جـامعـة الشـــارقـة
UNIVERSITY OF SHARJAH

Line Simplification for Efficient Approximate Join Queries On Big Geospatial Data

Dr. **Isam Mashhour Al Jawarneh,** Fatima Ahmed Alhammadi , Haya Almadhloum Alsuwaidi, Shooq Abdelrahman Alzarooni

Department of Computer Science, University of Sharjah, United Arab Emirates

International Conference on Intelligent Data Science Technologies and Applications (IDSTA 2024)

Dubrovnik, Croatia, 25 September 2024

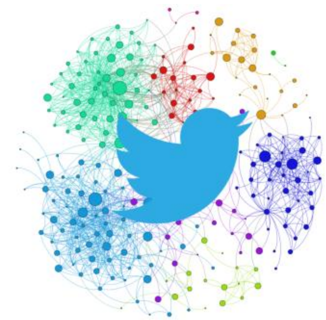
Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

Big data examples

- YouTube : Several petabytes (~**350 PB** of data in 2019)
- **500-700** million **tweets** a day,
  - which adds up to roughly **12 terabytes** of data every 24 hours.
- **Facebook**
  - on the verge of **500** daily **terabytes**,

Source: Forbes



Tweet with exact location

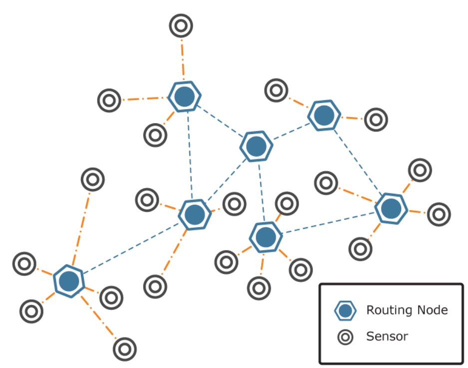

```
{
 "geo" : {
   "type" : "Point" ,
   "coordinates" : [
     40.74118764 ,
     -73.9998279
   ]
 } ,
```


- Most data (>60%) is **geo-referenced**!

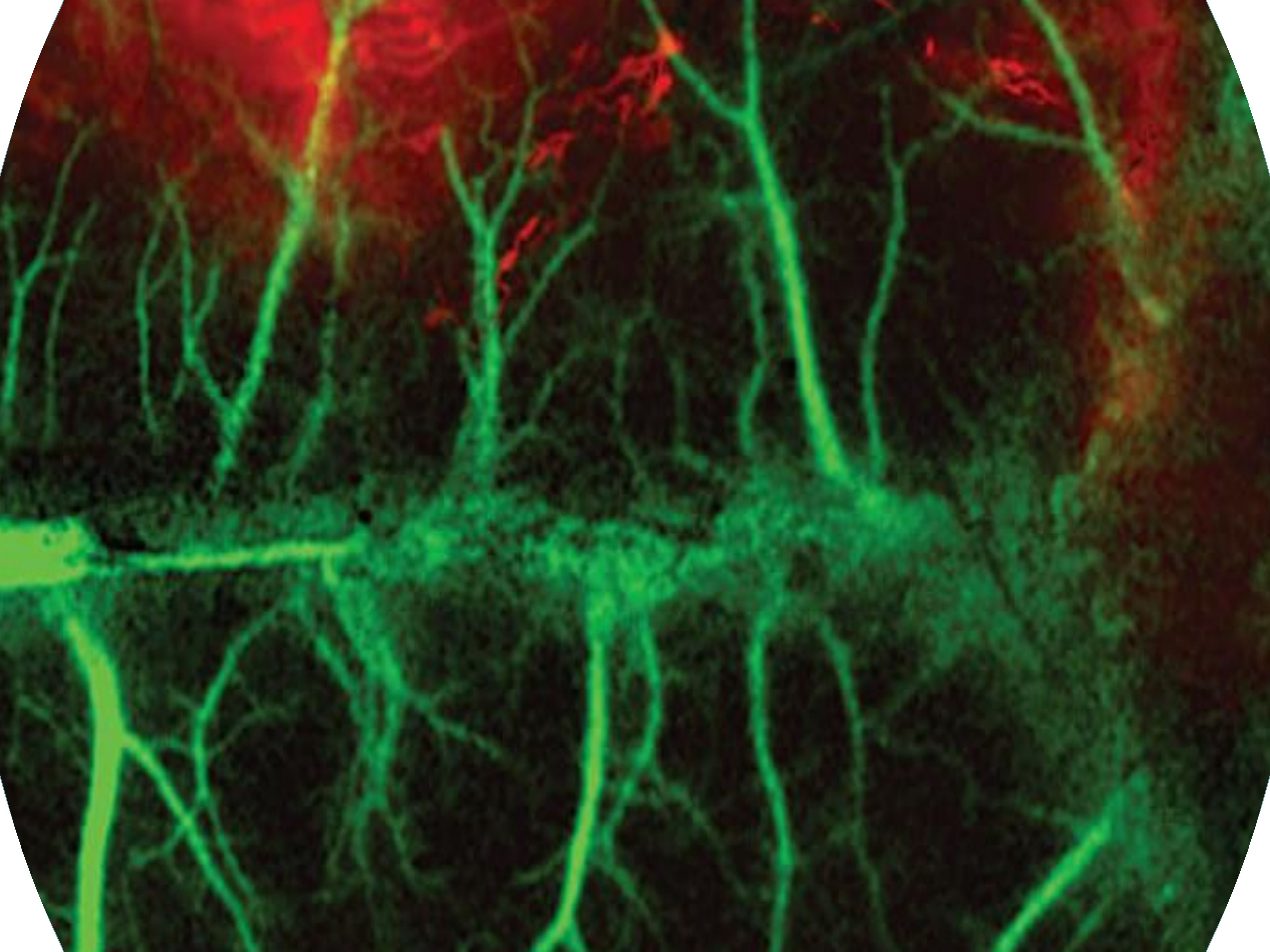
Geospatial Data is everywhere!

