Using road topology to improve cyclist path prediction

Pool, Ewoud; Kooij, Julian; Gavrila, Dariu

DOI
10.1109/IVS.2017.7995734

Publication date
2017

Document Version
Accepted author manuscript

Published in
Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)

Accepted Author Manuscript. Link to published article (IEEE): http://dx.doi.org/10.1109/IVS.2017.7995734

Using Road Topology to Improve Cyclist Path Prediction

Ewoud A. I. Pool [a,1], Julian F. P. Kooij [b,2] and Dariu M. Gavrila [a,b,3]

***Abstract*— We learn motion models for cyclist path prediction on real-world tracks obtained from a moving vehicle, and propose to exploit the local road topology to obtain better predictive distributions. The tracks are extracted from the Tsinghua-Daimler Cyclist Benchmark for cyclist detection, and corrected for vehicle egomotion. Tracks are then spatially aligned to local curves and crossings in the road. We study a standard approach for path prediction in the literature based on Kalman Filters, as well as a mixture of specialized filters related to specific road orientations at junctions. Our experiments demonstrate an improved prediction accuracy (up to 20% on sharp turns) of mixing specialized motion models for canonical directions, and prior knowledge on the road topology. The new track data complements the existing video, disparity and annotation data of the original benchmark, and will be made publicly available.**

I. INTRODUCTION

In recent years, Advanced Driver Assistance Systems (ADAS) have shown robust performance to detect and track traffic participants using a variety of on-board sensors, leading to active safety systems that can warn or intervene if a collision is imminent. This is especially important in the urban environment with many Vulnerable Road Users (VRUs). For automated driving however, the system should not only detect VRUs, but also predict their trajectories to anticipate and avoid potentially dangerous situations [1], [2]. Most literature on VRU path prediction focuses on pedestrians (e.g. [3], [4], [5], [6]), where various cues have been proposed to improve trajectory prediction, such as pedestrian attention, spatial layout, etc. Predicting cyclist tracks from a moving vehicle is also challenging, as cyclists move fast, and can be observed for long durations where high-level behavior results in distinct paths, e.g. 'take turn left' versus 'go straight'. Failure to detect the subtle onset of change in a cyclist's dynamics can therefore lead to large prediction errors, even at short time horizons (e.g. ~ 1 sec.).

While research on pedestrian path prediction yielded various publications and datasets, relatively few work currently focuses on cyclist [7], [8]. However, recently a large benchmark dataset on cyclist *detection* from a moving vehicle was made publicly available [9]. In this paper, we augment this dataset by extracting cyclist tracks that we compensate for vehicle egomotion, and use these tracks to learn viewpoint invariant cyclist dynamics in real-world settings. We define a coordinate system related to road topology to spatially align cyclists in the vicinity of curves and crossings where important changes in dynamics can occur. With these tracks, which are made publicly available, we evaluate standard motion models for path prediction, and propose an extension to leverage prior knowledge on the road topology to improve its predictions. Though not the focus of this paper, we note that good motion models can also benefit other tasks, such as data-association [10] and anomalous track detection [11], [12].

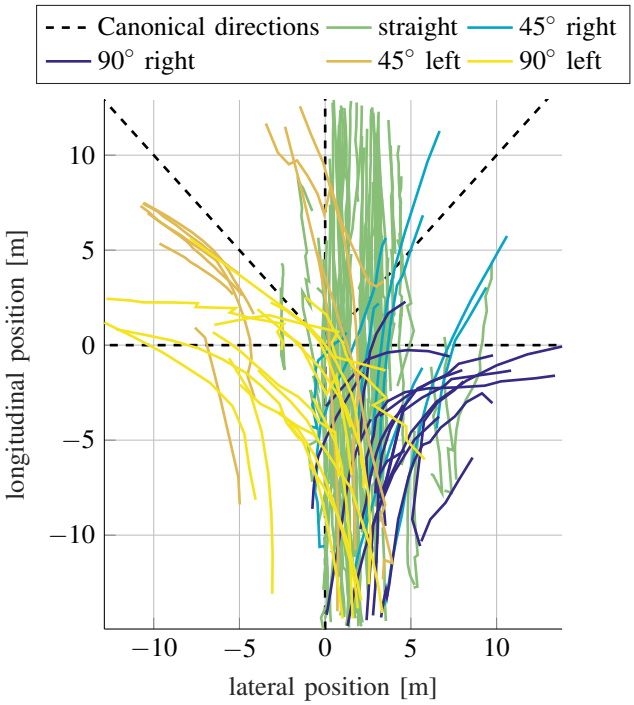


Fig. 1: Extracted real-world cyclist tracks, aligned with their local road topology which distinguishes five canonical directions. All tracks start at the bottom, and move upward. The figure shows that most (but not all) cyclists drive on the right side of the road. Note that some cyclists who plan to turn to the left are seen to cycle on the left side of the road, even before the crossing. Most tracks move straight.

