BEYOND DATABASES ARCHITECTURES AND STRUCTURES
10th INTERNATIONAL CONFERENCE

AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Map-matching in a real-time traffic monitoring service

Piotr Szwed and Kamil Pękala
AGH University of Science and Technology
Department of Applied Computer Science
e-mail: pszwed@agh.edu.pl

BDAS'2014 Ustroń, Poland, May 27-30, 2014

Agenda

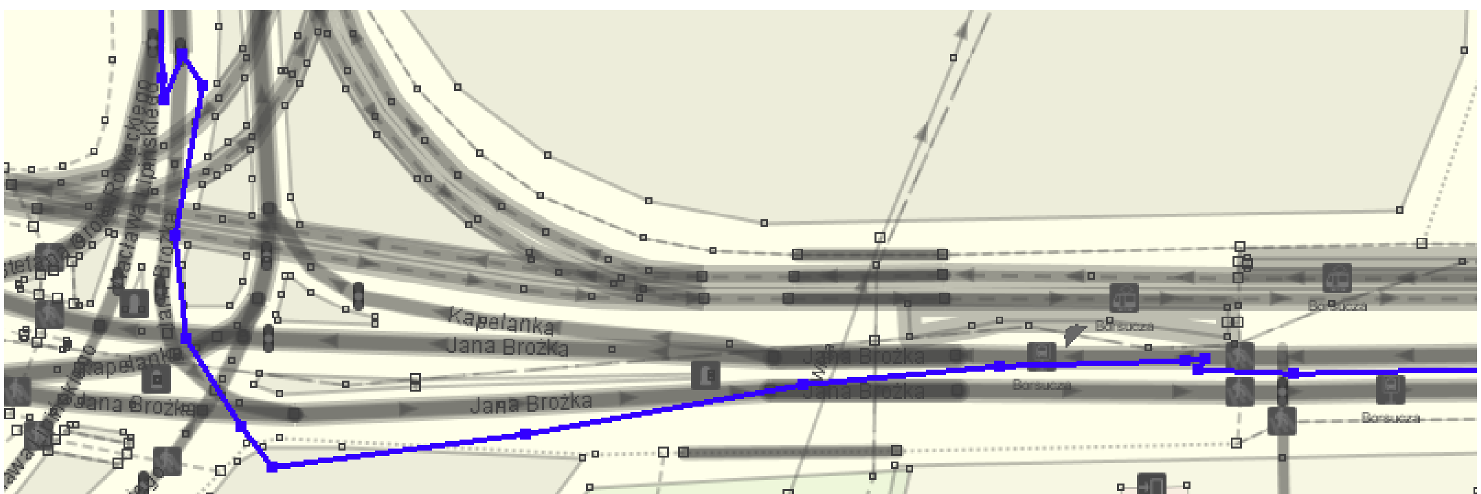
1. Motivation
2. Related works
3. Operational concept of GPS tracker (traffic monitoring system)
4. Hidden Markov Model set-up
5. Map matching algorithm
6. Test results
7. Conclusions

Motivation

- GPS tracker is a prototype implementation of real time traffic monitoring service within INSIGMA *(Intelligent System for Global Monitoring Detection and Identification of Threats)* project
- Traffic congestion: a serious problem in urban areas
- Intelligent Transportation Systems (ITS)
  - Concerns: safety, mobility and environemental performance
  - Services based on real-time traffic monitoring data
    - alerting,
    - navigation,
    - fleet management,
    - logistics
    - intelligent traffic control
- Data sources:
  - Sensors: inductive loops, cameras, microphone arrays
  - **Crowd-sourcing: smartphone devices transimitting positioning data over cellular networks**

Motivation 2

- Key issue: **map-matching - calculation of vehicle location on a road segment**
- Requirements for the map-matching algoithm
  - Should take into account roads connectivity
  - Should be *incremental*, i.e. capable of analyzing GPS trace on arrival of a new data



