

Newest Developments

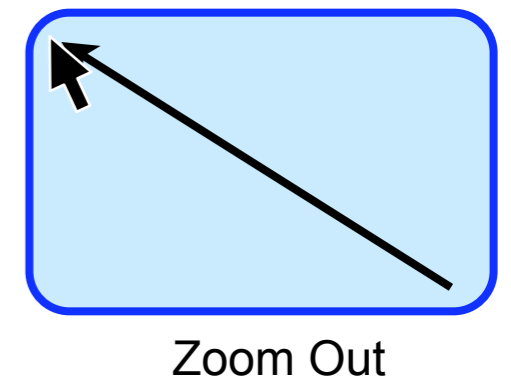
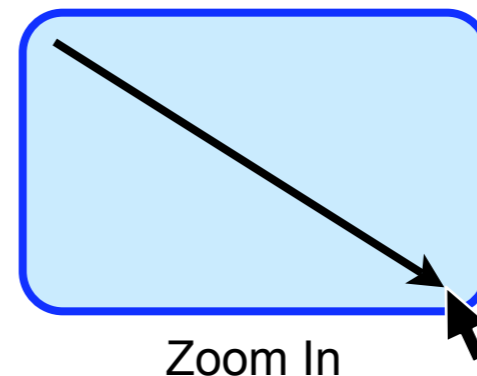
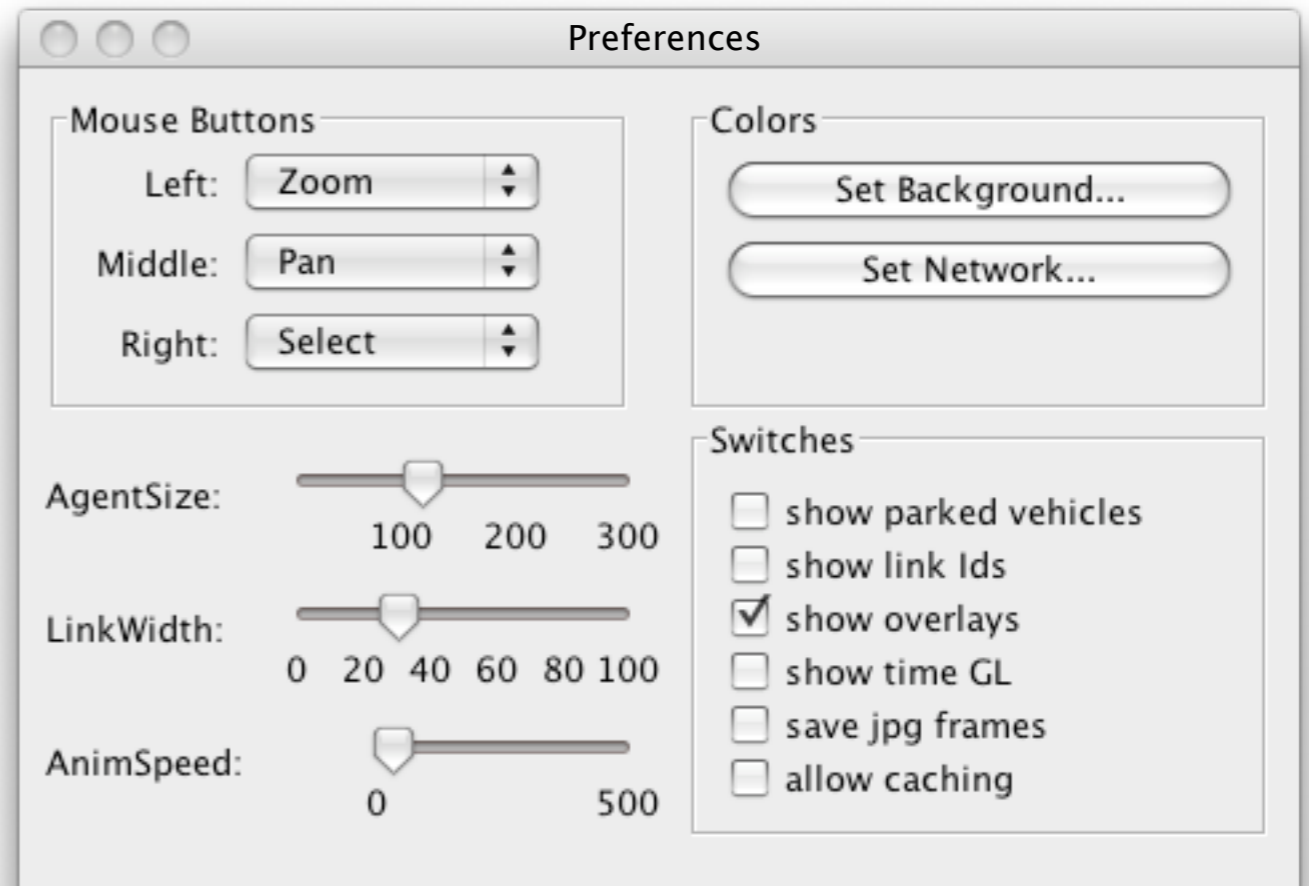
Overview

