

# Quality-aware Real-Time Tracking of Moving Objects' Trajectories

Ralph Lange

PhD graduate (Dr. rer. nat.) of the Universität Stuttgart

Supervisor: Prof. Dr. Kurt Rothermel

Institute of Parallel and Distributed Systems (IPVS)

# Motivation

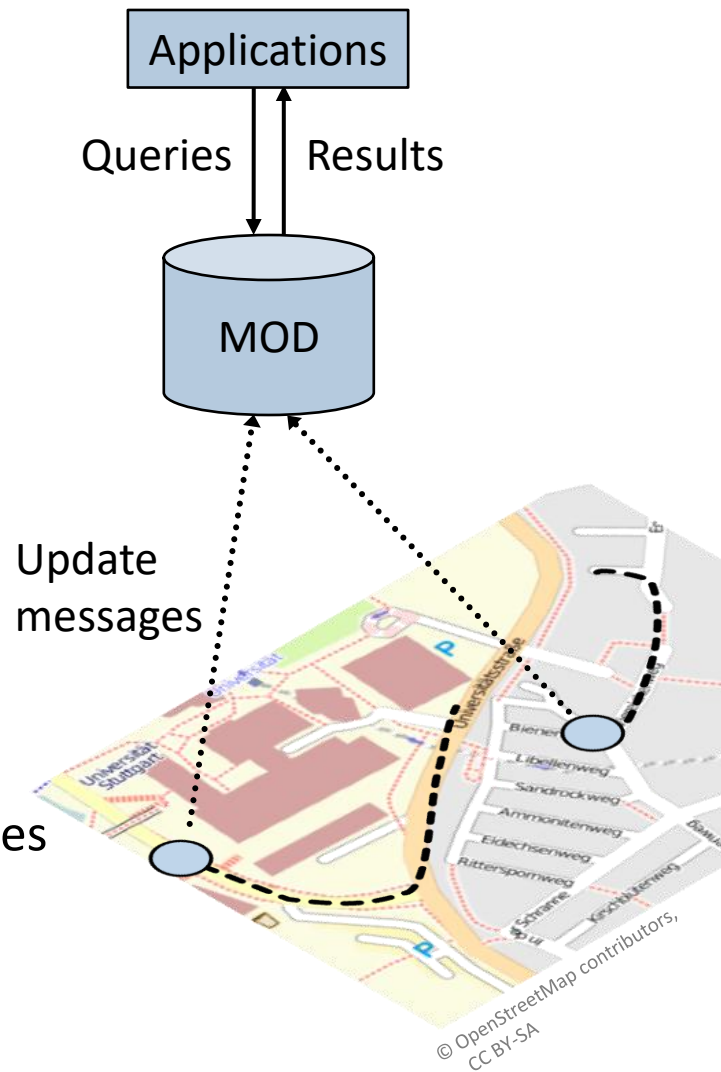
Importance of **position information** of moving objects for many applications

- Logistics, sports, wildlife monitoring, ...

Variety of requirements

- Position tracking in real-time
- Queries about large numbers of objects
- Queries on past positions
- Incorporation of different positioning technologies

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



# Motivation (2)

## Fundamental problems

- Uncertainty of position/trajectory information
  - Variety of sources: Physical limitations, interpolation, ...
  - Heterogeneity of mathematical models
- Large amounts of trajectory data
  - GPS receiver generates  **$3 \cdot 10^7$  records** per year
  - High communication and storage cost

1. How to provide a general  
**uncertainty-aware** query interface



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



# Overview

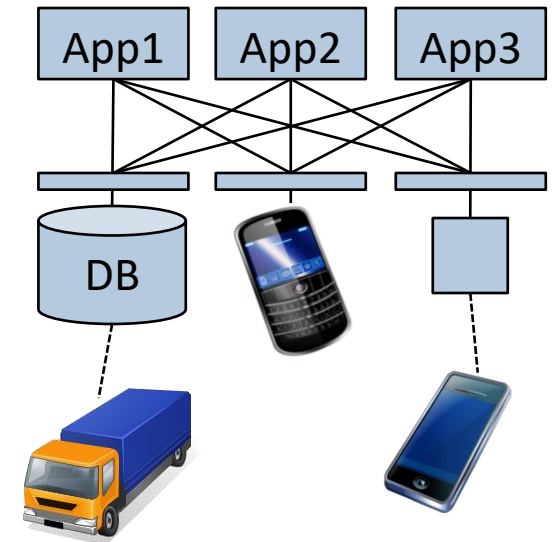
- Motivation
- Generic uncertainty model and query interface
  - Detailed problem statement
  - Survey and classification of existing models
  - Generic uncertainty model
  - Query interface
- Efficient real-time trajectory tracking
- Summary

# Problem Statement

Extended query interface based on a generic uncertainty model to enable source-**independent** ...

... but **uncertainty-aware** processing

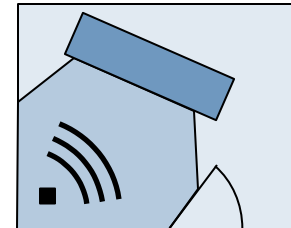
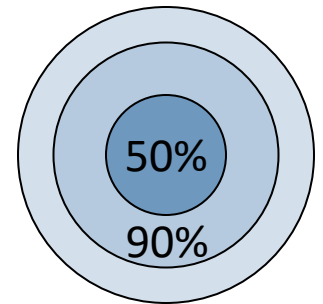
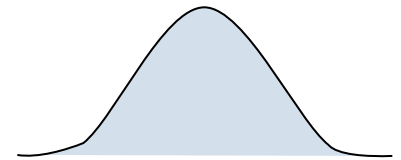
1. Requirements to generic uncertainty model
  - Generality, expressiveness, and directness
2. Requirements to extended query interface
  - Immediacy, comprehensiveness, and generality



# Survey of Existing Uncertainty Models

## Uncertainty models of positioning technologies

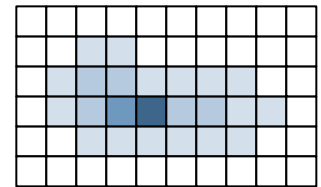
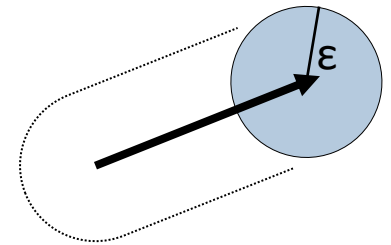
- Triangulation and -lateration (e.g., GPS, ultrasonic-based)
  - Complete **pdf**, mostly normal distribution
  - May be estimated by Dilution of Precision (DOP)
- Scene analysis (e.g., WiFi, camera-based)
  - Center point and one or more quantiles
- Proximity (e.g., IR-beacons, RFID)
  - Sensing range given as **geometric area** depending on setup



