Precise and Efficient Model-Based Vehicle Tracking Method Using Rao-Blackwellized and Scaling Series Particle Filters

Mengwen He[1*,3*], Eijiro Takeuchi[1,3], Yoshiki Ninomiya[2,3], and Shinpei Kato[3,4]

[1]Graduate School of Information Science, Nagoya University (* left since April, 2016)
[2]Institute of Innovation for Future Society (MIRAI), Nagoya University
[3]JST/COI, Nagoya (* left since April, 2016)
[4]Graduate School of Information Science and Engineering, the University of Tokyo

NAGOYA UNIVERSITY

Objective

- Develop a precise and efficient high-dimensional particle filter (PF) based vehicle tracking method for the intelligent vehicle to interact with the surrounding vehicles on the road (Fig. 1).

Figure 1: Precise and efficient tracking of a target vehicle can sensitively detect the driver's intension and will lead to reliable predictions for ensuring safe driving. For example: (Left) in the parking-lot, when a parked vehicle starts moving out, its small position change should be tracked for collision avoidance; (Middle) on the road, when a running vehicle tends to change lane, its small orientation change should be tracked for safe overtaking; (Right) at the intersection without traffic light, when a speedy vehicle comes out, its speed change should be tracked for deciding whether to pass the intersection.

Introduction

- Vehicle tracking technique enables the intelligent vehicle to interact with the surrounding vehicles (Fig. 1).
- Bayesian approach is a common and efficient method to solve object-tracking problems.
- PF is a type of non-parametric Bayesian filter:
  - Advantages: can represent complex beliefs and copes with non-linear motion models.
  - Disadvantages: computational cost increases exponentially with the tracking state's dimensionality.
- Rao-Blackwellized PF (RBPF) and Scaling Series PF (SSPF) can relief the curse of dimensionality:
  - RBPF: marginalize some parameters in the tracking state and use Gaussian estimate to track them.
  - SSPF: annealing-based interactive PF focuses limited particles on the regions with higher probability.
- This paper presents an improved vehicle tracking method by combining RBPF and SSPF (noted as RBSSPF).

Contributions

- We introduced the latest measurement into the motion estimate to get accurate pose update. (FastSLAM 2.0)
- With the new tracking method, we fully exploited the advantage of SSPF through the entire tracking process.
  - Motion: SSPF directly estimates the next pose using non-linear motion models with the latest measurement.
  - Geometry: SSPF derives the best geometry fitting results even with a partial observation (multi-mode problem).
- We conducted an experiment using two intelligent vehicles to evaluate our tracking method:
  - Both vehicles use Velodyne and 3D point cloud map to derive accurate localization results as ground-truth.
  - 10 km driving distance of public dataset with flat roads, narrow streets, and steep ramps.