- New Features, already available
- Recent Performance Improvements
- New Features, Currently in Development
- New Scenarios
- Growth-Problems
- Code / Package Structure

New Features already available

Recent Performance Improvements

- Visualize Networks only
- Change link-width
- Start multiple instances to compare networks / scenarios
- Zoom and Preferences can be saved to mvi file
- Multiple Zoom-Settings can be stored per mvi file
- Can disable cache for visualization of even larger scenarios (or for older/weaker graphic cards)
- Zoom out by spanning zoom-rectangle from bottom to top
- Start with Open File-Dialog



New Features already available

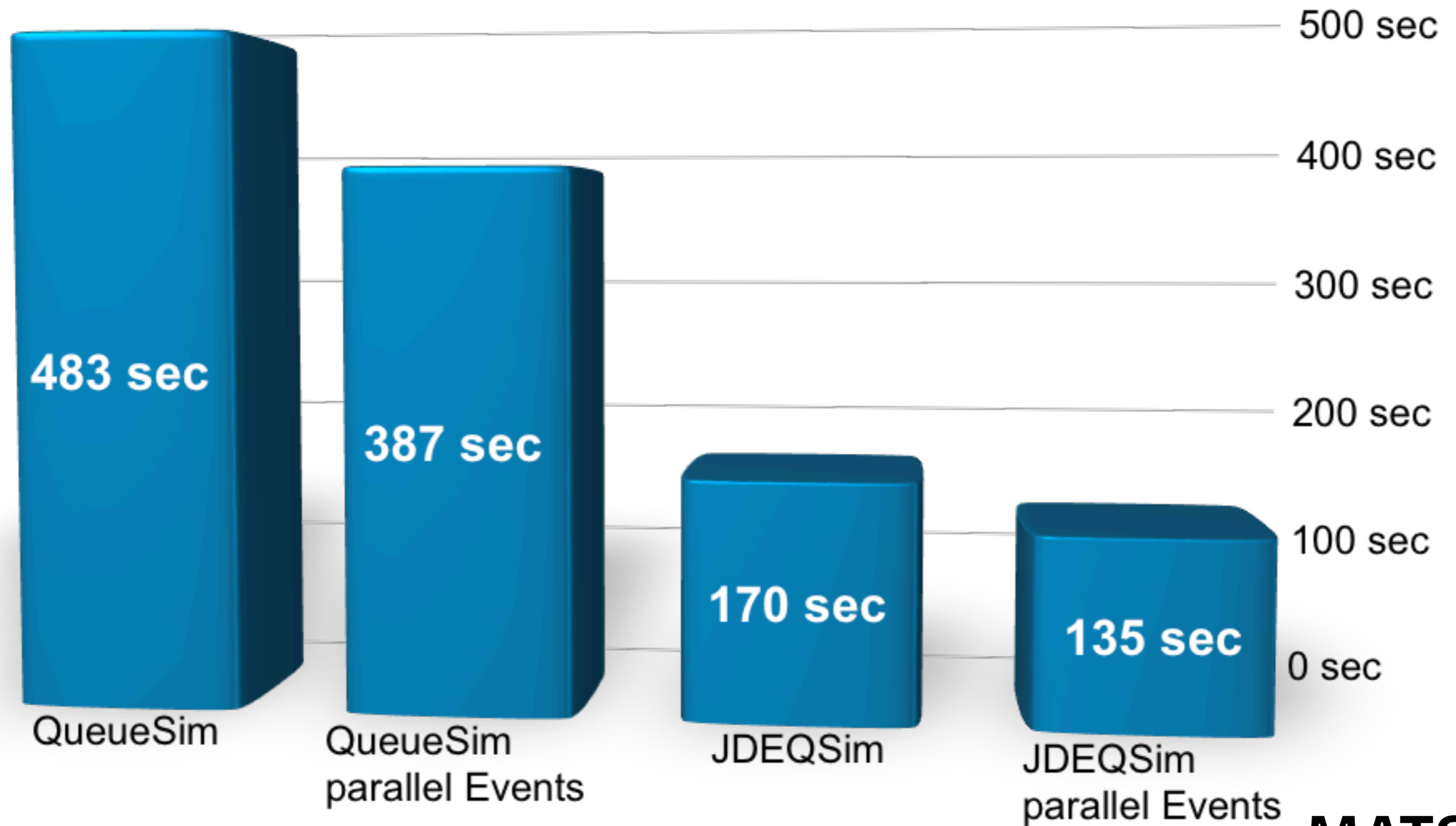
- Secondary Location Choice
 - Implemented by Andreas Horni, IVT, ETH Zurich
- Primary Location Choice
 - Implemented by Fabrice Marchal, LET Lyon
- Planomat (Combined Time-, Route-, Mode-Replanning)
 - Implemented by Konrad Meister, IVT, ETH Zurich
- Traffic Lights / Signal Systems
 - Implemented by Andreas Neumann & Dominik Grether, VSP, TU Berlin
 - fixed signal plans, turning restrictions
- Use network data from OpenStreetMap
 - Implemented by Marcel Rieser & Andreas Neumann, VSP, TU Berlin
 - Used in Tutorials and for new Berlin Scenario
 - Generate road network with different level of detail in different regions