Related works

Map matching algorithms

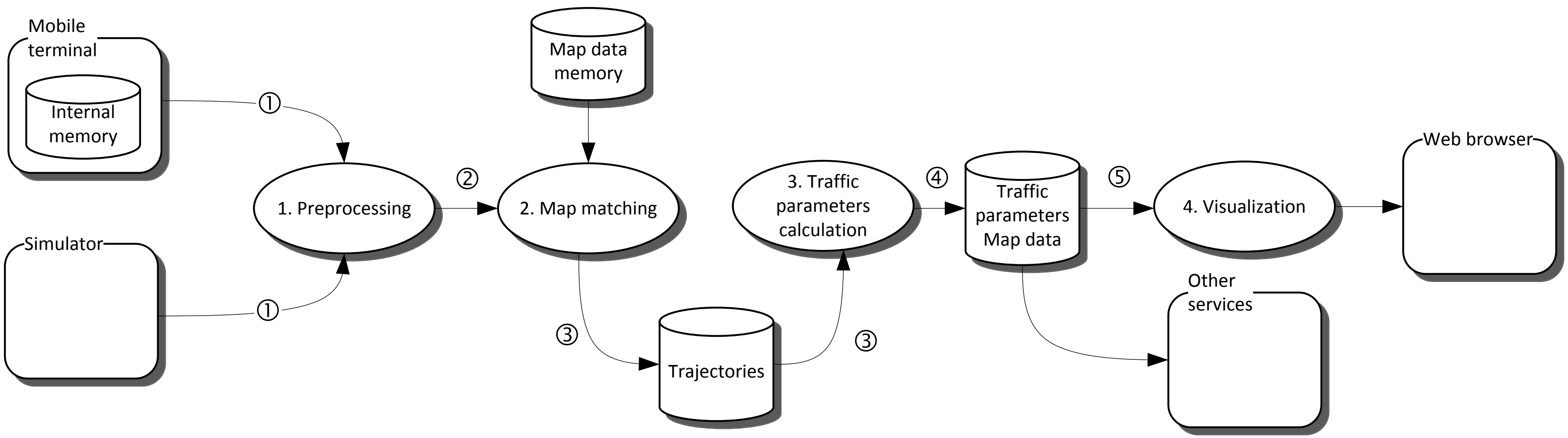
- *Geometrical* (point to curve or segment to curve matching) [White, Bernstein et al. 2000; Greenfeld 2002]
- *Topological*: utilize information about connections between road segments [Quddus, Ochieng et al. 2003].
- *Probabilistic*: use information on circular or elliptic confidence region associated with position reading [Ochieng, Quddus 2009]
- *Advanced*: Kalman filter, fuzzy rules, Particle filters (both topological and probabilistic) [Fu, Li et al. 2004; Gustafsson, Gunnarson et al., 2002]
- *Incremental* algorithms using tree-like structure [Marchal, Hackney et al. 2004; Wu, Zhu et al. 2007]
- Global algorithms based on *Hidden Markov Models* [Newson, Krumm 2009; Thiagarajan, Ravindranath et al. 2009]

Related works

Traffic monitoring systems:

- Mobile millenium project, University of California, Berkeley, http://traffic.berkeley.edu/
- INRIX, http://www.inrix.com/default.asp
- Google: The bright side of sitting in traffic: Crowd-sourcing road congestion data. http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html
- Gurtam: Commercial GPS solutions for vehicle tracking and fleet management: http://gurtam.com/en/

Operational concept



1. Preprocessing: trajectory smoothing with Kalman filter and interpolation of points between GPS readings.
2. **Map matching**: finding a seqence of projections on map segments forming a trajectory
3. Traffic parameters calculation (average speed and travel time). Includes data fusion and removal of aged data.
4. Other services (route planning, traffic control and Visualization)

