Combining 3D Shape, Color, and Motion for Robust Anytime Tracking

Paper by Held, Levinson, Thrun, and Savarese [1]

Frederik Zwilling

Seminar Current Topics in Computer Vision and Machine Learning

RWTH Aachen University

Agenda

Tracking for Autonomous Cars

Chances

- Free use of driving time
- Help disabled persons
- Computers do not get tired or drunk
- Faster reaction time

⇒ Safe 26k lives per year in EU

Tracking for Autonomous Cars

Chances

- Free use of driving time
- Help disabled persons
- Computers do not get tired or drunk
- Faster reaction time

⇒ Safe 26k lives per year in EU

Challenges

- Precise tracking
- Robustness
- Occlusion
- Real time

Tracking for Autonomous Cars

Subtasks of Tracking

- Segment sensor data into objects
- Associate objects in successive frames
- Position and velocity estimation
- Object and trajectory classification

Tracking for Autonomous Cars

Subtasks of Tracking

- Segment sensor data into objects
- Associate objects in successive frames
- Position and velocity estimation
- Object and trajectory classification

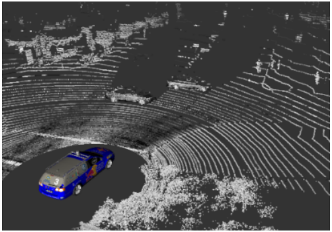
Topic of this presentation

Position and velocity estimation

Given Sensor Data

Sensor

- Dense 3D laser sensor
- Generates point cloud
- Additional panorama image
- Similar to stereo cameras
  but more precise and expensive



[2]

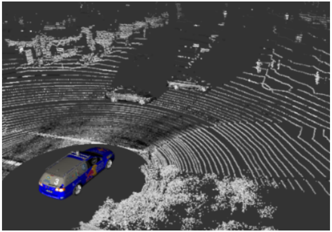
Given Sensor Data

Sensor

- Dense 3D laser sensor
- Generates point cloud
- Additional panorama image
- Similar to stereo cameras
  but more precise and expensive

Given for us

- Point clouds of detected objects
- Association between frames
  already done



[2]

Teaser Paper Ideas

How to find a precise alignment?

- Utilize whole object shape
- Additional cues from color
- Use motion model

Teaser Paper Ideas

How to find a precise alignment?

- Utilize whole object shape
- Additional cues from color
- Use motion model

How to search the state space fast?

- Histogram with coarse initial resolution
- Refine resolution important areas
- Consider resolution in the probabilistic model

Baseline Methods

[3]

Kalman Filter

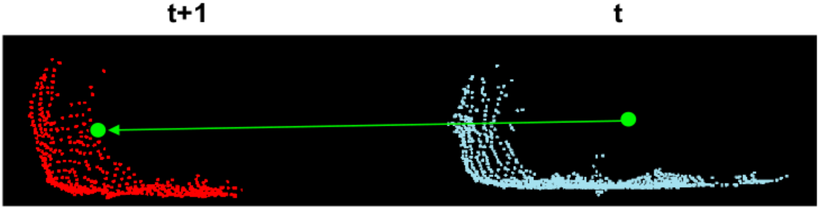
Baseline Methods

[3]

Kalman Filter

- Aligns centroids

Baseline Methods

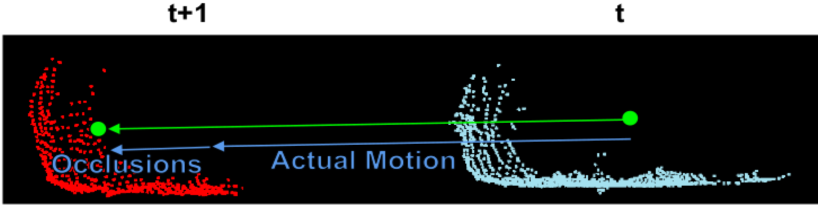


[3]