# Survey of Existing Uncertainty Models (2)

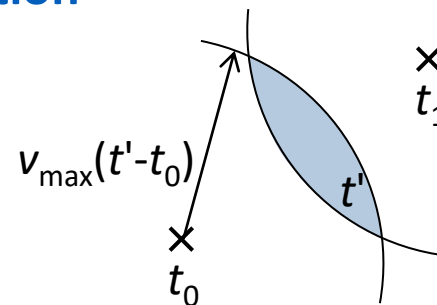
## Uncertainty models in update protocols

- Dead reckoning and remote trajectory tracking
  - Trade communication cost off against **accuracy**
  - Known and actual position do not deviate by more than accuracy bound  $\varepsilon$



## Uncertainty models for fusion and interpolation

- Bayes filters, possibly based on grid
- Fusion with multi-area probabilities
- Lense-based interpolation



# Classification of Specific Models

## Notation and assumptions

- $a_{i,t} \in \mathbb{R}^2$  denotes the **actual** position of a point-shaped object  $O_i$  at time  $t$
- $s_{i,t}$  denotes the **uncertain** position at time  $t$

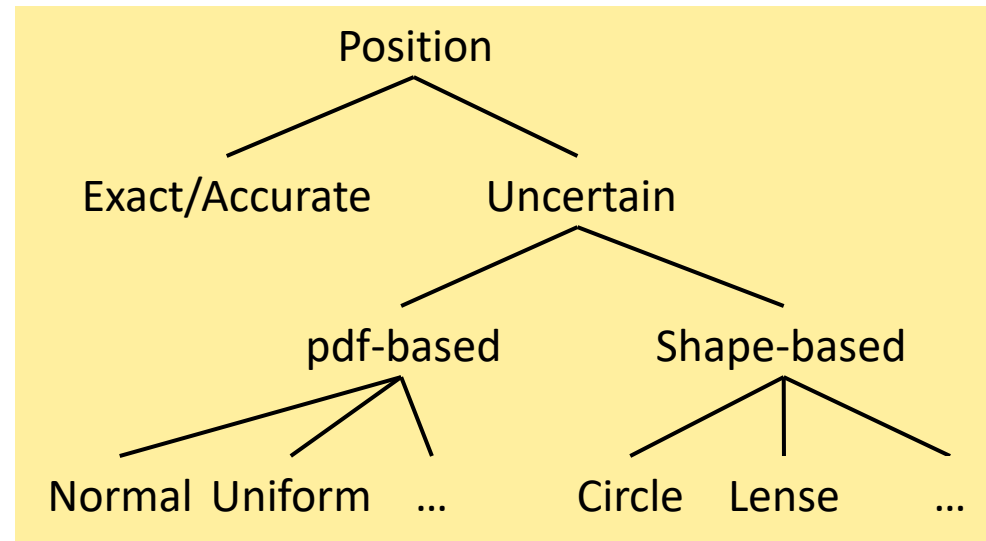
### 1. pdf-based models

- $s_{i,t} : \mathbb{R}^2 \rightarrow [0, \infty)$

### 2. Shape-based models

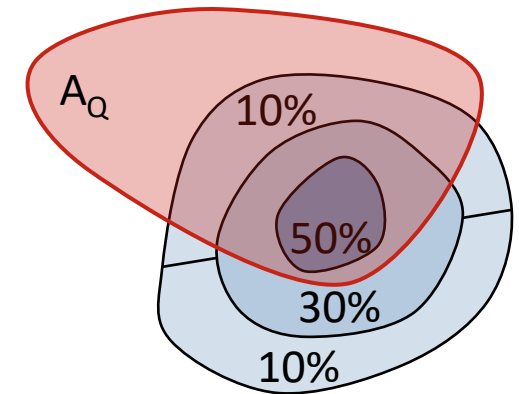
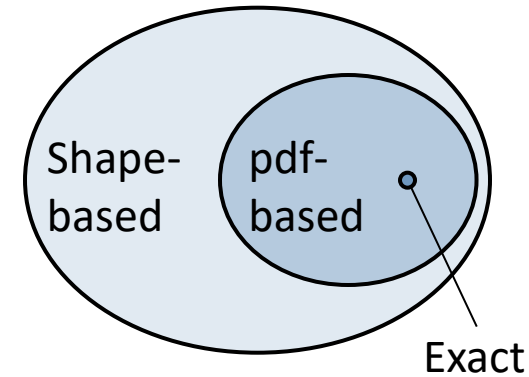
- $s_{i,t} = \{(A_1, p_1), (A_2, p_2), \dots\}$   
where  $A_i \subseteq \mathbb{R}^2$  and  $p_i \in [0, 1]$

### 3. Exact model ( $s_{i,t} = a_{i,t}$ )



# Generic Uncertainty Model

- All three classes provide probabilistic information
- All classes allow for mapping from one or more **shapes**  $A \subseteq \mathbb{R}^2$  to cumulative **probabilities**  $p$ 
  - $s_{i,t}: \mathcal{P}(\mathbb{R}^2) \rightarrow [0,1]$ , thus  $s_{i,t}(A) = P[A] = p$
- ➔ **Partial** spatial distribution function (PSDF)
  - PSDF even is complete for pdf-based models
- Given **any** shape  $A_Q$ , the PSDF allows for deriving estimates:  $0 \leq p_{\text{lower}} \leq s_{i,t}(A_Q) \leq p_{\text{upper}} \leq 1$



# Uncertainty-aware Query Interface

- Approach: Extend query types to account for **PSDF-based** uncertainties
  - Prevalent types: position, inside, range, distance, and next-neighbor query

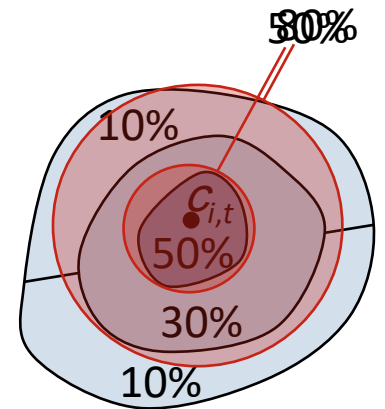
- PSDF map from shapes to probabilities

→ What about the **inverse**?