Hidden Markov Model

Hidden Markov Model: $\lambda = (Q, A, O, P_t, P_o, q_0)$

Q – set of states

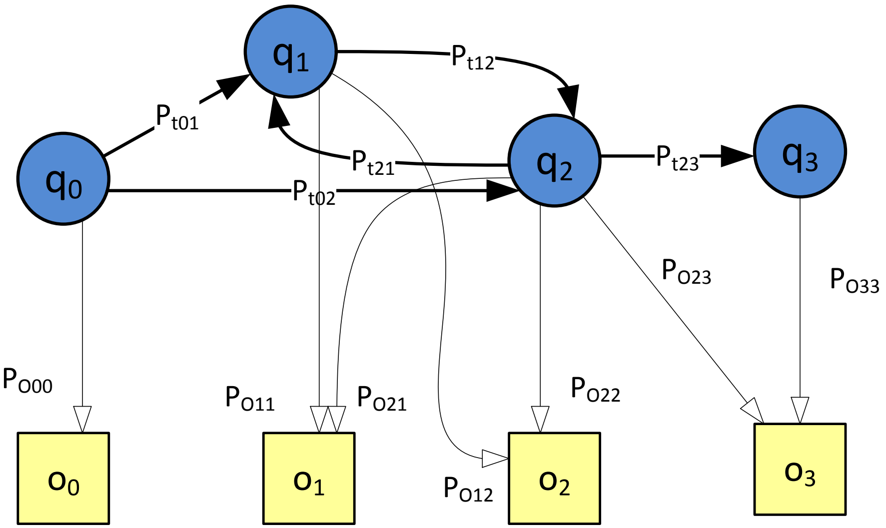
$A \subset Q \times Q$ – set of arcs

O - set of observations

$P_t : A \rightarrow (0,1]$ – state transition probability

$P_o : Q \times O \rightarrow [0,1]$ – emission probability

q_0 - initial state

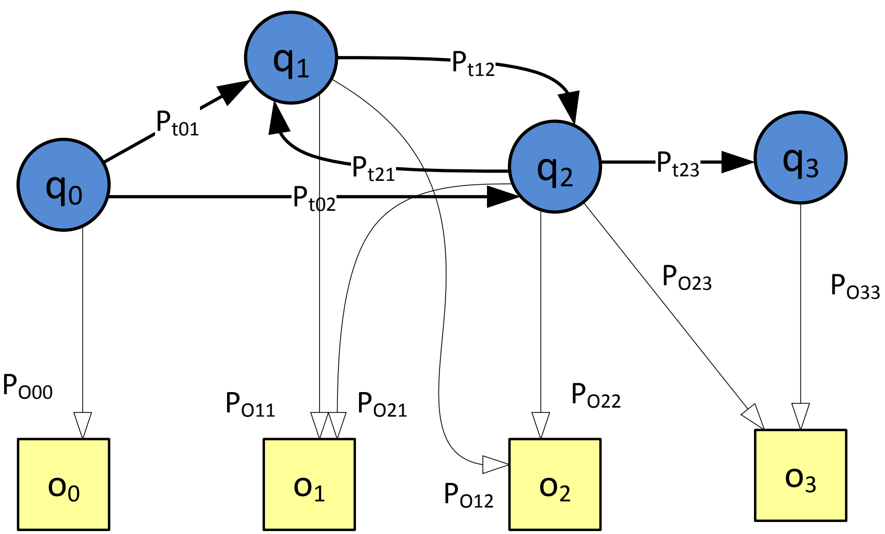


Hidden Markov Model – Decoding

Decoding problem:

- given a sequence of observations $o_{i1}, o_{i2}, \dots, o_{in}$
- find the most probable sequence of hidden states $q_{i1}, q_{i2}, \dots, q_{in}$

Resolved with Viterbi algorithm



Idea of application to map-matching:

- observations: readings from a location sensor (GPS, WiFi)
- hidden states: road segments

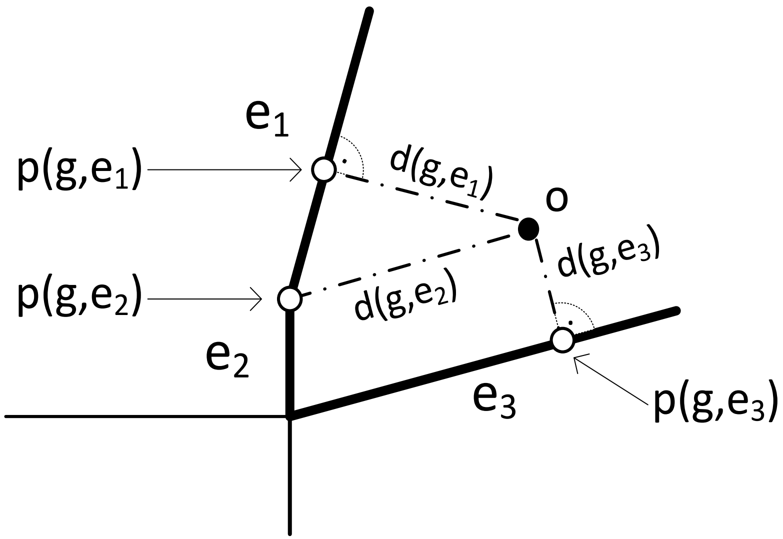
HMM model setup 1

Road network model

$G = (V, E, I)$, where

- $V \in \boldsymbol{R} \times \boldsymbol{R}$ – node (longitude, latitude)
- $E \subset V \times V$ – edge (straight segment)
- $I \subset E \times E$ – specify forbidden maneuvres at junctions

