NEXUS
Collaborative Research Center 627

Efficient Real-Time Tracking of Moving Objects' Trajectories

Ralph Lange, Frank Dürr, Kurt Rothermel

Institute of Parallel and Distributed Systems (IPVS)
Universität Stuttgart, Germany

mail@lange-ralph.de

Motivation

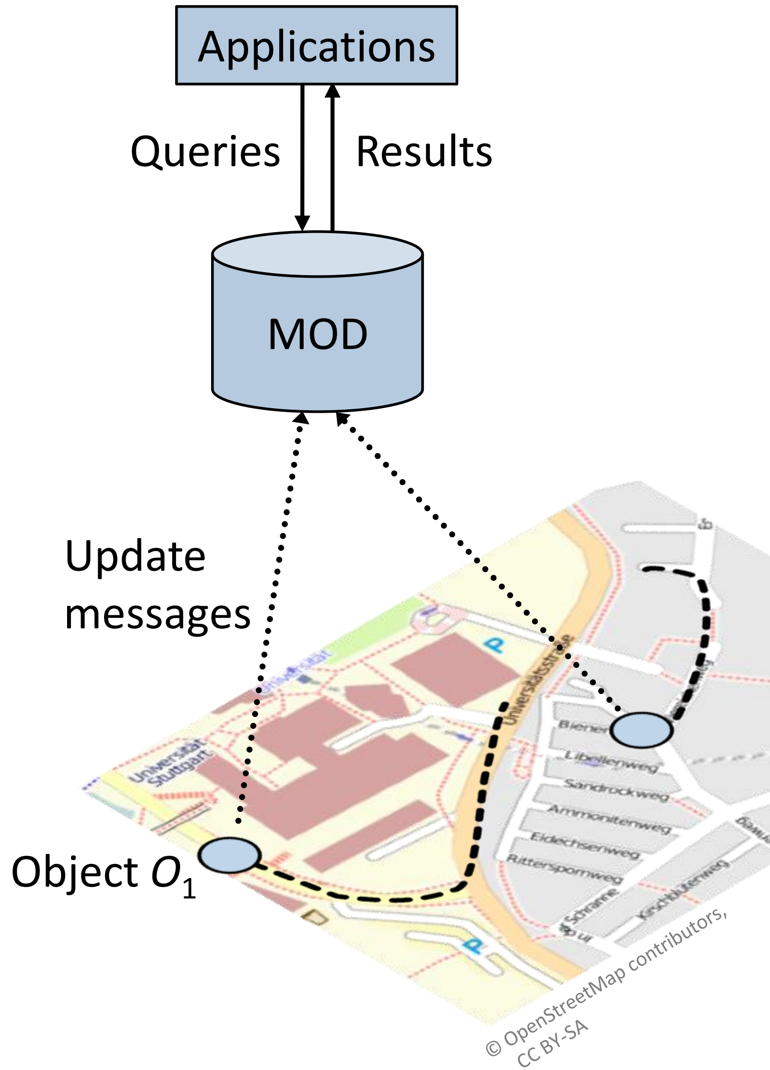
Importance of **position information** of moving objects for many applications

- Logistics, sports, wildlife monitoring, …

Variety of requirements

- Position tracking in real-time
- Queries about large numbers of objects
- Queries on past positions

➔ **Moving objects databases** (MODs) for real-time management of **trajectories**



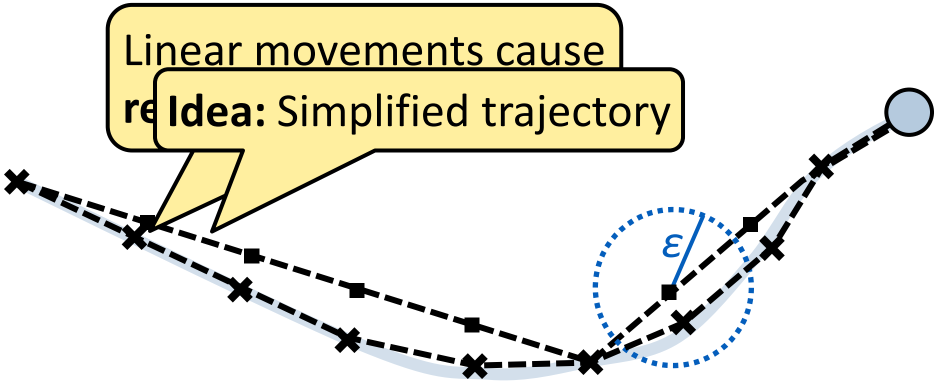
Motivation (2)

Problem: Large amounts of trajectory data

- GPS receiver generates **$3 \cdot 10^7$ records** per year
- High communication cost
- Consume a lot of storage capacity

