Mobility Data Management & Exploration

Ch. 03. Modeling and Acquiring Mobility Data

Nikos Pelekis & Yannis Theodoridis

InfoLab | University of Piraeus | Greece
infolab.cs.unipi.gr

v.2014.05

“It is a very sad thing that nowadays there is so little useless information." Oscar Wilde

Chapter outline

3.1. Modeling Mobility Data

3.2. Acquiring Trajectories from Raw Data

3.3. Trajectory Reconstruction and Simplification

3.4. Trajectory Data Generators

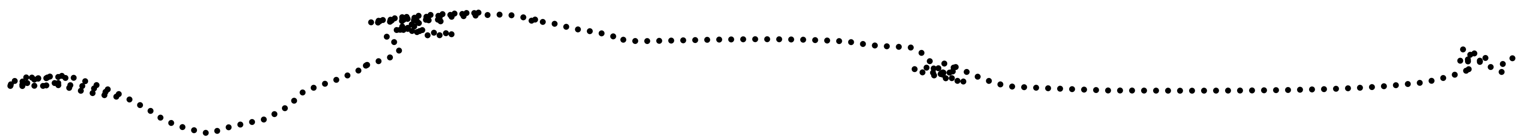
3.5. Summary

3.1.

Modeling mobility data

Trajectory definition

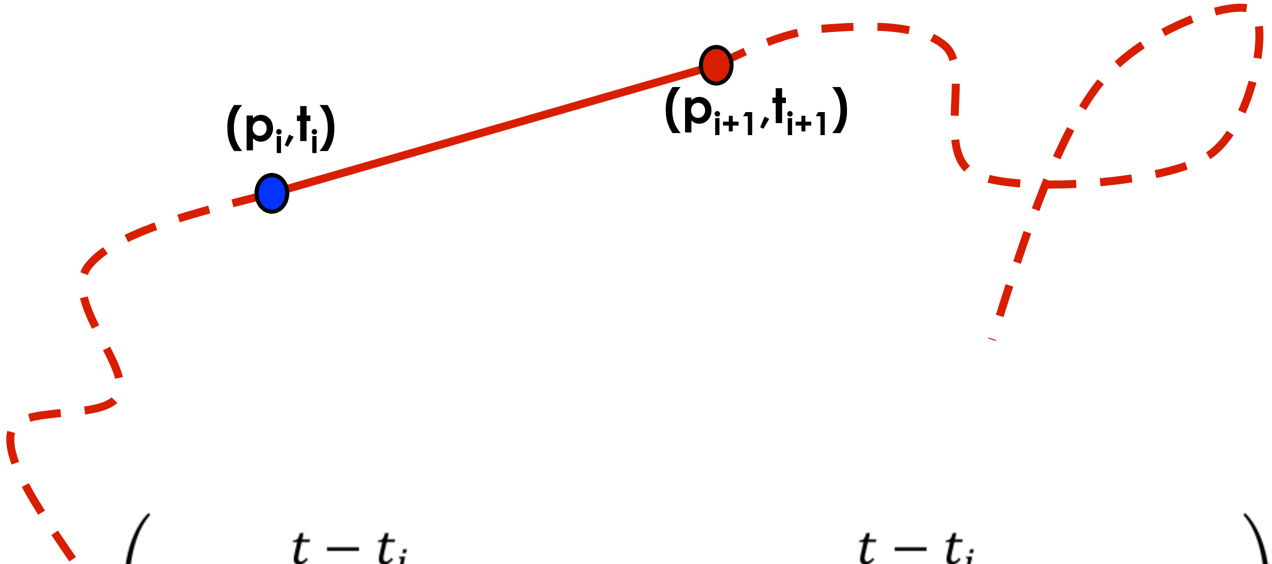
- A **trajectory** is a model for a motion path of a moving object (human, animal, robot, …)
  - (due to discretization) a sequence of sampled time-stamped locations (p_i, t_i) where p_i is a 2D point (x_i, y_i) and t_i is the recording timestamp of p_i



$$T = \{ <p_1, t_1>, <p_2, t_2>, \ldots, <p_n, t_n> \}$$

Trajectory definition (cont.)

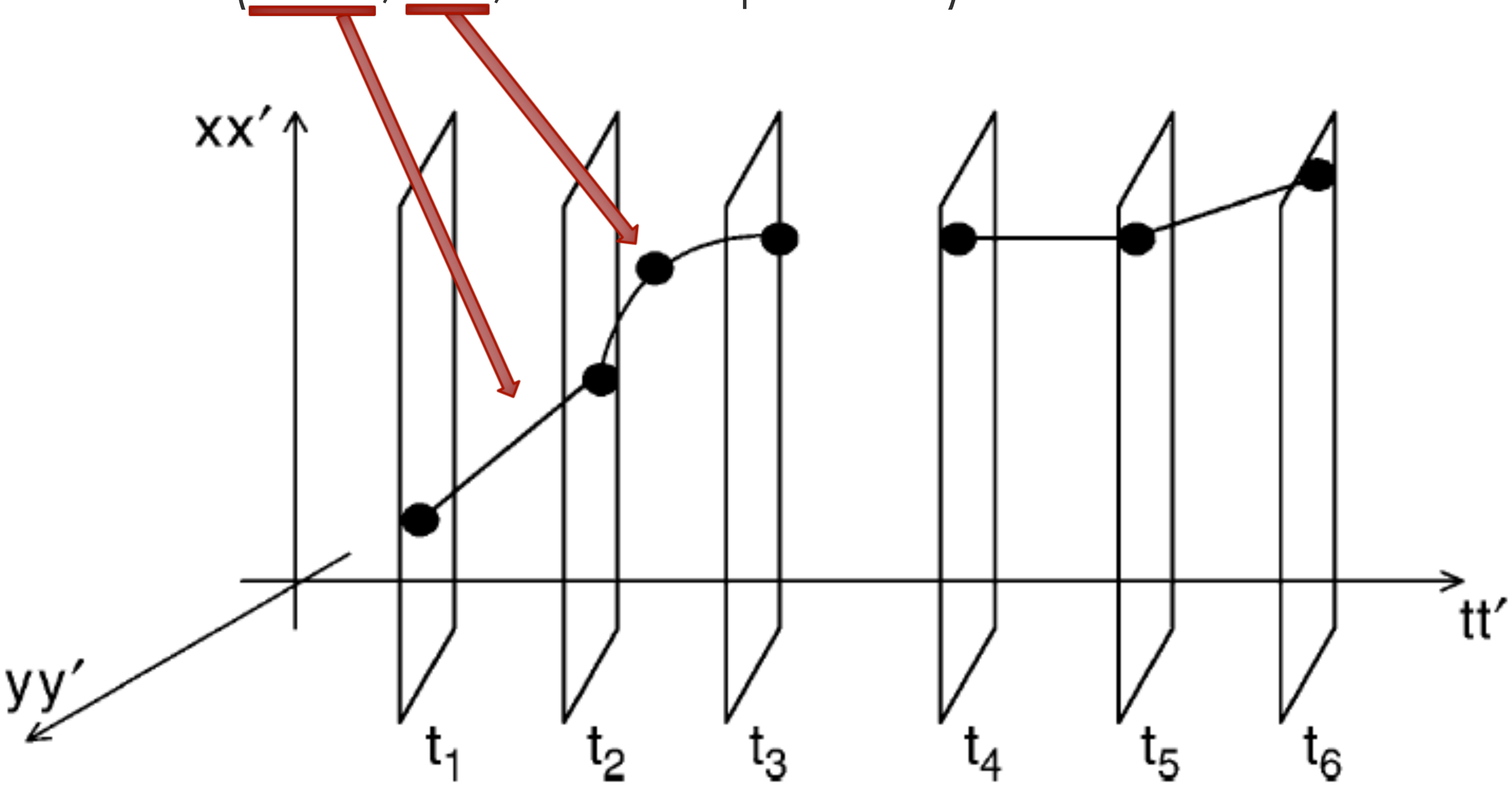
- A common representation of a trajectory is a **3D polyline** whose vertices correspond to time-stamped locations (p_i, t_i)
  - Usually, **linear interpolation** is assumed between (p_i, t_i) and (p_{i+1}, t_{i+1})



