Real-Time Movement Visualization of Public Transit Data

Master Thesis

Patrick Brosi

Albert-Ludwigs-Universität Freiburg
Department of Computer Science

March 31, 2014

Outline

Motivation

Real-time visualization?

- map of the entire network (schematic or real-world)
- shows each vehicle's real-time position
- live and interactive

Motivation

Why a real-time map?

- Where is my bus? How much longer do I have to wait?
- **Which vehicles are currently in my area?**
- What is the current overall status of the network?
- How good is the coverage of certain areas?
- Promotional value
- **Combination with traditional route planners**

Motivation

Existing maps

- swisstrains.ch (static)
- Zugradar Deutsche Bahn, Zugradar ÖBB (real-time)
- Live-Timetable S-Bahn Munich (real-time)
- Nahverkehrskarte.de (Stuttgart region, static)
- Stadtbahntracker (Freiburg region, static)

But: no universal, scalable toolset available.

Challenges

Goal efficient live map that is able to visualize public transit of the whole world

Problem even public transit data of single cities is usually very complex

	trips	stations	arr/dep events	vertices	vehicles 8am
Netherlands	111,537	73,293	2,438,857	3,843,780	4,8 k
New York City	81,950	34,948	3,126,850	3,482,713	4,7 k
Switzerland	77,949	21,689	2,092,196	—	2 k
Turin	16,399	7,250	498,524	198,946	820
Freiburg	7,595	1,611	97,535	—	~ 150
Vitoria-Gasteiz	1,766	338	35,867	4,198	65

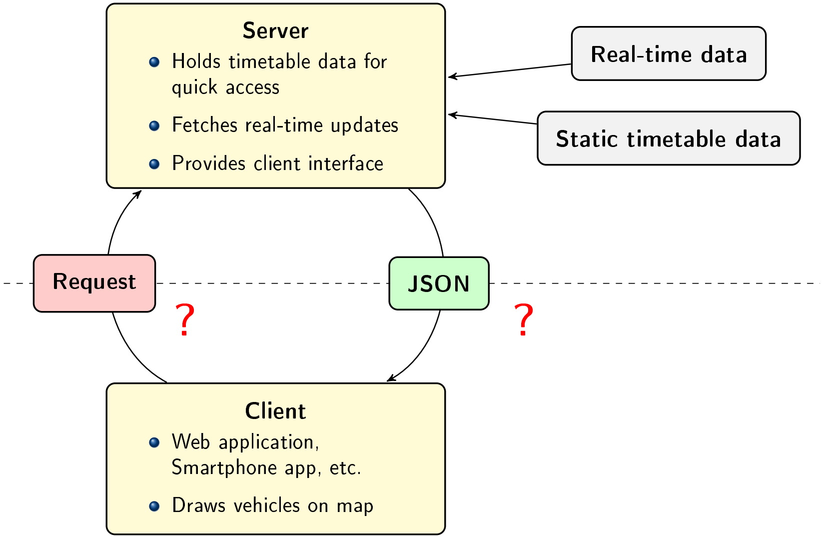
Basic Approaches

Two basic approaches to realize live transit maps:

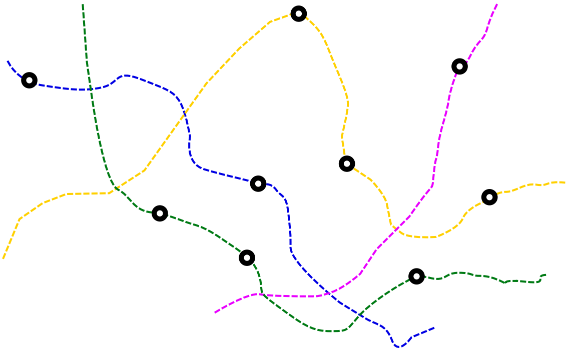
1. **Static timetable visualization**. Interpolation of static timetables → no delay information
2. **Periodic position updates**, positions obtained through GPS devices in vehicles → high traffic, no fallback

Third approach: Combination of interpolated static timetable and delay information → possibility of look-ahead requests.