1. How to **scale** MODs to large numbers of servers **?**

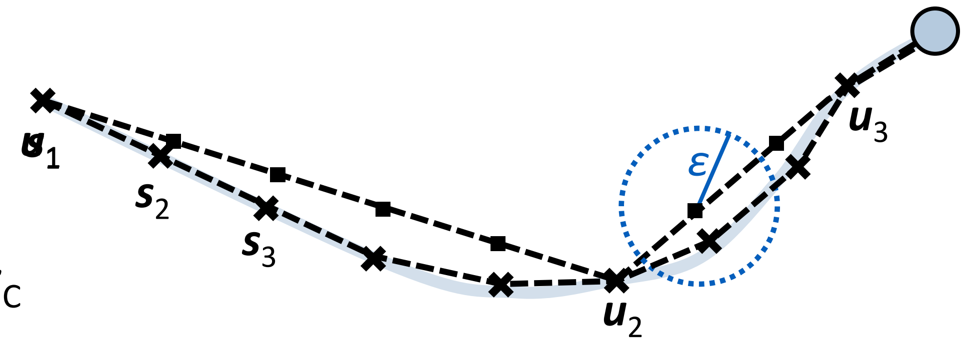
2. How to **track** trajectories efficiently in real-time **?**



Overview

- Formal problem statement
- Related work
- Connection-Preserving Dead Reckoning (CDR)
- Generic Remote Trajectory Simplification (GRTS)
- Evaluation
- Summary

Formal Problem Statement



Kinds of trajectories

- Actual: $\boldsymbol{a}(t)$ is function $\mathbb{R} \to \mathbb{R}^d$
- Sensed: $\boldsymbol{s}(t)$ with vertices $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_C$
  - Attribute $\boldsymbol{s}_i.\boldsymbol{p}$ denotes position at time $\boldsymbol{s}_i.t$
  - Differs from $\boldsymbol{a}(t)$ due to δ_{sense} and movement during T_{sense}
- Simplified: $\boldsymbol{u}(t)$ with vertices $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$

Definition: **Efficient real-time trajectory tracking**

- Goals: Minimize $|(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots)|$ and communication cost
- Simplification constraint: $|\ \boldsymbol{u}(t) - \boldsymbol{a}(t)\ | \leq \varepsilon \ \ \forall\ t$
- Real-time constraint: $\boldsymbol{u}(t)$ is available $\ \forall\ t \in [\boldsymbol{s}_1.t, t_C]$

Related Work

Line simplification algorithms

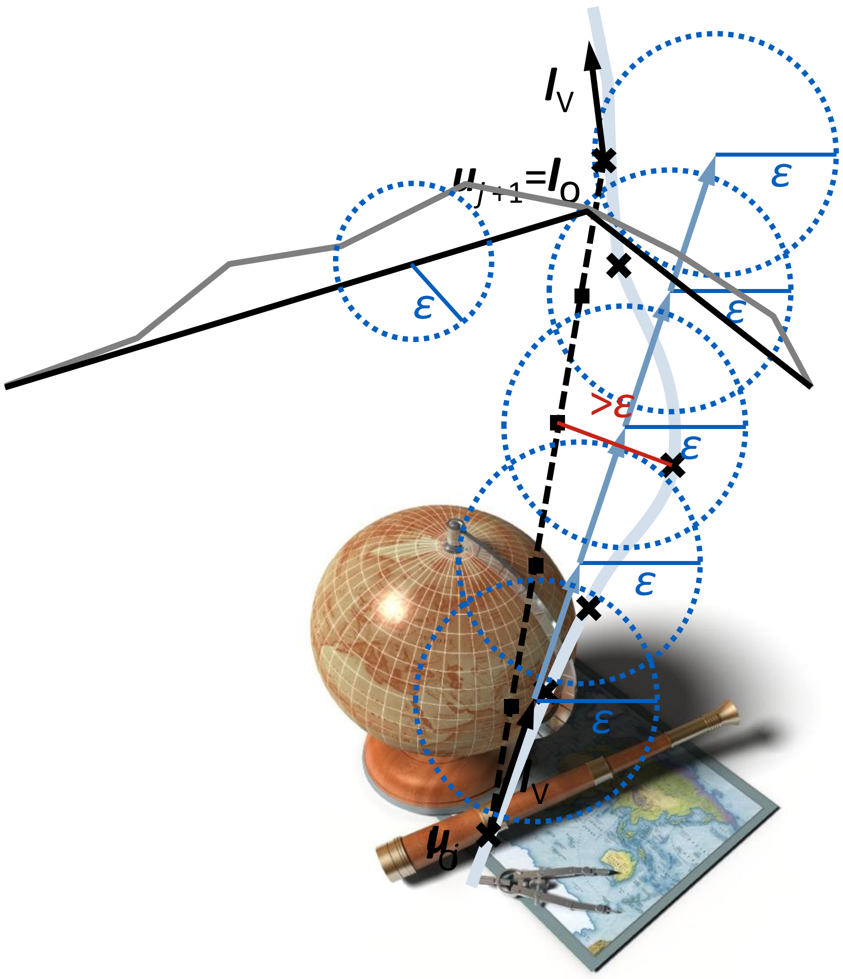
- E.g. Douglas-Peucker heuristic

➔ High communication cost,
no real-time behavior

Position tracking protocols

- Best mechanism: Dead Reckoning (DR)