- **Defined point**  $c_{i,t}$  for object  $O_i$  at time  $t$

- Inherently given with most specific uncertainty models
- Unambiguous mapping  $\hat{c}_{i,t} : [0,1] \rightarrow \mathcal{P}(\mathbb{R}^2)$

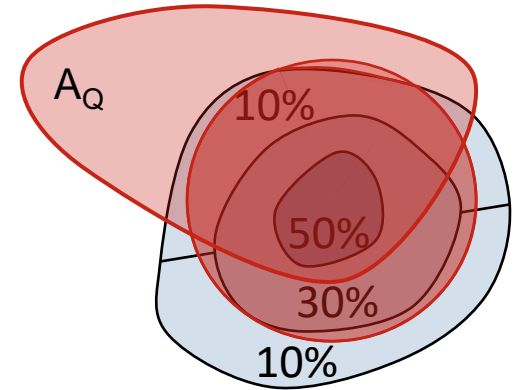
- $\hat{c}_{i,t}(p) = A^C$ , circle with minimum radius and center  $c_{i,t}$  such that  $s_{i,t}(A^C) \geq p$



# Uncertainty-aware Query Types

## Position query

- Expects probability  $p$  besides  $O_i$  and  $t$
- Signature:  $\mathbf{PQ}(O_i, t, p) \rightarrow (A, \hat{c}_{i,t}(p), p_{\text{lower}})$   
... where  $A := \hat{c}_{i,t}(p) \cap A_{i,t}^1$  and  $s_{i,t}(A) \geq p_{\text{lower}} \geq p$



## Inside query

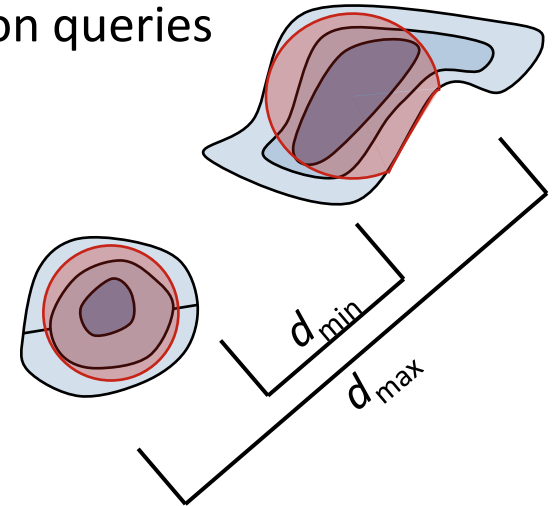
- Based on three-valued logic derived from two bounds  $p_{\text{false}} < p_{\text{true}}$
- Signature:  $\mathbf{IQ}(O_i, t, A_Q, p_{\text{true}}, p_{\text{false}}) \rightarrow (\{\text{true}, \text{maybe}, \text{false}\})$ 
  - Estimates  $p_{\text{lower}} \leq s_{i,t}(A_Q) \leq p_{\text{upper}}$  to test whether  $p_{\text{lower}} \geq p_{\text{true}}$  or  $p_{\text{upper}} \leq p_{\text{false}}$

**Range query** by multiple inside queries on set  $\{O_1, \dots, O_n\}$

# Uncertainty-aware Query Types (2)

## Distance query

- Based on possible distances between results from position queries
- Signature:  $\mathbf{DQ}(O_i, O_j, t, p) \rightarrow (d_{\min}, d_{\max})$ 
  - Distances between  $\hat{c}_{i,t}(p) \cap A_{i,t}^1$  and  $\hat{c}_{j,t}(p) \cap A_{j,t}^1$



## Nearest-neighbor query

- Based on distance queries
  - Signature:  $\mathbf{NNQ}(O_i, t, p) \rightarrow (O_j, d_{\min,j}, d_{\max,j})^*$ 
    - Result set contains all  $O_j$  where  $\min(d_{\max,1}, d_{\max,2}, \dots) = \delta \geq d_{\min,j}$
- Set of all **possible** nearest neighbors w.r.t. possible positions given by  $p$

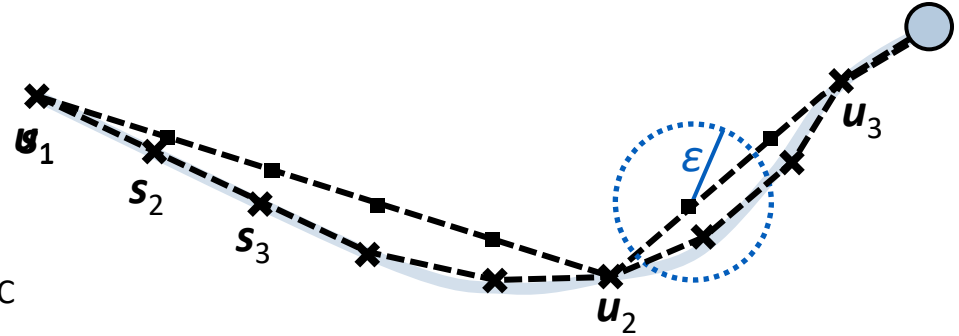
# Overview

- Motivation
- Generic uncertainty model and query interface
- Efficient real-time trajectory tracking
  - Formal problem statement for real-time tracking
  - Connection-Preserving Dead Reckoning (CDR)
  - Generic Remote Trajectory Simplification (GRTS)
  - Evaluation
- Summary

# Formal Problem Statement

## Kinds of trajectories

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



## Definition: Efficient real-time trajectory tracking

- Goals: Minimize  $|\mathbf{u}_1, \mathbf{u}_2, \dots|$  and communication cost
- Simplification constraint:  $|\mathbf{u}(t) - \mathbf{a}(t)| \leq \epsilon \quad \forall t$
- Real-time constraint:  $\mathbf{u}(t)$  is available  $\forall t \in [s_1.t, t_C]$

