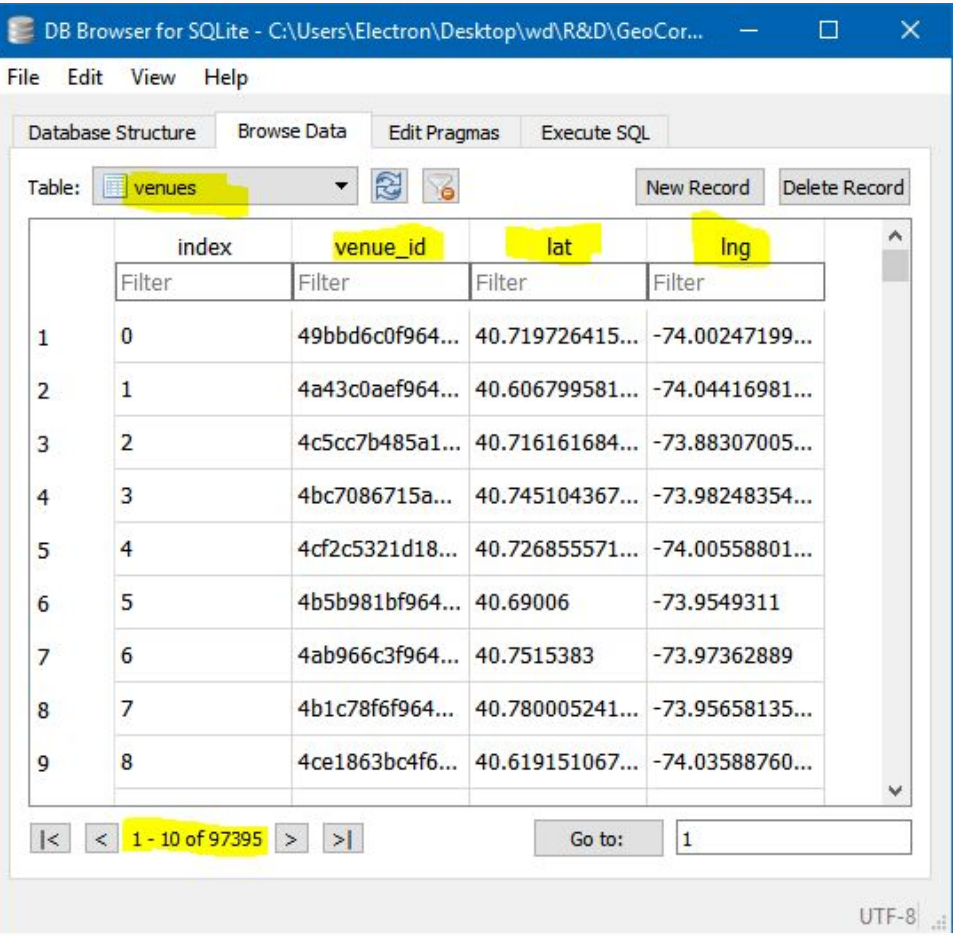
Latitude	Longitude
40.7197264158111	-74.0024719994671