$$p(t) = \left(x_i + \frac{t - t_i}{t_{i+1} - t_i}(x_{i+1} - x_i), y_i + \frac{t - t_i}{t_{i+1} - t_i}(y_{i+1} - y_i) \right)$$

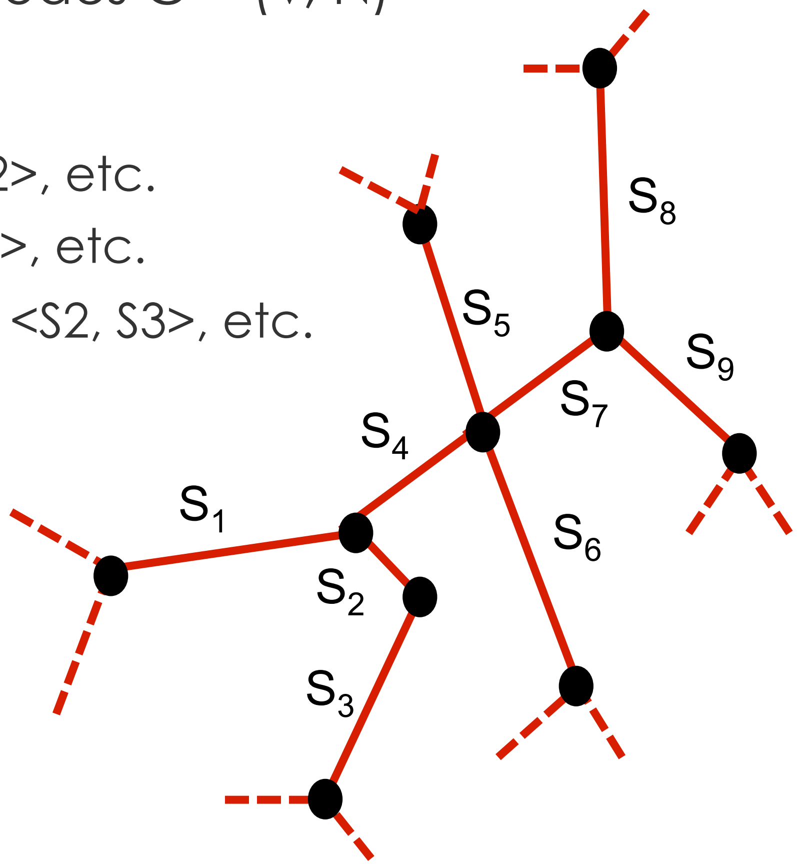
Sliced representation

- Decomposes the temporal development into ‘**slices**’
- Within each slice, the movement is modeled by a ‘simple’ function (linear, arc, etc. interpolation)



Movement on a network

- Network-constrained movement assumes an underlying network / graph of vertices and nodes G = (V, N)
- Alternative models:
  - **Segment-oriented model**: <S1>, <S2>, etc.
  - **Edge-oriented model**: <S1>, <S2, S3>, etc.
  - **Route-oriented model**: <S1, S4, S7>, <S2, S3>, etc.
- The location of an object is represented by:
  - the entity (segment / edge / route) it is located on and
  - an offset in [0, 1] denoting the relative location in the entity



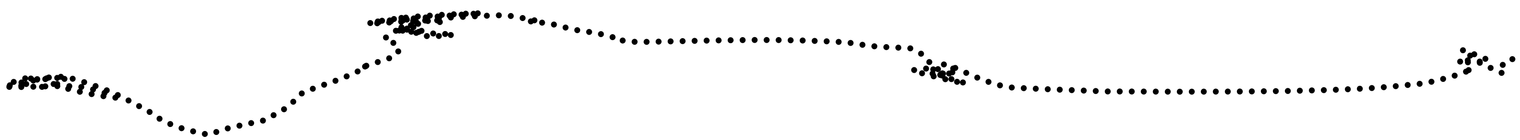
3.2.

Acquiring trajectories from raw data

Raw Data

- Example of GPS recordings

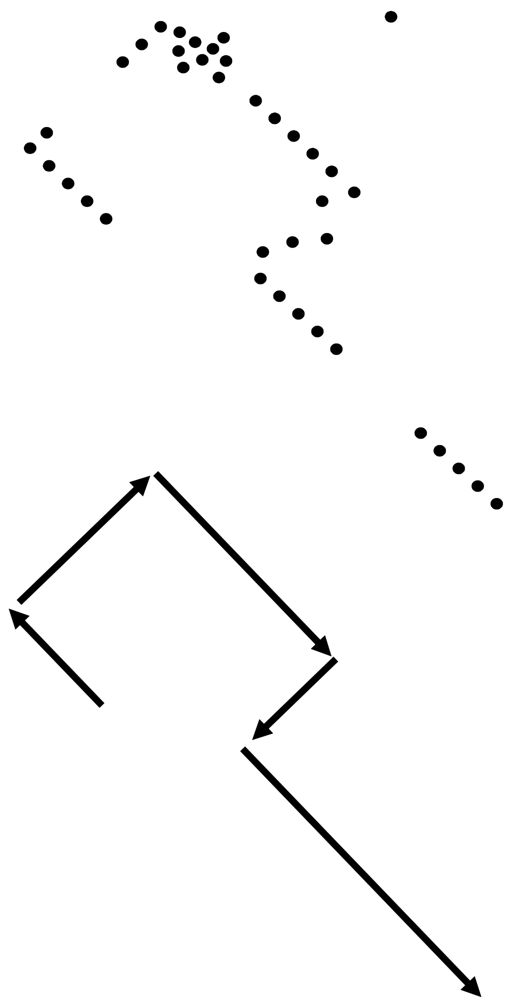
```
<trk>
...

<trkpt lat="38.17733919" lon="23.74038222”> <ele>862.62</ele>
        <time>2013-01-19T08:54:57.608Z</time> </trkpt>

<trkpt lat="38.17725880" lon="23.74043843"> <ele>1117.98</ele>
        <time>2013-01-19T08:55:21.609Z</time> </trkpt>

<trkpt lat="38.17717291" lon="23.74039676"> <ele>1129.98</ele>
        <time>2013-01-19T08:55:31.608Z</time> </trkpt>

<trkpt lat="38.17707471" lon="23.74038878"> <ele>1155.93</ele>
        <time>2013-01-19T08:55:45.584Z</time> </trkpt>

...
</trk>
```