II. RELATED WORK

Detection and tracking of VRUs has made great progress in recent years. [1] indicates that pedestrian tracking is becoming increasingly robust, and research is shifting to high-level tasks of predicting future traffic situations to inform automated decision making in ADAS. Besides pedestrians, a review of severe and fatal car-cyclist accidents showed that cyclists at crossings are another important safety case [2].

a) Amsterdam Machine Learning Lab, University of Amsterdam, The Netherlands; *b*) Intelligent Vehicles & Cognitive Robotics Group, Technical University Delft, The Netherlands; 1) E.A.I.Pool@uva.nl; 2) J.F.P.Kooij@tudelft.nl; 3) D.M.Gavrila@tudelft.nl

For pedestrians, various approaches focus first on classifying current traffic behavior [2], [4], [13], which can inform future behavioral events [6]. But predictive models of a pedestrian's path must represent spatial uncertainty too. A common approach leverages the motion models which are already an integral part of the tracker's filter [3]. More informed predictions are obtained by conditioning dynamics on additional cues, such as intent and awareness of the pedestrian [14], [15] or driver [16]. Others use the dynamics of the appearance (e.g. optical flow) to predict behavior [4].

The surroundings of a pedestrian provide another interesting cue for future behavior. Social factors, such as proximity to other VRUs, can be an informative factor [17], [18], [19]. Others condition the dynamics on the environment, e.g. on the spatial location within the scene [5], [12], or on the pedestrian's destination and semantic regions [19], [20], [21], [22]. For road users, the topological and geometric layout of crossings can be a powerful cue for future behavior, especially for crossings [11], [13], [23].

Cyclists at crossings are an especially important safety case for moving vehicles [2], which means it is important to study the problem from a vehicle's point of view. There has been work on how cyclist motion evolves at a crossing [7], [8], but this was from static viewpoints outside of the vehicle. Furthermore, [19] proposed a novel datasets for learning social dynamics in crowded scenes, which in addition to 11216 pedestrians includes 6364 cyclist tracks [19]. Tracks were collected using a drone at 6 locations with a fixed top-down viewpoint, so unfortunately no road-level images are available, and the variety in traffic environments is limited.

The KITTI dataset offers benchmarks for a wide variety of vision tasks in the intelligent vehicle domain [10], but the raw data only contains 43 labeled cyclist trajectories. The publicly available real-world Tsinghua-Daimler Cyclist (TDC) benchmark [9] offers color and disparity video from a moving vehicle and object bounding boxes. It contains many annotated cyclist tracks, however it currently lacks egomotion estimation and road-layout annotations.

Instead of creating yet another dataset, we extend the existing TDC benchmark. This results in real-world cyclist tracks for which annotated video and disparity data is available to support future research on various behavioral cues, similar to the pedestrian case. Unlike [19], these tracks are obtained from a moving vehicle in on-road traffic and at various crossings, making it directly relevant for path prediction in the intelligent vehicle domain. Our work therefore offers the following contributions:

1) We study cyclist path predictions using probabilistic filters, and extend it to a mixture model. We show that this approach can exploit prior information on the topological road layout.
2) We provide and describe ego-motion compensated cyclist tracks, extracted from the public TDC benchmark [9], and make it available for the community[1].

[1]This dataset is available for non-commercial research purposes. Follow the links from http://www.gavrila.net or contact the authors.

III. CYCLIST TRACK DATASET

The publicly available real-world TDC benchmark [9] is recorded with a stereo-camera setup in a moving vehicle in the Tsinghua city area. It contains annotated bounding boxes for cyclists, together with dense disparity maps of each frame, and camera parameters. Since our objective is to study predictive motion models instead of object detection, we extract ground truth track data from the bounding boxes. Furthermore, all tracks are spatially aligned based on the road topology to ensure that all tracks have a more similar initial state, which could aid in path prediction.

A. Extracting tracks from TDC benchmark

By taking the median disparity in each bounding box, we can obtain 2D ground plane positions (lateral, longitudinal) relative to the egovehicle. Since the annotated bounding boxes also contain track ids, cyclist tracks can be extracted as sequences of 2D positions.

We combine tracks from both the training and test set of the detection benchmark (our experiments will instead use Leave-One-Out cross-validation to separate training and test tracks). In the test set bounding boxes were provided at 5 fps, but in the training set at 2.5 fps. The training tracks are therefore interpolated to 5 fps to ensure constant time intervals for all tracks. All occluded bounding boxes, bounding boxes smaller than 30×30 pixels, or with a distance greater than 60 meters from the egovehicle, are removed.