Results of the work completed

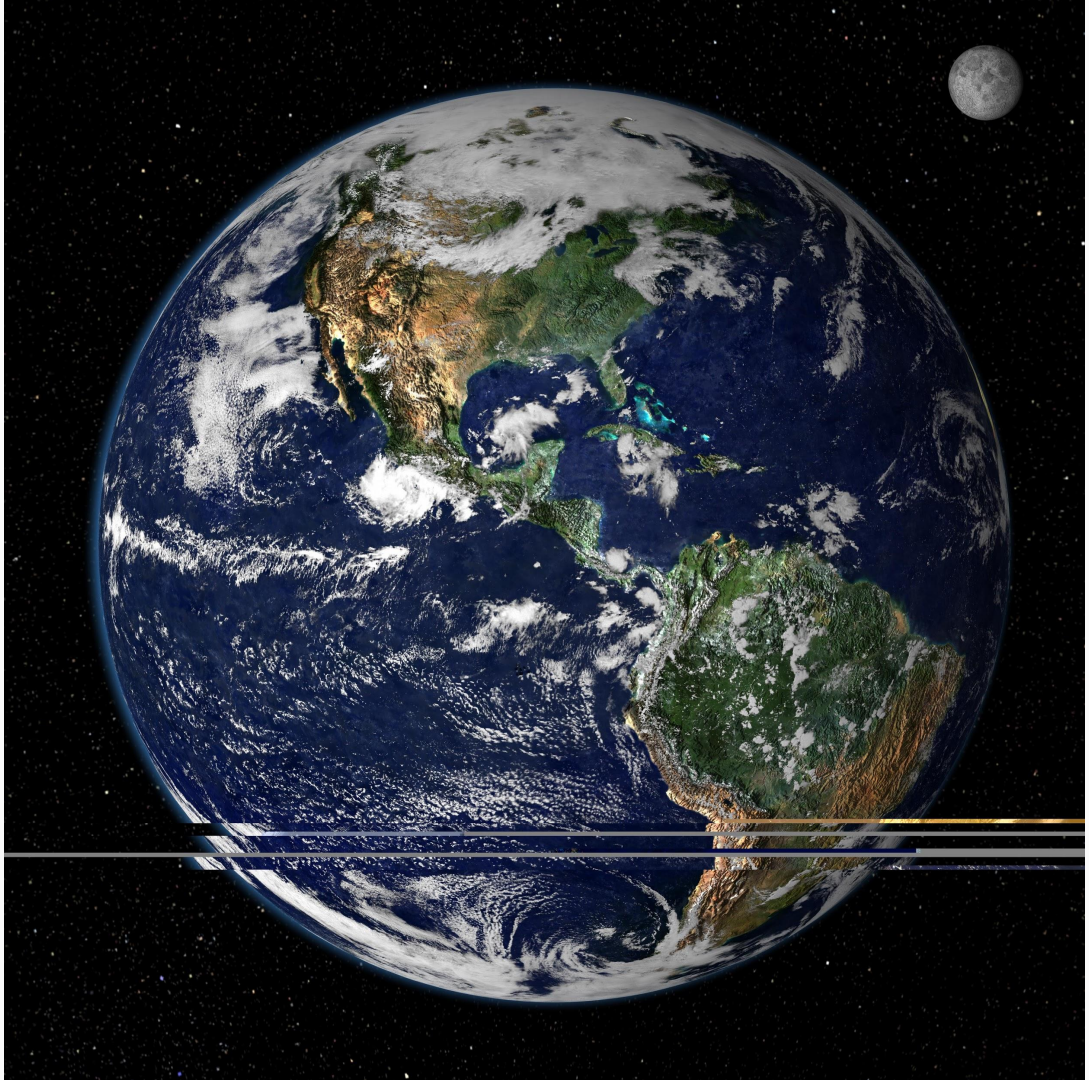
Dataset created

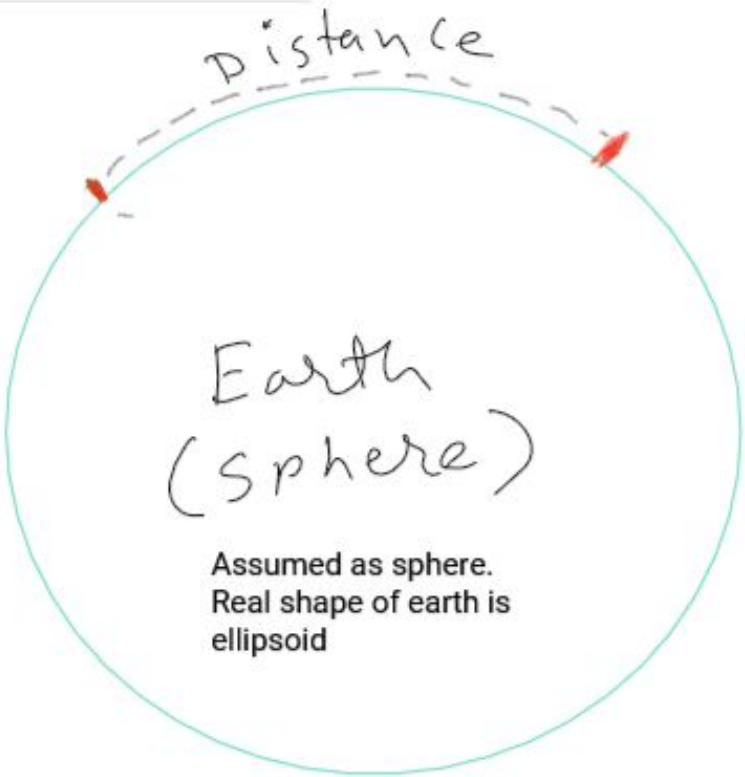
- 783511 Checkins and 97395 Unique locations
- Have Foursquare venue_id which can be used to get more data if required
- Data extraction, cleaning and conversion scripts is written in Python

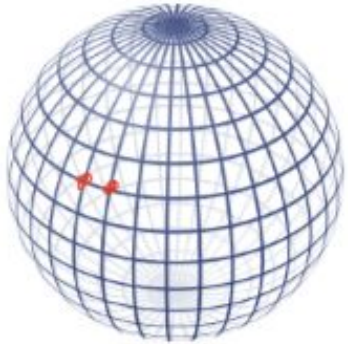

DB Browser for SQLite - C:\Users\Electron\Desktop\wd\R&D\GeoCor...
File Edit View Help
Database Structure Browse Data Edit Pragmas Execute SQL
Table: checkins
New Record Delete Record
index venue_id lat lng
Filter Filter Filter Filter
1 0 49bbd6c0f964... 40.7198103755 -74.0025810321
2 1 4a43c0aef964... 40.6067995814 -74.0441698103
3 2 4c5cc7b485a1... 40.7161616848 -73.8830700585
4 3 4bc7086715a... 40.7451638 -73.982518775
5 4 4cf2c5321d18... 40.7401038274 -73.9896583557
6 5 4b5b981bf964... 40.6904271181 -73.9546867751
7 6 4ab966c3f964... 40.7515914313 -73.974121401
8 7 4ce1863bc4f6... 40.6191510676 -74.0358876006
9 8 4be319b321d... 40.6190059409 -73.990374726
1 - 10 of 783511
Go to: 1
UTF-8


DB Browser for SQLite - C:\Users\Electron\Desktop\wd\R&D\GeoCor...
File Edit View Help
Database Structure Browse Data Edit Pragmas Execute SQL
Table: venues
New Record Delete Record
index venue_id lat lng
Filter Filter Filter Filter
1 0 49bbd6c0f964... 40.719726415... -74.00247199...
2 1 4a43c0aef964... 40.606799581... -74.04416981...
3 2 4c5cc7b485a1... 40.716161684... -73.88307005...
4 3 4bc7086715a... 40.745104367... -73.98248354...
5 4 4cf2c5321d18... 40.726855571... -74.00558801...
6 5 4b5b981bf964... 40.69006 -73.9549311
7 6 4ab966c3f964... 40.7515383 -73.97362889
8 7 4b1c78f6f964... 40.780005241... -73.95658135...
9 8 4ce1863bc4f6... 40.619151067... -74.03588760...
1 - 10 of 97395
Go to: 1
UTF-8