Location-based Services

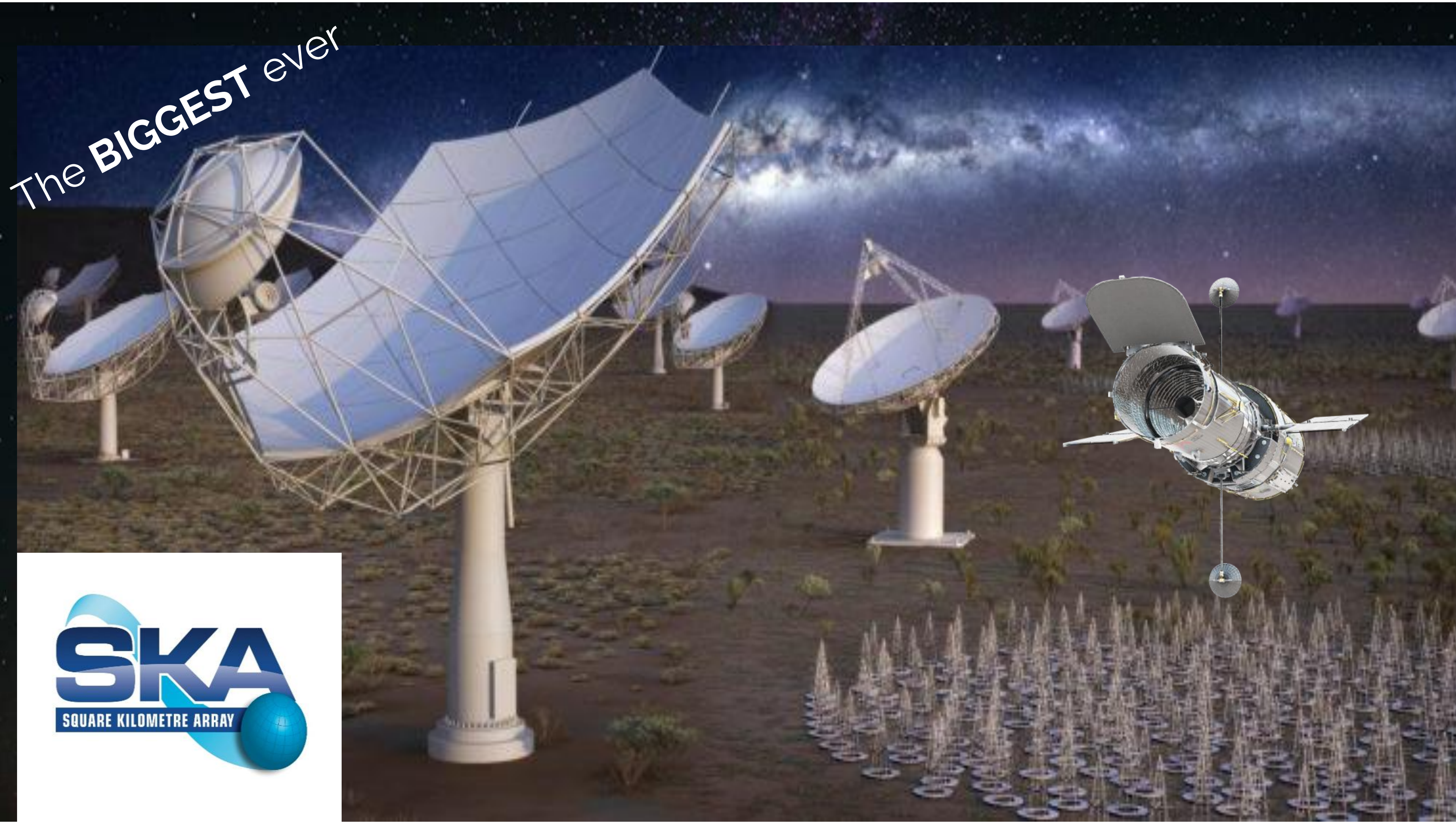
IoT Projects & Sensor Networks



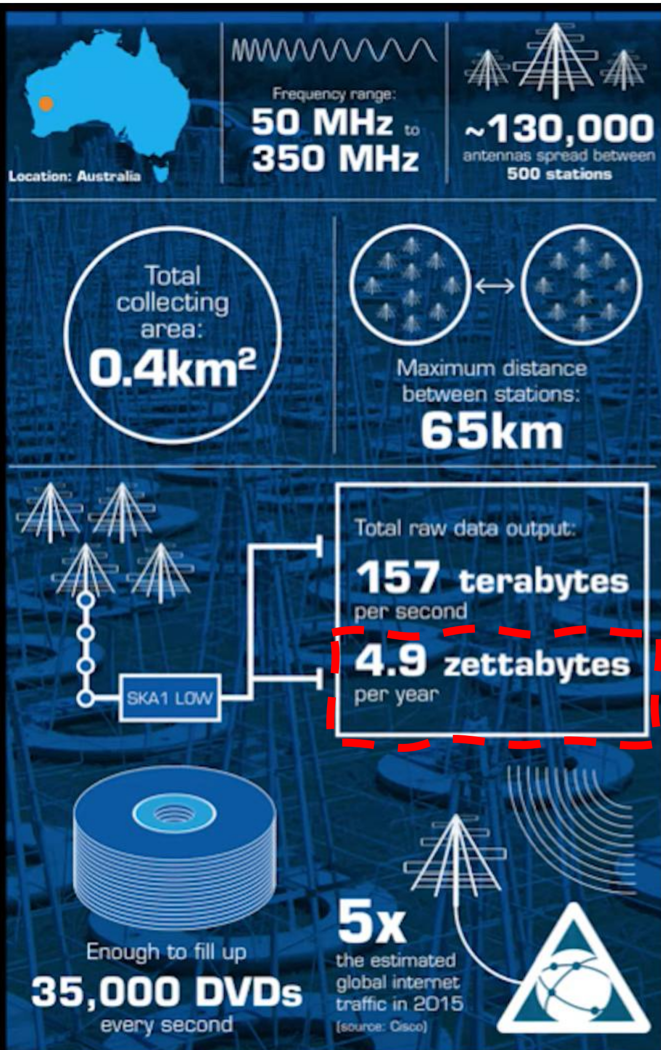
Social Media



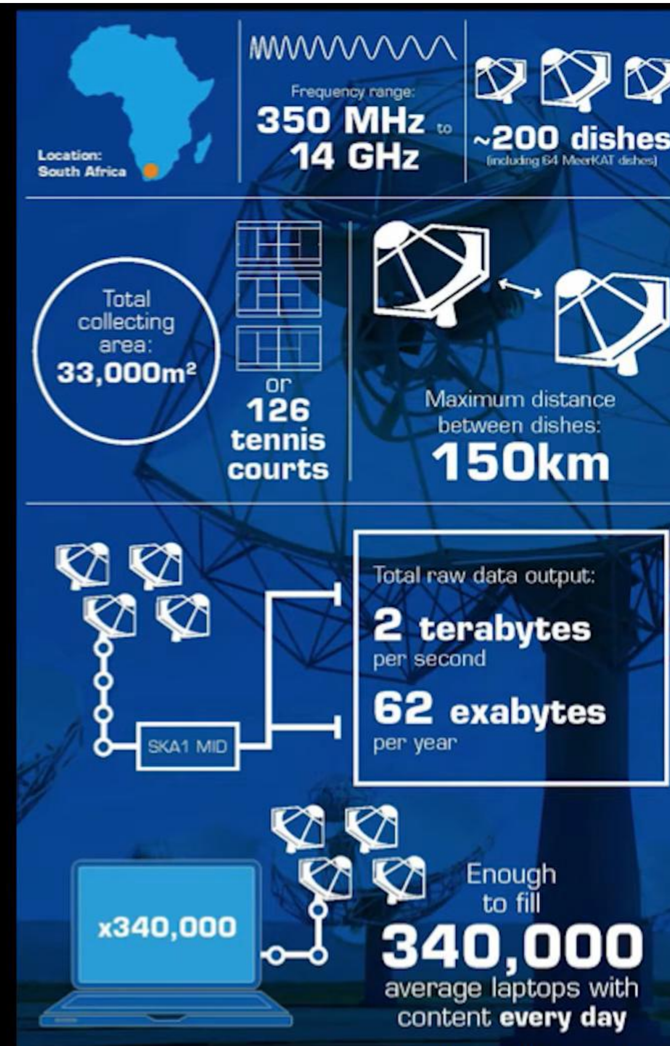
جامعة الشارقة
UNIVERSITY OF SHARJAH



جـامعـة الشـارقـة
UNIVERSITY OF SHARJAH



97 ZB of Global digital data was created up to 2022



Courtesy: SKAO, modifications by MJ-H

Spatial Data-intensive applications

- Spatial Data is the primary **challenge**
  - **Volume (size)**,
  - **Complexity**,
  - **Speed** of arrival & **change** (**uncertainty**)