Kalman Filter

- Aligns centroids

Baseline Methods

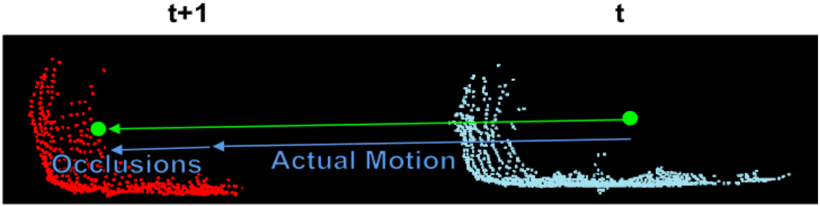


[3]

Kalman Filter

- Aligns centroids
- Problems with occlusion

Baseline Methods

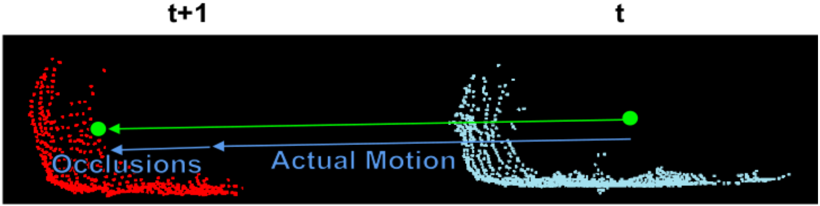


[3]

Kalman Filter

- Aligns centroids
- Problems with occlusion
- Robustness through motion model

Baseline Methods



[3]

Kalman Filter

- Aligns centroids
- Problems with occlusion
- Robustness through motion model
- Very fast

Baseline Methods

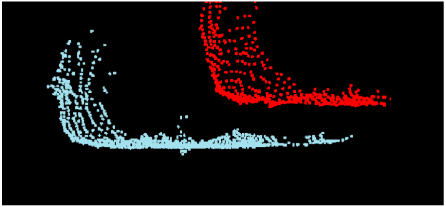
Iterative Closest Point (ICP)

- Iterative hill climbing approach
- Minimizes quadratic distance of closest points
- Uses whole point cloud

Baseline Methods

Iterative Closest Point (ICP)

- Iterative hill climbing approach
- Minimizes quadratic distance of closest points
- Uses whole point cloud
- Depends on good initialization
- Problem: local optima

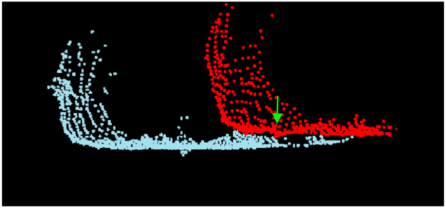


[3]

Baseline Methods

Iterative Closest Point (ICP)

- Iterative hill climbing approach
- Minimizes quadratic distance of closest points
- Uses whole point cloud
- Depends on good initialization
- Problem: local optima

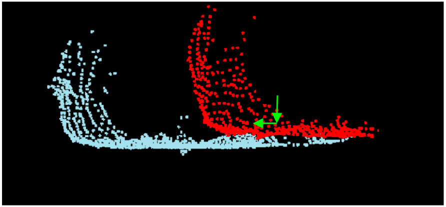


[3]

Baseline Methods

Iterative Closest Point (ICP)

- Iterative hill climbing approach
- Minimizes quadratic distance of closest points
- Uses whole point cloud
- Depends on good initialization
- Problem: local optima

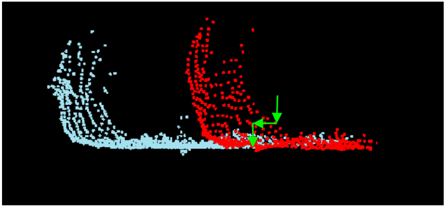


[3]

Baseline Methods

Iterative Closest Point (ICP)