Recent Performance Improvements

- Parallel Events Handling
 - Implemented by Rashid Waraich, IVT, ETH Zurich
 - Make use of multi-core processor / systems to reduce time needed to run simulations
- JDEQSim (Implementation of DEQSim in Java)
 - Implemented by Rashid Waraich, IVT, ETH Zurich
 - Fast, (yet) single-threaded traffic simulation
 - slightly different simulation characteristics, requires “re-calibration” of scenarios
 - does not support all features of QueueSim
- Compressing Route Information
 - Implemented by Marcel Rieser, VSP, TU Berlin
 - Route Information uses huge amounts of memory with detailed networks
 - Compressing this information makes simulation a bit slower, but enables running larger scenarios

Recent Performance Improvements

average time for 1 iteration of 10% Zurich-Scenario
(187 484 agents, 24 180 nodes, 60 492 links, ~9mio events)



New Features Currently in Development

Disclaimer: This presentation includes information about features currently in development. No guarantee can be given about when (or if at all) a feature will be an official part of MATSim.

“Real” Transit / Public Transportation

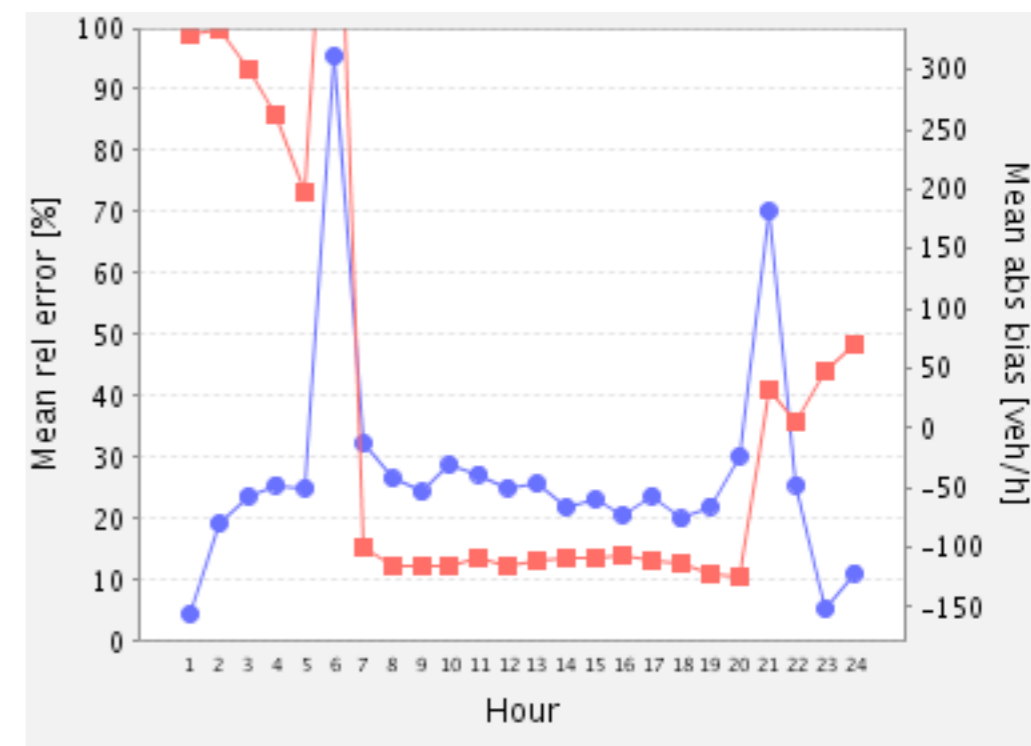
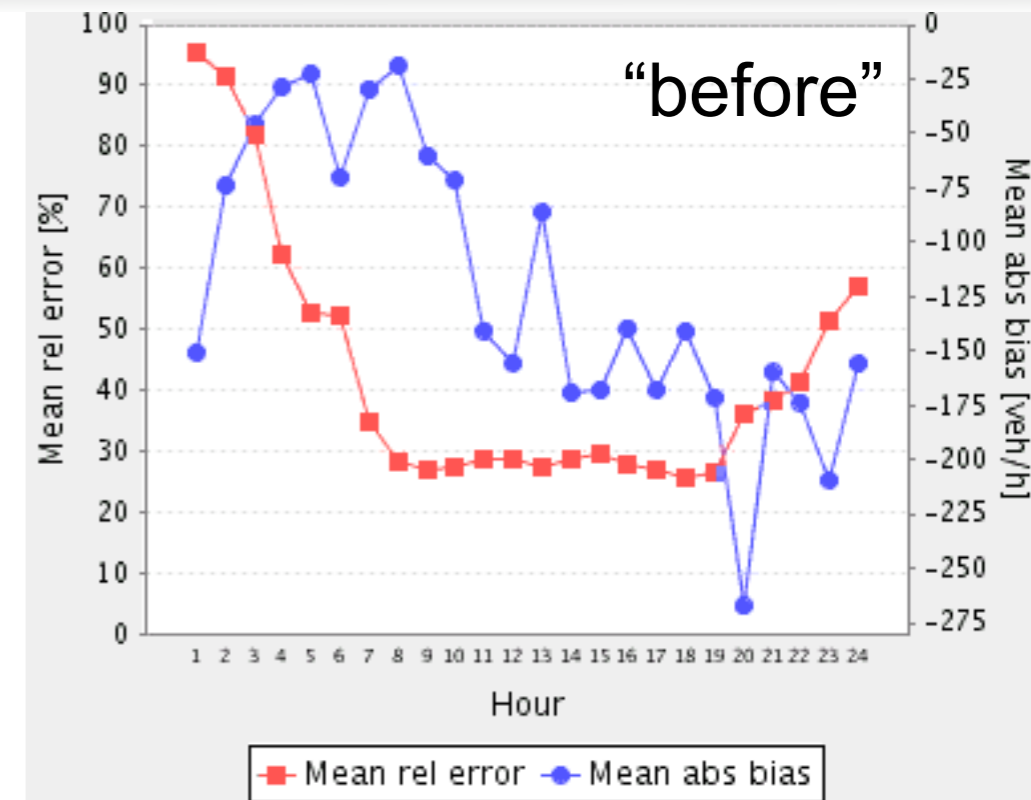
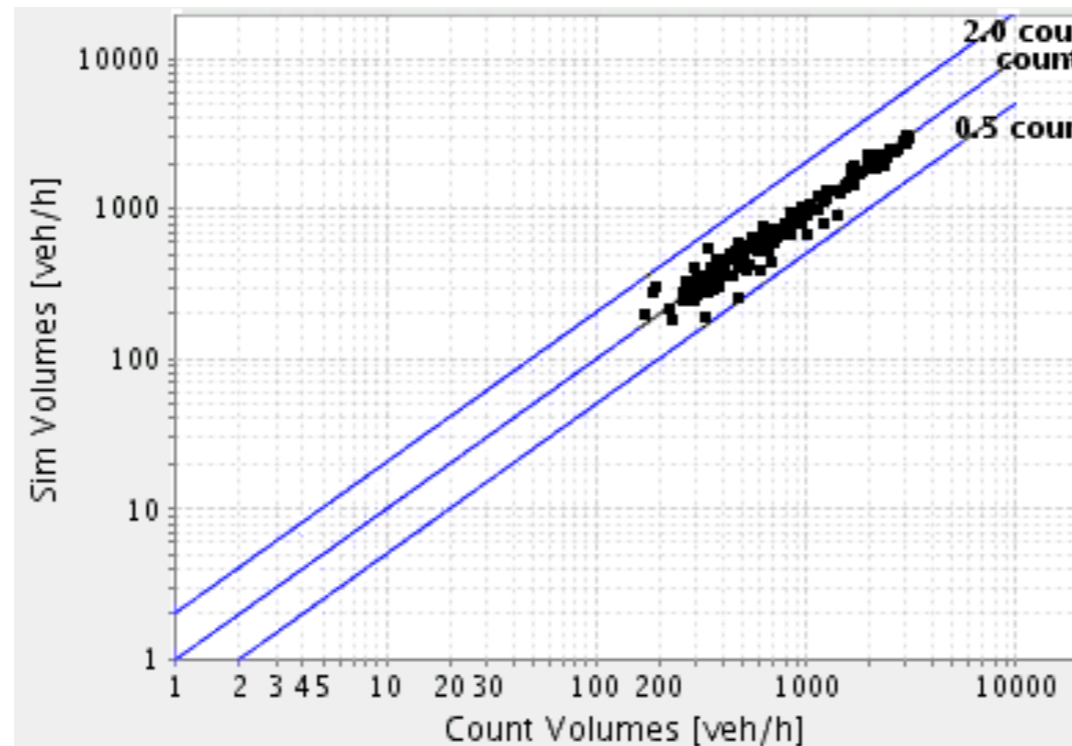
- Support time-table information for better route-finding
- Simulate transit vehicles according to time-table
- Agents can enter/leave transit vehicles
- Statistics on transit vehicle occupancy, delays, ...
- Advanced route-finding:
changing lines vs. longer travel time, comfort, on time, ...
- Vehicle-Types