Projection of a point g on a segment $\boldsymbol{e} = (\boldsymbol{v_b}, \boldsymbol{v_e})$

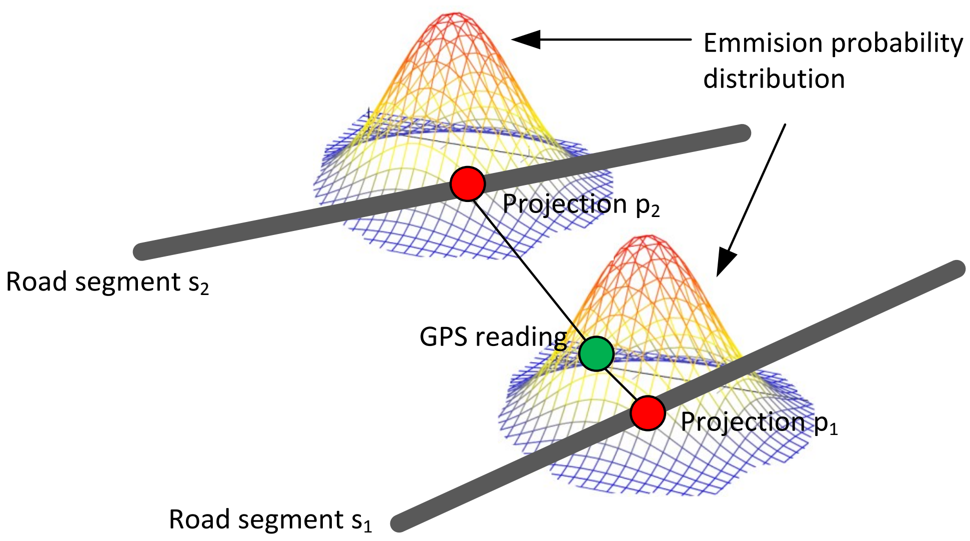


$$p(e, g) = \underset{g' = v_b + t(v_e - v_b) \wedge t \in [0,1]}{\arg\min} d(g, g')$$

$d(g, g')$ – distance (haversine formula)

HMM model setup 2

- HMM state $q = (e, p, i)$ e- road segment, p – projection point, i – sequence number
- Transition probability $P(q_1, q_2)$ – equal at junctions, low probability for forbidden meneuvres, dead recokonning on speed



- Observation $o = (lon, lat, time)$
- Normal distribution for the emmision probability:

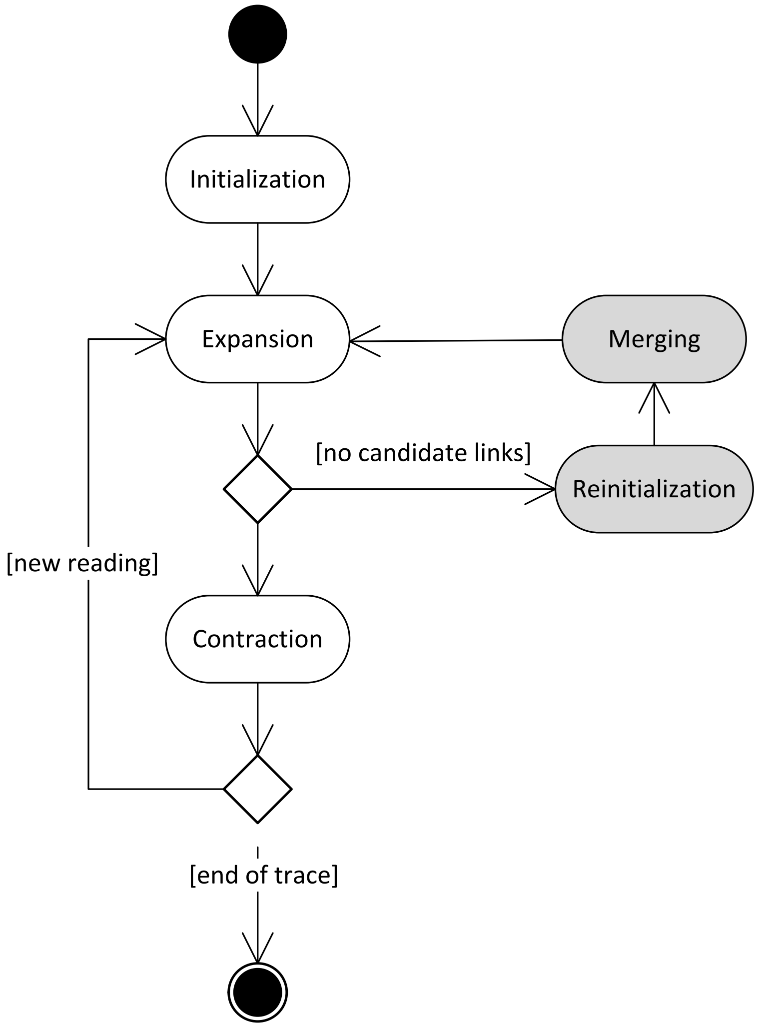
$$P(x,y) = \frac{1}{D} e^{-k\left((x-x_p)^2+(y-y_p)^2\right)}$$