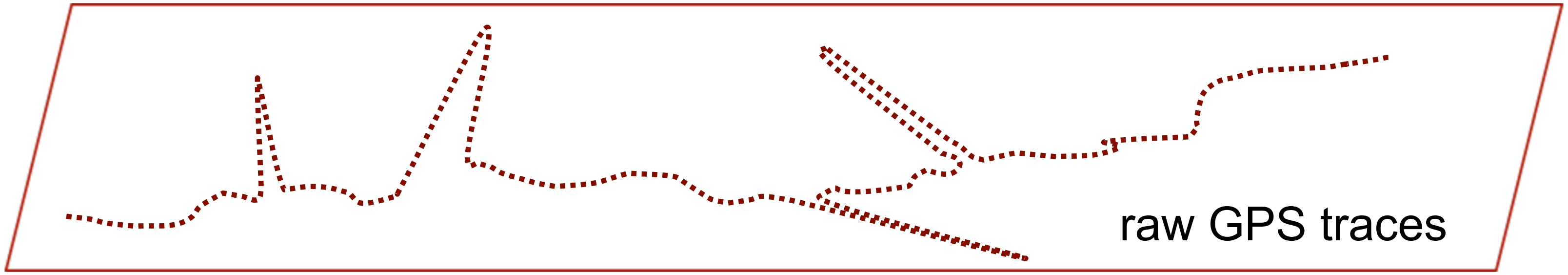
Acquiring Trajectories from Raw Data

The problem:

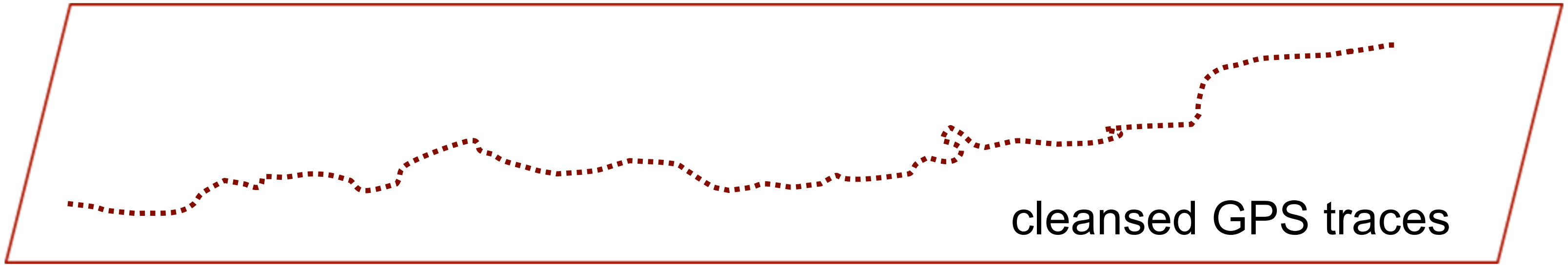
- **From raw data**, i.e., successive time-stamped locations …
- … **to meaningful trajectories**, i.e., continuous development of movement