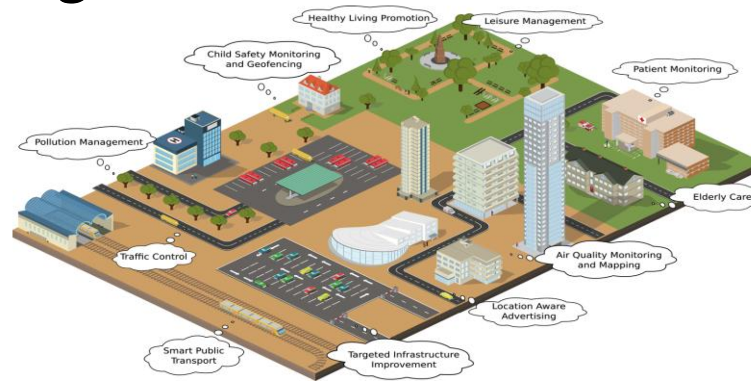
Smart City and Big Data Context

Smart City

Advanced technological services

Geographic Big Data

Huge amount of information

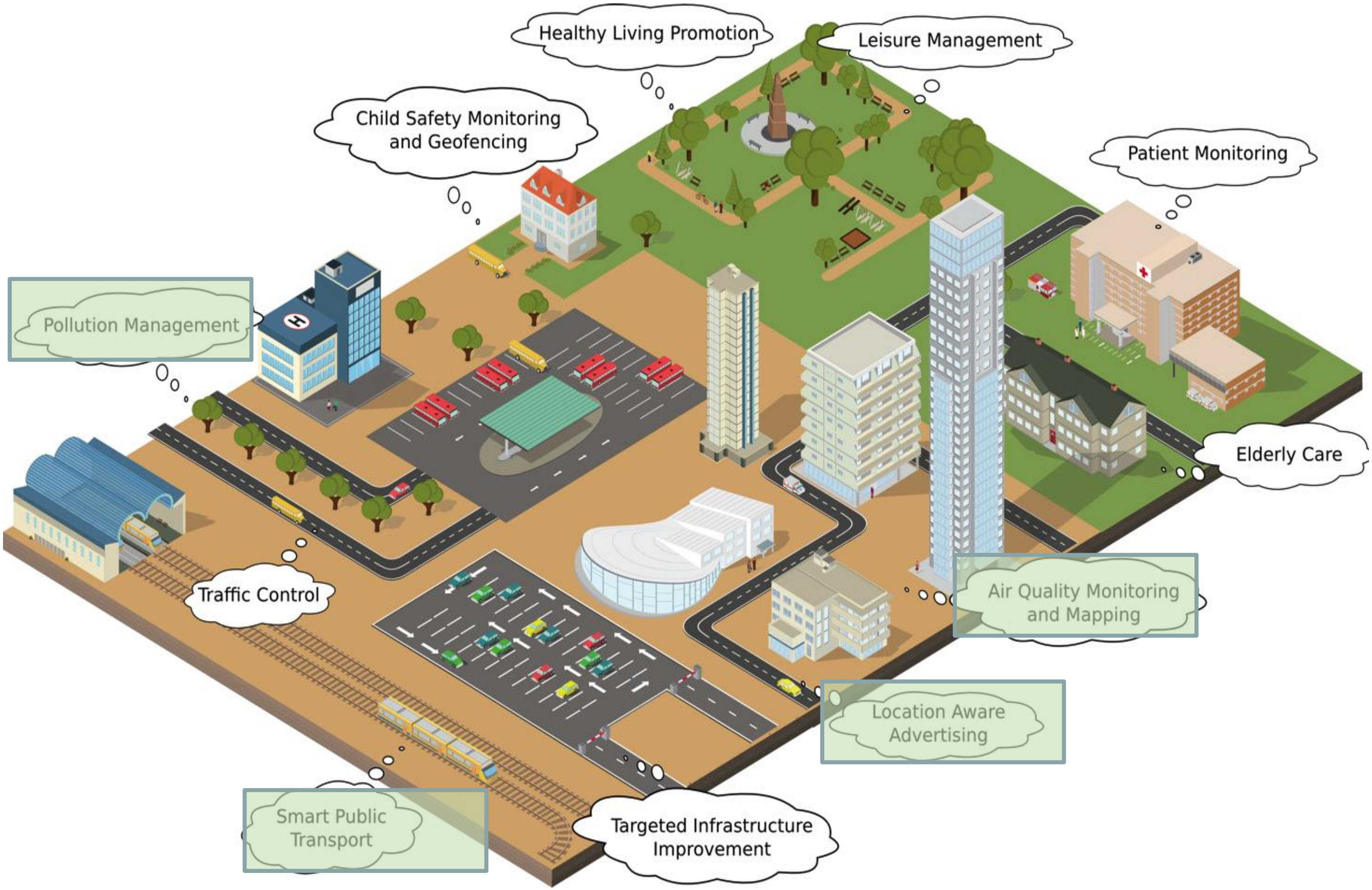


Mobile Sensing

Automatic collection of data

Billions of GPS-enabled handheld devices collect massive data amounts

Location-based services



Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

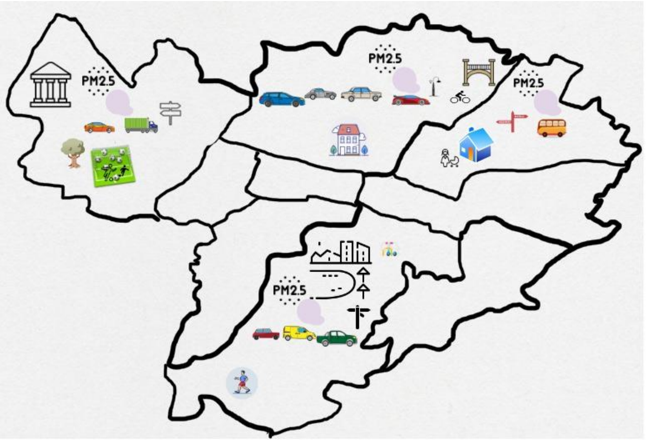
Why Spatial Join?

- Urban Computing
  - Improves **urban environment, human life quality**, and **city operation systems.**
- e.g., **“Planning Bike Lanes based on Sharing-Bike’s Trajectories”**
  - **Spatial join**

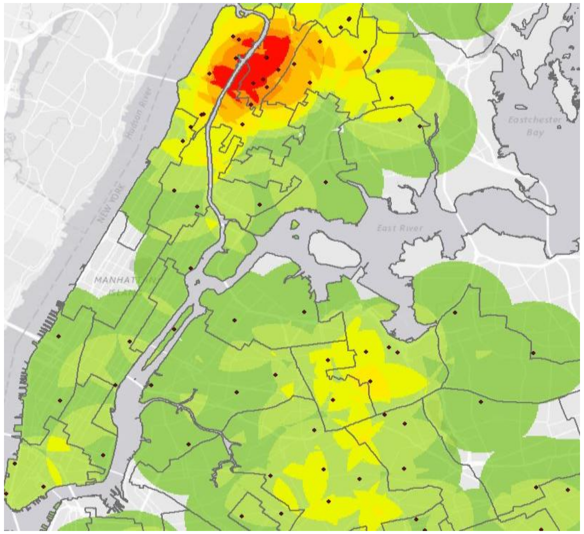


Generating Geo-maps

- Data is subjected to Exploratory Spatial Data Analytics (**ESDA**)
  - Generating geo-maps (e.g., **region-based** maps such as choropleth)
    - Requires **Spatial join** **(costly)**

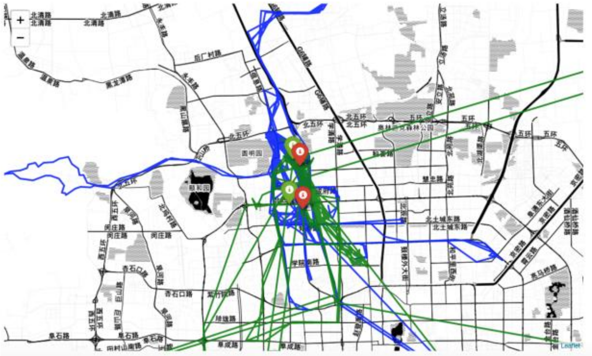


- **Geospatial** aggregation
  - Air pollutants **density** in each **zone**,
  - **Autocorrelation** between nearness and pollution



Visualizing georeferenced data requires aggregation

- **line-based**
  - time-series trajectory visualization of spatial data
  - Requires **aggregations** and **group-by, spatial join**

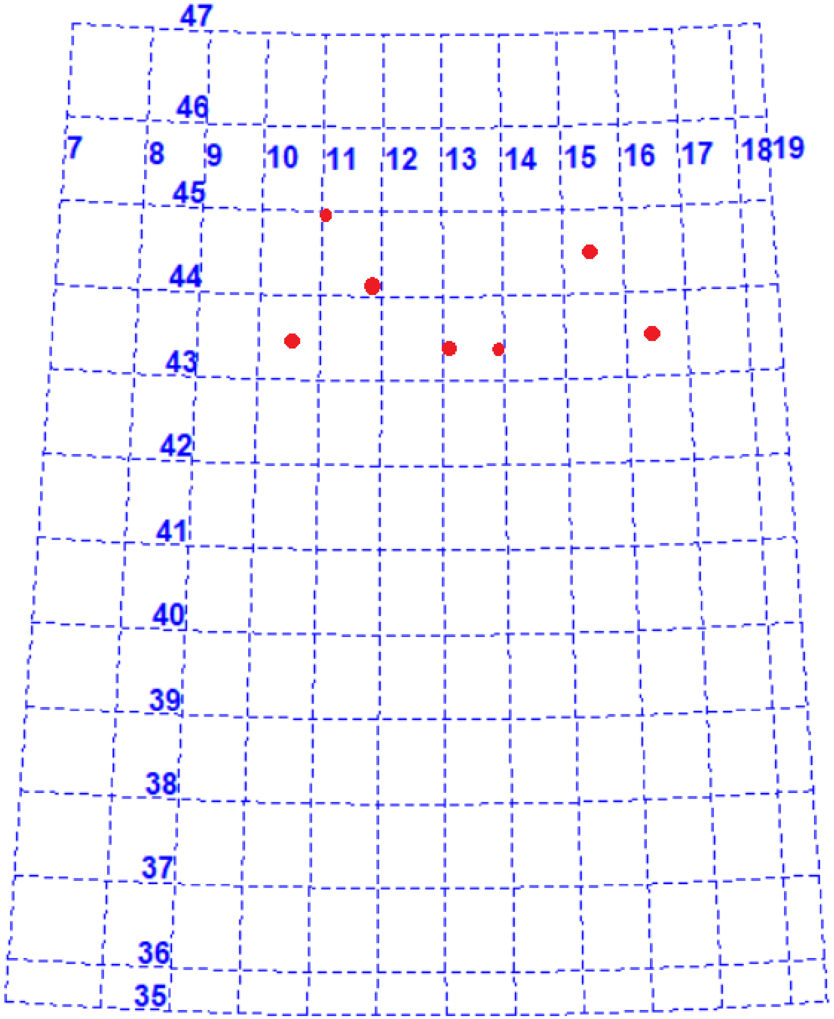


- **region-based**
  - Tessellating geographic regions into grid cells, then, **grouping** data by region-based **aggregations, requires spatial join**
  - e.g., Choropleth maps generation

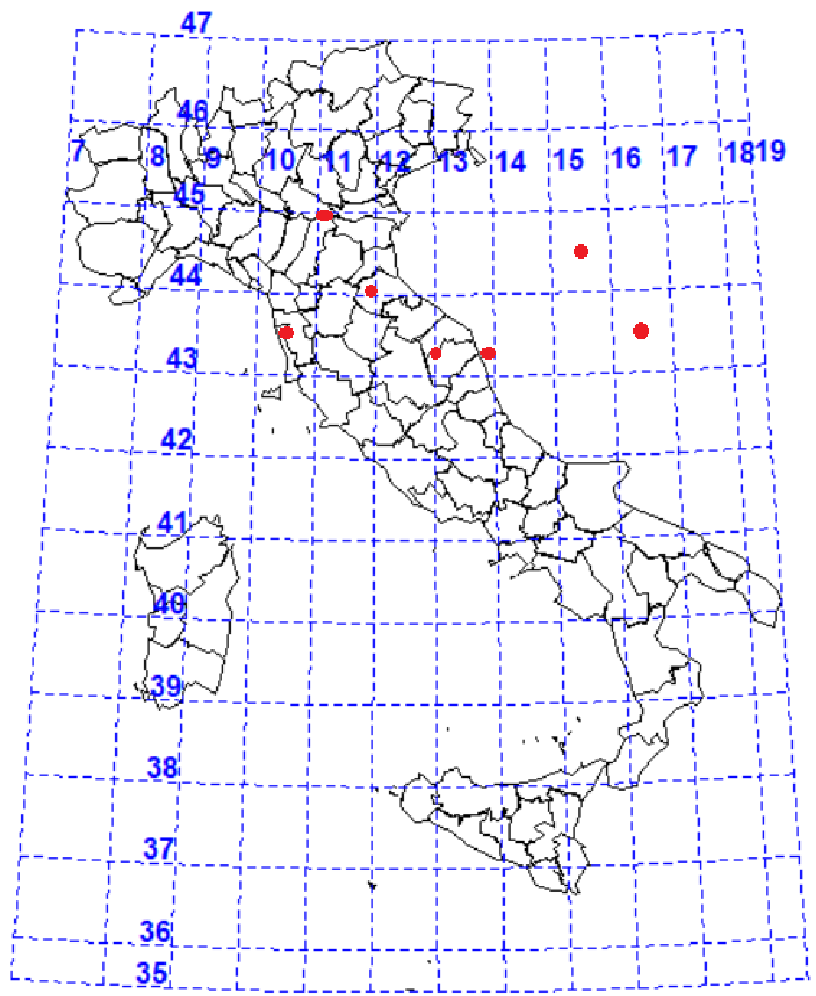
Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

Where is that!