(x_p, y_p) - projection point

k – depends on a sensor

$D = \iint_{-\infty}^{\infty} P(x,y)dx\,dy$ – normalization

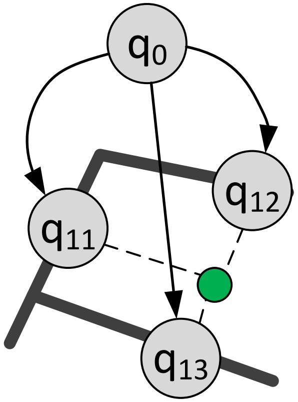
Map matching algorithm

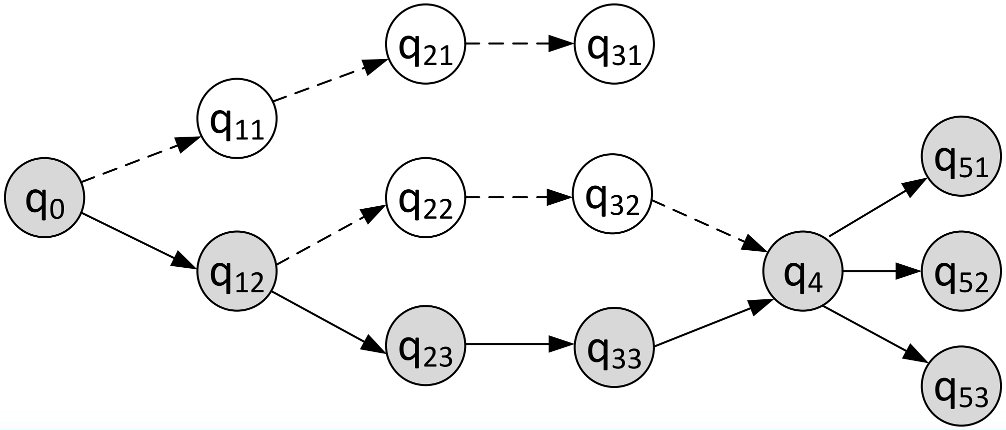


- Input: seqence of observations $\omega = (o_i : i = 1, n)$
- Internal data: a seqence of HMMs $\Lambda = (\lambda_i : i = 1, n)$
- Output: sequence of HMM states (projections of observations on road segments)