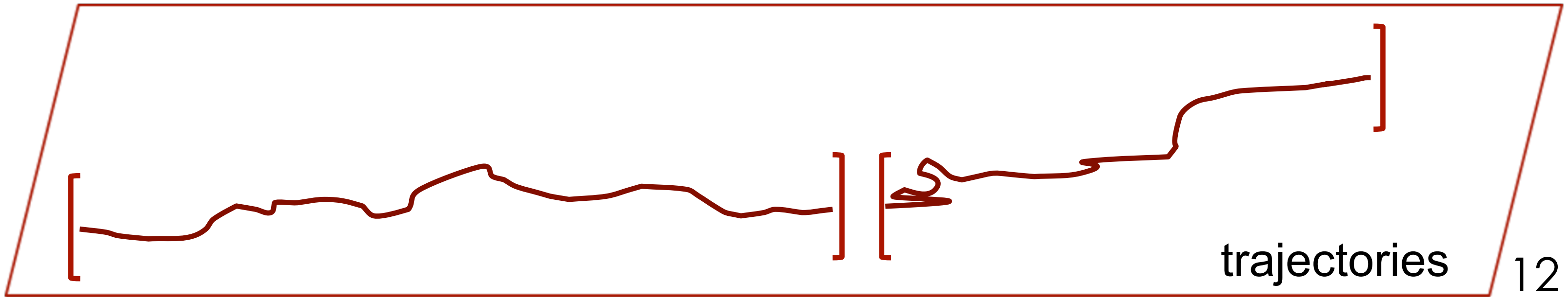
A two-step methodology



- Step 1: **data cleansing**

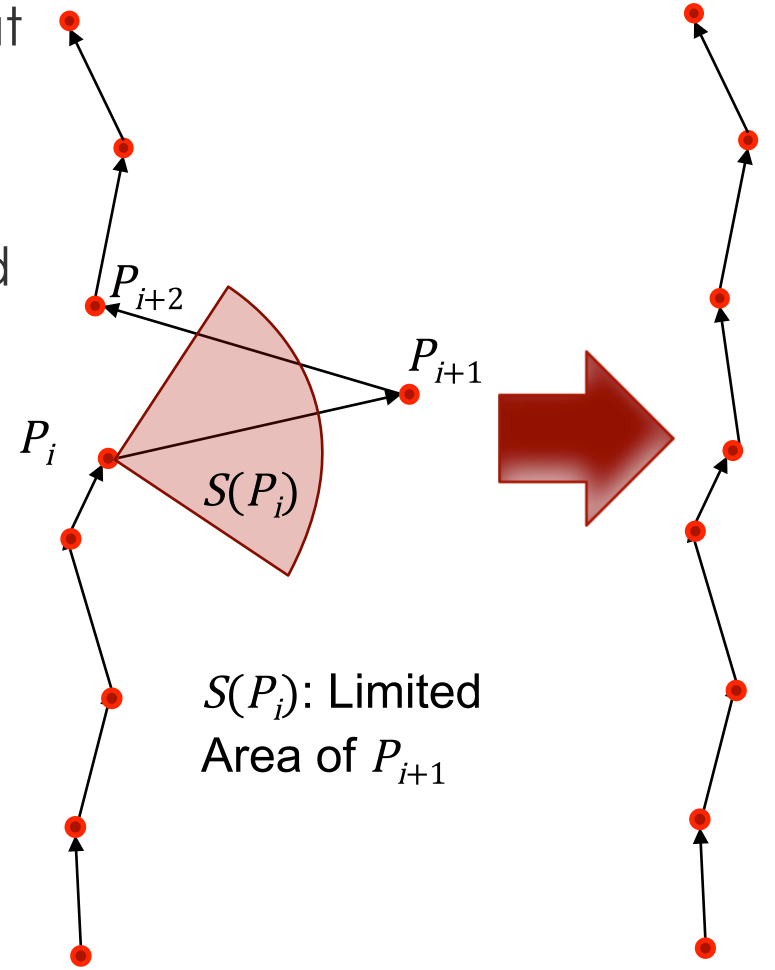


- Step 2: **trajectory identification**



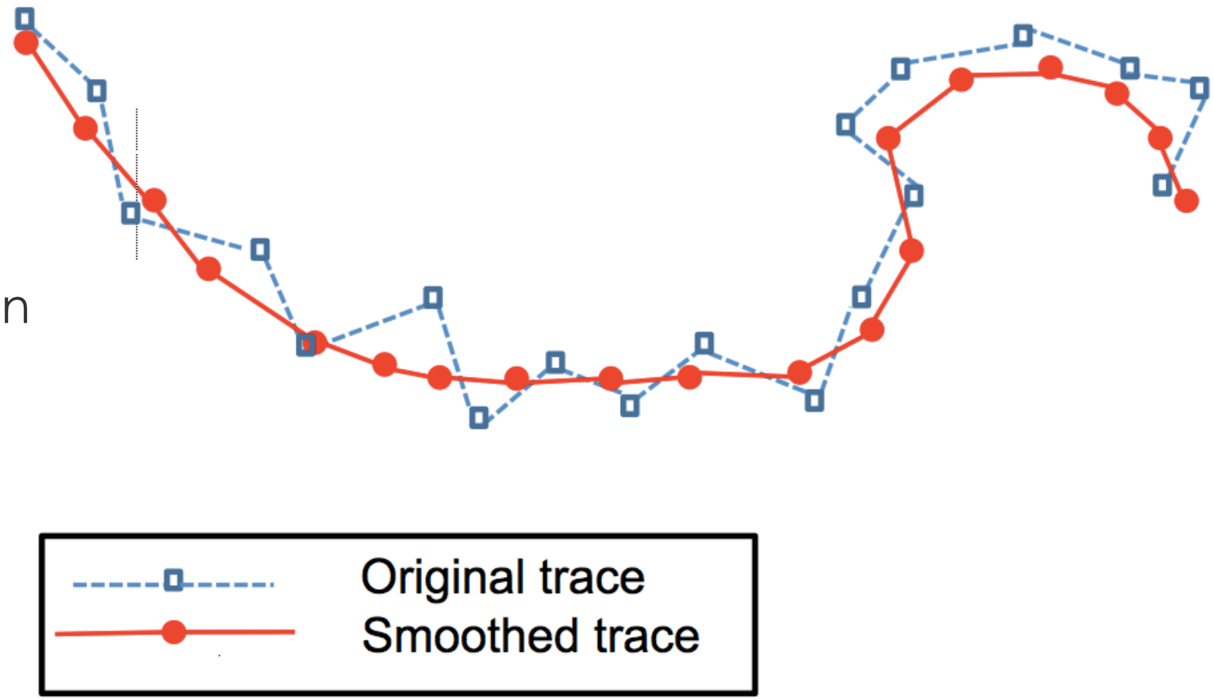
GPS Data Cleansing

- Erroneous recordings: noise vs. random errors
- **Noise** corresponds to values that are 'impossible' to appear
- Can be detected and removed using appropriate filters
  - e.g. maximum speed



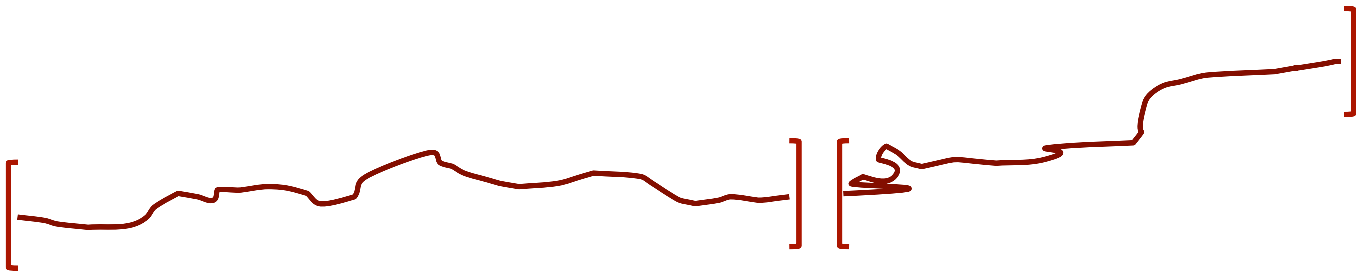
GPS Data Cleansing (cont.)

- Erroneous recordings: noise vs. random errors
- **Random errors** correspond to 'possible' values that appear to be small deviations from actual ones
- Can be smoothed using statistical methods
  - e.g. least squares spline approximation



Trajectory identification

- Goal 1: **Interpolate successive points** in order to simulate the continuous nature of movement
  - Linear interpolation (the most popular), Bezier curves, etc.



- Goal 2: **Segment sequences of points** in homogeneous sub-sequences (= trajectories)
  - Identification via raw (spatial / temporal) gap
  - Identification via prior knowledge (e.g. office hours, sleeping hours)
  - Correlation-based identification (ideas from time-series segmentation)

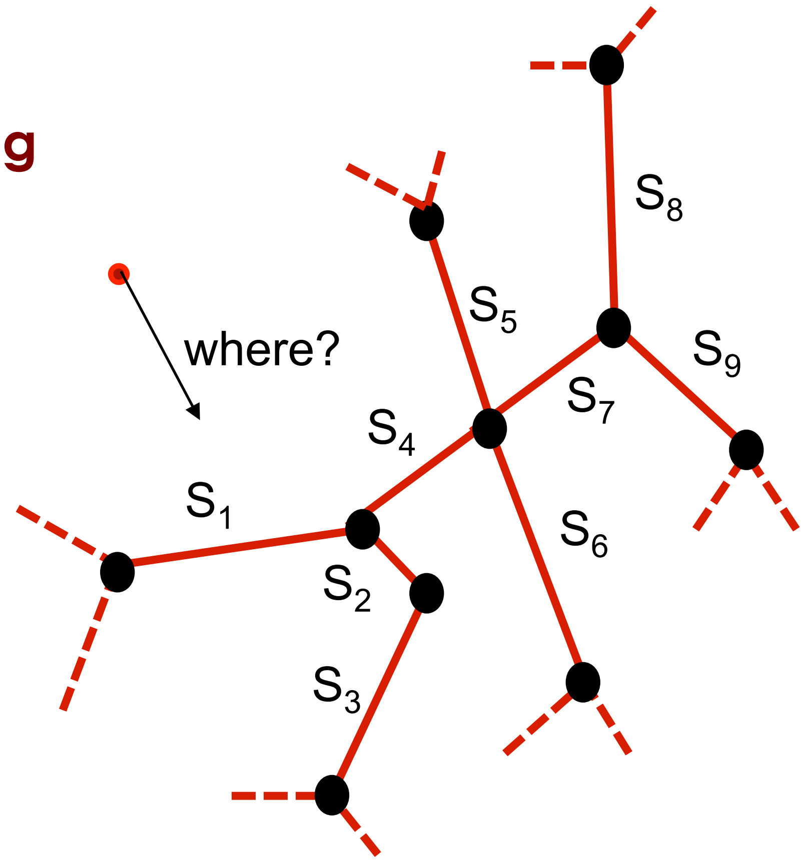
3.3.