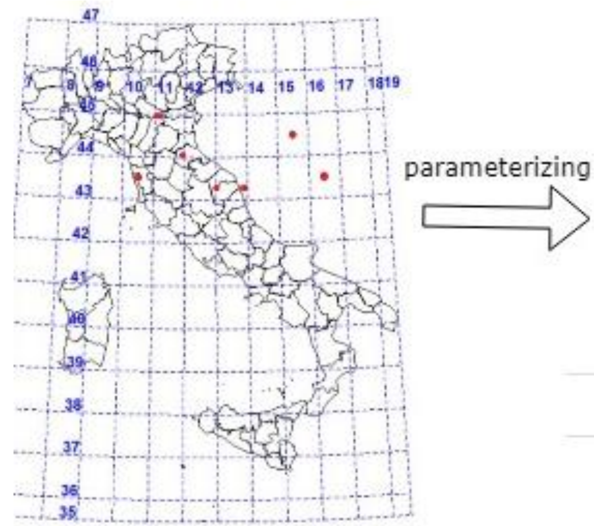


Welcome to Italy (benvenuti!) ☺



Geospatial data representation

- A spatial point is **parametrized** and represented as coordinates (longitude and latitude)
- **Geometry** inherent in the data will be **lost** by such a transformation
- Spatial reconstruction is **expensive**
  - **Spatial Join**



Longitude	Latitude
11.3709	44.5185
11.4081	44.4963
11.3477	44.499

Expensive geometry (point in polygon)

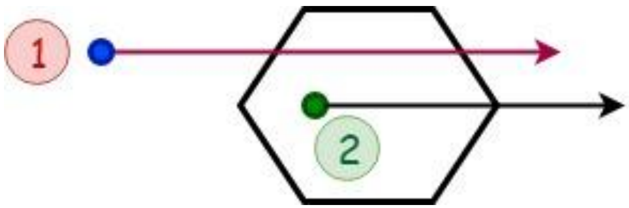
- Point-in-polygon (PIP)
  - **Ray casting algorithm**

(1) Pass a ray out from the test point

(2) Count the number of times that the ray intersects with the boundaries of the polygon

- Even → outside
- Odd → inside

easier said than done!



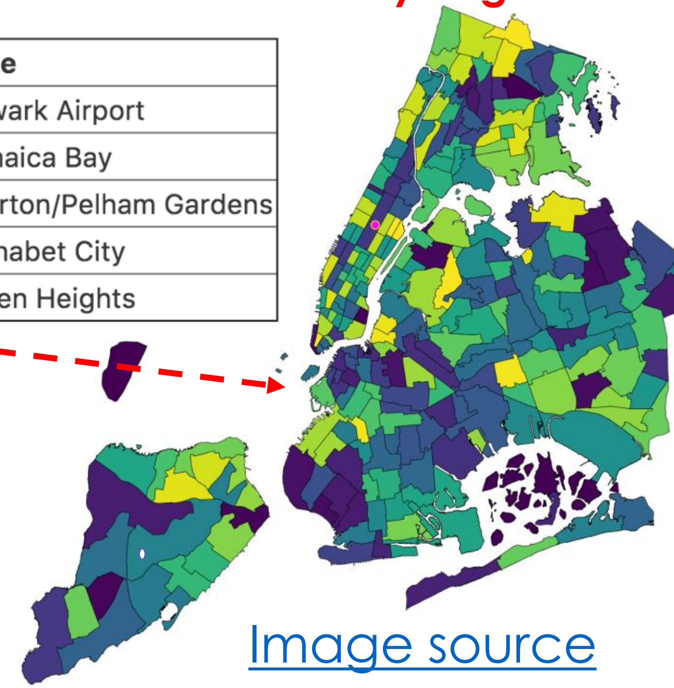
Ray casting for PIP

Spatial data analytics challenges

Shapefile, NYC

Polygons **normally huge in size**

	LocationID	borough	geometry	zone
0	1	EWR	POLYGON ((-74.18445299999996 40.6949959999999,...	Newark Airport
1	2	Queens	(POLYGON ((-73.82337597260663 40.6389870471767...	Jamaica Bay
2	3	Bronx	POLYGON ((-73.84792614099985 40.87134223399991...	Allerton/Pelham Gardens
3	4	Manhattan	POLYGON ((-73.97177410965318 40.72582128133705...	Alphabet City
4	5	Staten Island	POLYGON ((-74.17421738099989 40.56256808599987...	Arden Heights



[Image source](#)

taxi dataset

	tpep_pickup_datetime	tpep_dropoff_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
0	2016-05-01 00:00:00	2016-05-01 00:17:31	-73.985901	40.768040	-73.983986	40.730099
1	2016-05-01 00:00:00	2016-05-01 00:07:31	-73.991577	40.744751	-73.975700	40.765469
2	2016-05-01 00:00:00	2016-05-01 00:07:01	-73.993073	40.741573	-73.980995	40.744633
3	2016-05-01 00:00:00	2016-05-01 00:19:47	-73.991943	40.684601	-74.002258	40.733002
4	2016-05-01 00:00:00	2016-05-01 00:06:39	-74.005280	40.740192	-73.997498	40.737564

Points (parametrized)
Projected Coordinate System (PCS)

assigning trips pickups to city zones (districts) is an example of a **spatial join (expensive** computationally costly workload**)**

	geometry	index_right	LocationID	borough	zone
0	POINT (-73.96599999999999 40.78)	42	43	Manhattan	Central Park

Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

Coping up with geo-data loads

- **Scalability**
    - Hardware scalability. **Overprovisioning** resources
    - Scaling **up/out**
- **Approximate Query Processing** (**AQP**).
    - Data **reduction**
    - **Spatial** Approximate Query Processing (**SAQP**)

Our focus!

QoS Tension

Spatial (Approximate) Query Processing (S(A)QP)