Projection

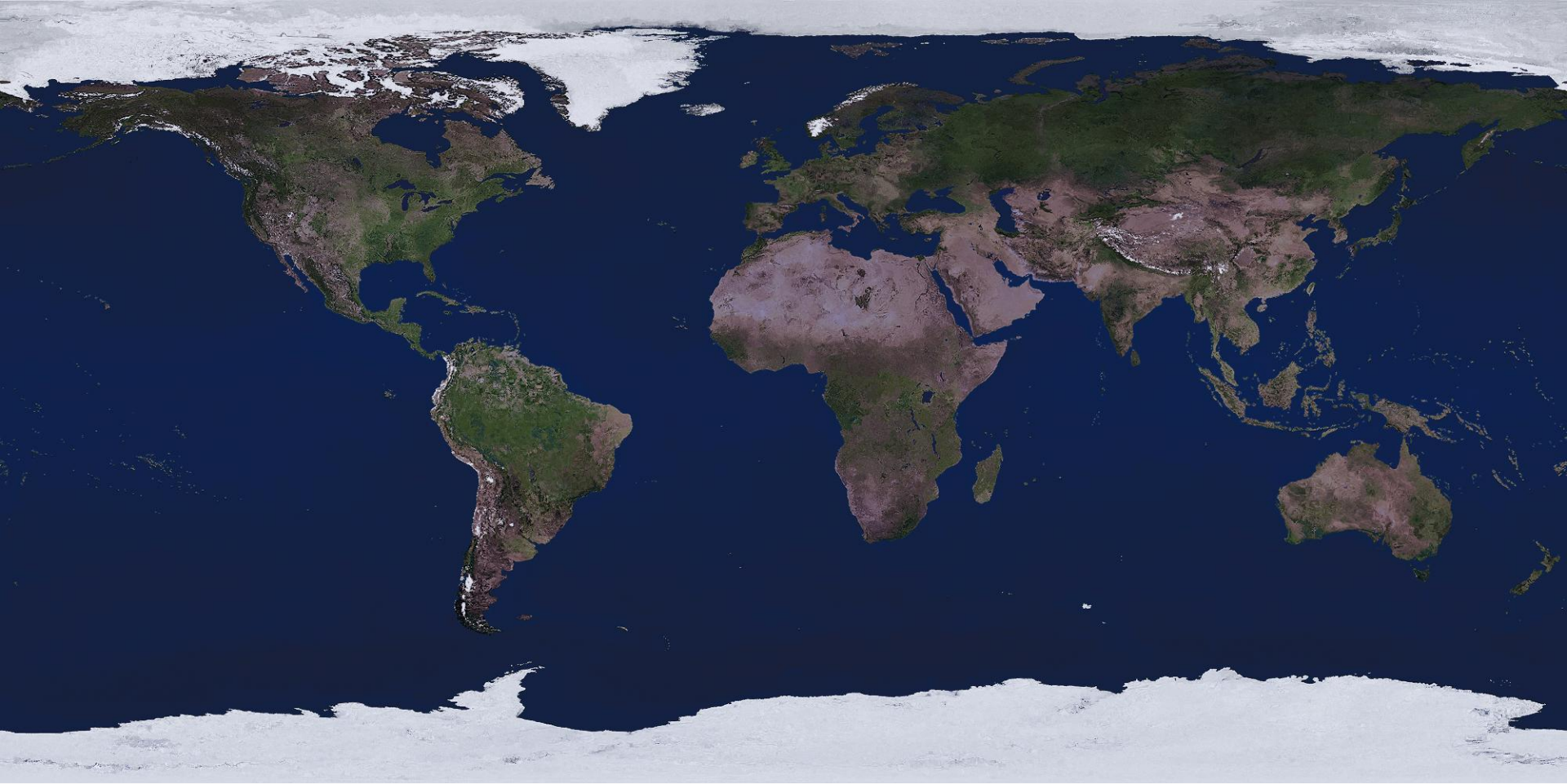




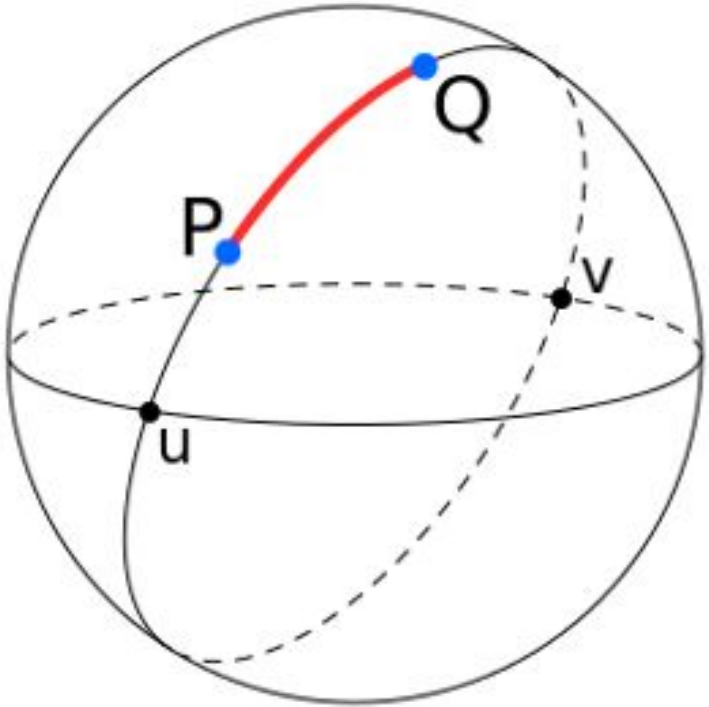


The Earth's equatorial radius *a*, or semi-major axis, is the distance from its center to the equator and equals 6,378.1370 km
The Earth's polar radius *b*, or semi-minor axis, is the distance from its center to the North and South Poles, and equals 6,356.7523 km