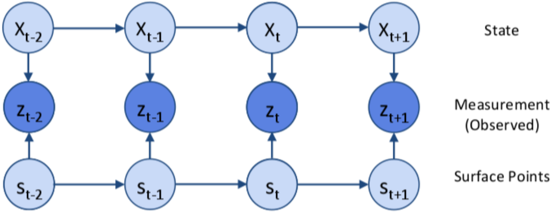
- Iterative hill climbing approach
- Minimizes quadratic distance of closest points
- Uses whole point cloud
- Depends on good initialization
- Problem: local optima



[3]

- No motion model

Probabilistic Model



[1]

Dynamic Bayesian Network

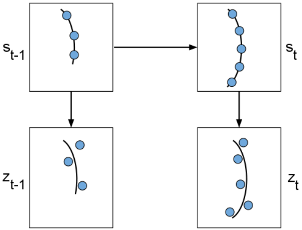
- Relates variables over successive frames
- State x_t (position relative to last frame and velocity)
  Surface points $s_t = \{s_{t,1}, ..., s_{t,n}\}$
  Measured point cloud $z_t = \{z_{t,1}, ..., z_{t,n}\}$
- Rotation not considered

Probabilistic Model

Surface points

- Sampled from the visible surface
- Indirectly observable
- Visible surface varies due to occlusion and viewpoint changes
- $p(s_{t,i}|s_{t-1}) =$
  $p(V) * p(s_{t,i}|s_{t-1}, V) +$
  $p(\neg V) * p(s_{t,i}|s_{t-1}, \neg V)$

$\Rightarrow$ $p(s_t|s_{t-1}) =$
$\eta(\mathcal{N}(s_t; s_{t-1,i}, \Sigma_r) + k)$

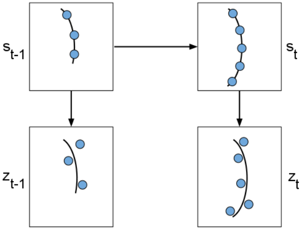


[1]

Probabilistic Model

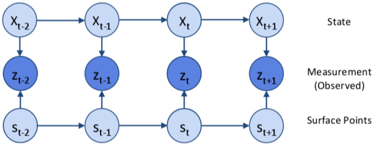
Measurement points

- Depending on surface points
- Gaussian sensor noise Σ_e
- $z_{t,i} \sim \mathcal{N}(s_{t,i}, \Sigma_e) + x_{t,p}$
  $z_{t-1,i} \sim \mathcal{N}(s_{t-1,i}, \Sigma_e)$



[1]

Probabilistic Model

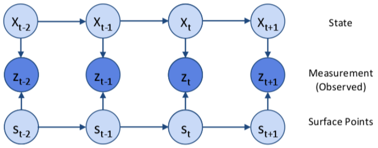


Measurement Model

$$p(z_t | x_t, z_1, \dots, z_{t-1})$$

Probabilistic Model

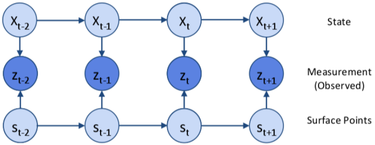
Measurement Model



$$p(z_t|x_t, z_1, ..., z_{t-1}) \approx p(z_t|x_t, z_{t-1})$$

Probabilistic Model

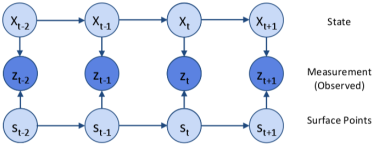
Measurement Model



$$p(z_t|x_t, z_1, ..., z_{t-1}) \approx p(z_t|x_t, z_{t-1})$$
$$= \int\int p(z_t, s_t, s_{t-1}|x_t, z_{t-1})\mathrm{d}s_t\mathrm{d}s_{t-1}$$