General Tracking Method Description

(1) The Dynamic Bayesian Network (DBN) of Our Tracking Method

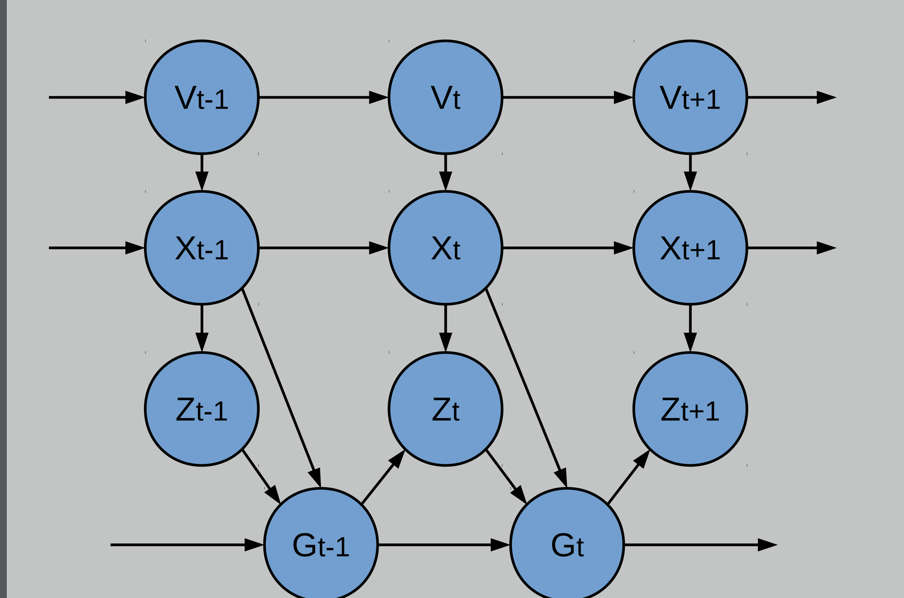


Figure 2: The modified DBN in our tracking method, which introduces the latest measurement into the motion estimate. The representation of this DBN is that for a target target vehicle and a given sensor measurement $\mathbf{Z_t}$ at time $\mathbf{t}$, we estimate the vehicle's pose $\mathbf{X_t}$, motion $\mathbf{V_t}$, and geometry $\mathbf{G_t}$.

The Update Equations of The DBN in Fig. 2

- Upper part: motion estimate:
  - Motion model: $\begin{cases} p(V_t|V_{t-1}) \\ p(X_t|X_{t-1}, V_t) \end{cases}$
  - Measurement model: $p(Z_t|X_t, G_{t-1})$
    - The estimate of V_t and X_t considers Z_t.
    - Z_t is related to the true geometry G^{**}.
    - Both G^{**} and G_t are still unknown.
    - But G_t will converge to constant G^{**}.
    - We use G_{t-1} for the motion estimate at $\mathbf{t}$.
- Lower part: geometry estimate:
  - Geometry update: $p(G_t|X_t, G_{t-1}, Z_t)$
- The Bayesian belief at time $\mathbf{t}$ according to the history of the estimate of the pose, motion, and geometry based on a set of sensor measurement:

$$Bel_t = p(X^t, V^t, G^t|Z^t)$$

(2) The Solution of The Bayesian Belief Bel_t

$$Bel_t = p(G_t|X^t, V^t, G^{t-1}, Z^t) \cdot p(X_t, V_t|X^{t-1}, V^{t-1}, G^{t-1}, Z^t) \cdot p(X^{t-1}, V^{t-1}, G^{t-1}|Z^t) = S_t \cdot R_t \cdot Bel_{t-1}$$

Step 1: Motion Estimate

- The motion Posterior R_t:

$$\begin{aligned} R_t &= p(X^{t-1}, V^{t-1}, G^{t-1}|Z^t) \\ &\propto p(Z_t, X_t, V_t|X^{t-1}, V^{t-1}, G^{t-1}, Z^{t-1}) \\ &= p(Z_t|X^t, V^t, G^{t-1}, Z^{t-1}) \\ &\quad \cdot p(X_t|X^{t-1}, V^t, G^{t-1}, Z^{t-1}) \\ &\quad \cdot p(V_t|X^{t-1}, V^{t-1}, G^{t-1}, Z^{t-1}) \\ &= p(Z_t|X_t, G_{t-1}) \cdot p(X_t|X_{t-1}, V_t) \cdot p(V_t|V_{t-1}) \\ &\propto p(X_t, V_t|X_{t-1}, V_{t-1}, G_{t-1}, Z_t) \end{aligned}$$

- We can use the SSPF (an optimization method) to directly optimize R_t based on X_{t-1}, V_{t-1}, G_{t-1}, and Z_t. Then we can get the updated motion V_t and pose X_t.
- The SSPF optimization for the motion estimate is similar to the scan-matching in FastSLAM 2.0.