Trajectory reconstruction and simplification

Trajectory reconstruction

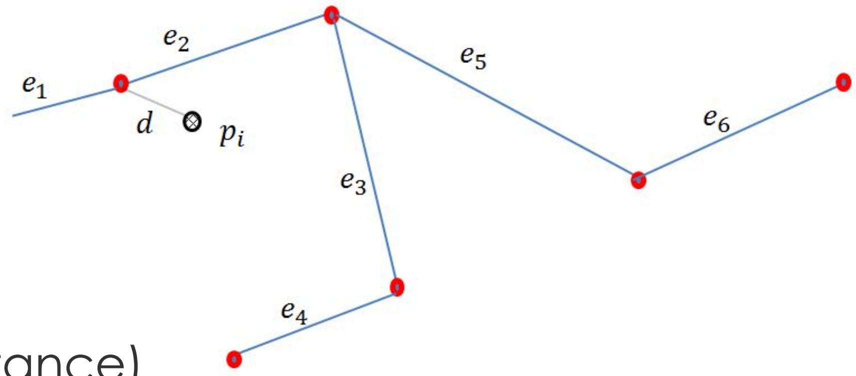
… considering network-constrained movement

- An additional step: **map-matching**
  - Geometric map-matching
  - Topological map-matching
  - Probabilistic map-matching
  - Hybrid map-matching

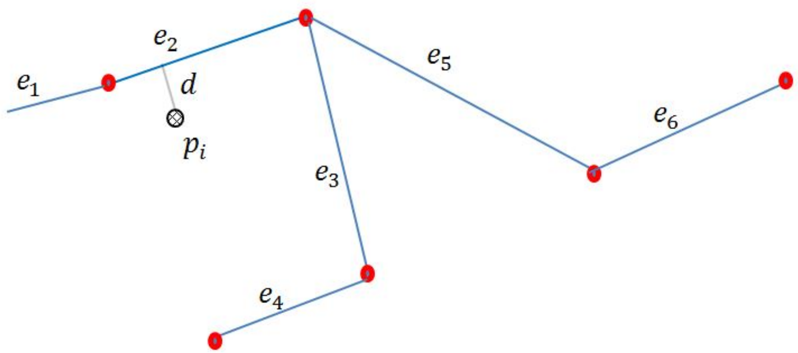


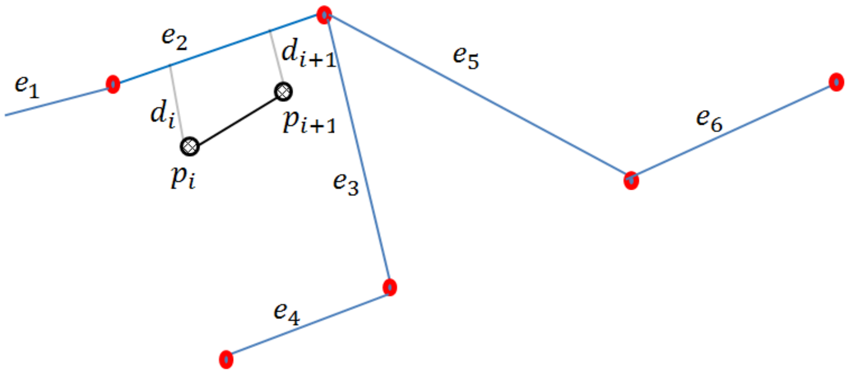
Geometric map-matching

- The basic idea: map a point into its closest position on the network



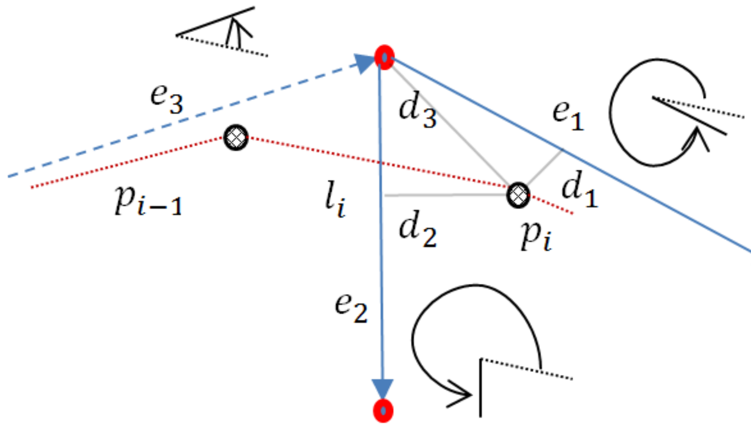
- Three types:
  - Point-to-point (e.g. Euclidean distance)
  - Point-to-curve (e.g. perpendicular distance)
  - Curve-to-curve (e.g. Fréchet distance)

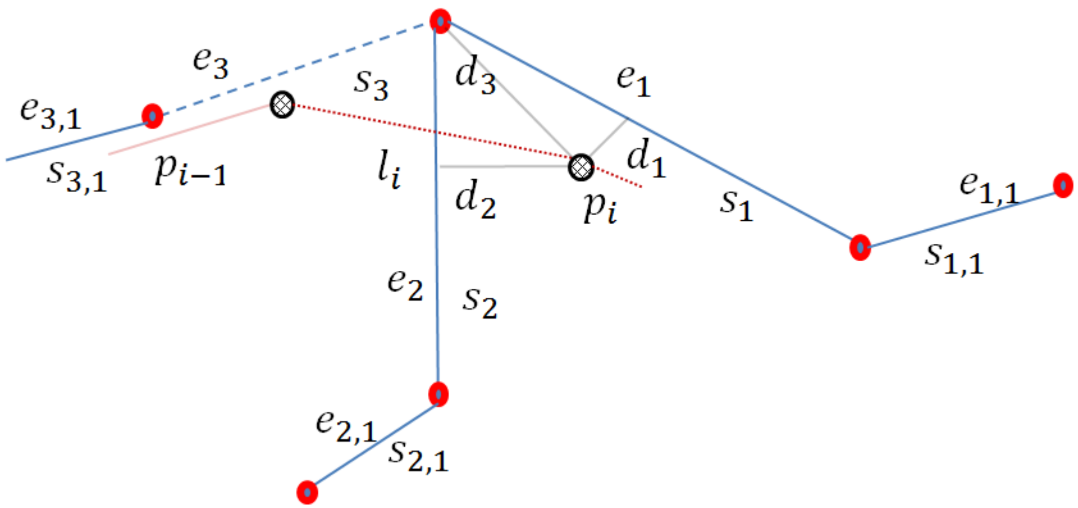




Topological map-matching

- Utilize both the geometry and the connectivity / adjacency of the graph
- Two steps:
  - Choose the most suitable node(s) of the graph
  - Match the point
- Could be enhanced by a "look-ahead" approach