Tension

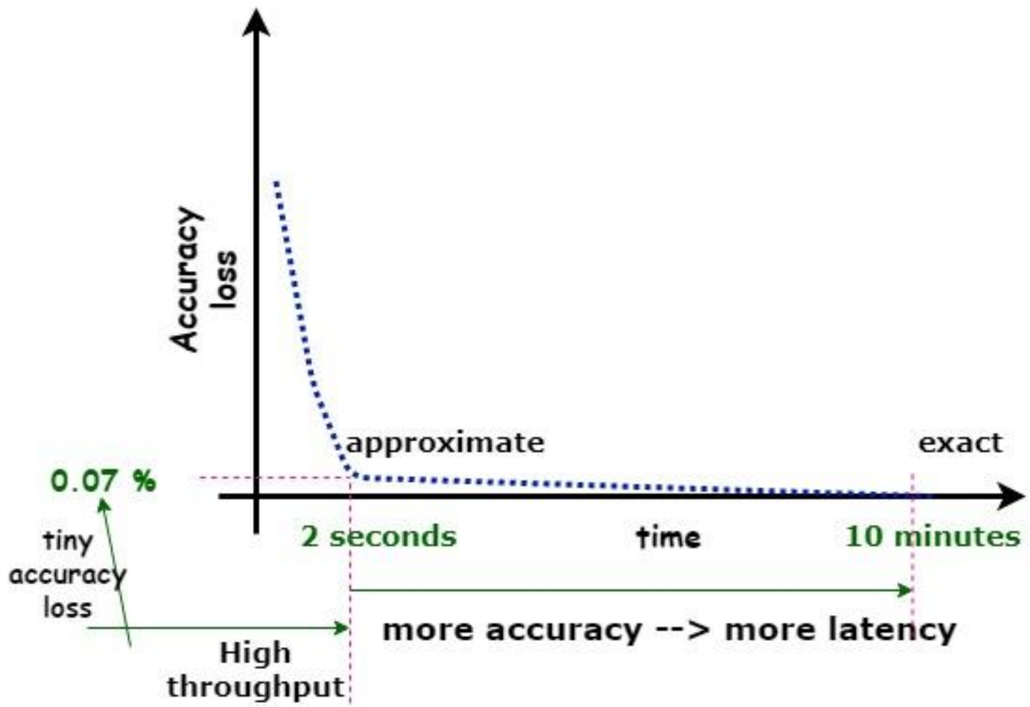
latency

Throughput

Accuracy

Spatial Approximate Query Processing

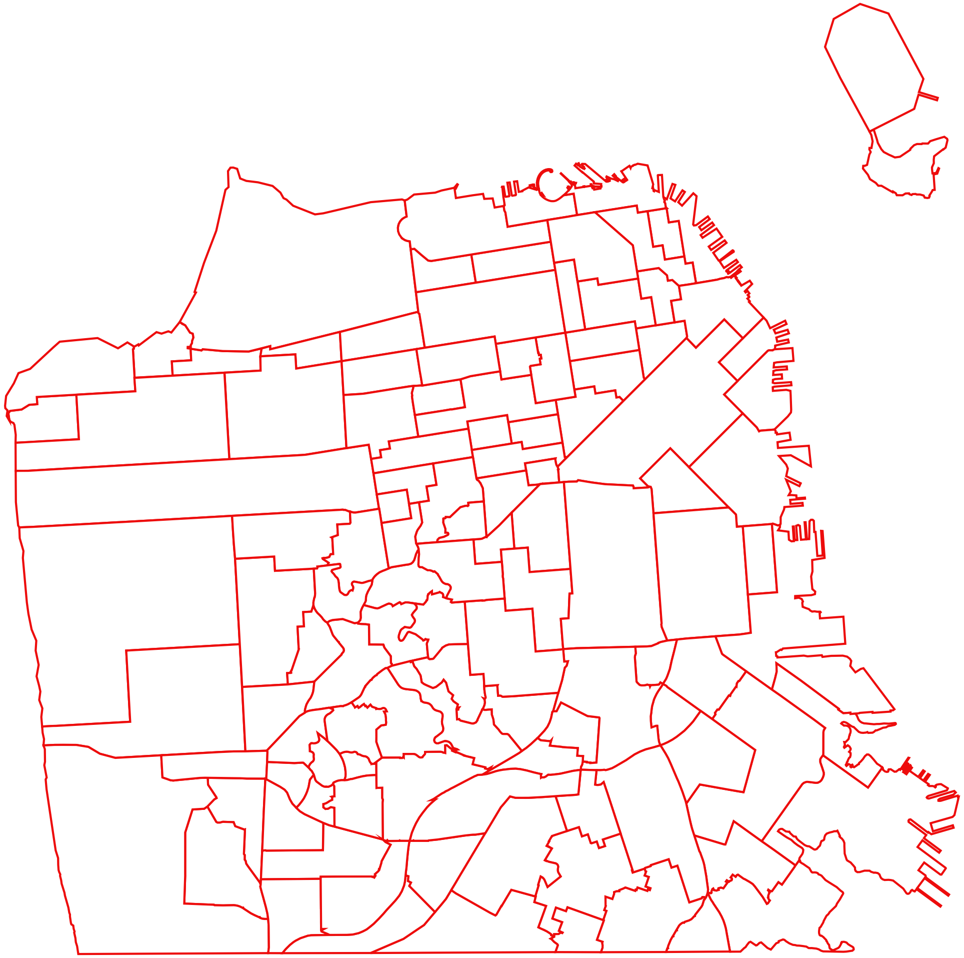
- Decision makers accept tiny **loss** in **accuracy** in exchange for a **throughput gain**



Problem

a beautiful shape of SF, USA

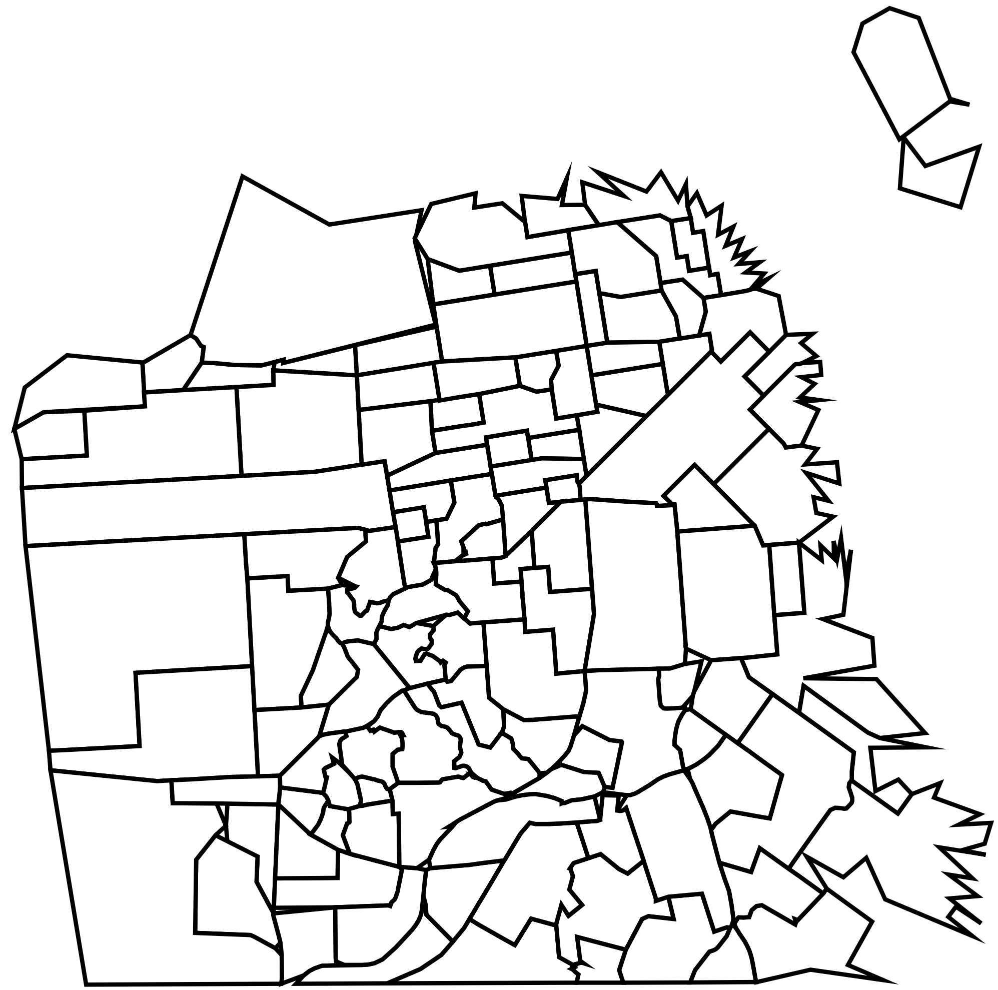
- polygon has **too many** points
- loads **slowly**
- consumes **a lot** of **memory**
  - & we don't even see the full detail



Solution

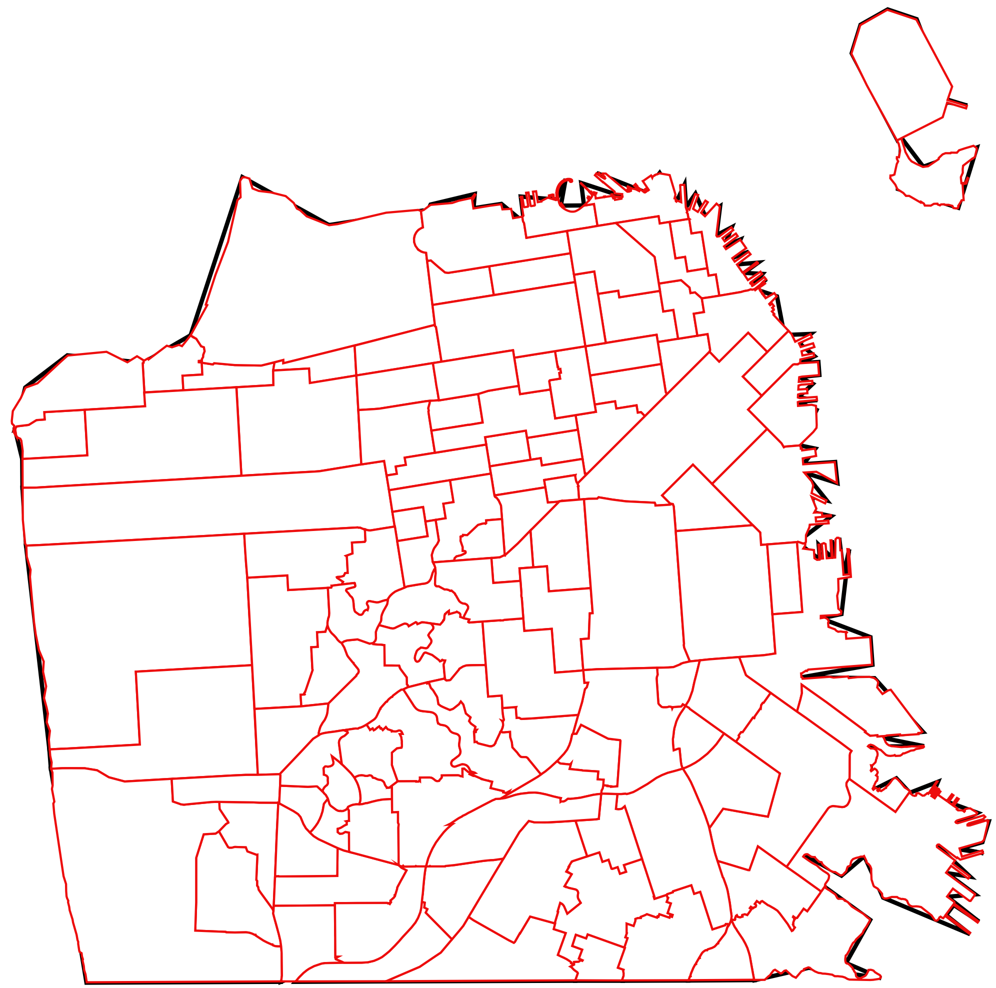
Simplify!

- express same geometry with **fewer** points
  - preserve original shape as much as possible
- Douglas-Peucker (**DP**) & Visvalingam-Whyatt (**VW**)



Difference?