Basic Concept



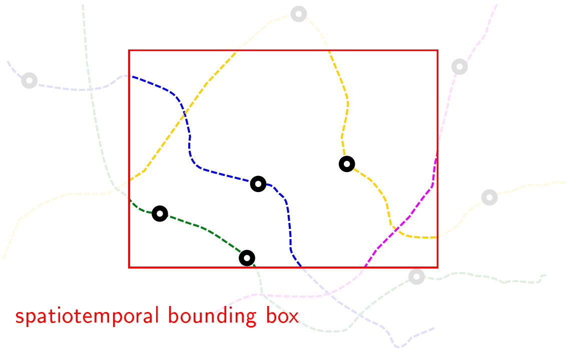
Trajectories



Definition

A trajectory $\mathcal{T}$ is a 2-tuple $(\mathcal{P}, \alpha)$ where α is the activity function and $\mathcal{P} = \{p_1, ..., p_n\}$, $p_i = (x_i, y_i, t_i)$ the ordered set of waypoints.

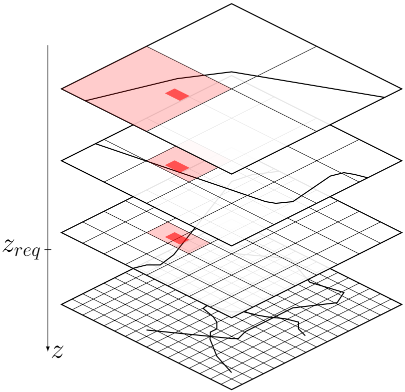
Partial Trajectories



Definition

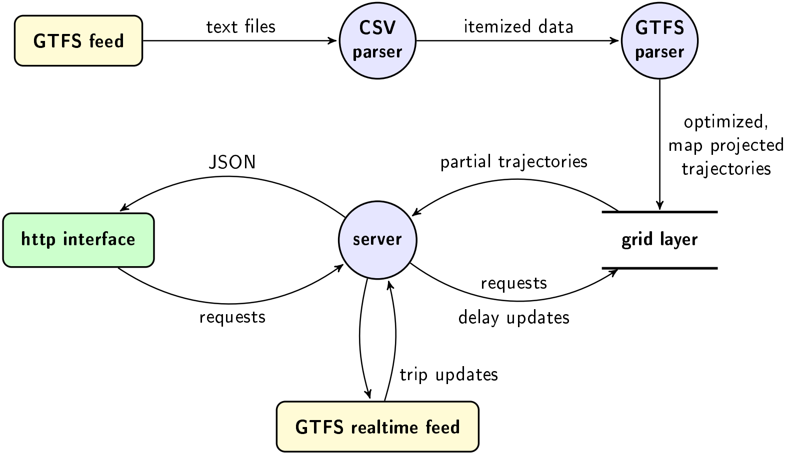
Given a trajectory $\mathcal{T} = (\mathcal{P}, \alpha)$, we call $\mathcal{T}^{par} = (\mathcal{P}^{par}, p_b, p_e, \alpha)$ a partial trajectory of $\mathcal{T}$. $\mathcal{P}^{par} \subseteq \mathcal{P}$.

Multilayer Grid



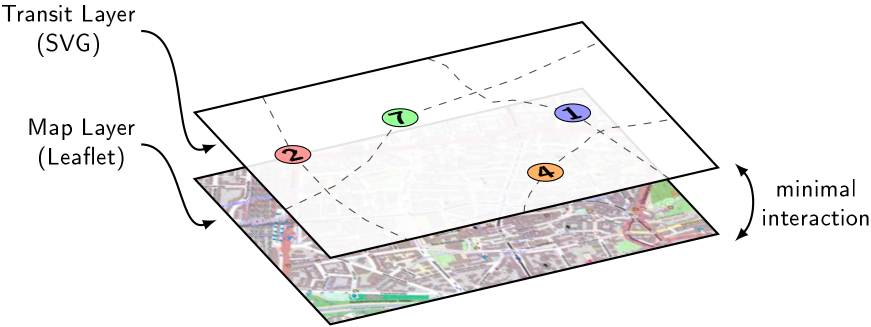
- Each cell has exactly 4 child cells
- Resembles a tree
- One layer per zoom level
- Trajectories are indexed spatially and temporally according to their activity function

General Workflow



TRAVIC

- a JavaScript webclient for TrajServ
- circumvents map API's JS projection functions
- draws directly to HTML5 vector layer (SVG)
- interpolates (visible) partial trajectories



Testing results (Netherlands)