Mercator projection



Mercator projection used by map makers to project earth on flat plane



great-circle distance (drawn in red) between two points on a sphere, P and Q

Algorithm

1. Project 3-D space with check-ins to 2-D using great circle distance
2. Make the min(x), min(y) of all points as origin, this would result in a single quadrant
3. Project points on X and Y as last algorithm
4. Find median of X and Y, this will give us the new point.
5. Find that point and corresponding coordinates in dataset and find the coordinates latitude and longitude of those X and Y points
6. These coordinates of X and Y will be new corrected location

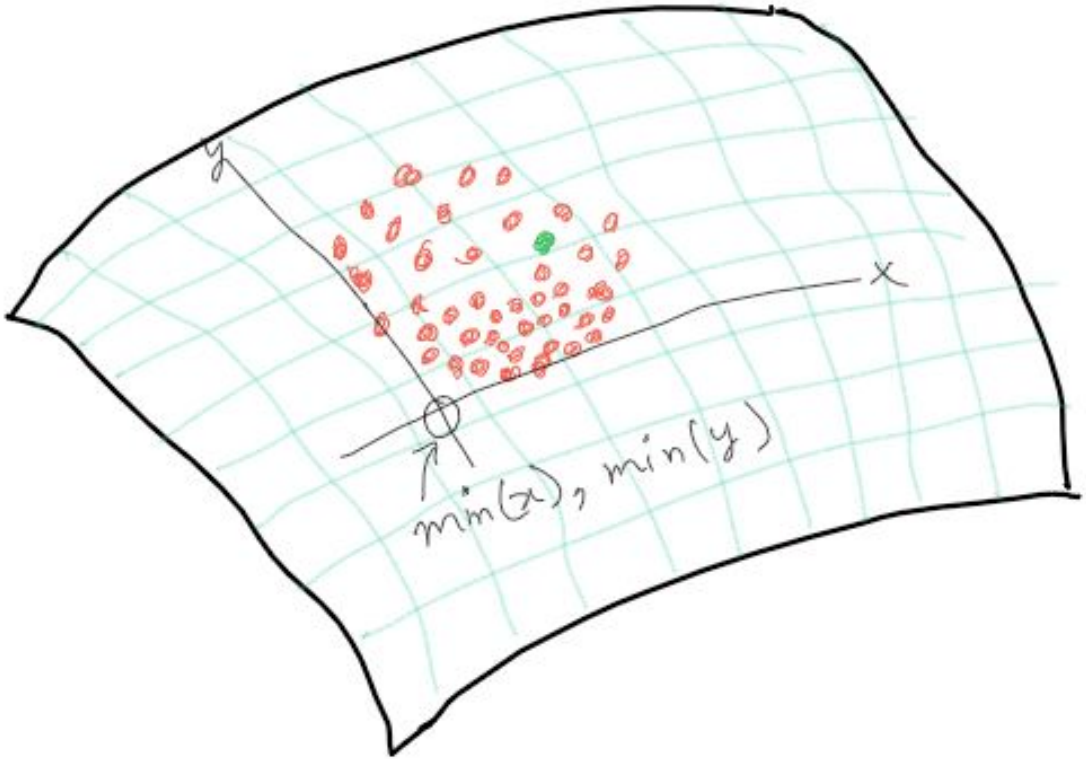


Fig: 3D cross-section of area is divided in quadrants based on current-location(center- *min(x), min(y)*