To learn cyclist motion models, their position and velocities should be expressed in a ground plane coordinate system independent of vehicle egomotion. Unfortunately, the TDC benchmark does not provide egomotion information. We therefore estimate the vehicle egomotion once in an offline process by applying the Iterative Closest Point (ICP) algorithm (using the Point Cloud Library [24]) on the disparity maps of each pair of subsequent frames, and accumulating the resulting 3D transformations. For reference, in recording "2014-11-20_074640" the vehicle starts and finishes at the same spot (i.e. 'loop closure') after driving 1.55 km. The traveled distance found by ICP was 1.55 km, with a deviation of only 12.8 m between start and end point.

B. Aligning tracks with road topology

We have also manually annotated the road layout and intersection topology of the driven routes. This was done by marking points in the video along the centerline of the driven road throughout the sequences, as well as any side- or crossroads that a cyclists' path followed. For each crossing on a cyclists' path, we also label which directions are available and which direction the cyclist actually takes, as five canonical direction classes: a 90° left and right bend, a 45° left and right bend, and straight. We also label which main and sideroad segments each track follows, hence it is known where each track passes a crossing. An example of the annotated scene is shown in fig. 2.

In practice, ADAS could access such information about road layout and intersections from map data. In section IV

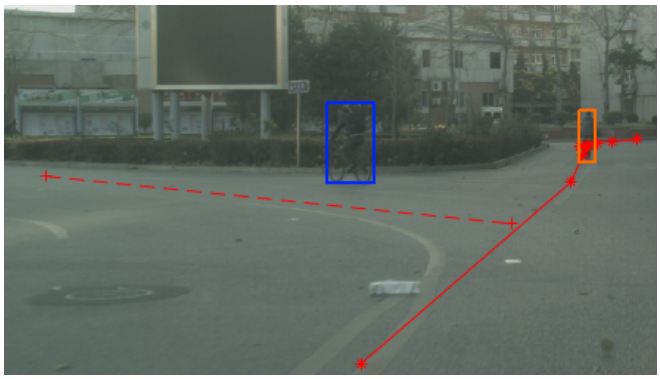


Fig. 2: An example of two cyclists, together with the road annotation. The dotted line shows center lane of the road that the cyclist in the blue rectangle is cycling onto, while the solid line shows the center lane of the road that both the egovehicle and the 'orange' cyclist follow. Every star is an annotated point on the main road, the dashed line is the sideroad annotation.

TABLE I: The total amount of tracks extracted from the dataset. In total, there are 119 tracks, extracted from 108 cyclists.

	90° left	45° left	straight	45° right	90° right
Track count	16	8	68	10	17
Frame count	136	99	1128	135	167

we shall propose a method to exploit this topological prior knowledge for path prediction. We locate all locations where a track follows a bend in the road, either by taking a turn at an intersection, or by following a curved road. If a cyclist track had more than one turn in it, the tracks are cut into two segments, one for each turn. If a cyclist travels straight longer than 10 seconds (50 frames), it is cut into smaller segments, and track segments shorter than 1 second are discarded. For the remainder of this paper, the term 'tracks' shall now refer to the processed track segments.

All tracks are then categorized by the direction that they take on their respective crossings. The total track count per class label is given in table I. There are many more straight than bended tracks, therefore we extract bended tracks from the full TDC benchmark, but straight tracks only from the TDC test set.

Tracks are then aligned with respect to their local road topology through translation and rotation. For all curved tracks, the translation is done based on the intersection point of the center lanes of the incoming and outgoing roads. This point is selected to be the origin point for the curved track. For straight tracks, the origin is the point at the center lane that is closest to the average of the track's start and end position. After the translation, the tracks are rotated such that the direction from the incoming road towards the intersection point is pointed directly upward, when viewed in a 2D x-y graph. The process is illustrated in fig. 3. The resulting spatially aligned real-world tracks are shown in fig. 1.

For temporal alignment, we follow existing literature [3], [14] and express frames in 'time to event' (TTE), where the frame with TTE = 0 is the when the track crosses the line of equal lateral and longitudinal distance to the origin (see dotted lines in fig. 3(right)). Earlier frames have negative TTE (e.g.. cyclist approaching intersection), later frames a positive TTE (e.g. cyclist leaving intersection).