- **Loads faster**
- **Memory efficient**



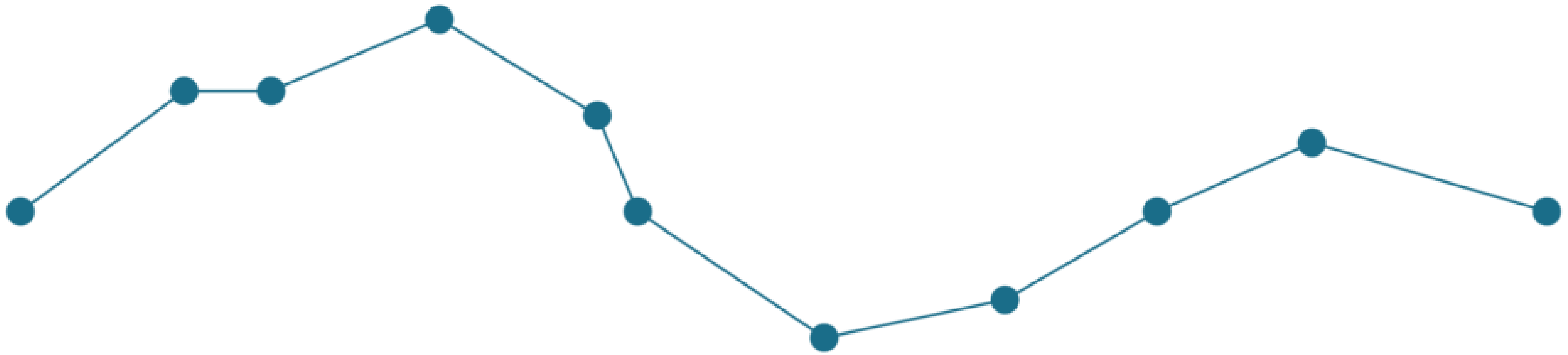
Spatial Approximate Computing

Computing over a sample instead of the whole population

Service Level Objectives:
Latency/throughput targets

Geospatial data → **sample** → SAQP → **Approximate Result** w/ rigorous Error bounds (**accuracy loss**)

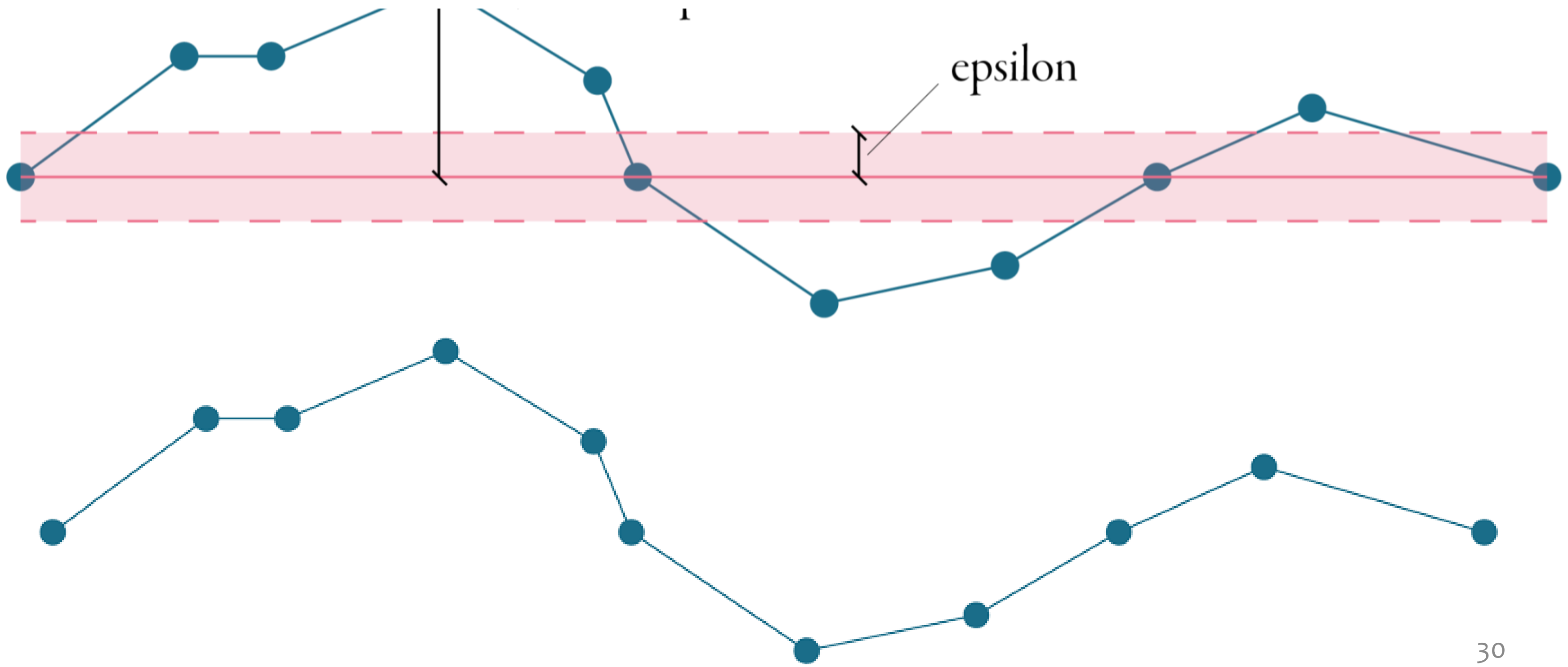
Example: Line simplification



- A **complex** line with **11** points
  - needs to be **simplified**

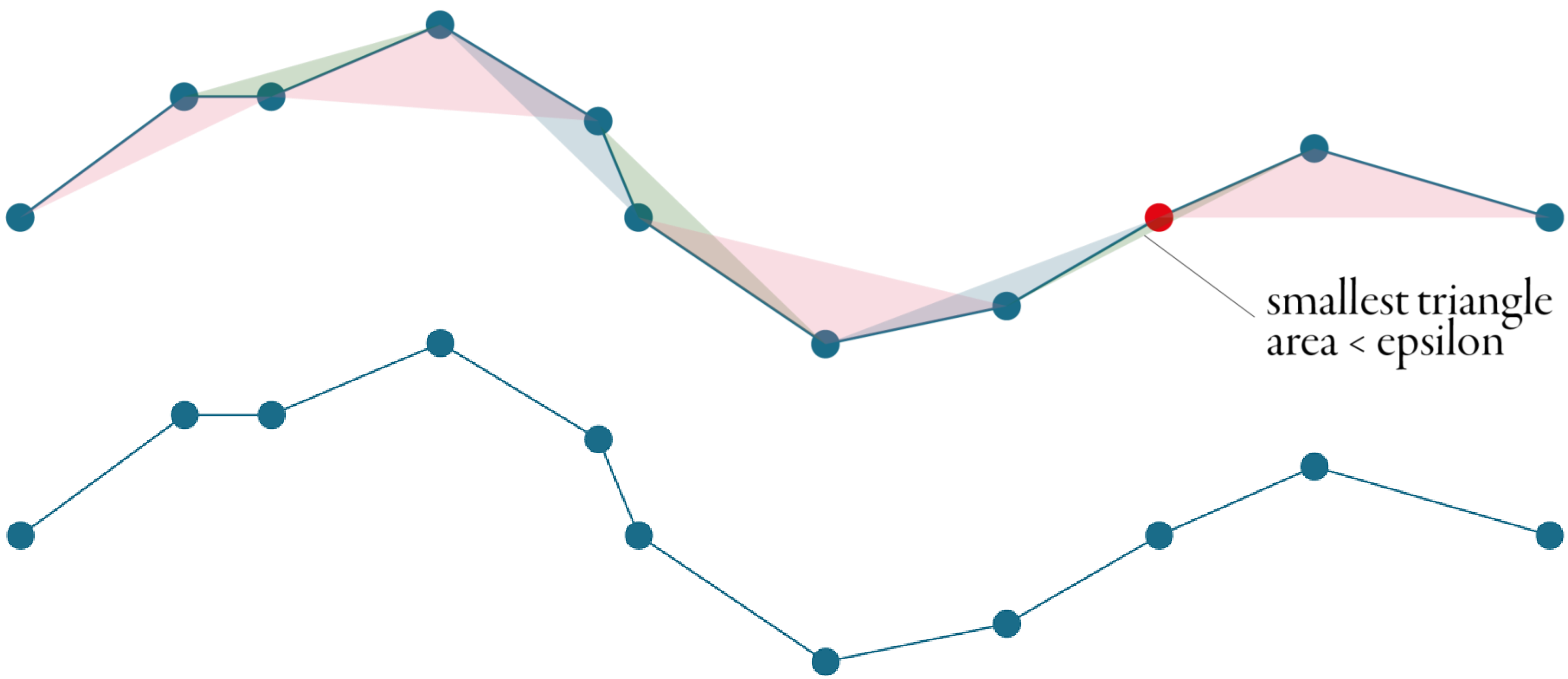
Douglas-Peucker

- remove points that are less important for overall shape
  - **No** new points
- One parameter, tolerance (*epsilon*)