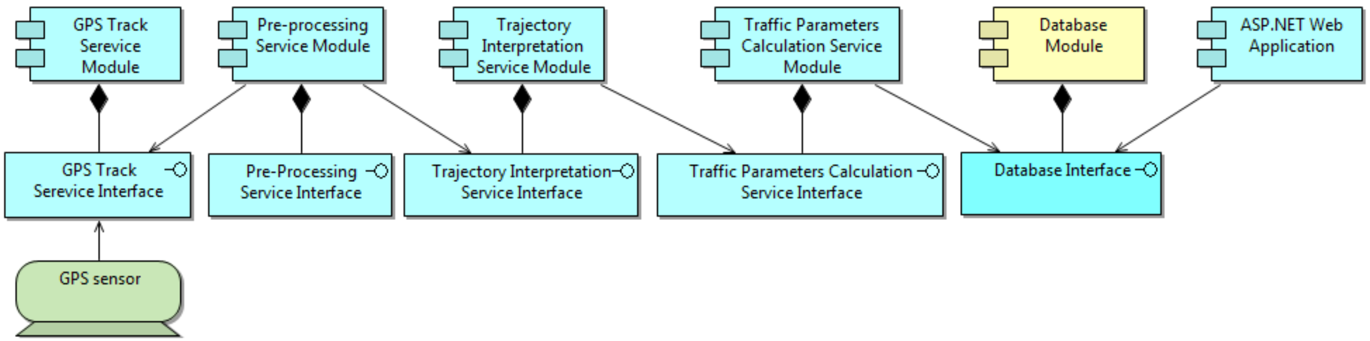
Map matching algorithm

- **Initialization**: the first model λ_1 is built by linking an artificial state with projections of o_1 on road segments
- **Epansion**: observation o_i is projected on road segments. Then, obtained new states are linked with the last states (segments) from λ_{i-1}
- **Contraction**:
  - orphan nodes without successors are removed
  - the HMM root is moved forward and a next part of the trajectory is output.





Implementation



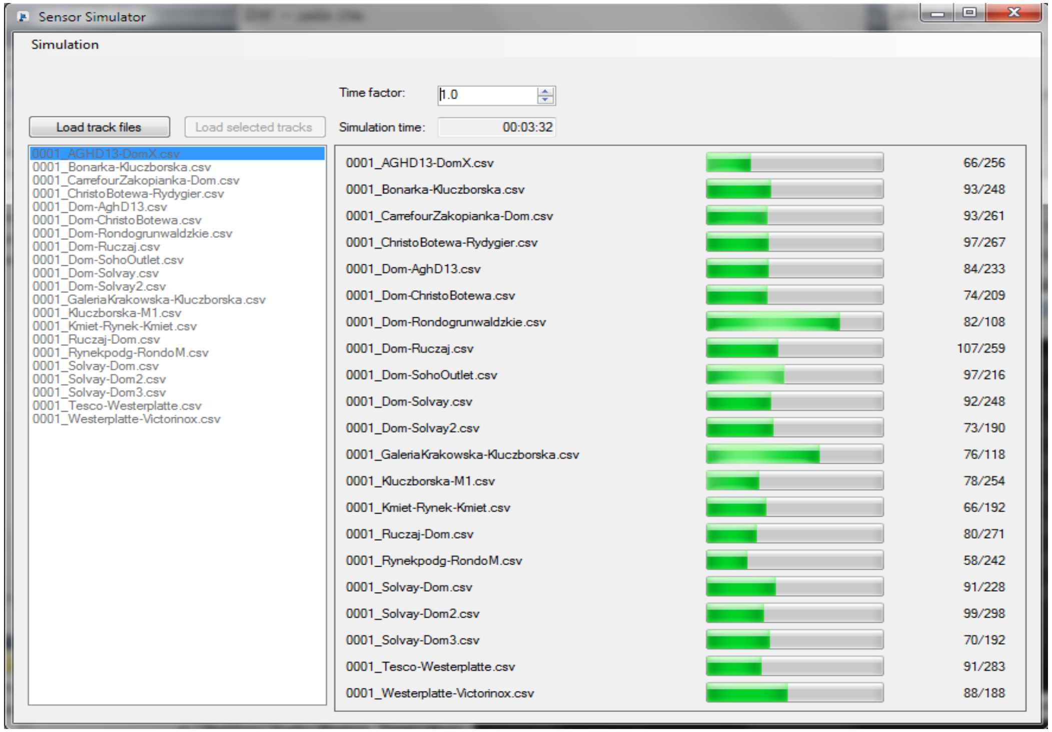
- System is implemnted in C# on .NET 4.0 platform
- Distributed into several components
- Communication via WCF web-services (SOAP and JSON)
- OpenLayers for visualization

Test results (accuracy)

- Map data source: Open Street Map (OSM)
- 20 GPS traces registered with EasyTrials GPS software on iPhone 5 (148.6km, 4482 readings)
- Criterion: number of reinitializations

	RI - Number of reinitializations	RI/sample	Av. distance between RI
Normal	24	0.005	6.18 km
Noise (20m)	73	0.016	2.03 km
HS (Half-sampled)	23	0.005	6.44 km
HS+Noise	45	0.010	3.29 km