Step 2: Geometry Estimate

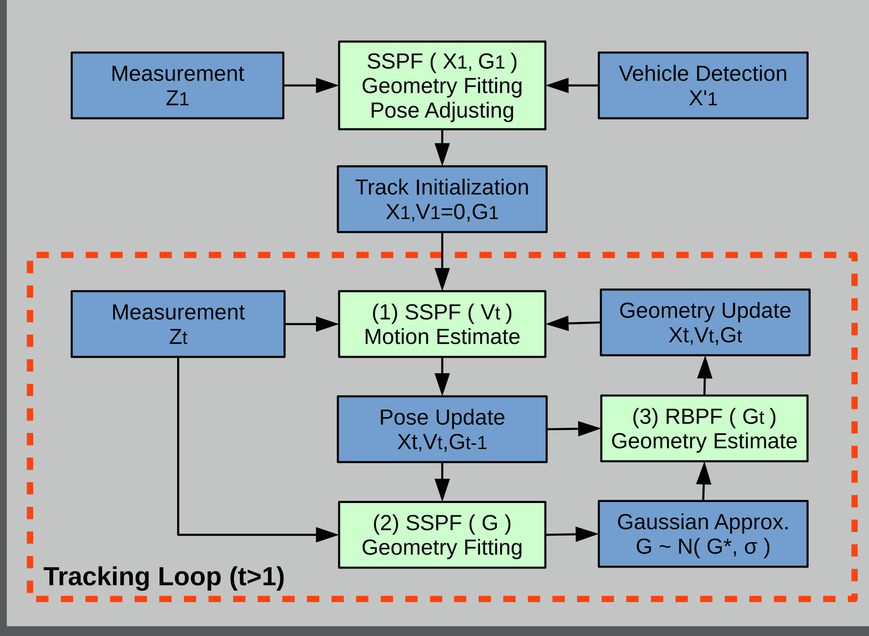
- The geometry posterior S_t conditioned on motion:

$$\begin{aligned} S_t &= p(G_t|X^t, V^t, G^{t-1}, Z^t) \\ &\propto p(Z_t, G_{t-1}, X_t, G_t|X^{t-1}, V^t, G^{t-2}, Z^{t-1}) \\ &= p(Z_t|X^t, V^t, G^t, Z^{t-1}) \\ &\quad \cdot p(G_{t-1}|X^t, V^t, G_t, G^{t-2}, Z^{t-1}) \\ &\quad \cdot p(X_t|X^{t-1}, V^t, G_t, G^{t-2}, Z^{t-1}) \\ &\quad \cdot p(G_t|X^{t-1}, V^t, G^{t-2}, Z^{t-1}) \\ &\propto p(Z_t|X_t, G_{t-1}) \cdot S_{t-1} \end{aligned}$$

- G_{t-1} is out of date after the motion estimate.
- Replace $p(Z_t|X_t, G_{t-1})$ with the measurement likelihood $p(Z_t|X_t, G)$ to obtain better geometry estimate G^* from geometry fitting done by SSPF.
- Linearize $p(Z_t|X_t, G)$ on G in Gaussian form $\mathcal{N}(G^*, \sigma)$ to perform RBPF on G_t estimate:

$$S_t \propto p(Z_t|X_t, G) \cdot S_{t-1} \sim \mathcal{N}(G^*, \sigma) \cdot S_{t-1}$$

(3) The Method Framework



eps Summary of The General Tracking Method

- Bayesian belief of tracking and update equation
  - $Bel_t = p(X^t, V^t, G^t|Z^t)$
  - $Bel_t = S_t \cdot R_t \cdot Bel_{t-1}$
- Motion estimate and pose update (X_t and V_t)
  - SSPF on $R_t \propto p(X_t, V_t|X_{t-1}, V_{t-1}, G_{t-1}, Z_t)$
- Geometry fitting and linearization ($p(Z_t|X_t, G)$)
  - SSPF on $p(Z_t|X_t, G)$ with respect to $G \Rightarrow G^*$
  - $p(G|X_t, Z_t) \sim \mathcal{N}(G^*, \sigma)$ (Laplace approx.)
- Geometry estimate (G_t)
  - RBPF on $S_t \propto N(G^*, \sigma) \cdot S_{t-1}$