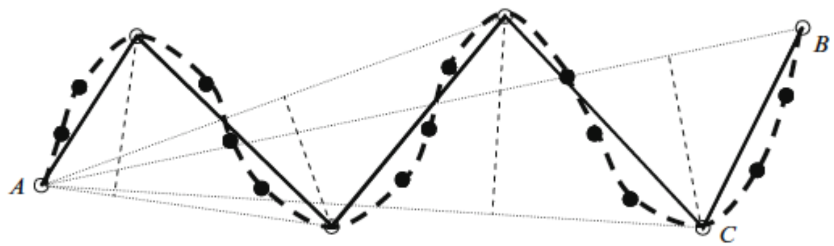


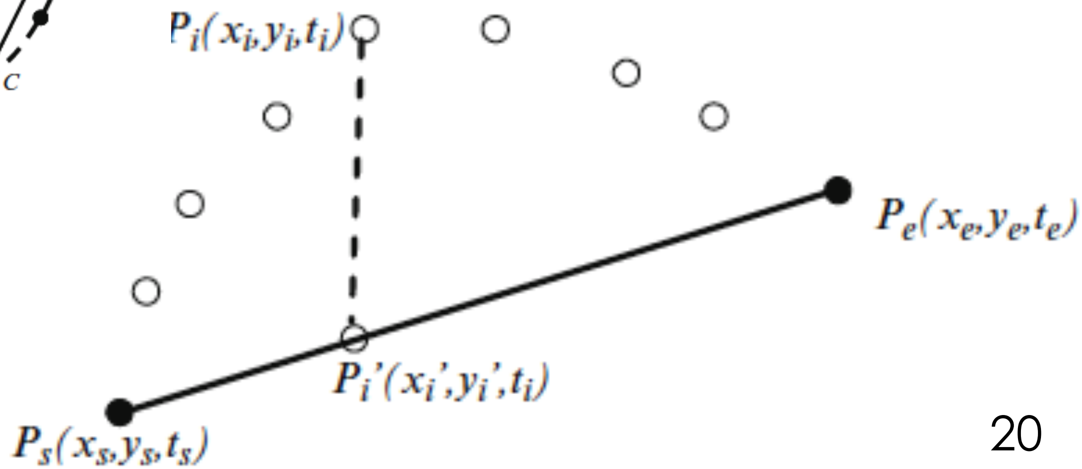
Trajectory simplification

… via data compression

- Top-down vs. bottom-up vs. sliding window vs. opening window methods

- From line (e.g. Douglas-Peucker) to trajectory simplification (e.g. Synchronous Euclidean Distance – SED)





3.4.

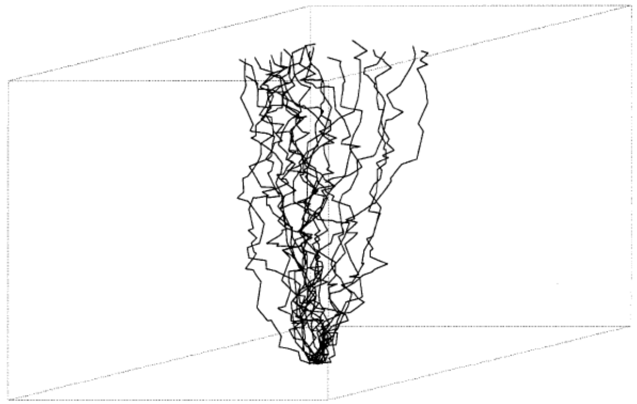
Trajectory data generators

Why data generators?

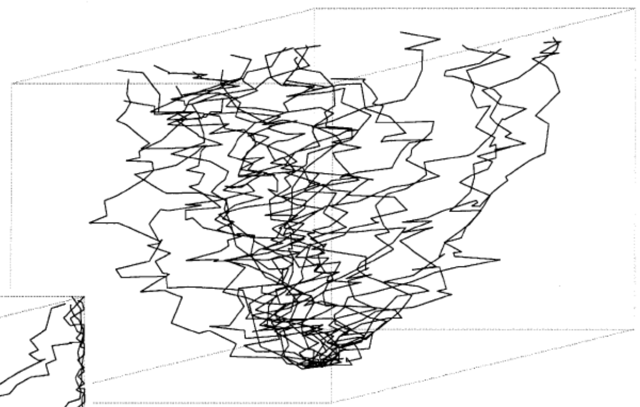
- Necessary for performance evaluation purposes
- **Microscopic** (i.e., dealing with single moving objects) vs. **macroscopic** (i.e., dealing with the traffic flow rather than single moving objects)
- Support **free** vs. **network-constrained** movement
  - Examples of generating free movement: **GSTD** (1999), **CENTRE** (2005)
  - Examples of generating network-constrained movement: **Brinkhoff** (2002), **Sumo** (2002), **Gamma** (2005), **BerlinMOD** (2008), **MWGen** (2012), **Hermoupolis** (2013)
  - We present a few next …

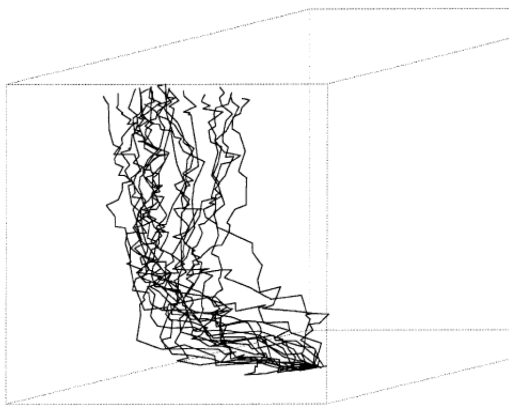
GSTD (Generating spatio-temporal data)

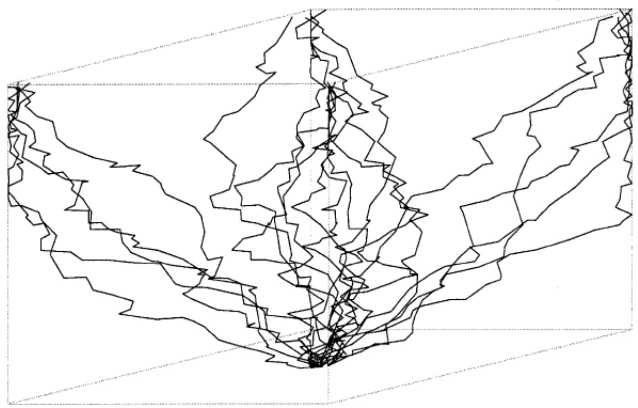
- Methodology
  - Define starting positions
  - Repeatedly, compute new time-stamped locations



- Thus, simulates different movement behaviors, e.g.
  - slow vs. fast moving objects
  - directed movement towards south vs. the four corners of the horizon