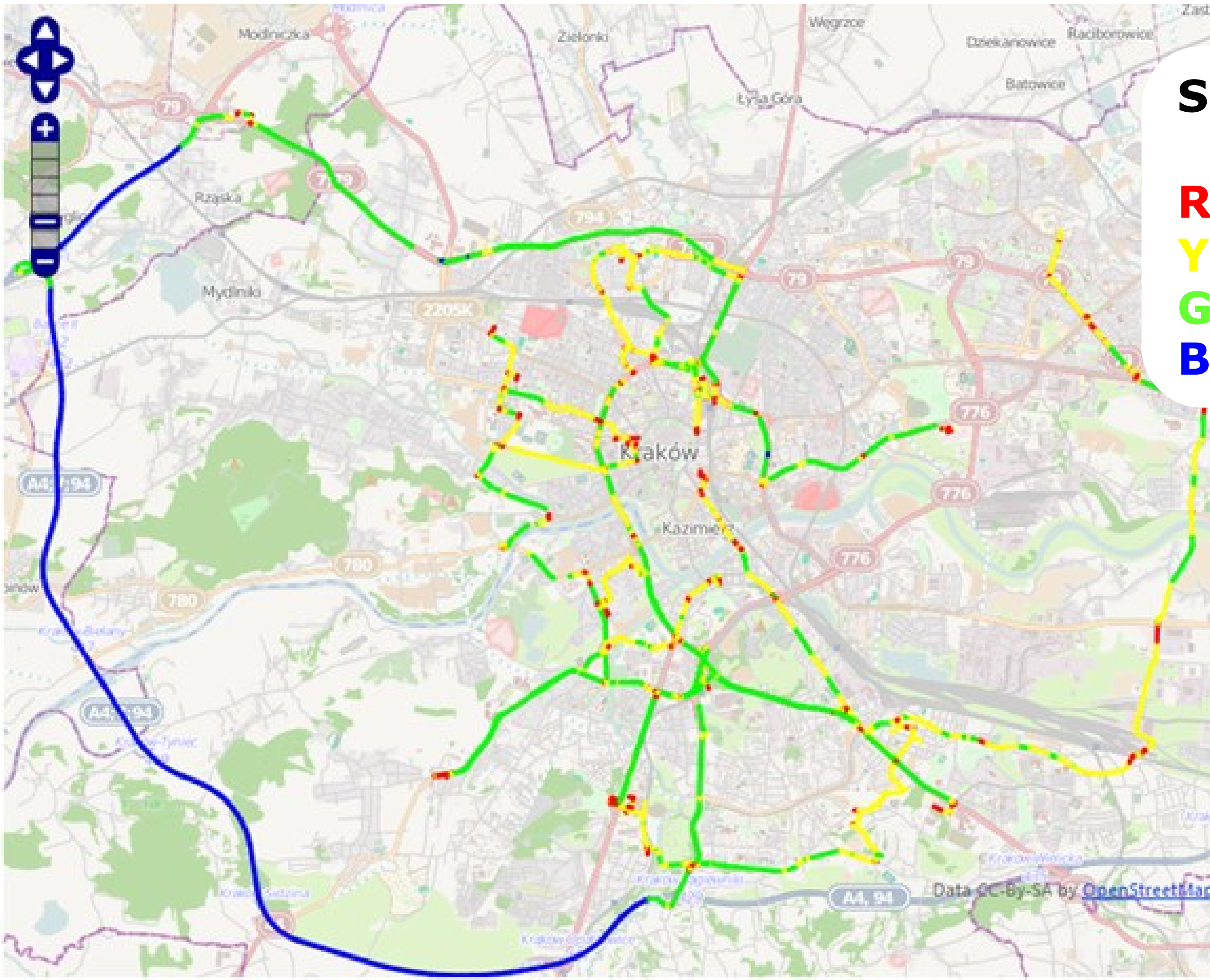
Test results (performance)



Mock client:

- 20 simultaneous feeds
- Speed-up 50 x
- Equivalent to 1000 mobile sensors

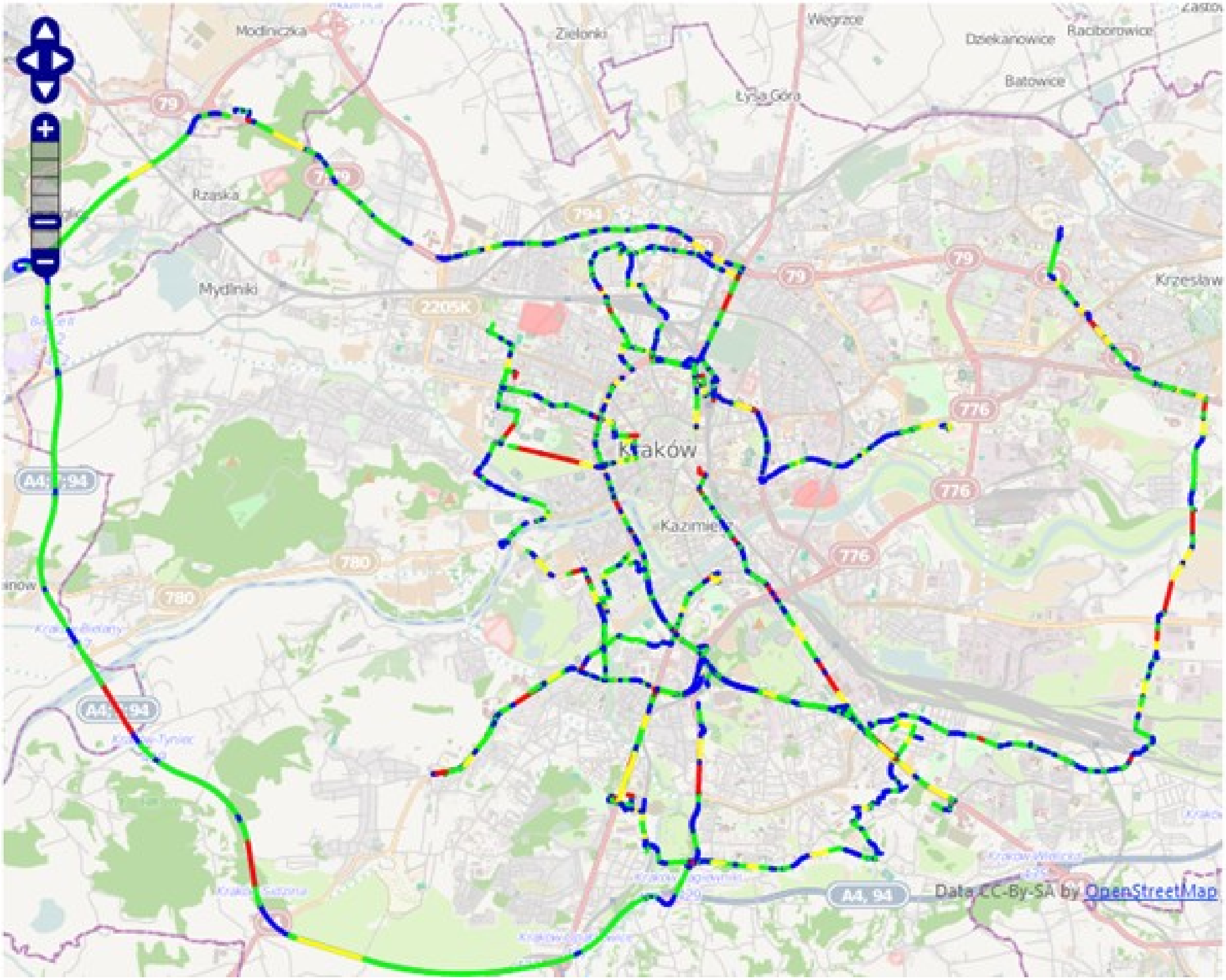
Test results: speed map



Speed km/h

Red	[0,20)
Yellow	[20,50)
Green	[50,90)
Blue:	[90,∞)

Test results: travel time



Conclusions 1

GPS tracker

- Real-time traffic monitoring system based on GPS positioning information originating from traveling vehicles.
- Stages:
  - Kalman filtration,
  - interpolation,
  - map-matching
- Vehicle trajectories are the basis for calculation of traffic parameters

Conclusions 2

Map-matching algorithm

- Based on Hidden Markov Model
- In each iteration HMM is
  - expanded by adding new states (projections on road segments)
  - contracted to output a next part of a vehicle trajectory.
- Structure of HMM forms in most cases a tree [Wu, Zhu et al. 2007] but parallel roads are supported
- Viterbi algorithm used only, if parallel roads are encountered
- The algorithm is *incremental* (required for real-time services)

Thank you