➔ Linear DR with ½ε allows for trajectory
tracking [Trajcevski et al. 2006]



Connection-Preserving DR

CDR extends linear DR

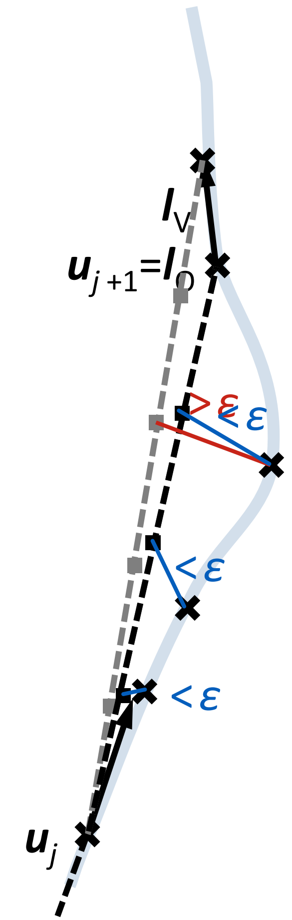
- Object manages $\boldsymbol{l}_O$, $\boldsymbol{l}_V$, and **sensing history** $\mathbb{S}$
- Second update condition:

  $$\exists\ s_i \in \mathbb{S} \text{ with } |\overline{\boldsymbol{l}_O\boldsymbol{s}_C}(\boldsymbol{s}_i.t) - \boldsymbol{s}_i.\boldsymbol{p}| > \varepsilon$$

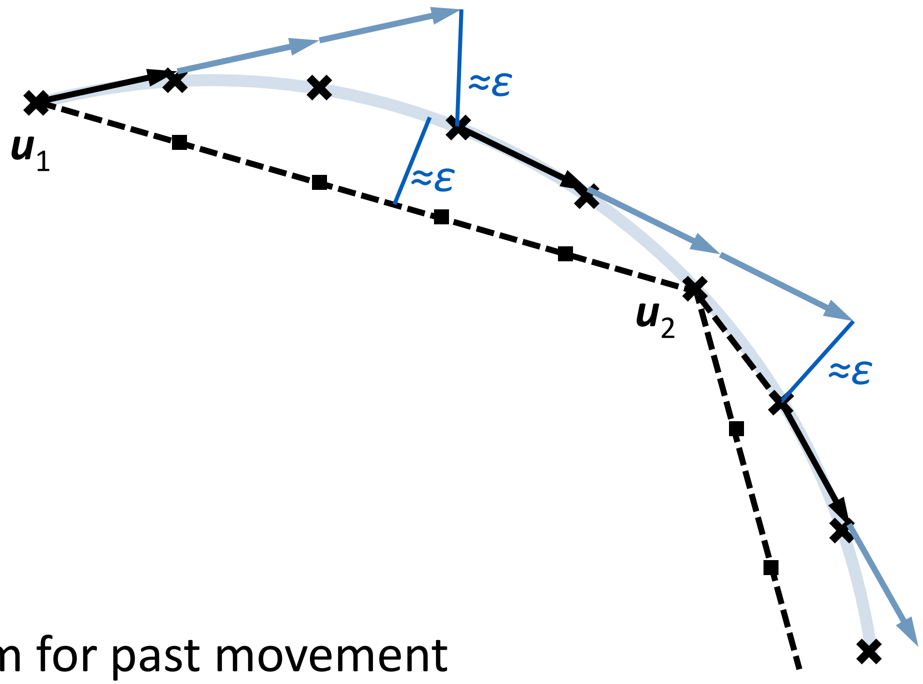
- New prediction starts at **last** sensed position