- Work by Manuel Moyo and Marcel Rieser, VSP, TU Berlin

Counts Calibration / Parameter Estimation

- Work based on Gunnar Flötteröd's PhD thesis (VSP, TU Berlin, 2008)
- Implemented by Gunnar Flötteröd (EPFL) & Yu Chen (VSP)
- “During Plan selection, prefer those plans that lead to better counts comparisons”
- Estimate parameters based on plan selection

with calibration
7:00 – 20:00



The Future / Experimental Stuff

- Calculating Emissions from MATSim Output
Project ready, work will start this spring/summer
- Traffic Simulation on GPUs (Graphic Cards)
Planned PhD thesis by David Strippgen, VSP, TU-Berlin
GPUs offer huge amount of parallel computation power, but may be rather slow exchanging big amounts of data.
Rather conceptual/experimental work, will not be ready for a while.
- Households in MATSim
- Energy & Transport (Electro-Mobiles, Vehicle to Grid)
Project at ETHZ, already started
- Integrated Landuse and Transport Systems
(planned, KTI, MATSim meets UrbanSim)
- lean, always-running MATSim (@ AxonActive)

Scenarios

ETH

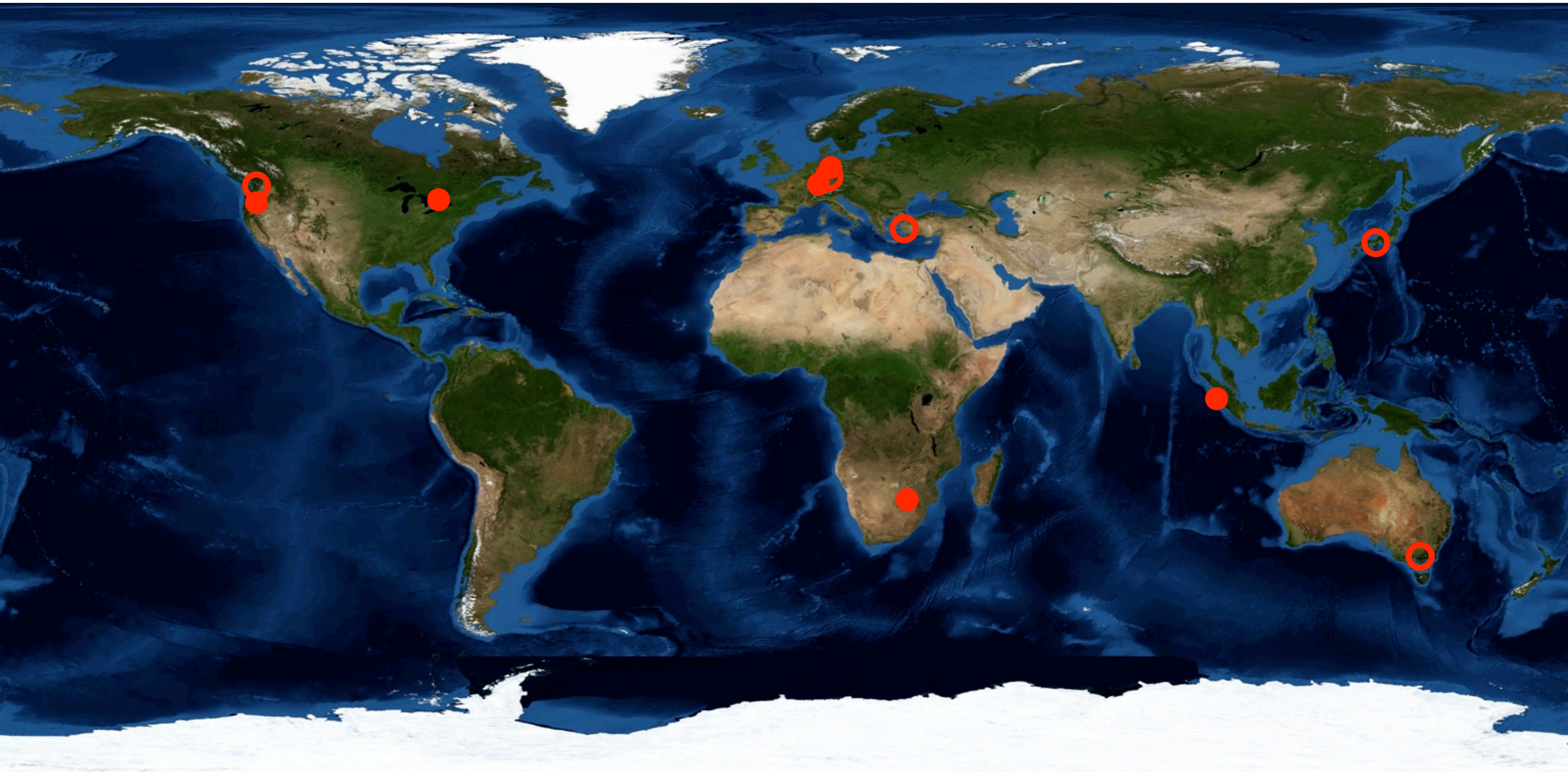
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Technische Universität Berlin



MATSim
Multi-Agent Transport Simulation

MATSim Scenarios



● Scenario up and running

○ Scenario currently built / in planning / shown interest

Existing Scenarios

- Berlin, Germany (v1)
 - Never really successful
 - a lot of required data missing