Probabilistic Model

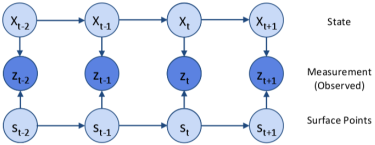
Measurement Model



$$
\begin{aligned}
p(z_t|x_t, z_1, \dots, z_{t-1}) &\approx p(z_t|x_t, z_{t-1}) \\
&= \int \int p(z_t, s_t, s_{t-1}|x_t, z_{t-1}) \mathrm{d}s_t \mathrm{d}s_{t-1} \\
&= \underbrace{\int p(z_t|s_t, x_t) \underbrace{\int p(s_t|s_{t-1}) \eta p(z_{t-1}|s_{t-1}) \mathrm{d}s_t}_{\text{convolution}} \mathrm{d}s_{t-1}}_{\text{convolution}}
\end{aligned}
$$

Probabilistic Model

Measurement Model



$$
\begin{aligned}
p(z_t|x_t, z_1, ..., z_{t-1}) &\approx p(z_t|x_t, z_{t-1}) \\
&= \int\int p(z_t, s_t, s_{t-1}|x_t, z_{t-1}) \mathrm{d}s_t \mathrm{d}s_{t-1} \\
&= \underbrace{\int p(z_t|s_t, x_t) \underbrace{\int p(s_t|s_{t-1}) \eta p(z_{t-1}|s_{t-1}) \mathrm{d}s_t}_{\text{convolution}} \mathrm{d}s_{t-1}}_{\text{convolution}} \\
&= \eta(\mathcal{N}(z_t; z_{t-1} + x_{t,p}, \Sigma_r + 2\Sigma_e) + k)
\end{aligned}
$$

Probabilistic Model

Measurement Model Computation

- $ccp(z_{t,i})$: closest correspondence point in $z_{t-1} + x_{t,p}$

Measurement Probability

$$p(z_t|x_t, z_{t-1}) = \eta \prod_{z_{t,i} \in z_t} \exp\left(-\frac{1}{2}(z_{t,i} - \mathrm{ccp}(z_{t,i}))^T \Sigma^{-1} (z_{t,i} - \mathrm{ccp}(z_{t,i}))\right) + k$$

Probabilistic Model

Measurement Model Computation

- $ccp(z_{t,i})$: closest correspondence point in $z_{t-1} + x_{t,p}$
- Covariance matrix $\Sigma = 2\Sigma_e + \Sigma_r$

Measurement Probability

$$p(z_t|x_t, z_{t-1}) = \eta \prod_{z_{t,i} \in z_t} \exp\left(-\frac{1}{2}(z_{t,i} - \mathrm{ccp}(z_{t,i}))^T \Sigma^{-1} (z_{t,i} - \mathrm{ccp}(z_{t,i}))\right) + k$$

Probabilistic Model

Measurement Model Computation

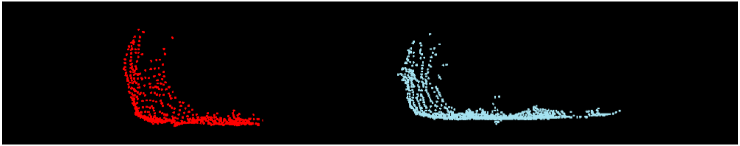
- $ccp(z_{t,i})$: closest correspondence point in $z_{t-1} + x_{t,p}$
- Covariance matrix $\Sigma = 2\Sigma_e + \Sigma_r$
- Normalization constant η
- Smoothing factor k

Measurement Probability

$$p(z_t|x_t, z_{t-1}) = \eta \prod_{z_{t,i} \in z_t} \exp\left(-\frac{1}{2}(z_{t,i} - \mathrm{ccp}(z_{t,i}))^T \Sigma^{-1} (z_{t,i} - \mathrm{ccp}(z_{t,i}))\right) + k$$