CDR$_m$ limits computational cost by $|\mathbb{S}| \le m$

- Compression approach to prevent periodic updates



GRTS – Generic Remote Trajectory Simplification



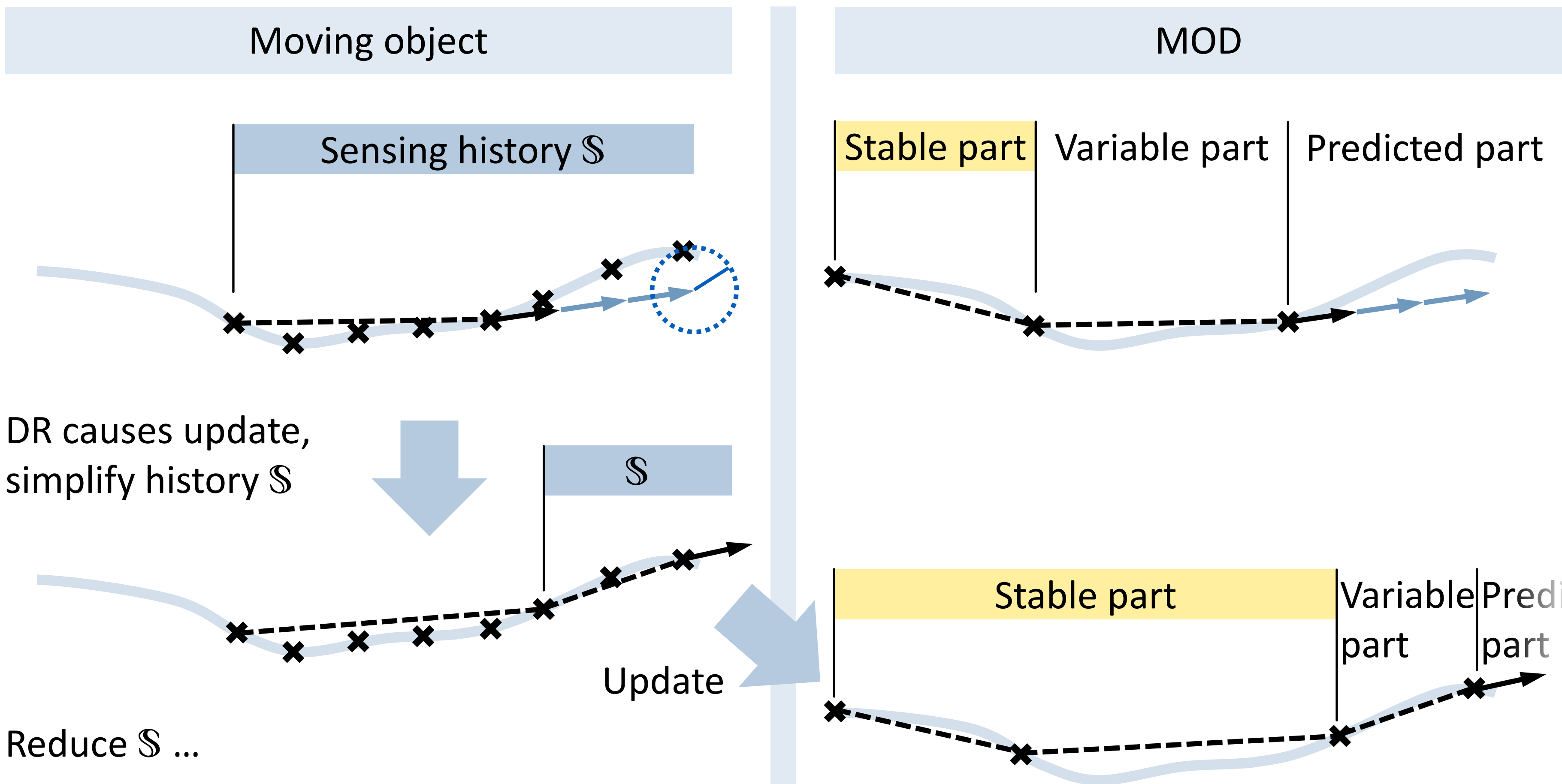
Tracking and simplification are **different concerns**

Basic approach of **GRTS**

- DR to report latest movement
- Arbitrary line simplification algorithm for past movement
  - Computational cost ↔ reduction efficiency

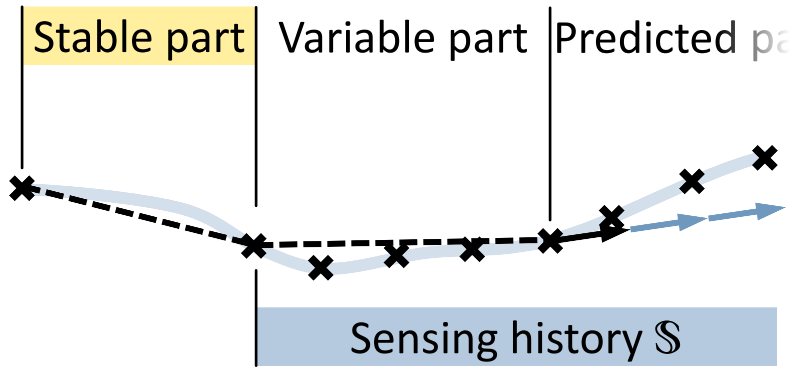
➔ Tracking and simplification must be synchronized!