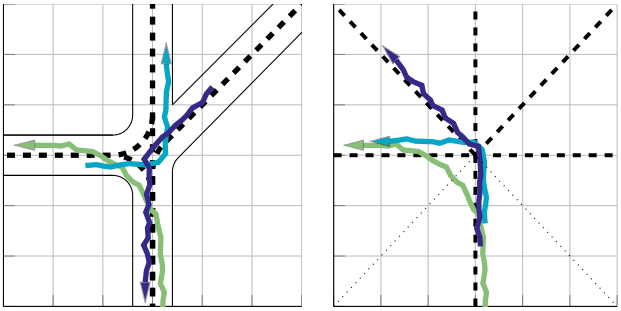


Fig. 3: Three tracks, before (left) and after (right) they have been transformed to the general coordinate system. The general coordinate system ensures a much more similar initial state between all tracks. The frame where a track is closest to the thin dotted line is where the Time To Event (TTE) of that track is defined to be 0.

IV. METHODOLOGY

This paper will compare three probabilistic motion models based on linear dynamics for path predictions. As in [3], [14], observations are filtered online using a recursive Bayesian filter with the selected motion model. At any frame, a predictive distribution for future positions is obtained by executing a filter's 'predict' step several times without any 'update'.

Below, section IV-A will first introduce the considered motion models and explain how to exploit road topology. Then section IV-B will explain how model parameters are estimated from the track data, section IV-C will detail online path prediction. Finally section IV-D defines the metrics on which the learned models are evaluated.

A. Motion models for path prediction

The following probabilistic motion models are considered:

a) Linear Dynamical System (LDS): Previous research on pedestrian path prediction found no significant benefit of higher-order or constant turn motion models over a constant velocity model with white noise acceleration [3]. Hence we pick the constant velocity LDS as our baseline. The used recursive Bayesian filter is therefore a common Kalman filter. The predictive distribution is Gaussian.

b) Uninformed MoLDS (U-MoLDS): Next we consider that the cyclist has a latent *intent* to move into one of the five canonical directions that could occur in the road topology (see section III-B) Given this intent, more specific dynamics might be applied. The baseline model is extended to a Mixture of 5 LDSs (MoLDS), one for each of the five canonical directions.

Since the cyclist's intent is unobserved, it must be estimated online. We place a uniform prior distribution over the intent, and call this model *uninformed* with respect to the latent intent. During online inference, both a distribution on the continuous state and the latent intent is inferred from the past observations.

c) Informed MoLDS (I-MoLDS): The third model is similar to the U-MoLDS, but includes prior information on which canonical directions are present in a track's local road topology (e.g. obtained from map data). Namely, the intent prior is set to zero for road directions that are not in track's local topological layout, the other directions have equal prior probability.

More formally, at every time step t, the model is defined by two variables: the state $x_t \in \mathbb{R}^m$ and the observation $y_t \in \mathbb{R}^n$. Their relations are defined by linear dynamics and Gaussian noise, namely

$$x_t = Ax_{t-1} + B\eta_t \qquad \eta_t \sim \mathcal{N}(\mu_\eta^z, \Sigma_\eta^z) \tag{1}$$

$$y_t = Cx_t + \varepsilon_t \qquad \varepsilon_t \sim \mathcal{N}(0, \Sigma_\varepsilon^z). \tag{2}$$

Here the vector $\eta_t \in \mathbb{R}^{m'}$ is an unknown noise signal affecting the system, and matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{m\times m'}$ define the linear state transition. The measurements $y \in \mathbb{R}^n$ are related to the state through matrix $C \in \mathbb{R}^{n\times m}$. The variables η and ε are Gaussian noise, and assumed to be dependent on the intent $z \in [1, \cdots Z]$ of the cyclist. Here, $\mu_{(\cdot)}^z$ and $\Sigma_{(\cdot)}^z$ show that the process mean and covariance of the noise is dependent on z. In the remainder of this paper, the LDS baseline is considered a special case with only one possible intent, i.e. $Z = 1$, while $Z = 5$ for the U-MoLDS and I-MoLDS.

The observations are positions in 2D, and the model is a constant velocity model. The noise that acts on the system is modeled as an acceleration as given in eq. (3):

$$B = \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & \Delta T & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 & \Delta T \end{bmatrix}^\top. \tag{3}$$

Here, ΔT is the time difference between consecutive frames.

B. Offline parameter learning

For each motion model, its parameters consist of the initial state distribution $(\mu_{x_1}^z, \Sigma_{x_1}^z)$, process noise parameters $(\mu_\eta^z, \Sigma_\eta^z)$, and observation noise Σ_ε^z for each intent z. During training, the model parameters must be determined from the training data, but each track's intent z is set to its class label (i.e. the canonical direction the cyclist actually takes).

Due to the difficulty of obtaining large amounts of track data, rare motion patterns could only have a few examples, and maximum likelihood parameter estimation could overfit the more complex models with more parameters. Therefore, we follow the approach in [5] and use fully Bayesian approximate inference to integrate out the model parameters in our experiments. More precisely, conjugate priors distributions are placed on the parameters for regularization, namely Normal-Inverse-Wishart (NIW) distributions on $(\mu_{x_1}^z, \Sigma_{x_1}^z)$ and on $(\mu_\eta^z, \Sigma_\eta^z)$, and Inverse-Wishart (IW) on Σ_ε^z. Given the training data, Gibbs sampling is used to sample several probable parameter combinations from their joint posterior. The sampling procedure is explained in Appendix A The same priors will be used for all motion models.