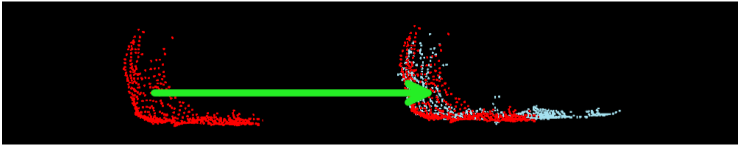
Probabilistic Model

- Align smaller point cloud in larger one



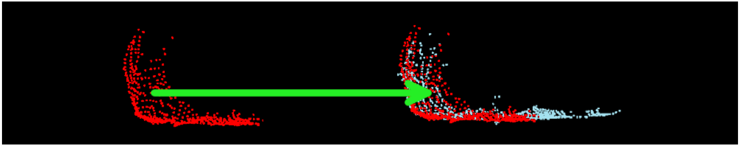
Probabilistic Model

- Align smaller point cloud in larger one

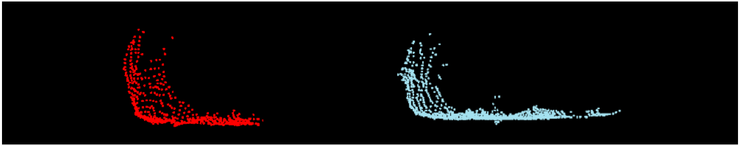


Probabilistic Model

- Align smaller point cloud in larger one

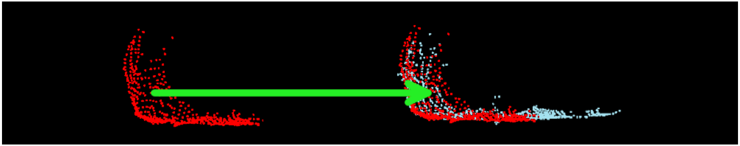


- Aligning larger point cloud in smaller one

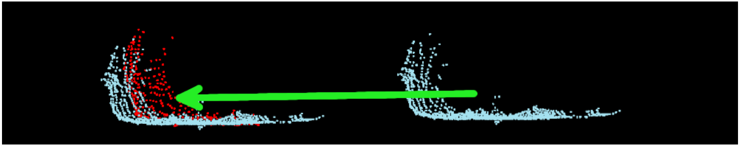


Probabilistic Model

- Align smaller point cloud in larger one

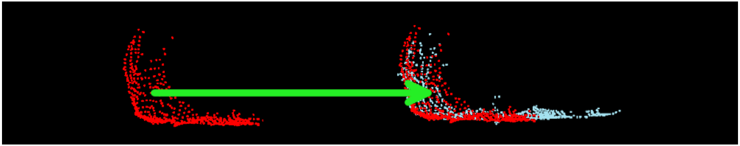


- Aligning larger point cloud in smaller one

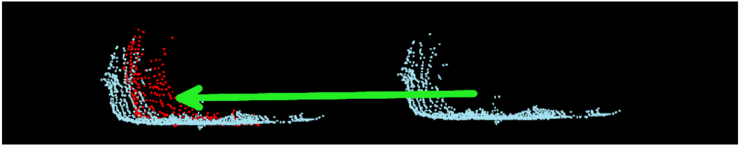


Probabilistic Model

- Align smaller point cloud in larger one



- Aligning larger point cloud in smaller one



$\Rightarrow$ Exchange z_t and z_{t-1} when $|z_t| > |z_{t-1}|$

Probabilistic Model

Adding Color

- Use available color information of measurement points
- Better alignment despite occlusions
- Embed color matching into measurement model

$$p_c(s_{t,i}|s_{t-1},V) = p(C)p_c(s_{t,i}|s_{t-1},V,C) + p(\neg C)p_c(s_{t,i}|s_{t-1},V,\neg C)$$

- $p(C)$: Color-model applicable
- $p(\neg C)$: Color-model not applicable (lens flare, reflections)
- $p_c(s_{t,i}|s_{t-1},V,\neg C) = \frac{1}{255}$: Color mismatch
- $p_c(s_{t,i}|s_{t-1},V,C)$: Color match

Probabilistic Model

Motion Model

- Take whole measurement history into account
- Robustness against single imprecise alignments

Probabilistic Model

Motion Model

- Take whole measurement history into account
- Robustness against single imprecise alignments
- Kalman filter with constant velocity model
- Measurement step: update with Gaussian distribution
  $\mu_t = \sum_i p(x_{t,i}|z_1, ..., z_t)x_{t,i}$
  $\Sigma_t = \sum_i p(x_{t,i}|z_1, ..., z_t)(x_{t,i} - \mu_t)(x_{t,i} - \mu_t)^T$
- Update step: apply velocity to position