# Related Work

## Line simplification algorithms

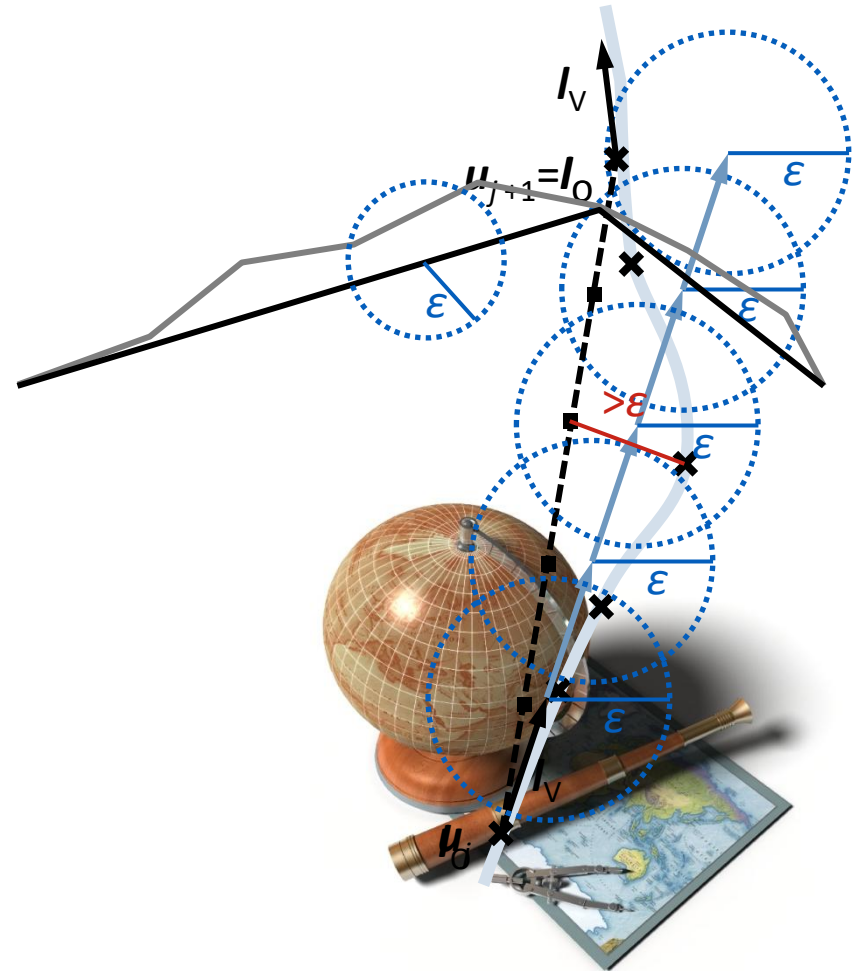
- E.g. Douglas-Peucker heuristic

➔ High communication cost,  
no real-time behavior

## Position tracking protocols

- Best mechanism: Dead Reckoning (DR)

➔ Linear DR with  $\frac{1}{2}\epsilon$  allows for trajectory tracking [Trajcevski et al. 2006]



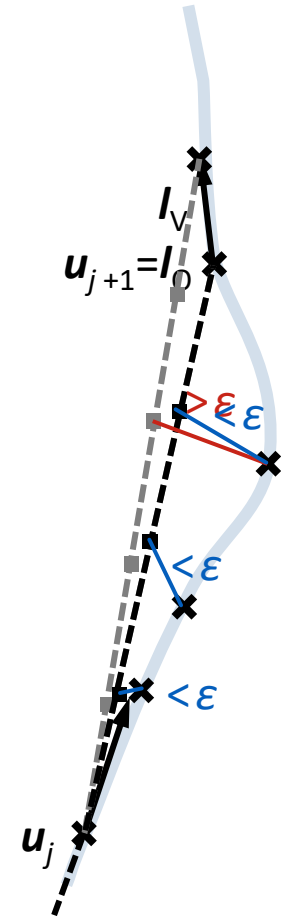
# Connection-Preserving DR

**CDR** extends linear DR

- Object manages  $I_O$ ,  $I_V$ , and **sensing history**  $\mathbb{S}$
- Second update condition:  
$$\exists s_i \in \mathbb{S} \text{ with } |\overline{I_O s_C}(s_i.t) - s_i.p| > \epsilon$$
- New prediction starts at **last** sensed position