C. Online path prediction

For a given set of sampled model parameters, online path prediction can proceed by running 'predict' steps, as outlined in the start of this section. Conditioned on the intent, all models reduce to a Kalman filter for which prediction is straightforward. However, since a track's intent is unknown during test time, we must consider the posterior distribution on z given all past observations.

At every time step t, the posterior state distribution $p(x_t|y_{1:t}, z)$ is computed separately for each LDS z, using Z separate Kalman filters. The posterior on z can be computed from the past observations and the prior distribution $p(z)$,

$$p(z|y_{1:t}) \propto p(y_{1:t}|z)\, p(z)\,. \tag{4}$$

Recall that the U-MoLDS assumes a uniform prior $p(z)$ over all 5 intents, and I-MoLDS over only the possible intents. The LDS has only one intent, which always has probability 1.

The posterior state distribution for time t is a mixture of Z Gaussians, $p(x_t|y_{1:t}) = \sum_{z=1}^{Z} p(x_t|y_{1:t}, z)\, p(z|y_{1:t})$. To get a prediction δ time steps in the future at time t, the Kalman prediction step is applied δ times to all Z filters, each resulting in a predictive distribution $p(x_{t+\delta}|y_{1:t}, z)$. The complete predictive distribution for future time step $t+\delta$ is thus again a mixture of Z Gaussians with weights $p(z|y_{1:t})$:

$$p(x_{t+\delta}|y_{1:t}) = \sum_{z=1}^{Z} p(x_{t+\delta}|y_{1:t}, z)\, p(z|y_{1:t})\,. \tag{5}$$

By applying the observation model once more to each filter, we can also obtain the predictive distribution $p(y_{t+\delta}|y_{1:t})$ for the future observation $y_{t+\delta}$.

D. Evaluation

To evaluate results on this dataset, we apply two metrics. For both metrics, the prediction δ is set to five time steps into the future, which equals one second. The first metric is the Mean Error (ME) between the expected future observations $\mathbb{E}\left[y_{t+\delta}^i | y_{1:t}^i\right]$ and the actual observations. For a track i of length T, let $y_{t+\delta}^i$ be the true future observations, δ time steps ahead of the current time t. For a particular model, a track's ME at a certain time step is then

$$ME_t(i) = |y_{t+\delta}^i - \mathbb{E}\left[y_{t+\delta}^i | y_{1:t}^i\right]|. \tag{6}$$

For this measure, a lower score indicates a better performance. When the ME of entire tracks is given, it shows the average ME of every time step.

The second evaluation metric is the Log-Likelihood (LL) of the predictions, which is a unitless measure, but is indicative of both accuracy and certainty. The LL considers

the probability of the actual observation at time step $t+\delta$. For each track i, the LL is defined as

$$LL_t(i) = \log\big(p\,(y_{t+\delta}|y_{1:t})\big). \tag{7}$$

For this measure, a higher score indicates a better performance. When the LL of entire tracks is given, it shows the summed LL of every time step. The two measures are also shown how they evolve over time, based on their TTE which was explained in section III-B.

To better assess how our model improves path prediction, we also test the underlying assumption that cyclist dynamics are distinct for different intents. The U-MoLDS and I-MoLDS have a separate intent for the five given directions because we expect that these five directions have distinct dynamics. This assumption will be tested as a classification problem: We evaluate how likely each track's observations for each of the $Z = 5$ intents, and comparing the most likely intent to the track's class label. If the assumption of linear dynamics is reasonable, and the dynamics are distinct, then we expect good classification results.

V. EXPERIMENTS

The models explained in section IV-B are trained and evaluated using Leave-One-Out cross-validation with the measures from section IV-D. For each model, and each leave-one-out iteration, the Gibbs sampler was run for 300 iterations, and every tenth parameters sample of the last 100 was selected. We always predict 1 second in the future, $\delta = 5$, and performance measures are computed for all sampled parameters. We then average the performance results of the sampled parameters at each time step.