 - inactive for some years
- Zurich / Switzerland
 - Westumfahrung Zürich – First real-world case study
 - Simulation of 100% complete Switzerland planned for this summer
 - High-resolution network: ~1.3mio links, ~800 000 nodes
- Padang, Indonesia
 - Tsunami evacuation simulations
 - Network generated from remote sensing data

Existing Scenarios

- Gauteng, South Africa
 - smallest province, but contains Johannesburg & Pretoria, ~9m inhabitants
 - future goal: simulate paratransit-like public transport offerings
- Toronto, Canada
 - combine with TASHA
 - comparison with EMME/2 model
 - network converted from EMME/2
- Seattle, USA
 - combine with OPUS / UrbanSim
 - still at an experimental level, but working
 - network converted from EMME/2
 - used to test MATSim for “sketch planning”: small population samples, fast turnaround, not much calibration/validation

New Scenarios / New Users

- Kyoto, Japan
 - Prof. Hiromitsu Hattori, Kyoto University, Dept. of Social Informatics
 - Simulation of Driver Behavior
 - Using MATSim for Framework around their own traffic simulation
- Munich, Germany
 - Prof. Regine Gerike, TU München, Lehrstuhl für Verkehrstechnik
 - Calculating emissions from MATSim output
 - (we're hiring! One open position in Berlin, one in Munich)
- Berlin, Germany (v2)
 - Prof. Kai Nagel, TU Berlin, VSP
 - possibly future collaboration with BVG (Berlin Transit Company)
 - likely to use network from OpenStreetMap