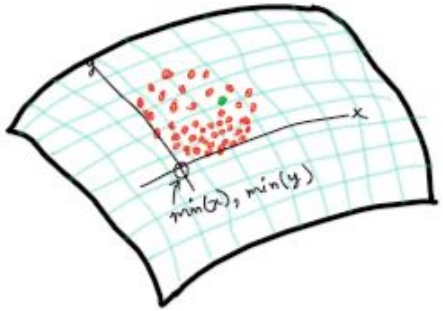
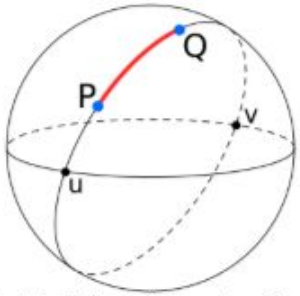
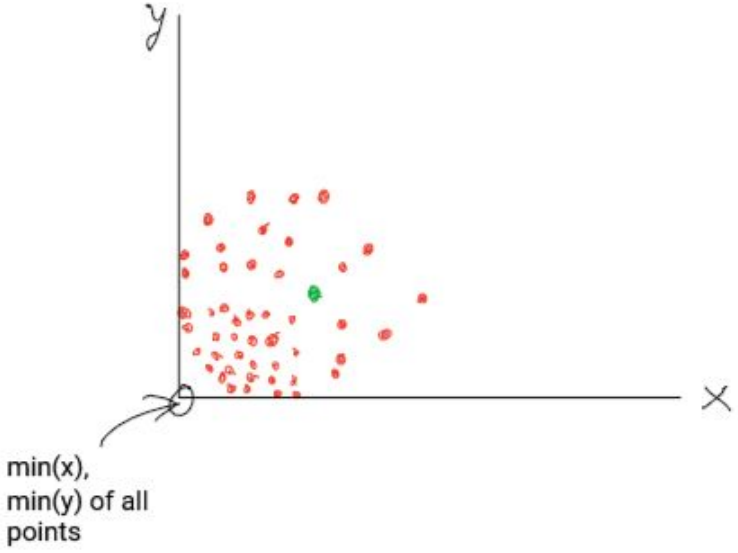
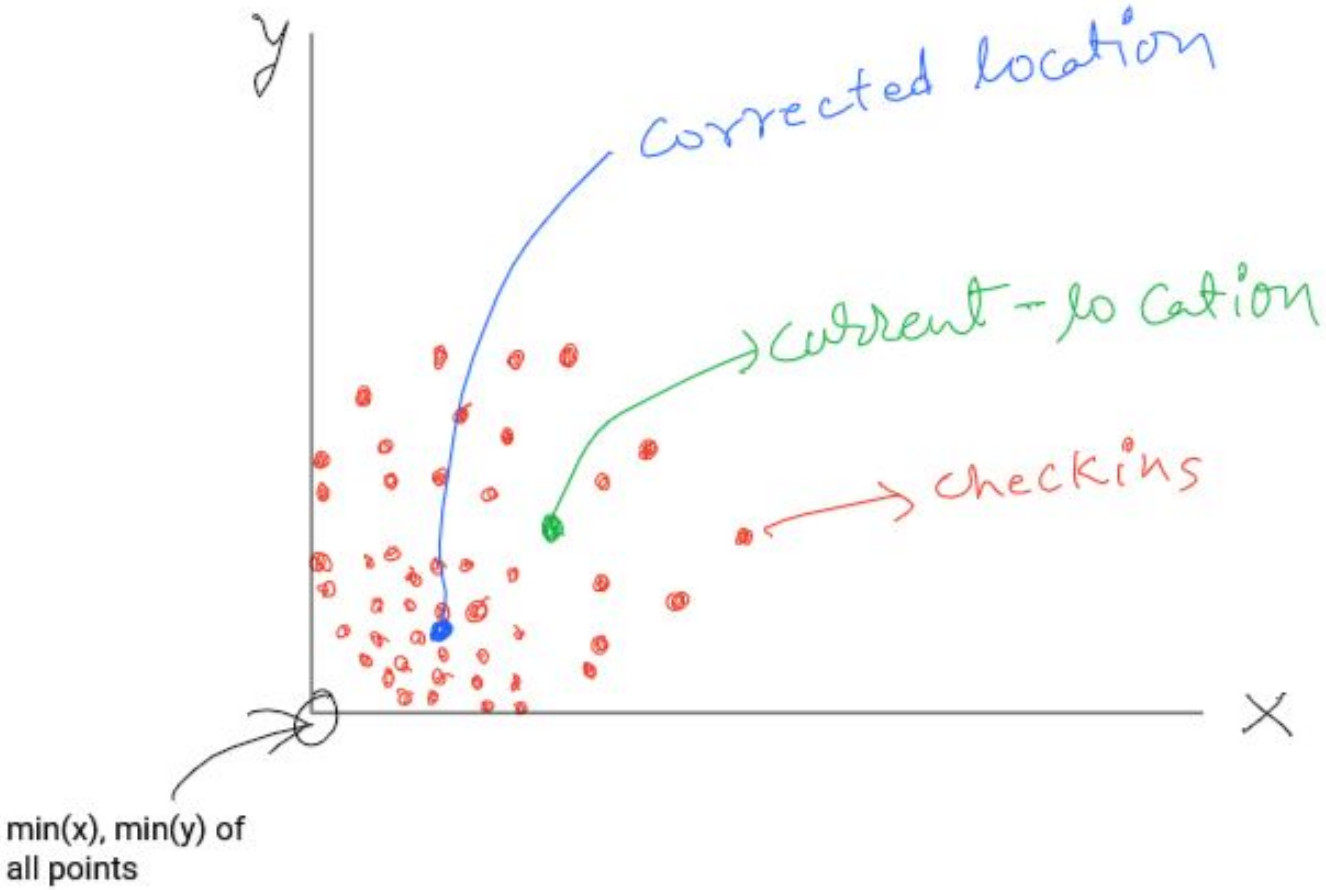


Fig: 3D cross-section of area is divided in quadrants based on current-location(*center-min(x), min(y)*