Searching the State Space

Search the State Space

Searching the State Space

Search the State Space

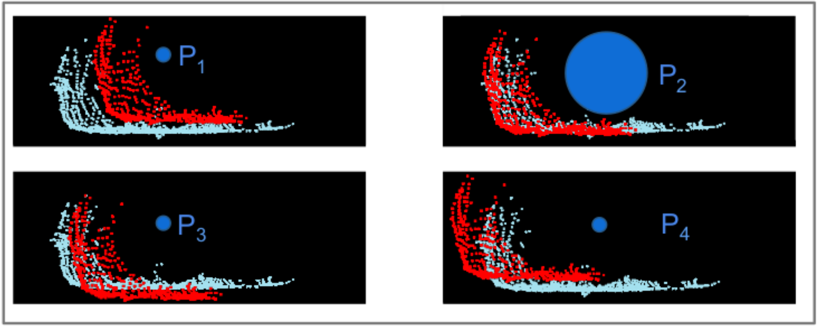
- For most likely state
- Globally, without getting stuck in local optima
- Allow multiple hypotheses

Searching the State Space

Search the State Space

- For most likely state
- Globally, without getting stuck in local optima
- Allow multiple hypotheses

⇒ Use histogram



[3]

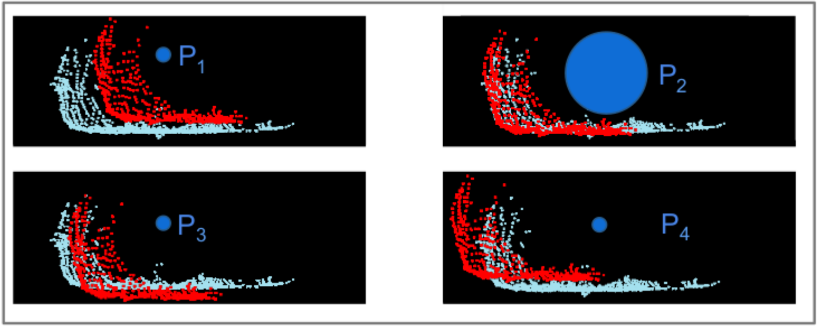
Searching the State Space

Search the State Space

- For most likely state
- Globally, without getting stuck in local optima
- Allow multiple hypotheses

⇒ Use histogram

⇒ Too slow for necessary resolution

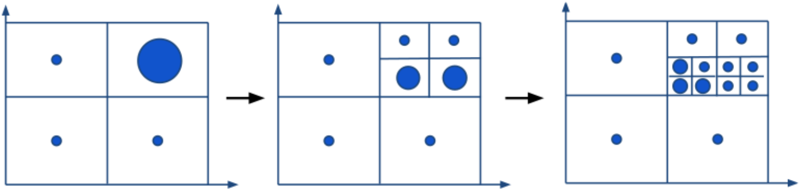


[3]

Searching the State Space

In Real Time

- Densely sample only areas with high probability
- Initial coarse resolution
- Stop at anytime during refinement



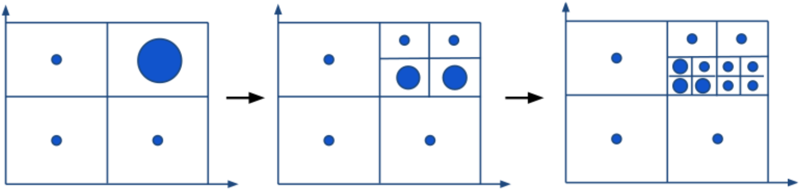
[1]

Searching the State Space

In Real Time

- Densely sample only areas with high probability
- Initial coarse resolution
- Stop at anytime during refinement

⇒ Use dynamic histogram

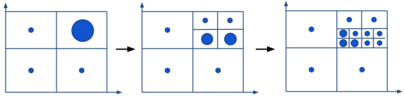


[1]