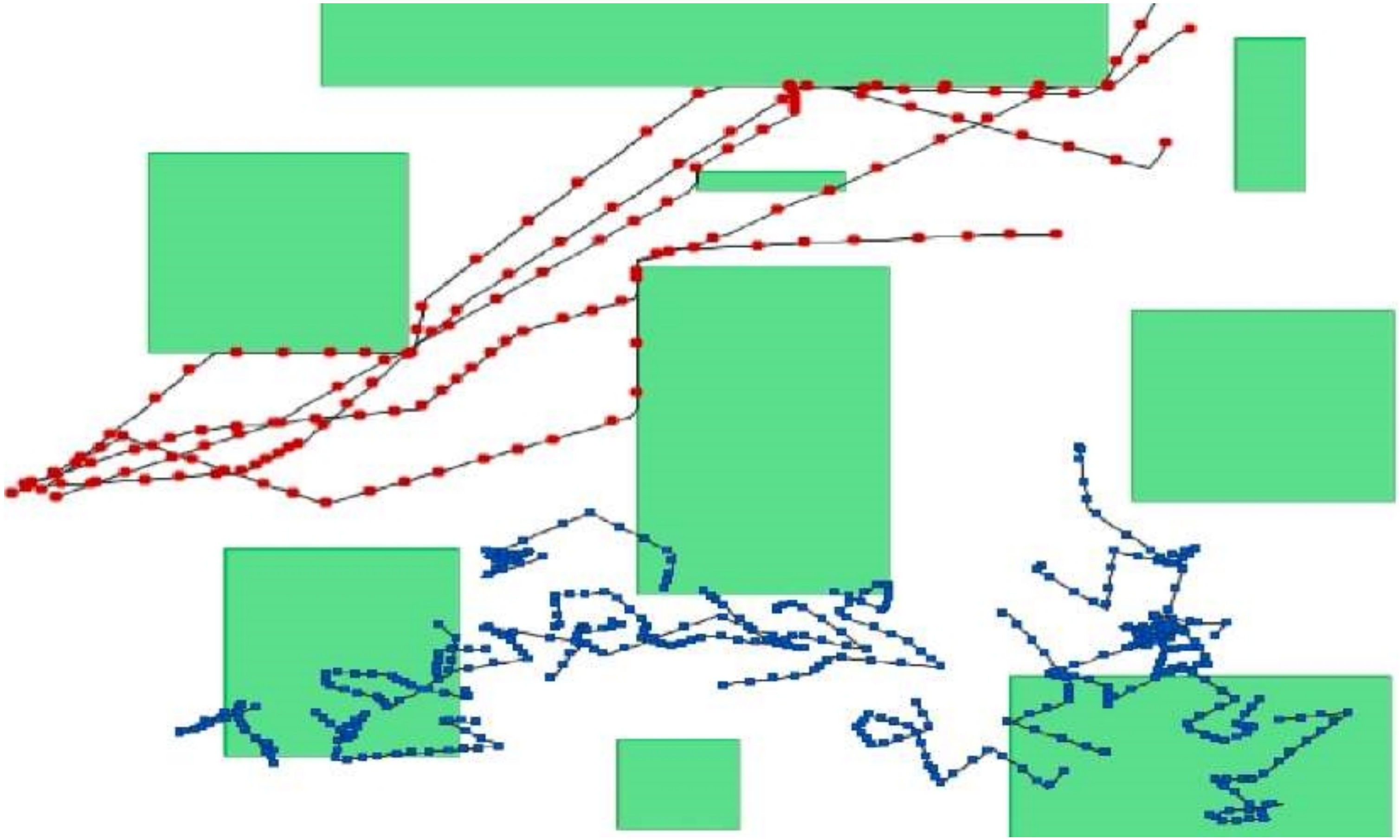






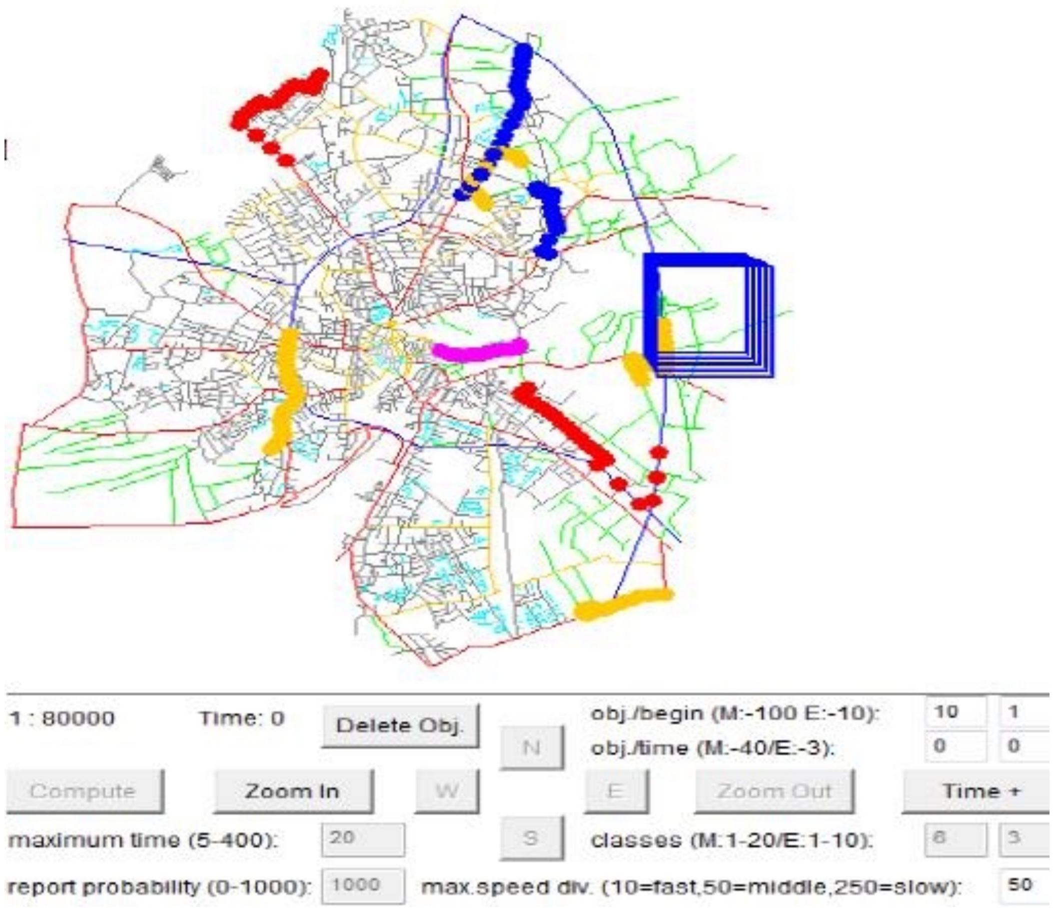
CENTRE (Cellular Network's Positioning Data Generator)

- Simulates groups of objects having different movement behavior and sensitivity to obstacles