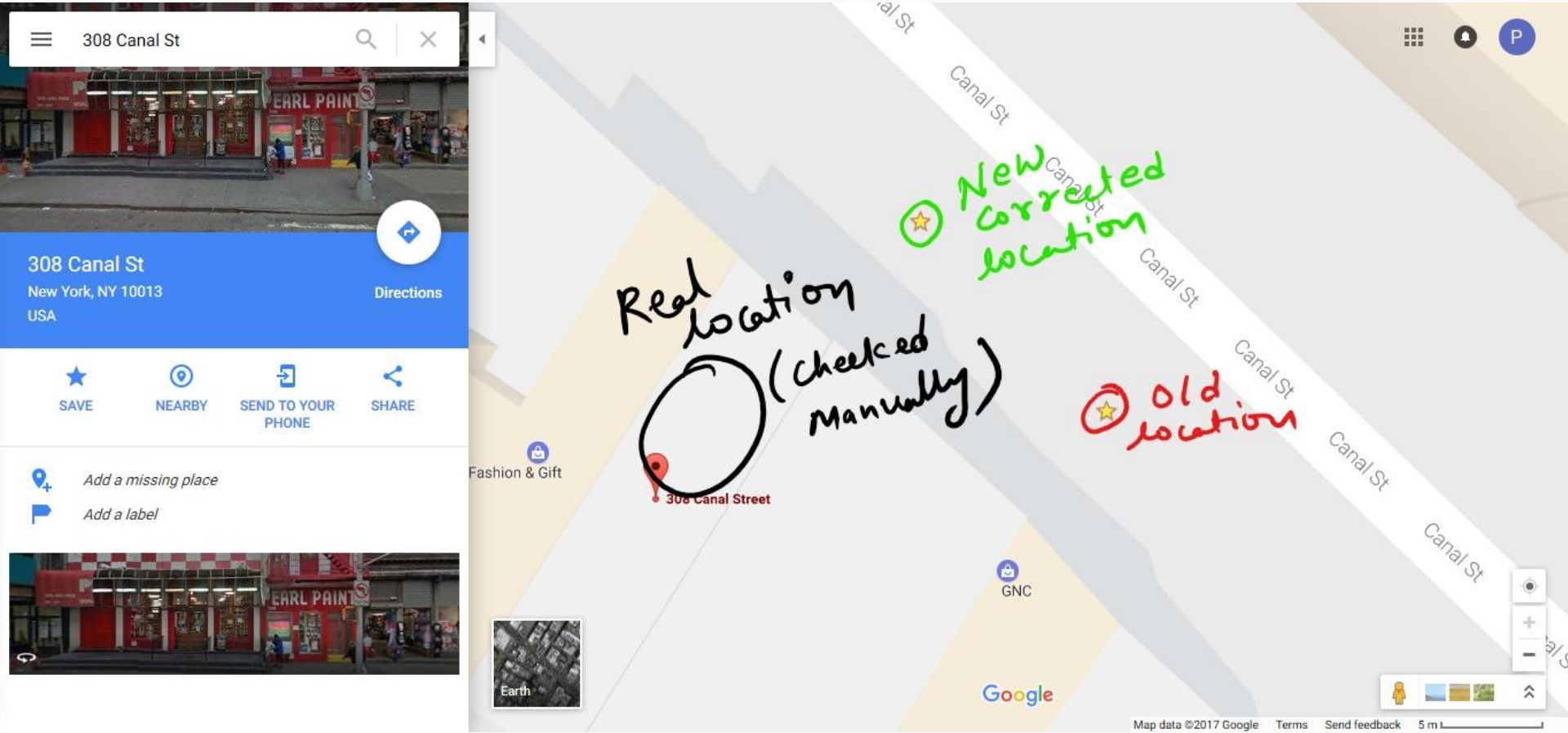


great-circle distance (drawn in red) between two points on a sphere, P and Q

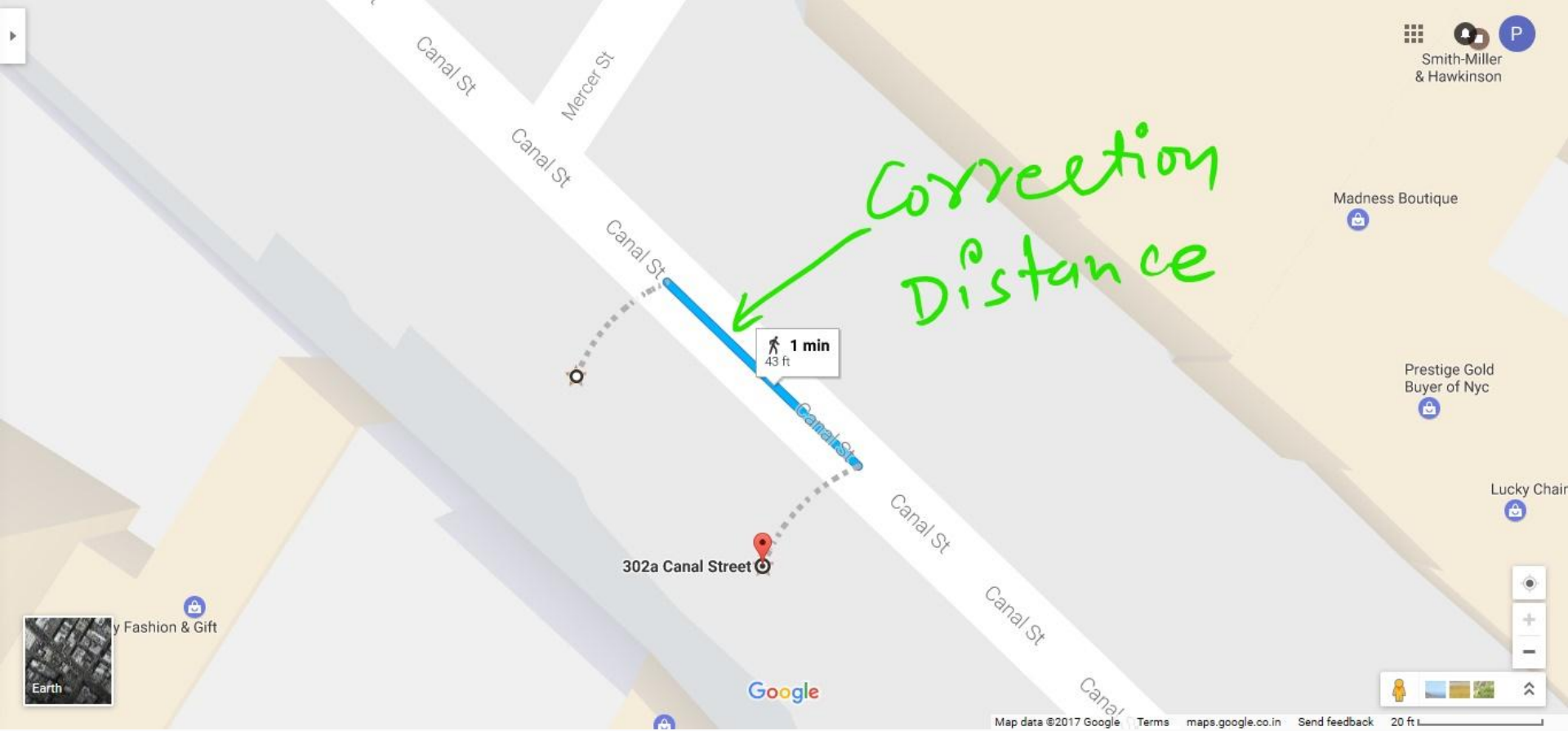



y
corrected location
current-location
checkins
X
min(x), min(y) of
all points

Results



Results from manual testing



On our Manual testing our algorithm corrected a location to 43 feet

FOURSQUARE

I'm looking for...

New York, United States

Find great places on the go.

Discover what's nearby, search for what you're craving, and get deals and tips along the way.

Get the app

Pearl Art & Craft Supply

(Now Closed)

Arts & Crafts Store

Tribeca, New York

Suggest an edit

Save Share

Tips 37 | Photos 85

37 Tips and reviews

Search tips...

Leave a tip for others...

Post

SORT: POPULAR RECENT

The Corcoran Group April 1, 2011

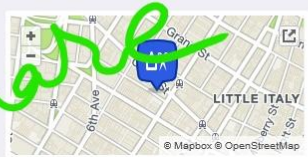