GRTS: Basic Protocol



GRTS: k/m Variants

So far, not defined how to reduce $\mathbb{S}$ …



GRTS_k limits variable part to *k* line sections

- Unlimited computational costs
- ➔ Only of theoretical interest

GRTS_m limits $|\mathbb{S}|$ to *m*

- Limits computational costs – essential for real-time behavior
- Adds vertex to $\boldsymbol{u}(t)$ at least every *m* sensing operations
- ➔ Compress $\mathbb{S}$ if its size exceeds *m*

GRTS: mc Variant

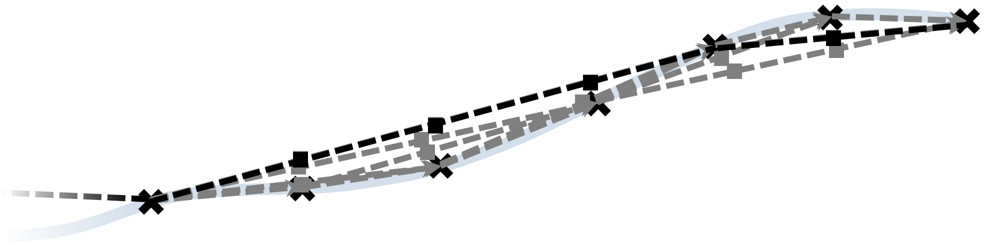
GRTS$_{mc}$

- $s_i.\delta$ gives **maximum deviation** along line section from s_{i-1} to s_i
- Number of compressed positions should be small (c = 1 or 2)
- With uncompressed positions, $s_i.\delta$ may represent varying δ_{sense}

$GRTS_*^{Sec}$ and $GRTS_*^{Opt}$

$GRTS_*^{Opt}$ – with optimal simplification algorithm [Imai and Iri 1988]

- Reduces simplification to shortest-path problem



- Segmentation by *k* or *m* influences reduction efficiency

$GRTS_*^{Sec}$ – with Section Heuristic [e.g. Meratnia and de By 2004]

- Online algorithm enables per-sense simplification
- Proposed improved version, optimizing $|\mathbb{S}|$

Evaluation: Setup

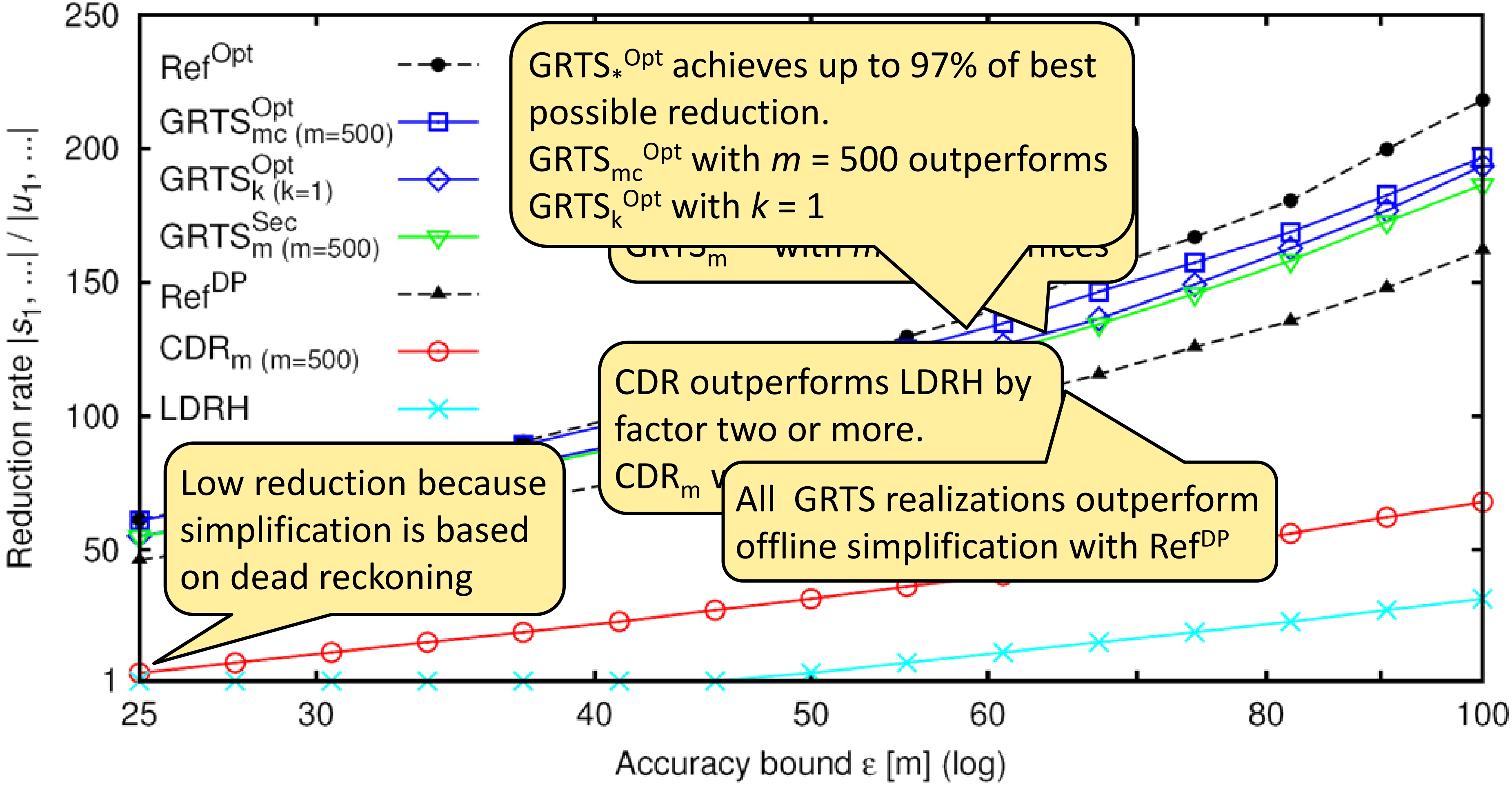
Comparing CDR variants and $GRTS_*$ realizations to …