	Naïve		Grid	
	8am	11pm	8am	11pm
Total area scan				
t (ms)	2.1 k	1.9 k	706	310
$\#T^{\text{par}}$	11.5 k	5.1 k	11.5 k	5.1 k
$\#T^{\text{aff}}$	548 k	548 k	18.1 k	8.2 k
req area (km^2)	25 M	25 M	25 M	25 M
Box request				
t (ms)	1.82 k	1.82 k	**51**	**33**
$\#T^{\text{par}}$	911	556	911	556
$\#T^{\text{aff}}$	548 k	548 k	2 k	1.2 k
req area (km^2)	10.7	10.7	10.7	10.7
Box req. $z = 9$				
t (ms)	1.28 k	1.28 k	**23**	**14**
$\#T^{\text{par}}$	707	451	707	451
$\#T^{\text{aff}}$	548 k	548 k	986	533
req area (km^2)	60 k	60 k	60 k	60 k

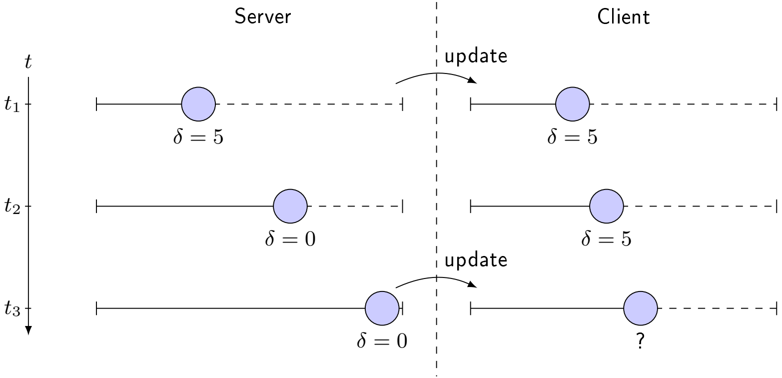
Testing results (20 feeds including NL)

	Naïve		Grid	
	8am	11pm	8am	11pm
Total area scan				
t (ms)	8 k	9 k	1.3 k	1.7 k
$\#T^{\text{par}}$	40.5 k	40.3 k	40.5 k	40.3 k
$\#T^{\text{aff}}$	2,5 M	2,5 M	67.7 k	67.1 k
req area (km^2)	~100 M	~100 M	~100 M	~100 M
Box request				
t (ms)	7.5 k	8.2 k	**51**	**33**
$\#T^{\text{par}}$	911	556	911	556
$\#T^{\text{aff}}$	2,5 M	2,5 M	2 k	1.2 k
req area (km^2)	10.7	10.7	10.7	10.7
Box req. $z = 9$				
t (ms)	5.9	5.9	**23**	**15**
$\#T^{\text{par}}$	707	451	707	451
$\#T^{\text{aff}}$	2,5 M	2,5 M	986	533
req area (km^2)	60 k	60 k	60 k	60 k

TRAVIC performance (New York City)

			Chrome		Firefox		IE	
z	$1/f$	$\#v$	t_r	t_{tot}/s	t_r	t_{tot}/s	t_r	t_{tot}/s
19	60	4	2.6	43.5	3.5	57.8	9.9	165.3
18	85	14	6.4	75.2	8.3	98.1	28.8	332.7
17	110	47	8.7	79.1	10.4	94.1	34.1	310.1
16	120	111	17.3	144.3	24.3	202.5	135.9	1.1 k
15	140	233	30.4	217.4	45.6	325.5	458.8	3.3 k
14	170	745	32.3	189.7	55.5	326.2	172.1	1 k
13	250	431	23.1	92.4	34.9	139.5	142	568.1
12	400	599	33.3	83.3	53.3	133.3	200.4	500.9
11	500	674	21.5	43	34.1	68.2	55	110
10-5	1 k	210	10.8	10.8	13	13	25.2	25.2

Asynchronous Delay Information



Summary

- **Goal:** scaleable live-map that is able to visualize public transit of whole world
- has been achieved with TrajServ and TRAVIC
- data input format: GTFS and GTFS realtime
- combined approach (interpolation of static schedule data + delay)
- Current index structure for trajectories: grid layer
- TRAVIC based on HTML5 vector layer, TrajServ outputs ready-to-go pixel coordinates

Future Work

- reduce memory usage
- solid solution to problem of asynchronous delay information
- **extraction of vehicle routes from geospatial data (map matching vs. iterative shortest-path)**
- combination with route planner

Any questions?