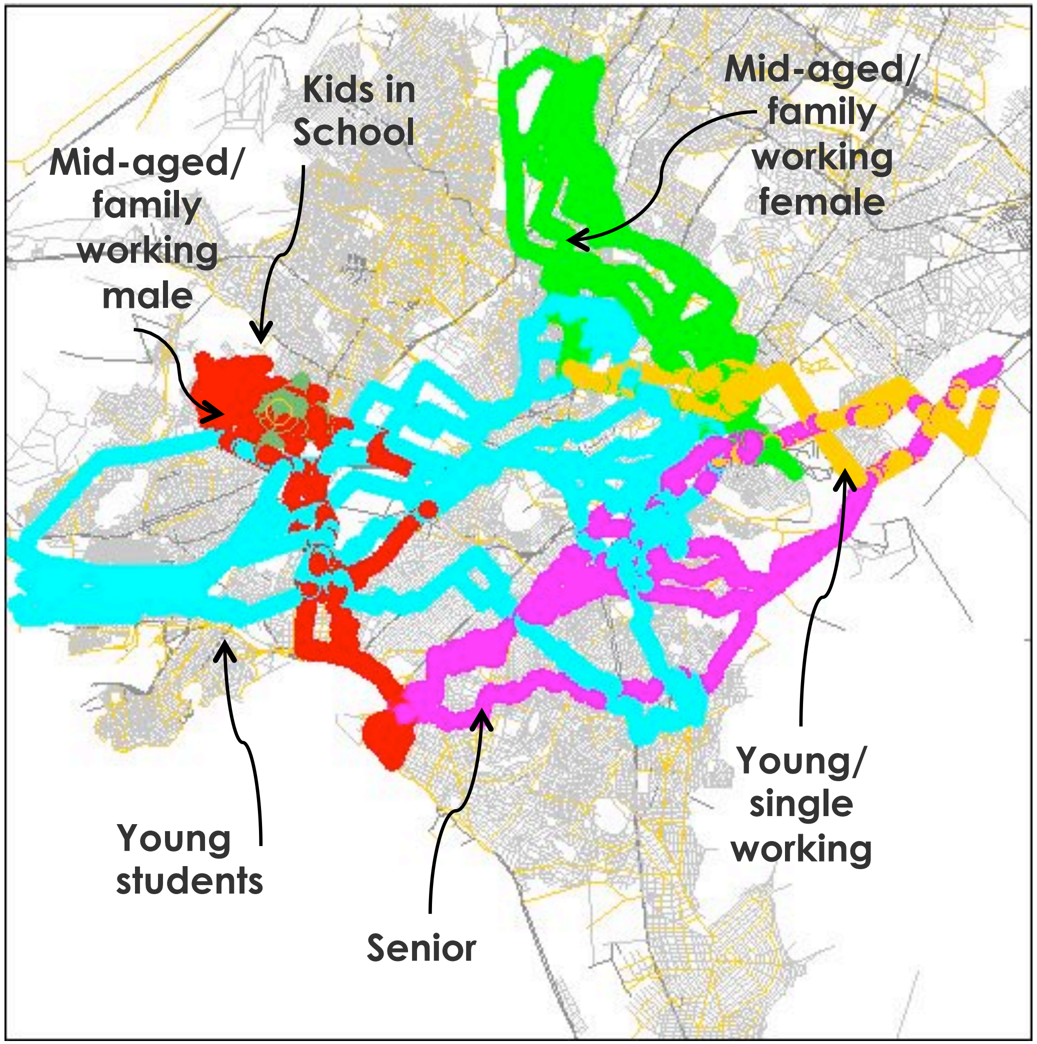
Brinkhoff's generator

- Methodology:
  - generate starting points
  - generate length of route (depending on object class)
  - generate destination for each object
  - compute the route
  - compute the trajectory by generating a random speed every time unit
    - based on capacity, weather, edge class, etc.



Hermoupolis

- Generate objects moving in an urban area …
- … according to different population profiles of given distribution
  - Kids in school: 20%
  - Young students: 10%
  - etc.
- Dual output: synchronized raw (GPS-like) + semantic (annotated) trajectories

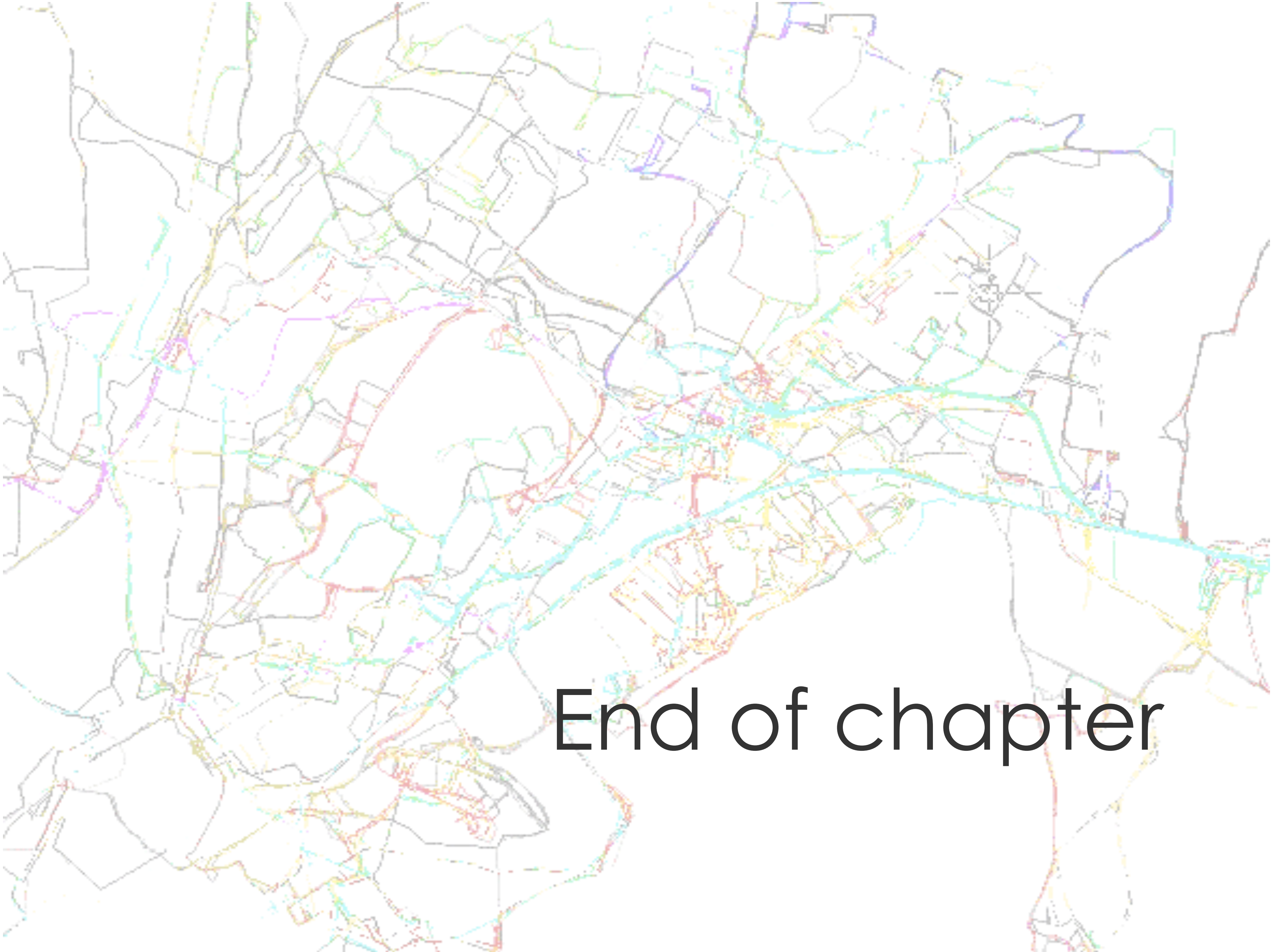


3.5.

Summary

Summarizing …

- In this chapter, we presented the various steps for building meaningful trajectories from raw (GPS-like) data:
  - Data cleansing (noise removal, random errors smoothening)
  - Trajectory identification (point interpolation, trajectory segmentation)
  - Trajectory map-matching
  - Trajectory simplification via data compression
- We also discussed trajectory data generators for evaluation purposes
  - Supporting free vs. network-constrained movement



End of chapter