**CDR<sub>m</sub>** limits computational cost by  $|\mathbb{S}| \leq m$

- Compression approach to prevent periodic updates

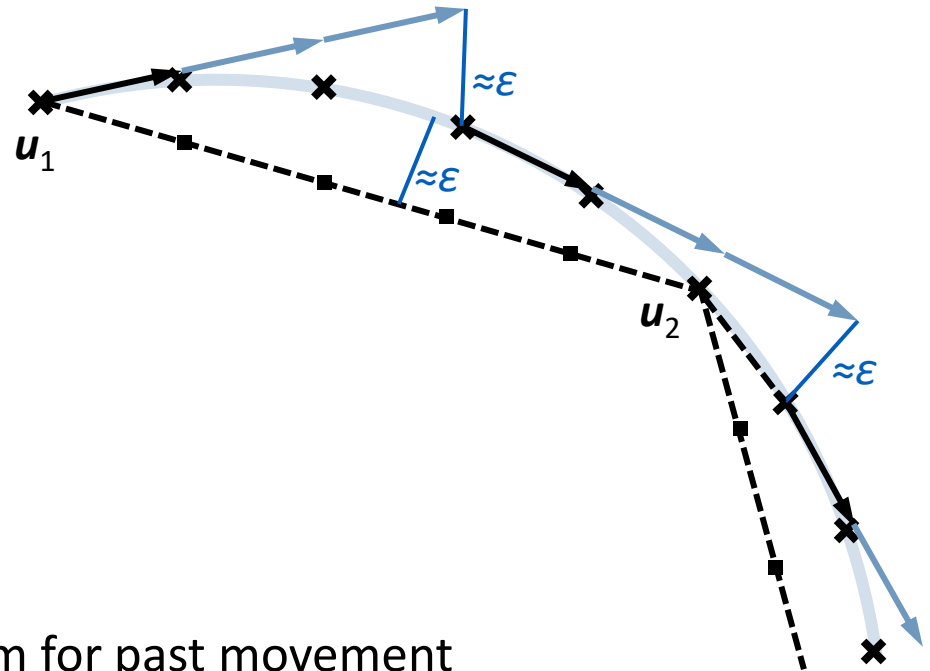


# GRTS – Generic Remote Trajectory Simplification

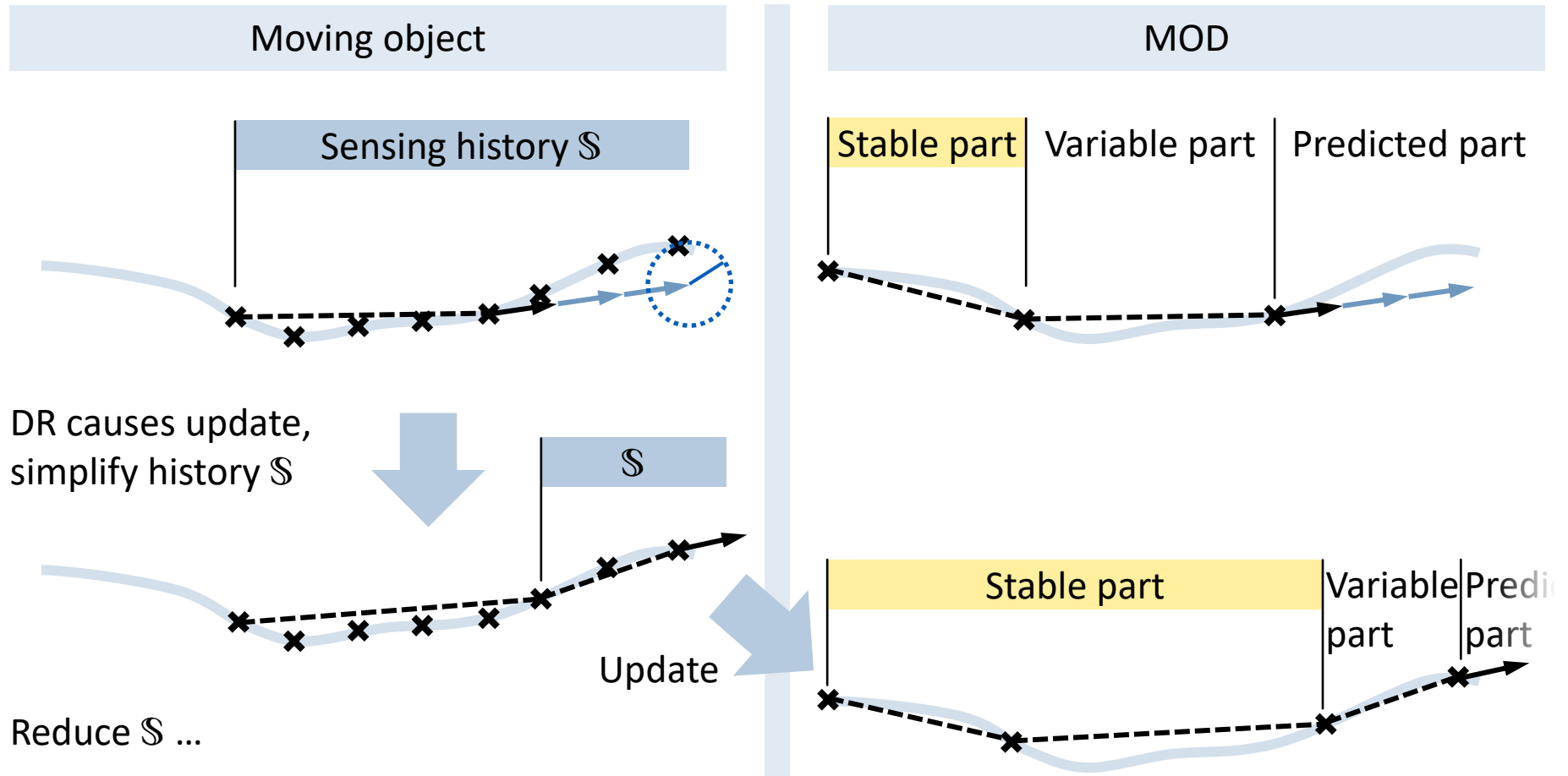
Tracking and simplification are **different concerns**

Basic approach of **GRTS**

- DR to report latest movement
  - Arbitrary line simplification algorithm for past movement
    - Computational cost  $\leftrightarrow$  reduction efficiency
- ➔ Tracking and simplification must be synchronized!



# GRTS: Basic Protocol



# GRTS: $k/m$ Variants

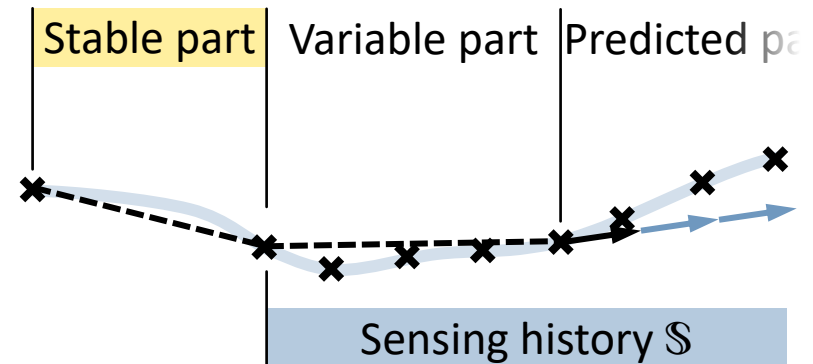
So far, not defined how to reduce  $\mathcal{S}$  ...

**GRTS<sub>k</sub>** limits variable part to  $k$  line sections

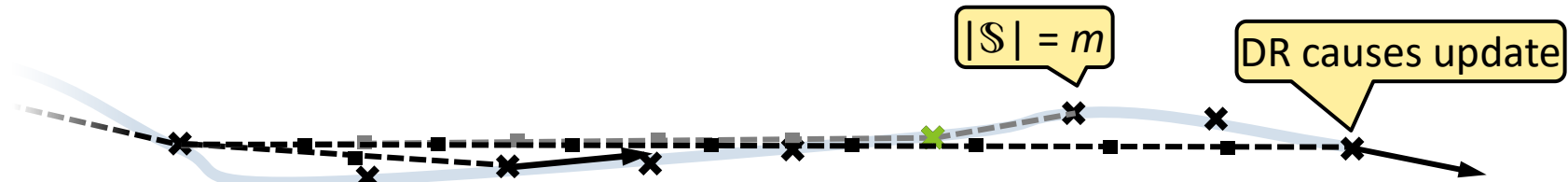
- Unlimited computational costs
- ➔ Only of theoretical interest

**GRTS<sub>m</sub>** limits  $|\mathcal{S}|$  to  $m$

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



# GRTS: mc Variant



$s_2$  with attributes  
 $p = (48.42^\circ, 9.01^\circ)$   
 $t = 2009-12-15\ 14:25:17.3$   
 $\delta = 7m$

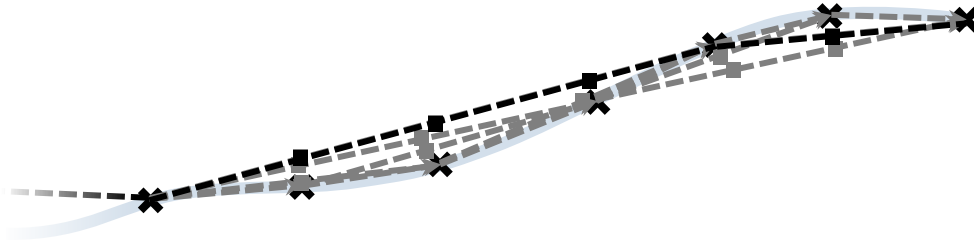
## GRTS<sub>mc</sub>

- $s_j \cdot \delta$  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,  $s_j \cdot \delta$  may represent varying  $\delta_{\text{sense}}$