The same priors are used for all models. The hyperparameters of these prior distributions (see appendix) are shown in eq. (8).

$$\begin{aligned}
\kappa^{z,x_1} &= 1 & \mu^{z,x_1} &= \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^\top \\
\nu^{z,x_1} &= 4 & \Psi^{z,x_1} &= \nu^{z,x_1}\mathrm{diag}\left(\begin{bmatrix}8 & 8 & 0.2 & 0.2\end{bmatrix}\right) \\
\kappa^{z,\eta} &= 1 & \mu^{z,\eta} &= \begin{bmatrix}0 & 0\end{bmatrix}^\top \\
\nu^{z,\eta} &= 4 & \Psi^{z,\eta} &= \nu^{z,\eta}\mathrm{diag}\left(\begin{bmatrix}0.01 & 0.01\end{bmatrix}\right) \\
\nu^{z,\varepsilon} &= 5\times 10^5 & \Psi^{z,\varepsilon} &= \nu^{z,\varepsilon}\mathrm{diag}\left(\begin{bmatrix}0.2 & 0.4\end{bmatrix}\right)
\end{aligned} \tag{8}$$

Here 'diag' is a shorthand for a diagonal matrix with the given entries on the diagonal. The parameter $\Psi^{(\cdot)}$ for both the IW and NIW are given as a matrix, m, multiplied by $\nu^{(\cdot)}$. Interpret these priors as if v samples are a-priori known, and their expected covariance is m.

The values for ν in eq. (8) mean that there is weak prior on initial state distribution ξ^{z,x_1} and the system noise $\xi^{z,\eta}$, while there is a strong prior for the observation noise $\xi^{z,\varepsilon}$. This encodes our belief that different models should have similar observation noise, though their dynamics may be distinct.

A. Model evaluation

We first assess the assumption that the dynamics of each intent are distinct by evaluating the classification performance given all observations. The U-MoLDS, which does not take the road topology into account, classifies 82% of all tracks correctly. To compare this with the I-MoLDS, one should consider that a part of the tracks in the available dataset have only one destination in their road topology, and as such the I-MoLDS cannot fail on these tracks. To make a fair comparison between the classification of the I-MoLDS and U-MoLDS, only the tracks with multiple destinations are considered in table II. On these tracks, the U-MoLDS classifies 76% correctly which shows that it is reasonable to assume the dynamics are distinctive for their respective intent. However, the I-MoLDS classifies 90% correctly, which means the model can still benefit from additional prior knowledge.

TABLE II: The confusion matrix for all tracks with multiple destinations. The value on the left/right shows the result for the U-MoLDS/I-MoLDS, respectively. The **bold** values highlight the best scoring model. Overall, the U-MoLDS classifies 76% correctly, whereas the I-MoLDS classifies 90% correctly.

		Estimate				
		90° left	45° left	straight	45° right	90° right
Ground truth	90° left	14 / **17**	3 / **0**	0 / 0	0 / 0	0 / 0
	45° left	0 / 0	2 / 2	0 / 0	0 / 0	0 / 0
	straight	**1** / 2	2 / **1**	13 / 13	1 / **0**	**0** / 1
	45° right	1 / 1	0 / 0	1 / 1	3 / **6**	3 / **0**
	90° right	**0** / 1	0 / 0	0 / 0	3 / **0**	13 / **15**

TABLE III: The average ME in meters over all tracks, grouped by true class label. The best performance is shown in **bold**.

	90° left	45° left	straight	45° right	90° right
LDS	1.75	1.15	**1.19**	1.23	2.36
U-MoLDS	1.59	1.11	1.38	1.16	1.99
I-MoLDS	**1.51**	**1.10**	1.20	**1.08**	**1.88**

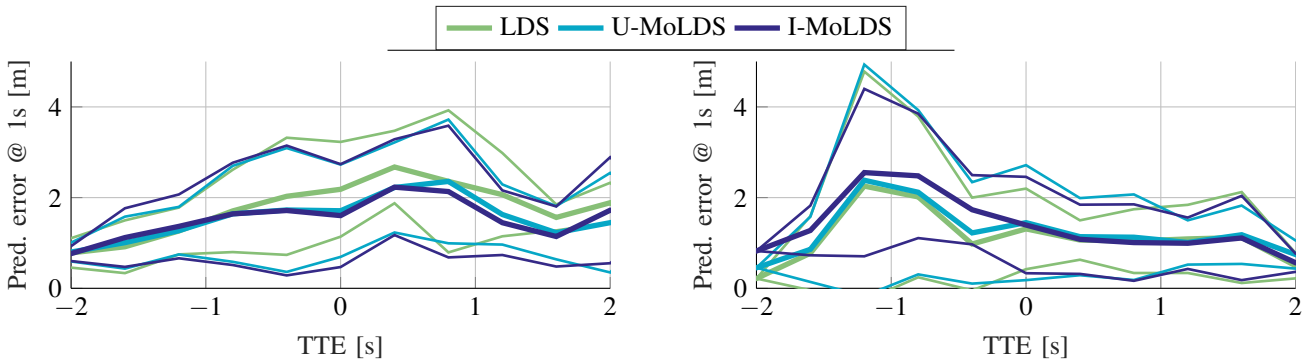
B. Path Prediction

Path prediction is evaluated on the two metrics explained in section IV-D, with the results shown in tables III and IV. The I-MoLDS has the lowest ME for all intents except straight. This is most evident for the 90 degree angles, where the average error decreases by 24 cm (14%) and 48 cm (20%) for left and right, respectively. On straight tracks, the LDS outperforms the I-MoLDS, although only minimally (1%).