Five floors of art supp heaven – they have absolutely everything! And the helpful staff will assist you in find everything on your list. Students get 10% off their order.

Upvote 4 Downvote

Keilon Lawrence August 11, 2013

Five floors filled with all the essentials. Be sure to browse a little harder for the best deals on the things you need. Numerous brands at varying price points.

Upvote Downvote



Pearl Art & Craft Supply Edit

308 Canal St (btwn Broadway & Mercer)

New York, NY 10013

United States

Get directions

+1 212-431-7932

pearlpaint.com

@pearlpaint

entery on four square

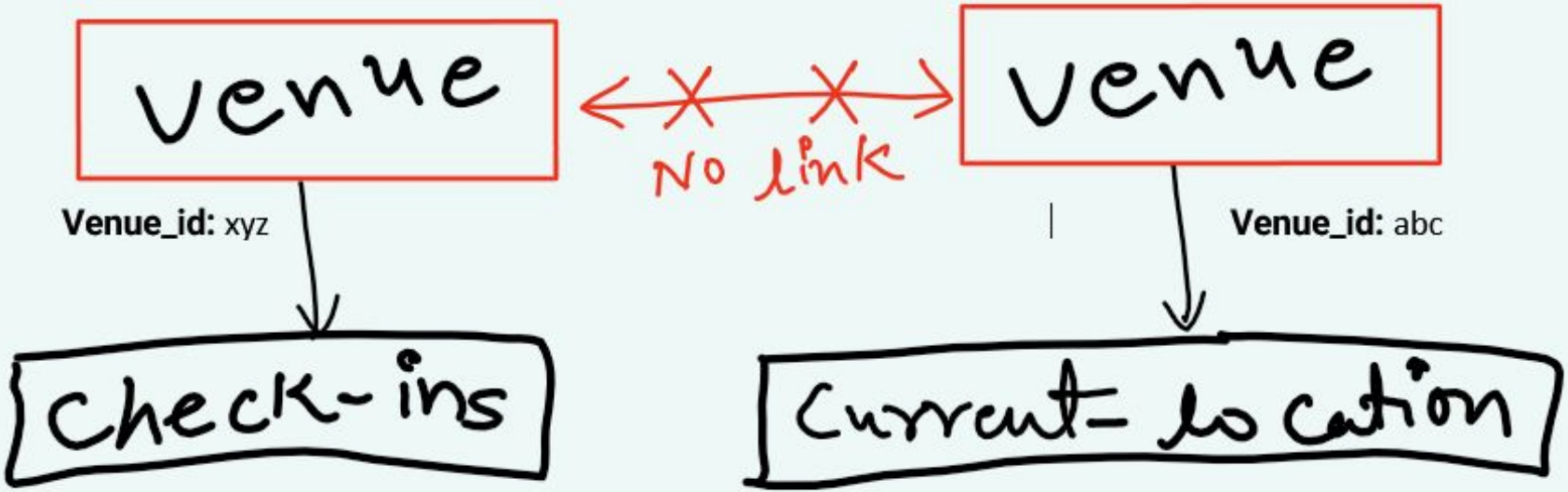
Manual verification of location via Google street view

Observations

- Algorithm works best when we have massive amount of check-in data
- Algorithm is tolerant to outliers and simple to implement in comparison with Algorithm we designed at first stage
- Algorithm will never introduce more error then current error, provided that check-in data is correct