# $GRTS_*^{Sec}$ 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  $|S|$

# Evaluation: Setup

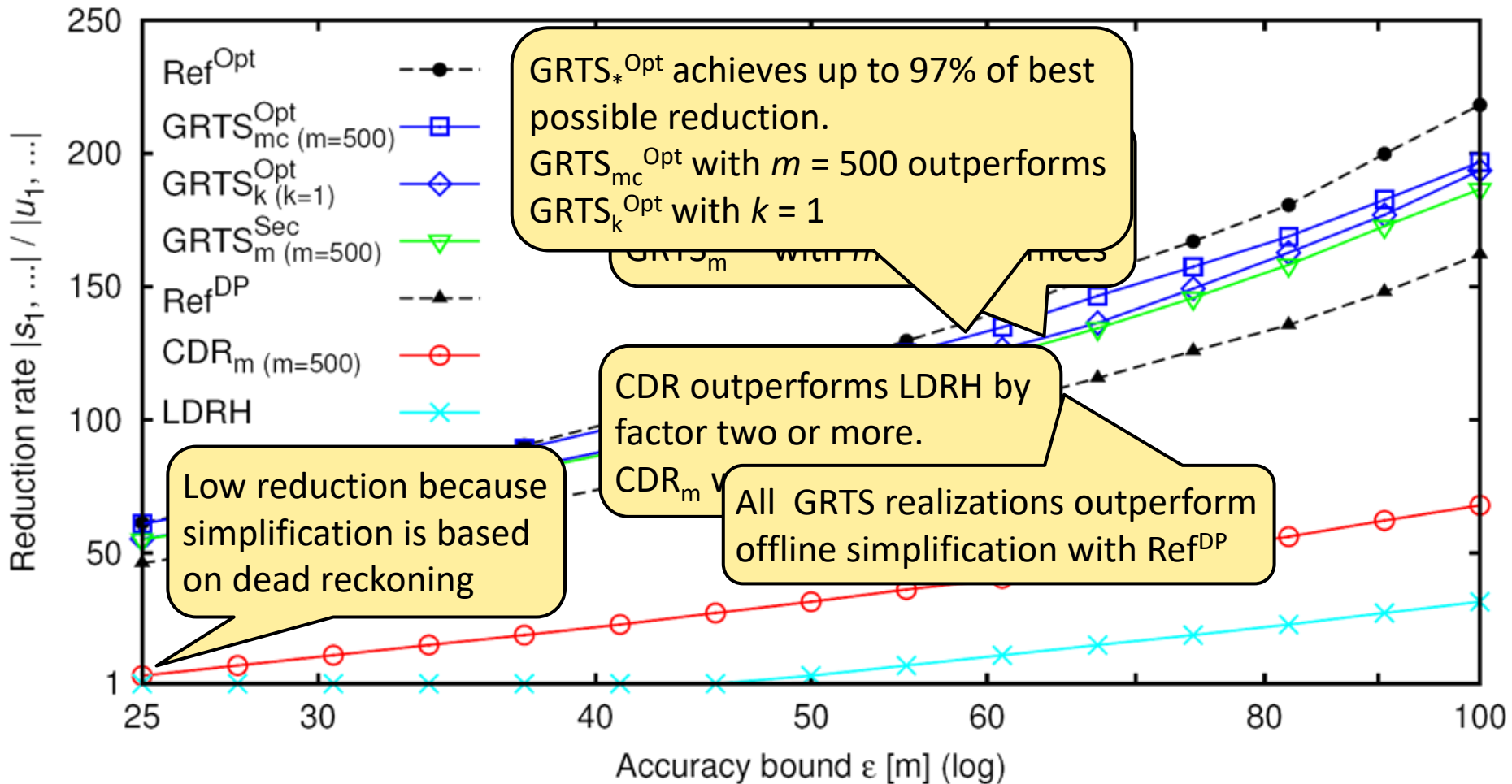
Comparing CDR variants and GRTS\* realizations to ...

- Linear DR with  $\frac{1}{2}\varepsilon$  (LDRH)
- Optimal offline simplification (Ref<sup>Opt</sup>)
- Douglas-Peucker algorithm (Ref<sup>DP</sup>)