On the LL, the I-MoLDS performs best on all directions, except for 45 degree turns to the left. Here, the U-MoLDS performs best. Furthermore, when the I-MoLDS performs best, it outperforms the LDS by a large margin, whereas the difference in performance is not so large for the 45 degree turn to the left. A closer inspection of the likelihoods also shows that where the LL for 90 degrees left and right are

TABLE IV: The mean LL for all tracks, grouped by true class label. The best performance is shown in **bold**.

	90° left	45° left	straight	45° right	90° right
LDS	-26.66	-27.27	-29.35	-21.91	-24.09
U-MoLDS	-21.72	**-26.57**	-26.24	-21.93	-19.99
I-MoLDS	**-20.62**	-28.05	**-23.65**	**-20.78**	**-19.73**



(a) Prediction error over time on 90° left turn.

(b) Prediction error over time on 45° left turn.

Fig. 4: The mean error (thick line) and standard deviation(thin line) over time for all tracks, with respect to the moment they were predicted. The tracks turning 90 degrees to the left are shown in (a). The tracks turning 45 degrees to the left that show an anomaly (see text) are shown in (b).

roughly the same, there is a large discrepancy between the LL of 45 degrees left and right.

A more complete picture is painted by plotting the error over time. Figure 4a shows the error over time for tracks bending at a 90 degree angle to the left. At $TTE = -1$ s, where the models are predicting the state for $TTE = 0$ s, the performance diverges. This shows that the I-MoLDS can predict the change in dynamics that is related to the 90 degree turns. Consequently, the I-MoLDS improves at a time where it matters the most. The same result was seen for the 90 degree turns to the right. The results for all classes can be found in the appendix.

The same cannot be said for the 45 degree left turn, however, as is shown in fig. 4b. For this angle, the LDS and U-MoLDS show more accurate predictions. A large spike in the standard deviation can also be seen around this time, indicating that the large error is not present for all tracks. Together, this indicates that there are tracks present in the 45 degrees left class whose dynamics are not represented by the others in their group during training. This is further illustrated by the anomaly seen in the LL in table IV, where, even though the I-MoLDS did not perform well on the 45 degree left turn, neither did the others. This suggests that more data is needed for this class.

VI. CONCLUSIONS

This paper presented a complementary dataset to the Tsinghua-Daimler Cyclist Benchmark. On this dataset, we trained a Mixture of Linear Dynamical Systems that can take the road topology into account. The prediction of the cyclist position one second into the future is shown to be comparable to a Linear Dynamical System on straight tracks, or improve 20% on average on sharp turns to the right. Because this dataset adds to an existing dataset, future research can consider incorporating the visual features from the video frames to further improve prediction.

APPENDIX

A. Gibbs sampling

The distributions over the unknown initial state, $\mu^z_{x_1}$ and $\Sigma^z_{x_1}$, and each type of dynamics, μ^z_η and Σ^z_η, have a prior NIW distribution, eqs. (9) and (10). The observation noise Σ^z_ε has an IW distribution as prior, eq. (11).

$$\left\{\mu^z_{x_1}, \Sigma^z_{x_1}\right\} \sim \mathcal{NW}^{-1}(\xi^{z,x_1}) \tag{9}$$

$$\left\{\mu^z_\eta, \Sigma^z_\eta\right\} \sim \mathcal{NW}^{-1}(\xi^{z,\eta}) \tag{10}$$

$$\Sigma^z_\varepsilon \sim \mathcal{W}^{-1}(\xi^{z,\varepsilon}) \tag{11}$$

The NIW and IW distributions are parametrized by $\xi^{(\cdot)} = \left\{\mu^{(\cdot)}, \kappa^{(\cdot)}, \Psi^{(\cdot)}, \nu^{(\cdot)}\right\}$ and $\xi^{(\cdot)} = \left\{\Psi^{(\cdot)}, \nu^{(\cdot)}\right\}$ respectively. The advantage of these two distributions is that when they are updated with new measurements, their posterior is the same type of distribution. So, from the prior $\xi^{(\cdot)}_-$, one can compute the posterior $\xi^{(\cdot)}_+$ after taking N more samples $(q_1, \ldots, q_N)$ from the normal distribution as given in eqs. (12) to (14). Here, $\bar{q}$ is the mean of all samples, and S is the scatter matrix created from all samples.