Searching the State Space

Derivation Step

- Recursively expend cells

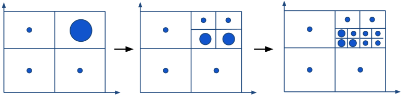


[1]

Searching the State Space

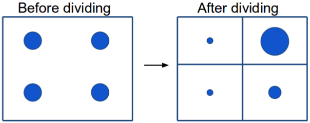
Derivation Step

- Recursively expend cells



[1]

- Maximize information gain / Minimize KL-divergence

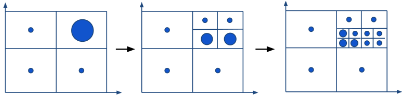


[1]

Searching the State Space

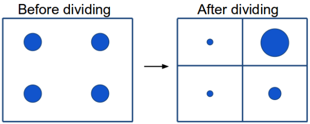
Derivation Step

- Recursively expend cells



[1]

- Maximize information gain / Minimize KL-divergence



[1]

- KL-divergence: information loss if A approximated by B

$$D_{KL}(A||B) = \sum_{j=1}^{k} p_j \ln\left(\frac{p_j}{P_i/k}\right)$$

Searching the State Space

Annealing Dynamic Histogram (ADH)

- No good alignment with coarse resolution
- Sampling resolution another error source

$\Rightarrow$ Consider sampling resolution in measurement model

- $\Sigma = 2\Sigma_e + \Sigma_r + \Sigma_g$
- Σ_g proportional to sampling resolution

Evaluation

What to evaluate

- Precision of the position and velocity estimation
- Root-Mean-Square error

$$e_{RMS} = \sqrt{\mathbb{E}((\hat{v}_t - v_t)^2)}$$

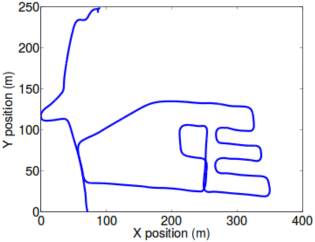
- Runtime (real time requirements)
- Comparison to baseline methods

$\Rightarrow$ Need for test data

Evaluation

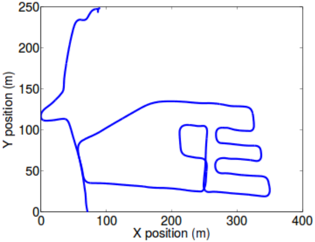
Relative Reference Frame Approach

- Record sensor data while driving
- Static environment
- Assume car is standing still



[1]

Evaluation



[1]

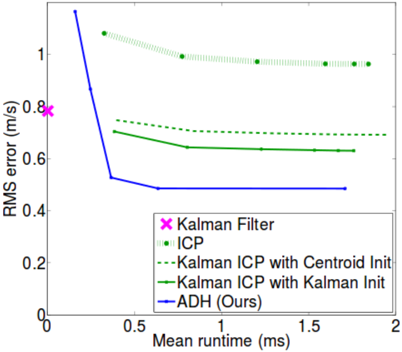
Relative Reference Frame Approach

- Record sensor data while driving
- Static environment
- Assume car is standing still
- Compute ground truth position and velocity
  from distance and car velocity

Evaluation

Kalman Filter

- Imprecise but very fast



[1]

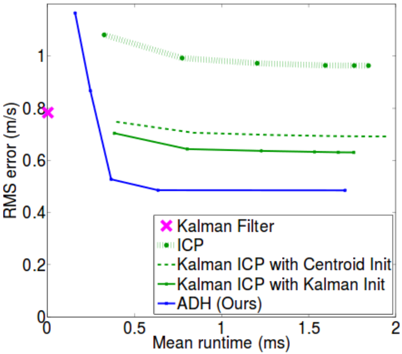
Evaluation

Kalman Filter

- Imprecise but very fast

ICP

- Slow and very imprecise
- Variants with motion model perform better



[1]