- Linear DR with ½ε (LDRH)
- Optimal offline simplification (Ref^{Opt})
- Douglas-Peucker algorithm (Ref^{DP})

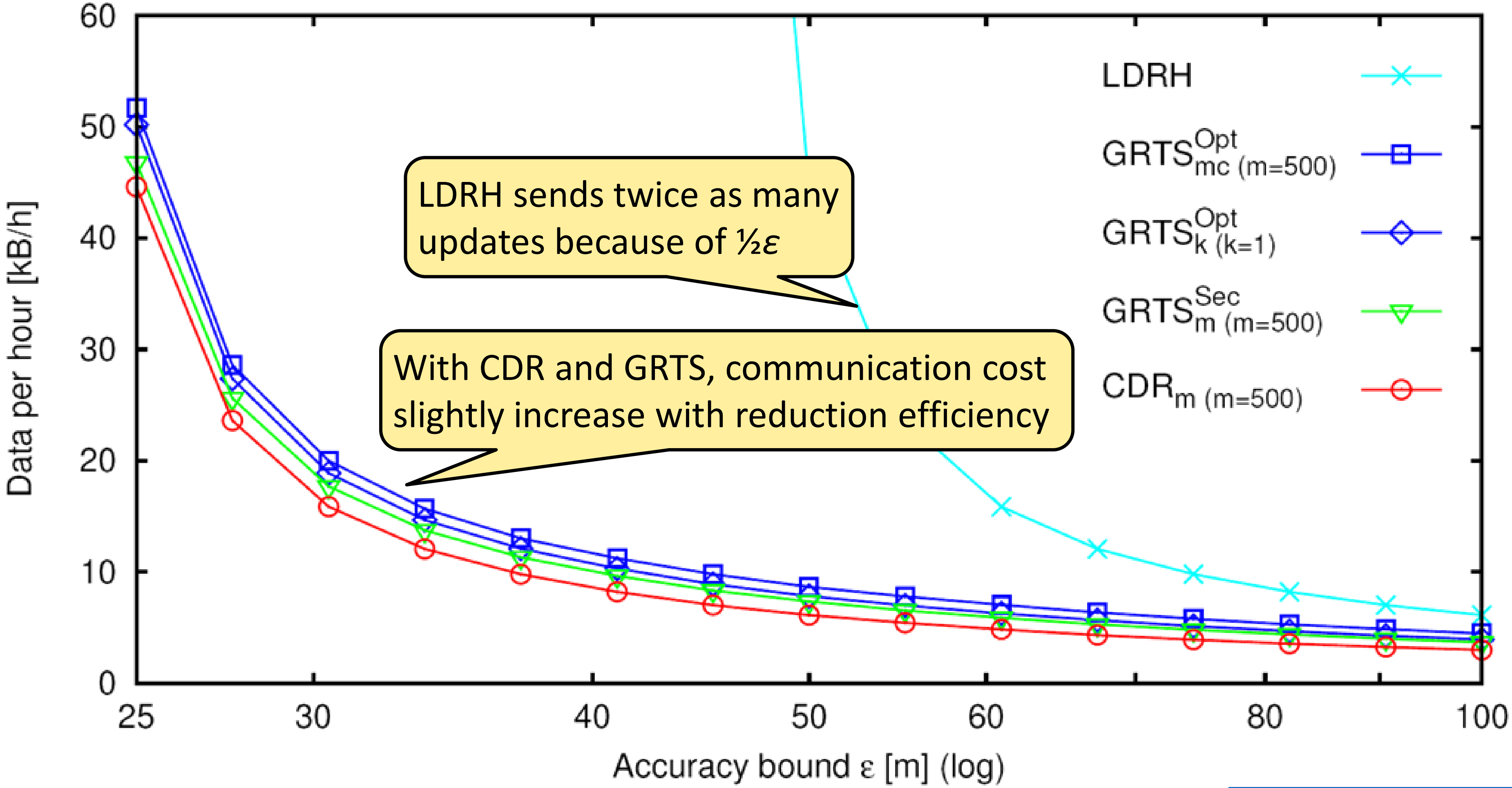
Simulated with **real** GPS traces from OpenStreetMap