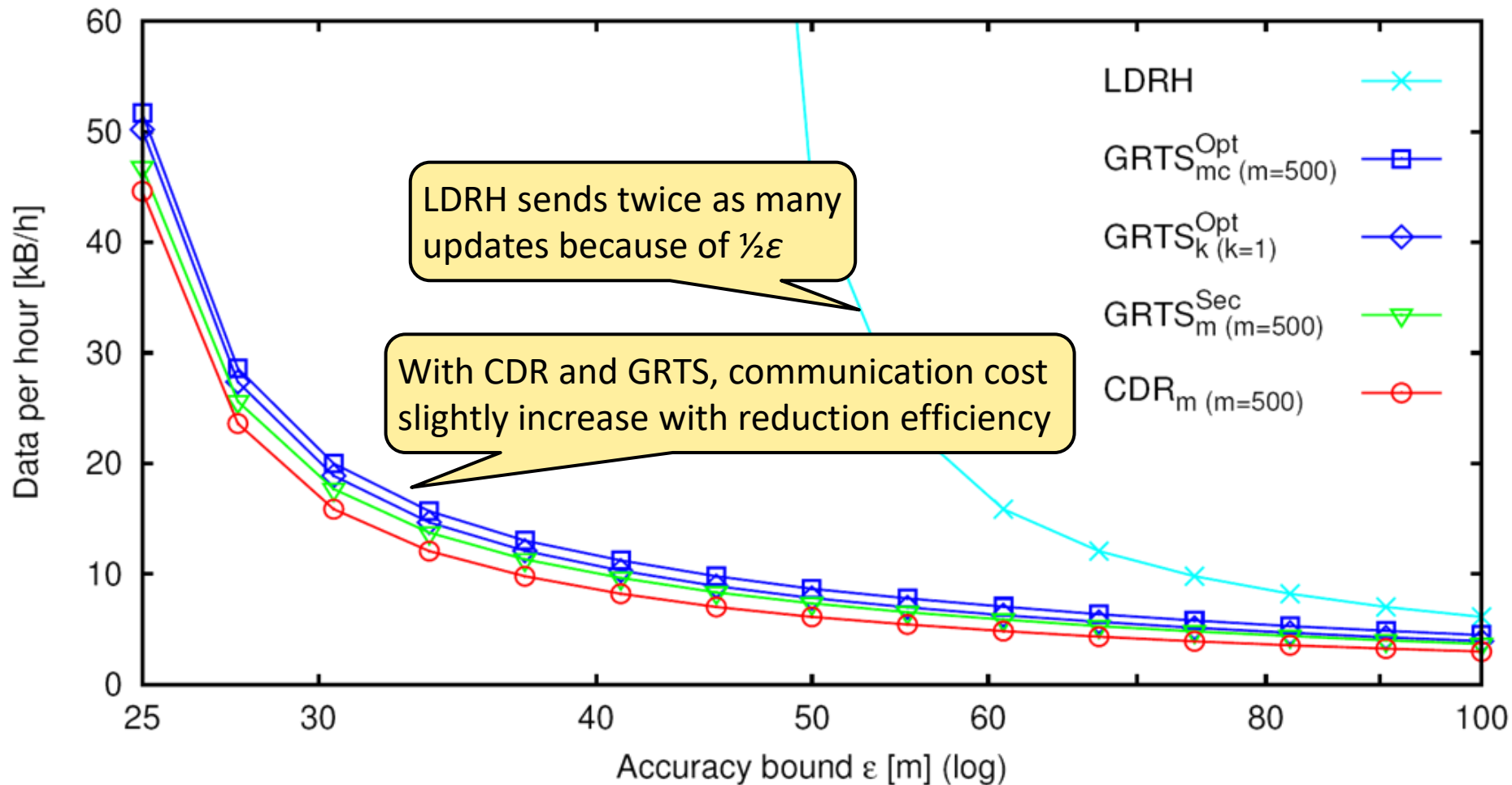
Simulated with **real** GPS traces from OpenStreetMap