Difficulties

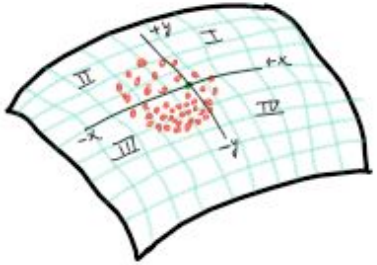
- Had to throw away all the work done in first few weeks due to anonymized foursquare dataset
- Not able find a non-anonymized dataset with all the attributes required
- Had to build a dataset using the attributes from NYC and Tokyo Check-in Dataset and Used the Foursquare API to fetch other Required attributes
- Algorithm we designed at first was prone to outliers and hence not so accurate, that resulted in redesign of algorithm
- We were all new to python and learning is from scratch was difficult.

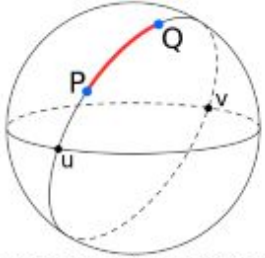


Latitude	Longitude
40.7198103755	-74.0025810333
40.7198103748	-74.0025810298
40.7198103765	-74.0025810328
40.7198103680	-74.0025810326
40.7198103790	-74.0025810327
40.7198103720	-74.0025810321

Latitude	Longitude
40.7197264158111	-74.0024719994671

```
e=conn.cursor()
f=conn.cursor()
y=0.0
z=0.0
countla=0.0
countlo=0.0
c.execute("SELECT index FROM venues")
for row in c:
    xe=c.fetchone()
    print xe
    xex=str(xe)
    xex=xex[1:-2]
    id=int(xex)
    print id
    if xe is not None:
        e.execute("SELECT latitude FROM checkins WHERE venue_id==id")
        for row in e:
            ye = e.fetchone()
            if ye is None:
                break
            yey=str(ye)
            yey=yey[1:-2]
            y= y+ float(yey)
            countla = countla + 1.0
        if countla>0:
                print(y/countla)
        f.execute("SELECT longitude FROM checkins WHERE venue_id==id")
        for row in f:
            ze = f.fetchone()
            if ze is None:
                break
            zey=str(ze)
            zey=zey[1:-2]
            z= z+ float(zey)
            countlo = countlo + 1.0
        if countlo>0:
```





great-circle distance (drawn in red) between two points on a sphere, P and Q

Fig: 3D cross-section of area

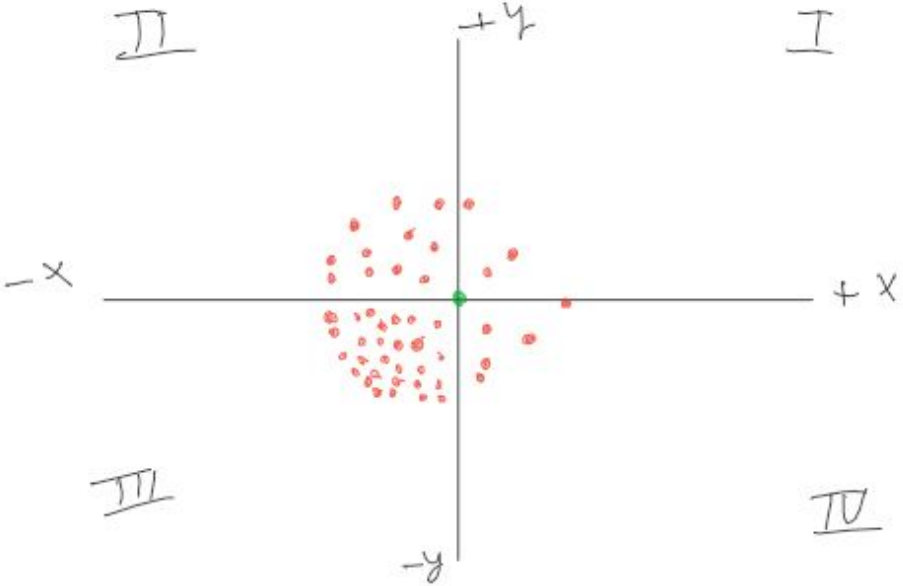


Fig: 2D projection of area by great-circle distance

Future plans

- Implementation and completion of algorithm
- Testing of algorithm on full dataset
- Analysis of results
- Visualization of results

References

"API Endpoints". *Developer.foursquare.com*. N.p., 2017. Web. 20 Mar. 2017.

Dingqi Yang, et al. "Modeling User Activity Preference By Leveraging User Spatial Temporal Characteristics In Lbsns". *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45.1 (2015): 129-142. Web.

"Earth Ellipsoid". *En.wikipedia.org*. N.p., 2017. Web. 20 Mar. 2017.

"Earth Radius". *En.wikipedia.org*. N.p., 2017. Web. 20 Mar. 2017.

"Electron0zero/Geocorrect". *GitHub*. N.p., 2017. Web. 20 Mar. 2017.

References

"Geopy/Geopy". *GitHub*. N.p., 2017. Web. 20 Mar. 2017.

Illustration Of Great-Circle Distance.. 2017. Web. 20 Mar. 2017.

“Calculating Latitude/Longitude X Miles From Point?". *Gis.stackexchange.com*. N.p., 2017. Web. 20 Mar. 2017.

"Understanding Latitude And Longitude". *Learner.org*. N.p., 2017. Web. 20 Mar. 2017.

Questions?

Thanks