Visvalingam-Whyatt

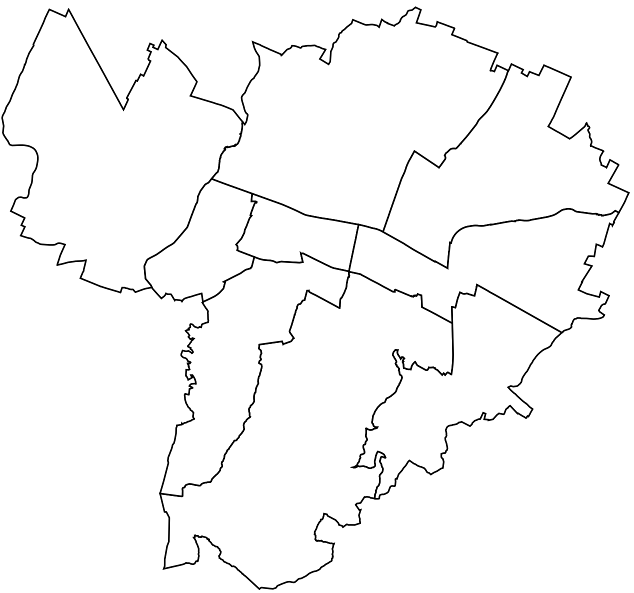
- principle is different.
  - Tolerance (epsilon) is an area, not a distance



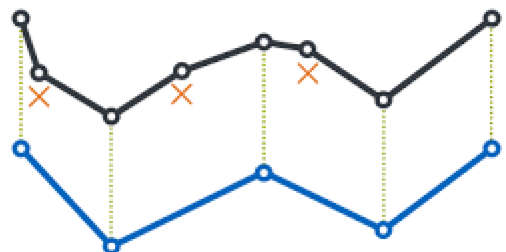
Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

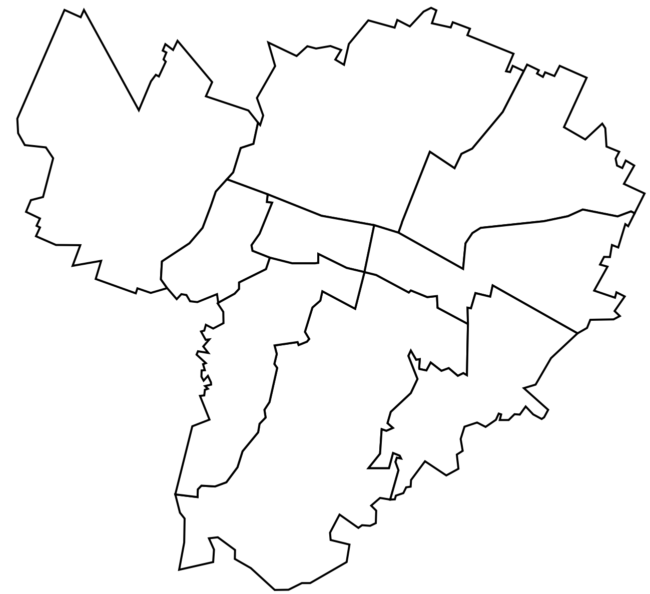
Boundary simplifier



Original polygons



Douglas-Peucker (DP)



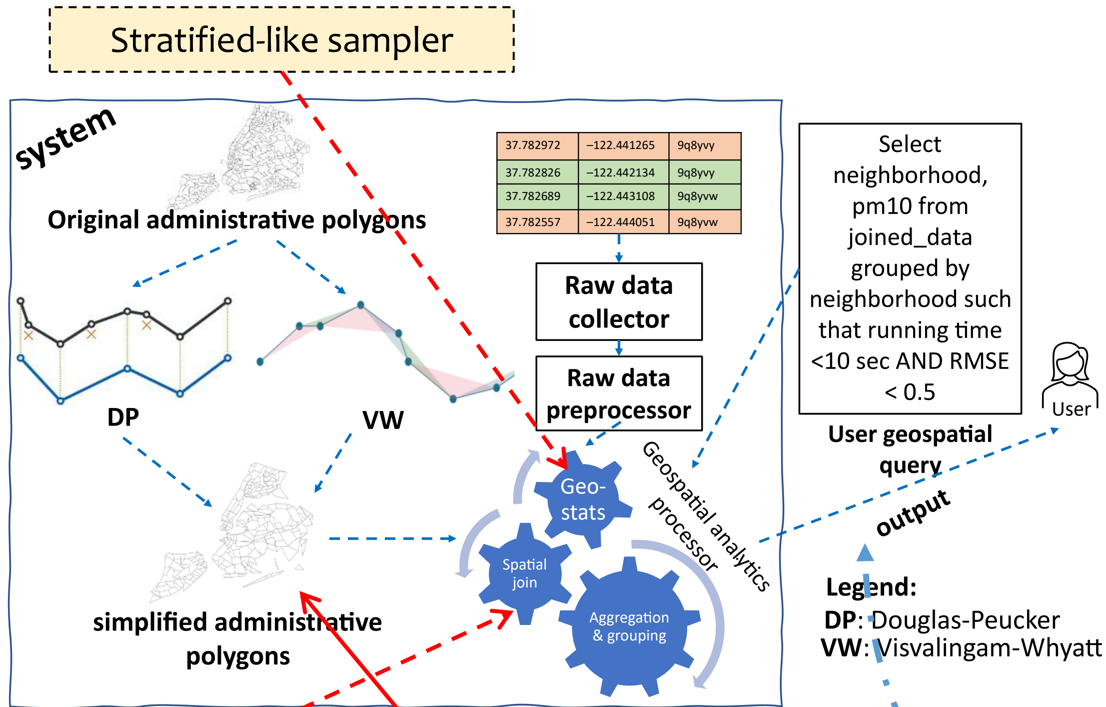
‘percentage of BPK’ 5%

Boundary **simplifier** function applied to polygons representing **Bologna** city, **Italy**

Architecture Overview: Geospatial join at Scale with QoS Guarantees

boundary **simplifier**

- DP & VW algorithms



Plain Filter-and-refine

Simplified polygons: a compact representation

Mean Absolute Percentage Error (**MAPE**), a measure of prediction accuracy, for geo-statistic group-by queries (specifically ‘mean’ queries).

$$MAPE = {}^{100\%}/_{n} \sum_{i=1}^{n} \left| {}^{AC_i - P_i}/_{AC_i} \right|$$

Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

Experimental setup

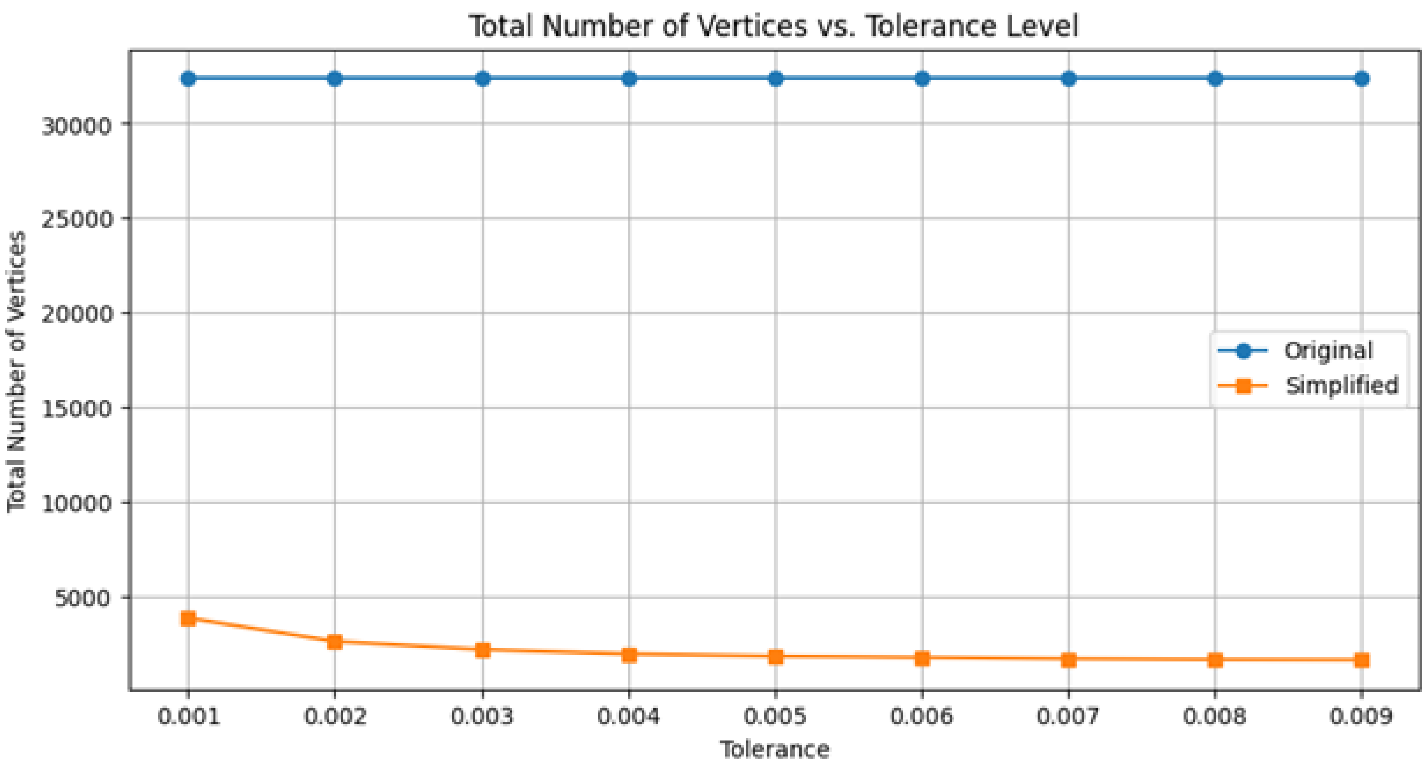
- **Evaluation metrics**
  - Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), Spearman Correlation, and Jensen-Shannon divergence (hereafter JSD for short)
- **Baselines**
  - Plain polygon without simplification
- **Testbed**
  - We have run **experiments** on google colab
  - **Datasets**
    - polygons representing New York City neighborhoods in the USA (GeoJSON)
    - geotagged air quality dataset (NYC) collected using low-cost air-quality sensors, consisting of 170K records

See the paper for explanation!

Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

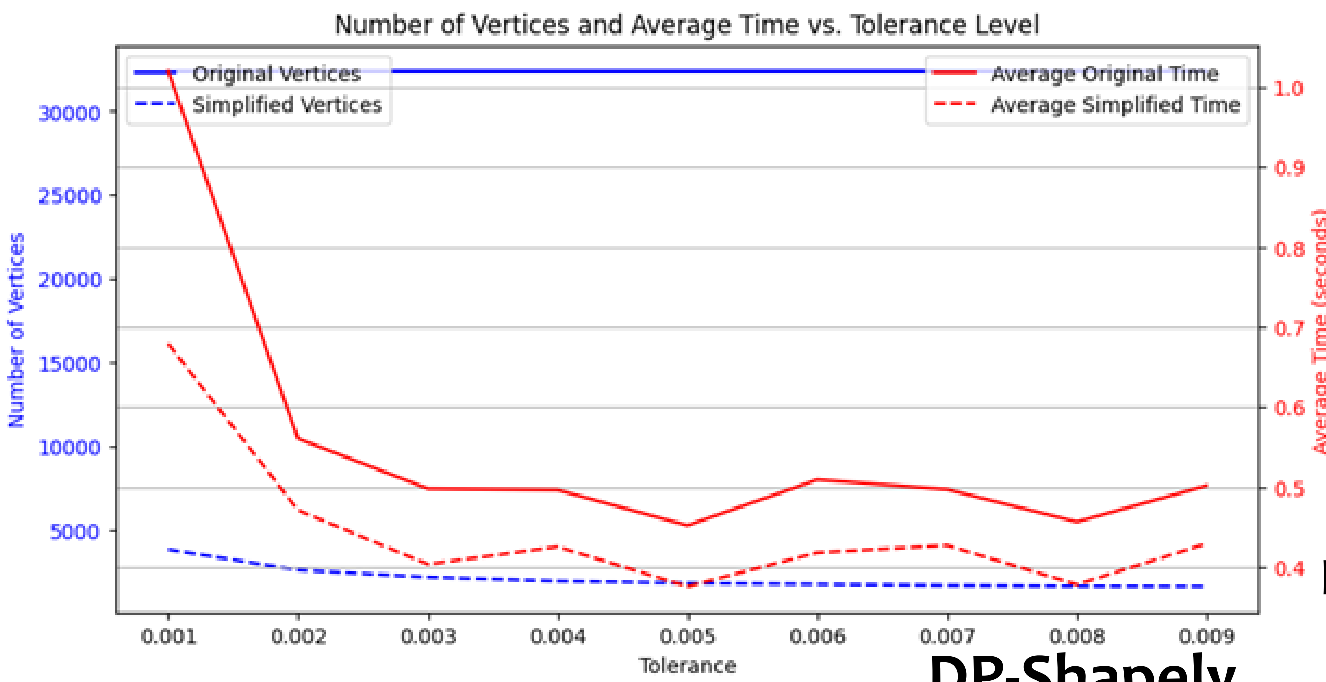
Number of Vertices vs. Tolerance, NYC polygons



Lower-better

- **Varying** the tolerance and
- **computing** total number vertices that are supplied to the system
  NYC polygons
- simplified number of vertices **decreases** with increasing tolerance, indicating a loss in detail as the number of vertices decrease

Number of Vertices vs. Tolerance vs. Average Time

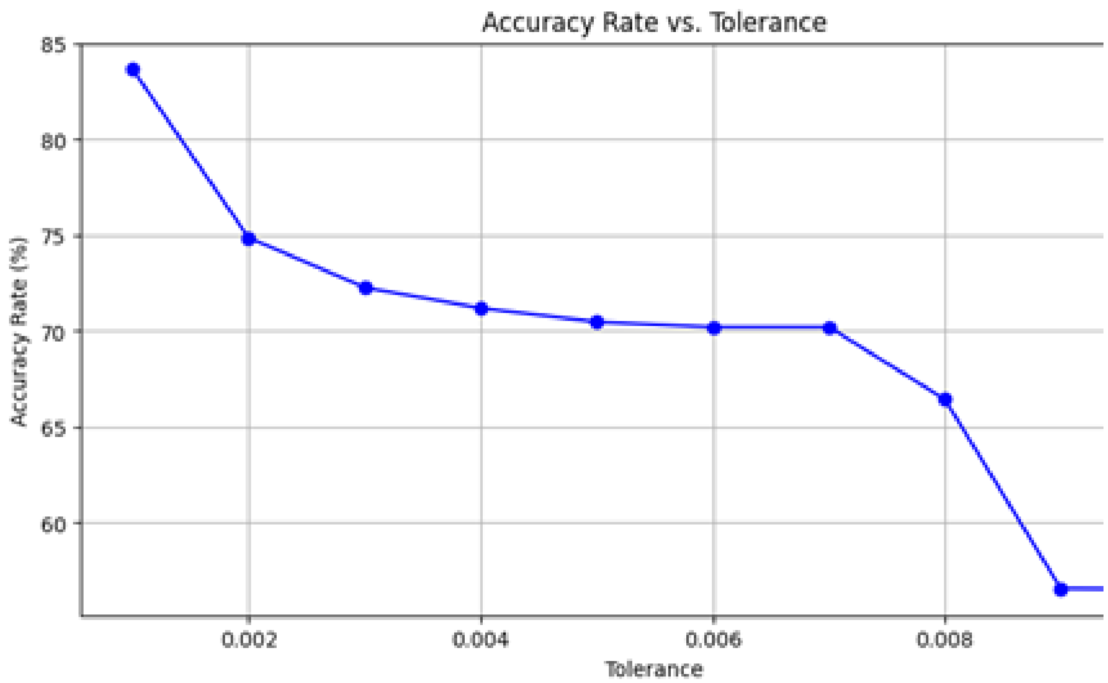


Lower-better

DP-Shapely

- **Varying** the tolerance and
- **computing** computational time required for performing the join operation (simplified Vs. baseline)
- tolerance is indirectly **proportional** to the average time of the spatial join
- as the tolerance increases, the average time and number of vertices decrease

Spatial Join Accuracy Rate vs. Tolerance



Higher
the better

- **Varying** the tolerance and
- **computing** accuracy rate
- accuracy **decreases** with increasing tolerance,
- A tradeoff between reducing complexity of data via simplification and maintaining accuracy
- significant drop in accuracy at around 0.007 tolerance
- optimal tolerance which seems to be the lowest at approximately 0.001

Spatial Join Accuracy

Metric	Algorithm	
	DP	VW
Tolerance	1%	1%
Area (m^2)	2792454.7	2260661.6
No. of Vertices	1,743	2,031
RMSE	62.69%	65.00%
MAPE	0.04758	0.04853
Spearman Correlation	0.88370	0.92074
JSD	0.33109	0.35564

- For **aggregation** workloads
- results are similar between the two algorithms (**DP** & **VW**)
    - both effective
- In terms of Spearman Correlation, **VW** performing slightly better,
    - statistically indicating that the original and simplified data are highly correlated and comparable
    - geospatial data is preserved despite reducing the number of vertices by approximately 94%

جـامعـة الشـارقـة
UNIVERSITY OF SHARJAH

Spatial Join Accuracy

Metric	Algorithm	
	DP	VW
Tolerance	1%	1%
Area (m^2)	2792454.7	2260661.6
No. of Vertices	1,743	2,031
RMSE	62.69%	65.00%
MAPE	0.04758	0.04853
Spearman Correlation	0.88370	0.92074
JSD	0.33109	0.35564

- low Jenson-Shannon Divergence (JSD)
  - similarity between original and simplified data
  - data is sufficiently well-preserved whilst decreasing computational cost.

Outline

- Introduction
  - Background & Motivations
  - Spatial data challenges & requirements
    - Spatial join
- A method for simplifying spatial join
  - Overview
  - Approximate spatial join
- Results and Discussion
  - Deployment: baselines & testing setup
  - Approximate spatial join Vs. baseline
- Summary & future research

Concluding remarks

- **Spatial join** is indispensable
  - computationally **expensive** in full form
  - Line **simplification** is essential
  - Significantly **reducing** data size, while preserving geometric characteristics
    - cutting down computational costs, efficiency improves
- Comparing the performance of **Douglas-Peucker** & **Visvalingam-Whyatt**
  - both effective,
  - However, somehow, DP performs slightly better, but VW produces nicer-looking geometry
- **Future research,** To parallelize spatial join with simplified polygons
  - Currently, requiring original polygons files to broadcast to all cluster computing nodes

Q&A and Contacts

Thanks for your attention!

Question's time...

Line Simplification for Efficient Approximate Join Queries On Big Geospatial Data

Dr. Isam Mashhour Al Jawarneh*,
Fatima Ahmed Alhammadi ,
Haya Almadhloum Alsuwaidi,
Shooq Abdelrahman Alzarooni

* *Assistant Professor*, Department of Computer Science, University of Sharjah, UAE (ijawarneh@sharjah.ac.ae)

IDSTA 2024
Dubrovnik, Croatia 25 September 2024