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

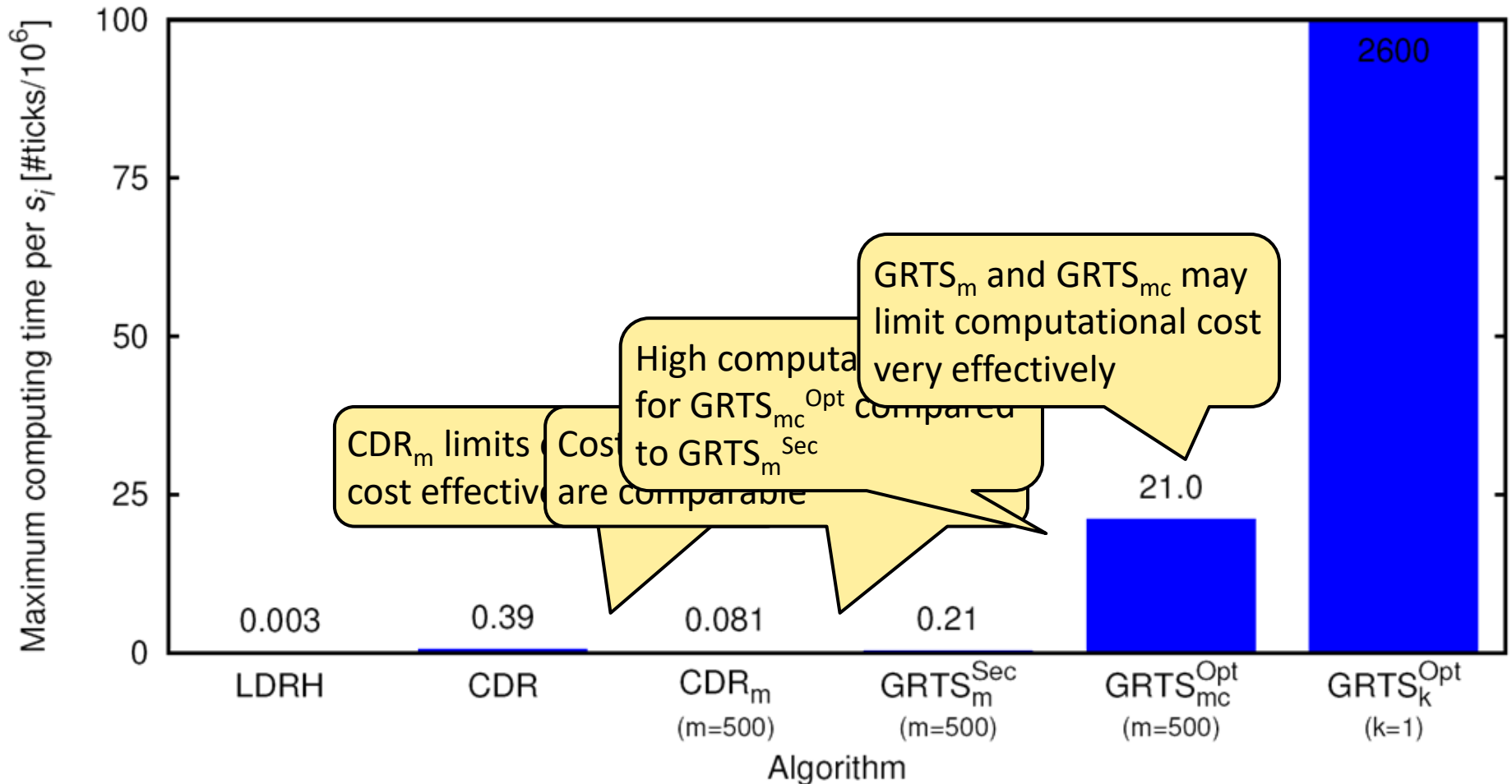
# Evaluation: Reduction



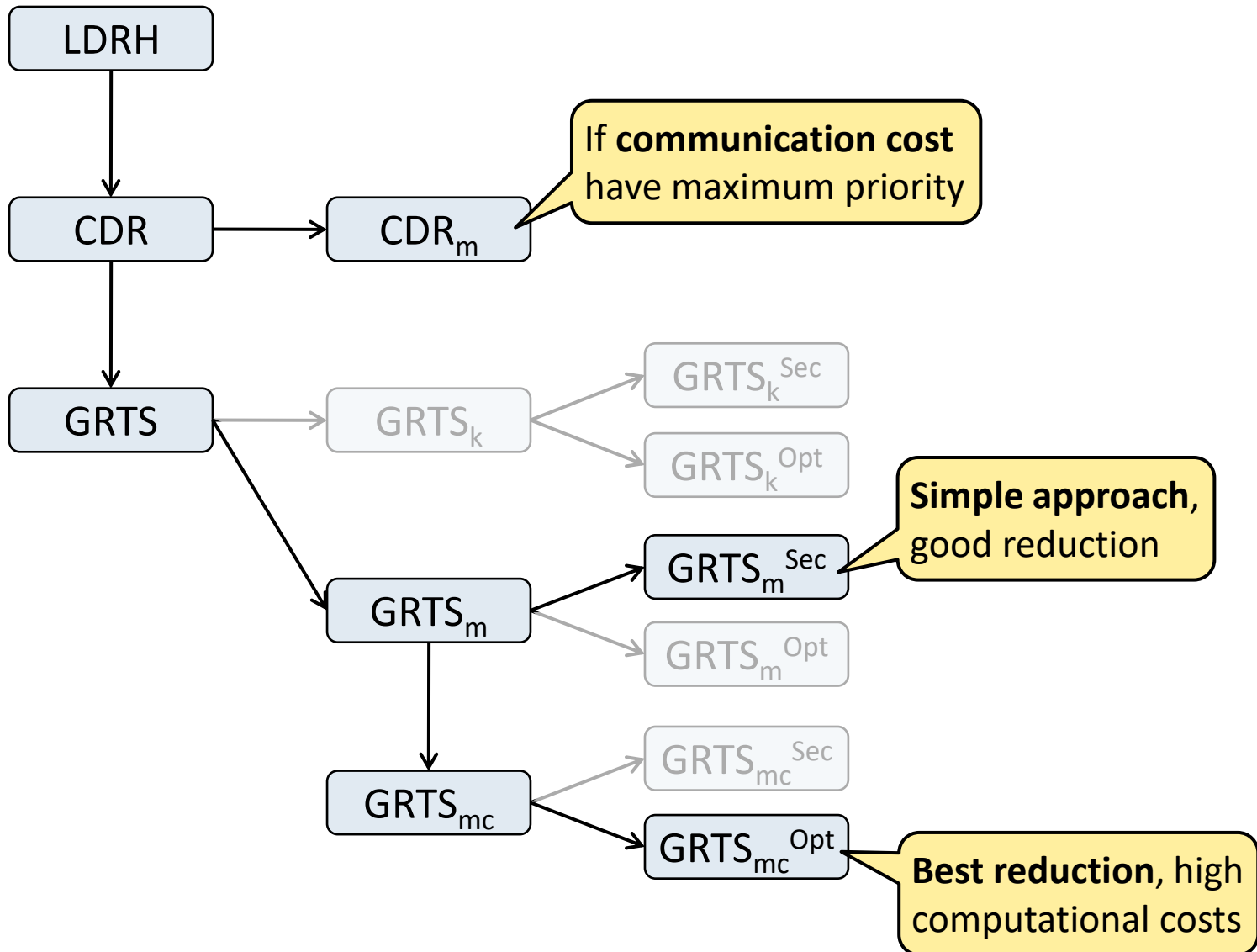
# Evaluation: Communication



# Evaluation: Computing Time



# Evaluation: Conclusion



# Summary

## Generic uncertainty-model for position information

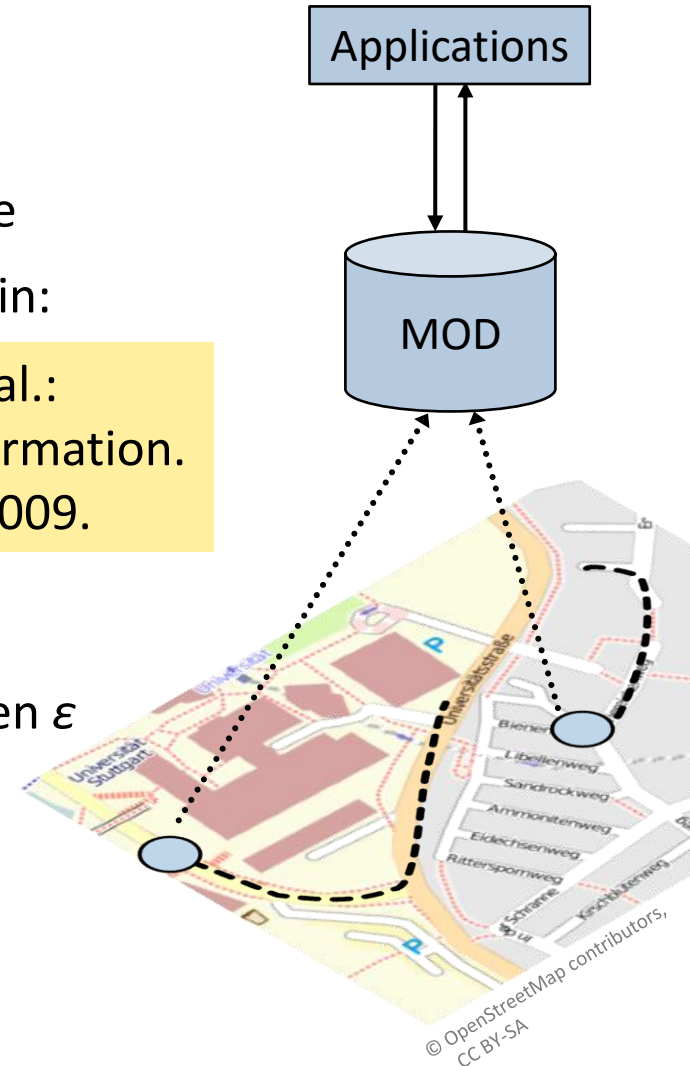
- Proposed extended and expressive query interface
- Further discussion with implementation sketches in:

Ralph Lange, Harald Weinschrott, Lars Geiger, et al.:  
On a Generic Uncertainty Model for Position Information.  
In *Proc. of 1st QuaCon*. Stuttgart, Germany. Jun 2009.

## Efficient real-time trajectory tracking

- Minimize communication and storage cost for given  $\epsilon$
- Algorithmic details, prototype and much more in:

Ralph Lange, Frank Dürr, and Kurt Rothermel:  
Efficient real-time trajectory tracking.  
*The VLDB Journal*, 20(5):671–694, Oct 2011.



**Thank you**  
for your attention!

**Dr. Ralph Lange**

[mail@lange-ralph.de](mailto:mail@lange-ralph.de)

[www.lange-ralph.de](http://www.lange-ralph.de)