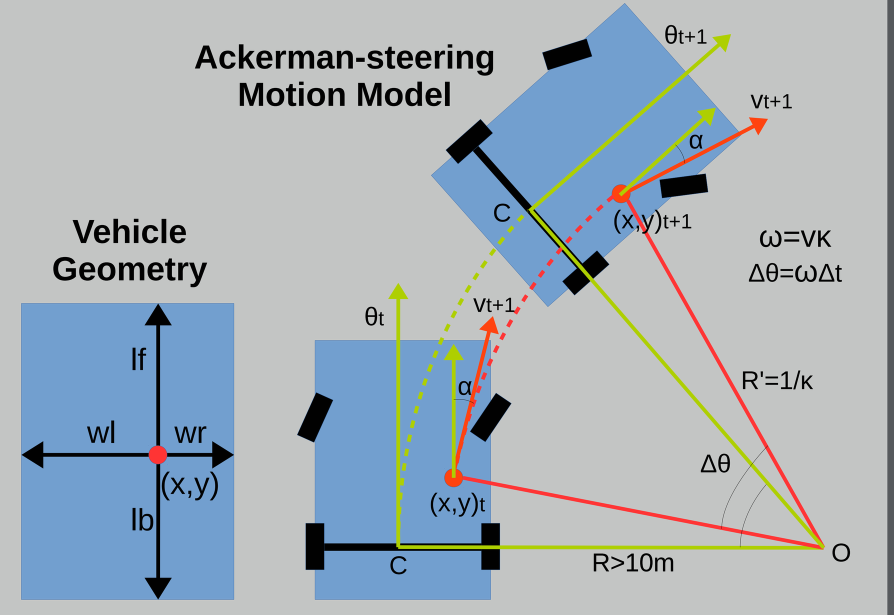
Tracking Method Implementation

(1) Tracking State Definition & Ackerman-Steering Motion Model

- Our tracking implementation works on a 2D coordinates with 10 dimensional state:

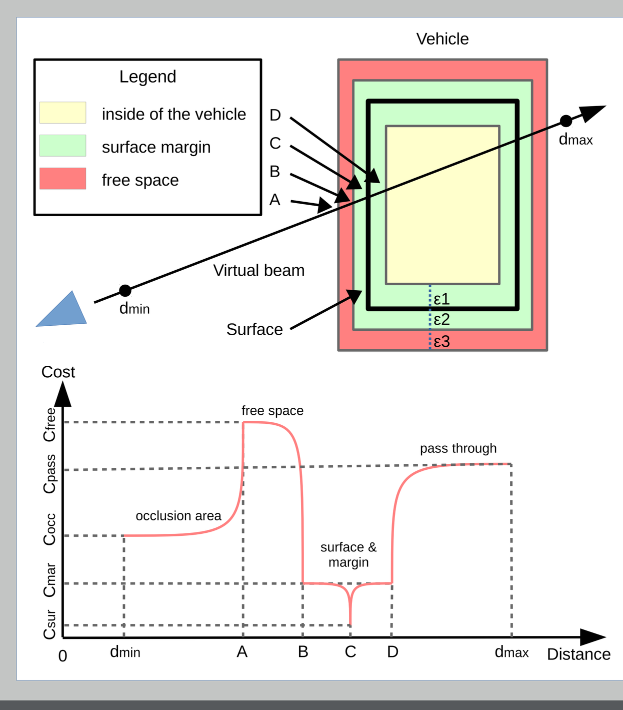
	Parameters	Note
Pose (X)	x, y	position of anchor point
	θ	orientation of vehicle
Motion (V)	α^*	orientation offset of velocity
	v	velocity
	κ	curvature
Geometry (G)	wl, wr	distance to left/back edge
	lf, lb	distance to front/back edge

* Because the anchor point is randomly selected from the detection result (Fig.1 upper) and the geometry is temporarily unknown, the variable α adjusts the velocity direction to be tangential to the circular trajectory (Fig.1).



(2) Virtual Scan & Measurement Model

- We developed a noval virtual scan algorithm to convert the 3D Velodyne point cloud to a 2D scan range array. (published on IV16: *"Robust Virtual Scan for Obstacle Detection in Urban Environments"*)
- The measurement model and the cost function are shown as right figure. $\mathbf{d_{min}}$–**A** corresponds to the occlusion area, **A**–**B** corresponds to the free space, **B**–**D** corresponds to the vehicle's surface, and **D**–$\mathbf{d_{max}}$ corresponds to the passing through the vehicle.
- The measurement model is used by the SSPF, which gives best results for continuous and differentiable beliefs with varying gradients; therefore, the common constant cost function is approximated by exponential functions.