Evaluation

Kalman Filter

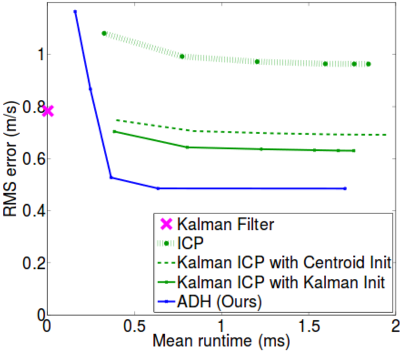
- Imprecise but very fast

ICP

- Slow and very imprecise
- Variants with motion model perform better

ADH

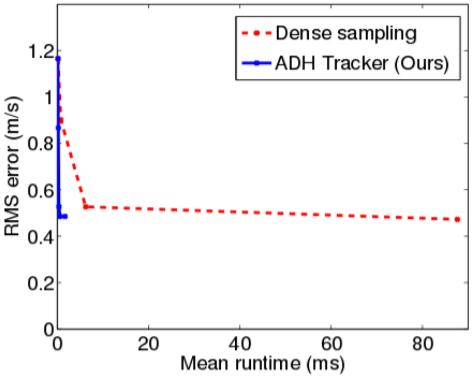
- Quick first result
- Outperforms all baseline methods



[1]

Evaluation

ADH acceleration



[1]

Evaluation

- Relative reference frame precise but not realistic

Evaluation

- Relative reference frame precise but not realistic

Model Crispness Approach

- Build a model of the tracked object
- Union object point clouds over all frames
- Point clouds aligned by position estimation

[1]

- Record sensor data in real traffic

Evaluation

Crispness Score

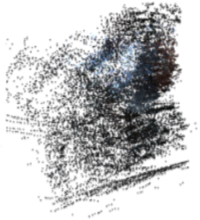
- Measure for distinctness of the build model

Evaluation

Crispness Score

- Measure for distinctness of the build model

[1]



[1]

- Presented tracker
- High crispness score

- Kalman ICP tracker
- Low crispness score

Evaluation

Tracking Method	Object Class		
	People	Bikes	Moving Cars
Kalman Filter	0.38	0.31	0.27
Kalman ICP	0.18	0.18	0.29
ADH (Ours)	**0.42**	**0.38**	**0.33**

[1]

- Highest score for all object classes
- Higher score for people than for cars

Evaluation

Improvement through color model

- RMS error decreased by 10.4%
- $p(C)$ very small (lens flare, heavy shadows)
- Works robust through day-times and seasons

Source Code

- Open Source
  http://stanford.edu/~davheld/anytime_tracking.html
- Easy to setup, integrate into C++
- Test-data available

Performance with stereo camera data?

- Cheaper sensor, more noisy data
- A lot of configuration values in the code

Conclusion

Combine 3D Shape, Color, Motion for Robust Anytime Tracking

Precise and robust tracking is possible in real-time

- Measurement model combines 3D shape, Color, Motion
- Derived from Dynamic Bayesian Network
- Annealed Dynamic Histogram for global and fast search
- Evaluation with local reference frame and model crispness
- Outperforms baseline methods

References I

- Held, D., Levinson, J., Thrun, S., Savarese, S.:
  Combining 3D Shape, Color, and Motion for Robust Anytime Tracking.
  In: Proceedings of Robotics: Science and Systems, Berkeley, USA (July 2014)
- Teichman, A., Levinson, J., Thrun, S.:
  Towards 3D Object Recognition via Classification of Arbitrary Object Tracks.
  In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE (2011) 4034–4041
- Held, D., Levinson, J., Thrun, S., Savarese, S.:
  Anytime Tracking.
  http://stanford.edu/~davheld/DavidHeld_files/RSS2014_Poster.pdf (2015)

Backup slide

Crispness score formula

- $$\frac{1}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}\frac{1}{n_i}\sum_{k=1}^{n_i} G(x_k - \hat{x}_k, 2\Sigma)$$

Backup slide: RoboCup Logistics League