Structure / Organisation of MATSim

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Technische Universität Berlin



MATSim
Multi-Agent Transport Simulation

New Scenarios / New Users: Implications

Up to now, nearly all users got write-access to SVN repository

- mostly in playgrounds: more than 2/3rd of code in MATSim is in playgrounds
- some “required” 3rd-party libraries only used in playgrounds
- hard to manage
- responsibilities / permissions unclear
- code quality sometimes poor
- “bad code” still got copied/pasted
- but people need commit-rights to repository because MATSim-code changes too quickly

Problems

- unclear which packages were stable, which experimental, which deprecated
 - people used classes from wherever it suited them
 - many unnecessary dependencies
 - no modularity, everything connects to / depends on everything
 - structural changes had often big impacts on others
 - many packages that were “common property”
 - e.g. population, network, controller: Many people just added their methods they thought they couldn't live without
 - many packages that were “nobody's property”
 - e.g. matrices, visum: who has ever used them?
-
- ▶ Problems get worse when even more people can commit
 - ▶ Too many support requests because of too many code dependencies
 - ▶ “core infrastructure” could not be maintained longer in this form

Code / Package Structure

Goal: Build a new structure such that:

- people can write code for MATSim that remains functional for some while
 - people do not have to commit code to repository to keep their code working
 - structural changes in core infrastructure affects as few people as possible
-
- ▶ A clear API must be made available
 - set of classes/interfaces that remain stable
 - ▶ Refactorings performed in March 2009

Code / Package Structure

src/	contains all sources
org/matsim/	the “official” part of MATSim contains code seen as stable
api/	the official interfaces for MATSim. Users writing code for MATSim should try to only use classes/interfaces from org.matsim.api.*
core/	implementations and infrastructure for running MATSim
run/	starting-point for running MATSim from outside
moduleA/ moduleB/ ...	contributions / extensions for MATSim e.g. Location Choice, Planomat, JDEQSim, OTFVis, ...
playground/	the “inofficial” part of MATSim experimental code with tight integration to MATSim (code for which the current API is not sufficient)

Code / Package Structure: Responsibilities

src/	contains all sources
org/matsim/	each package should have a file package-info.java with Javadoc mentioning package maintainer(s)
api/	Package-Maintainers: VSP
core/	Package-Maintainers: VSP
run/	Package-Maintainers: VSP
moduleA/ moduleB/ ...	Package-Maintainers: see package-info.java some packages (like <code>utils</code>) may not specify a maintainer, but delegate the responsibility to the single classes in the package
playground/	Package-Maintainers: package-owners (should be clear from name)

Code / Package Structure: Guidelines

- Classes / Interfaces in `org.matsim.api.*` are stable
 - Try to use only those MATSim classes / interfaces
 - Your code will *not* be influenced by future refactorings
 - Do not changes these classes / interfaces!
 - Speak to core-maintainers if you think you need an extension / modification
- Classes / Interfaces outside `org.matsim.api.*` are unstable
 - Your code *may be* influenced by future refactorings
 - Do not commit changes to other maintainers' packages, speak to them first
- Write a simple `package-info.java` for your packages in `org.matsim`
 - Take responsibility for your code

Four levels of involvement

- “outside”
 - just using stable API
- “playground”
 - using (unstable) core-api
- “org.matsim”
 - maintaining an official module
- “core”
 - contributing to code in org.matsim.core

Code / Package Structure: Future

- The latest refactorings were just a first step
- Current API will be extended
- New concepts may be added for offering higher modularity (e.g. not every class must be hardcoded in the Controller to be used from the Controller; kind of an extension to the current ControllerListeners)
- Higher flexibility for non-developers, and for “just running simulations”
- no time-frame to complete these features

How to Help / Contribute

- Try to use `org.matsim.api`
 - Talk to us if you don't succeed
- Enter bugs / problems in the issue tracker
 - <http://matsim.org/issuetracker>
- Developers: Have a look at the developer-pages
 - <http://matsim.org/developer>
 - <http://matsim.org/docs/devguide>
- Stay Informed: Join the mailing list
 - <https://lists.sourceforge.net/lists/listinfo/matsim-devel>

Thanks for your attention!

Questions? Comments?

More information about MATSim: <http://matsim.org/>

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Technische Universität Berlin



MATSim
Multi-Agent Transport Simulation