Experiment Results

Platforms & Driving/Tracking Trajectories

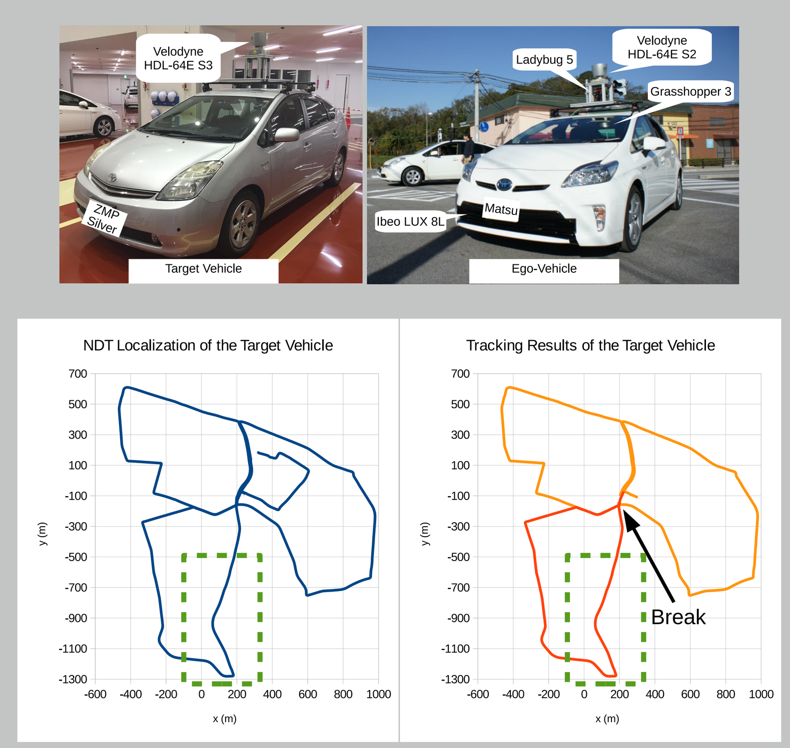
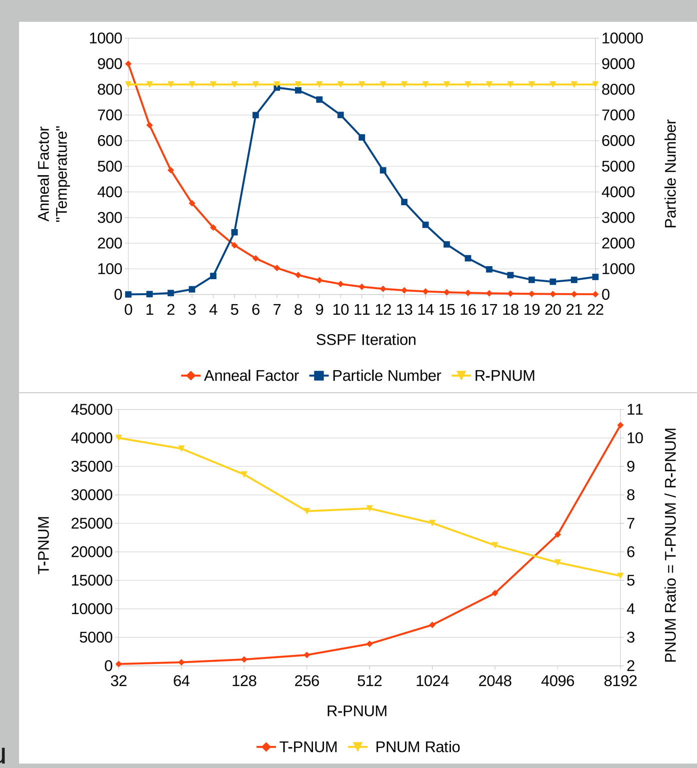


Figure 3: The target vehicle "ZMP Silver" and the ego-vehicle "Matsu". The green rectangle is the "challenge" area for evaluation.