- **> 1.2 million** sensed positions, i.e. **> 330 h** of data

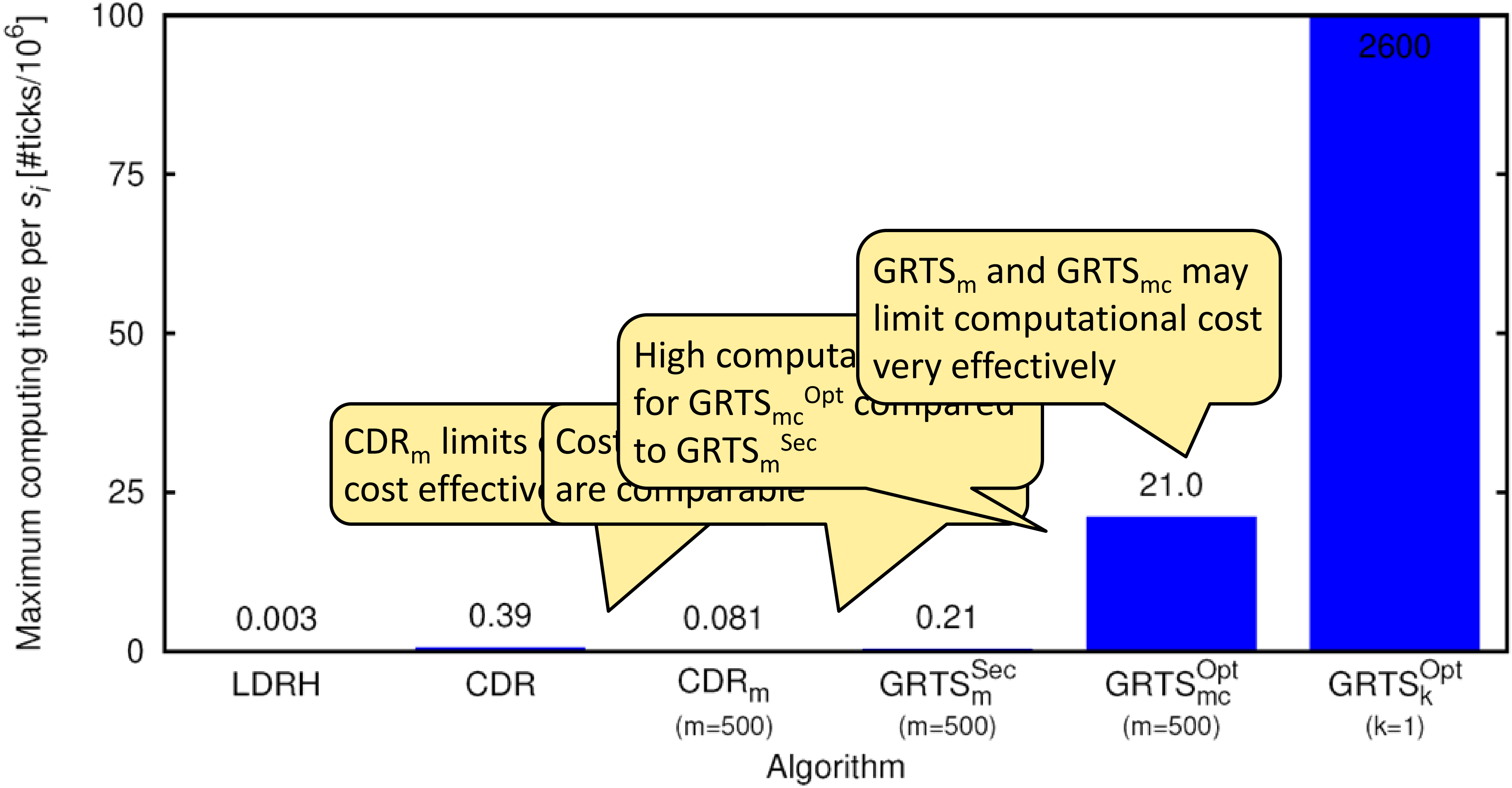
Evaluation: Reduction



Evaluation: Communication



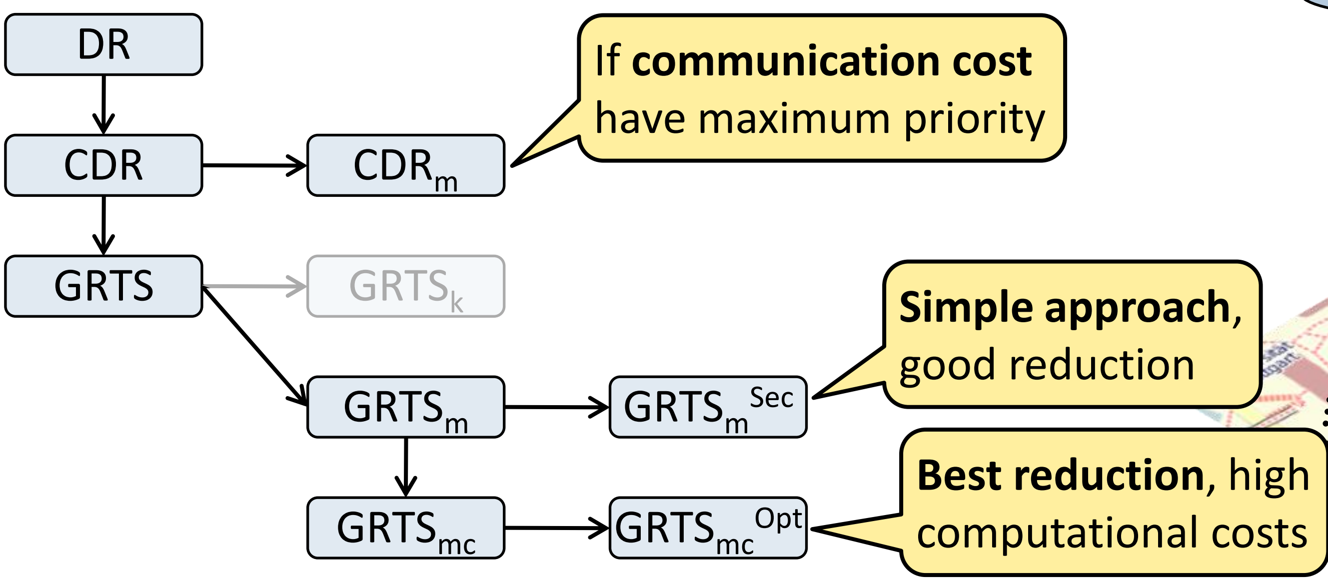
Evaluation: Computing Time

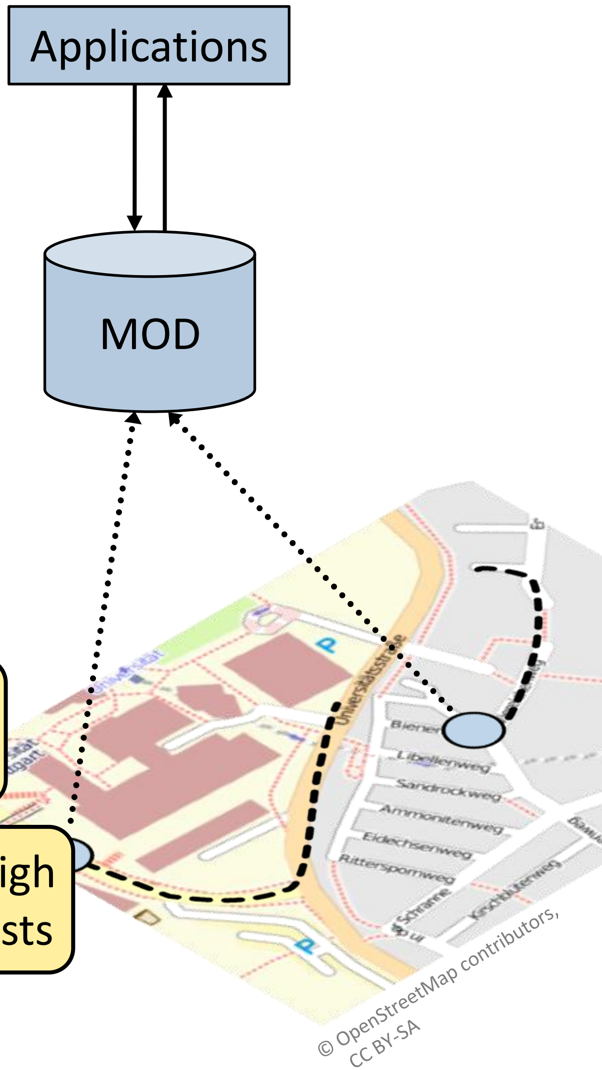


Summary

Efficient real-time trajectory tracking:

- Minimize communication cost and storage consumption under given accuracy ε





Useful for other sensor data – e.g. online flight recorder

Thank you

for your attention!

- Ralph Lange, Frank Dürr, Kurt Rothermel: "Online Trajectory Data Reduction using Connection-preserving Dead Reckoning". In: *Proc. of 5th MobiQuitous*. Dublin, Ireland. July 2008.
- Ralph Lange, Tobias Farrell, Frank Dürr, Kurt Rothermel: "Remote Real-Time Trajectory Simplification". In: *Proc. of 7th PerCom*. Galveston, TX, USA. March 2009.
- Ralph Lange, Frank Dürr, Kurt Rothermel: "Efficient Tracking of Moving Objects using Generic Remote Trajectory Simplification (Demo)". In: *Proc. of 8th PerCom Workshops*. Mannheim. March 2010.

➔ See www.lange-ralph.de