The Particle Number of RBSSPF

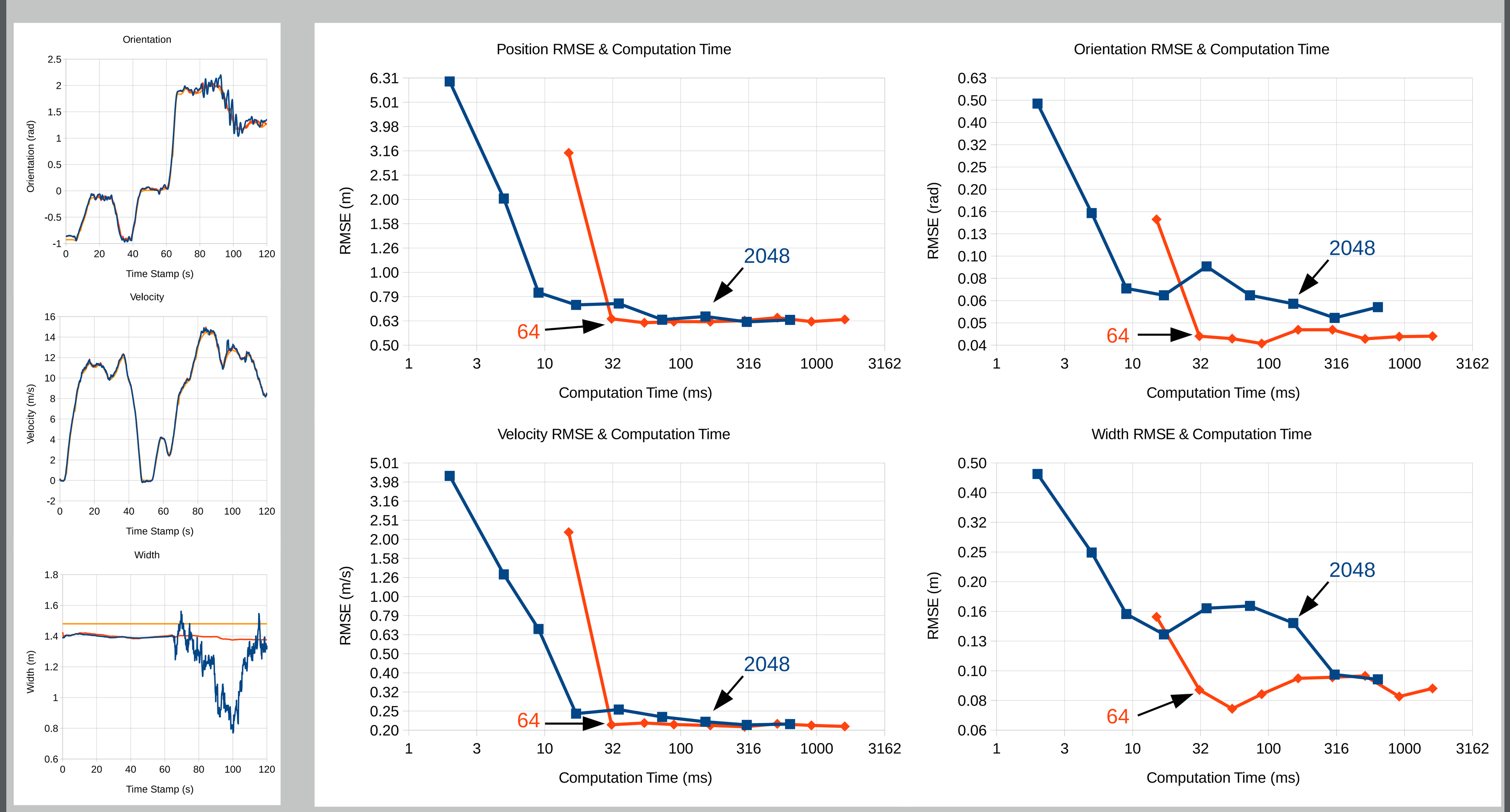


u

Figure 4: R-PNUM: required number of particles after each resampling. T-PNUM: total number of particles after all iterations.

The Evaluation and Comparison with RBPF Tracking Results (In The Green Rectangle Area)

- Orange: ground-truth; Blue: RBPF; Red: RBSSPF.
- Left: One example of the tracking results (RBPF: R-PNUM=2048; RBSSPF: R-PNUM=64).
- Right: The comparison of RMSE vs. CPU computational time (T-PNUM) between RBPF and RBSSPF.



The conclusion of evaluation results:

- The minimum RMSE of the RBSSPF is not greater than that of the RBPF, and specifically, the RBSSPF is more precise in orientation and width estimate than RBPF.
- The RBSSPF spends less iime (R-PNUM=64) to achieve the level of minimum RMSE than the RBPF (R-PNUM≥1024).
- If we constrain the computation time of tracking, the RBSSPF can obtain more precise results than RBPF, and if we require to reach a certain level of accuracy, the RBSSPF spends less time than the RBPF.

The GPU Acceleration for Real-time Tracking