$$\mu^{(\cdot)}_+ = \frac{\kappa^{(\cdot)}_- \mu^{(\cdot)}_- + N\bar{q}}{\kappa^{(\cdot)}_- + N}, \qquad \kappa^{(\cdot)}_+ = \kappa^{(\cdot)}_- + N \tag{12}$$

$$\Psi^{(\cdot)}_+ = \Psi^{(\cdot)}_- + S + \frac{\kappa^{(\cdot)}_- N}{\kappa^{(\cdot)}_- + N}\left(\bar{q} - \mu^{(\cdot)}_-\right)\left(\bar{q} - \mu^{(\cdot)}_-\right)^\top \tag{13}$$

$$\nu^{(\cdot)}_+ = \nu^{(\cdot)}_- + N \tag{14}$$

The intuitive explanation of the parameters is that $\Psi^{(\cdot)}$ is the sampled scatter matrix, taken from $\nu^{(\cdot)}$ samples. Similarly, $\mu^{(\cdot)}$ is the sampled mean, taken from $\kappa^{(\cdot)}$ samples. For the IW distribution, the same equations apply but it could be said that the prior taken samples $\kappa^{(\cdot)}$ of the mean $\mu^{(\cdot)}$, is infinite, thereby ensuring that additional measurements will not change the mean $\mu^{(\cdot)}$. In this paper, an IW or NIW distribution that is updated with additional measurements

will be written as a function of both its initial parameters $\xi^{(\cdot)}$ and the measurements $\varepsilon_{(\cdot)}$, e.g. $\mathcal{W}^{-1}(\xi_0^{z,\varepsilon}, \varepsilon_t)$ is the prior IW distribution of the observation noise covariance matrix Σ_ε^z that is parametrized by $\xi_0^{z,\varepsilon}$, updated with an additional sample of the observation noise, ε_t.

To reiterate, samples from the observation noise ε, the system noise η, and initial state distribution x_1 can be used to improve the distribution over their covariance and, for the system noise and initial state, their mean. However, it is not possible to sample from these directly because the true state is not known. Here we use Gibbs sampling to generate the posterior distributions, as is also done in [5]. An overview is given in algorithm 1, which is applied for each type of dynamics z, with its own example tracks.

Algorithm 1 The sampling algorithm.

Require: $\xi_0^{z,\eta}$, ξ_0^{z,x_1} and $\xi_0^{z,\varepsilon}$
  Sample initial covariances and means.
  $\Sigma_\eta^z, \mu_\eta^z \leftarrow \mathcal{NW}^{-1}(\xi_0^{z,\eta})$
  $\Sigma_{x_1}^z \mu_{x_1}^z \leftarrow \mathcal{NW}^{-1}(\xi_0^{z,x_1})$
  $\Sigma_\varepsilon^z \leftarrow \mathcal{W}^{-1}(\xi_0^{z,\varepsilon})$
  repeat
    for Each track i **do** $p(x_{1:T}|y_{1:T})$.
      Sample state from the posterior
      $x_{1:T}^i \leftarrow p(x_{1:T}|y_{1:T})$
      From eq. (15)
      $\eta_{1:T-1}^i \leftarrow B^+(x_{2:T} - Ax_{1:T-1})$
      $\varepsilon_{1:T}^i \leftarrow y_{1:T} - Cx_{1:T}$
    end for
    Update the inverse Wishart distributions using their initial distributions.
    $\mathcal{NW}^{-1}(\xi_+^{z,\eta}) \leftarrow \mathcal{NW}^{-1}(\xi_0^{z,\eta}, \eta_{1:T-1}^{1:N_{tracks}})$
    $\mathcal{NW}^{-1}(\xi_+^{z,x_1}) \leftarrow \mathcal{NW}^{-1}(\xi_0^{z,x_1}, x_1^{1:N_{tracks}})$
    $\mathcal{W}^{-1}(\xi_+^{z,\varepsilon}) \leftarrow \mathcal{W}^{-1}(\xi_0^{z,\varepsilon}, \varepsilon_{1:T}^{1:N_{tracks}})$
    Resample the covariances and means
    $\Sigma_\eta^z, \mu_\eta^z \leftarrow \mathcal{NW}^{-1}(\xi_+^{z,\eta})$
    $\Sigma_{x_1}^z \mu_{x_1}^z \leftarrow \mathcal{NW}^{-1}(\xi_+^{z,x_1})$
    $\Sigma_\varepsilon^z \leftarrow \mathcal{W}^{-1}(\xi_+^{z,\varepsilon})$
  until Satisfied

Initially, there is some prior knowledge over the IW and NIW distributions, given by $\xi_0^{z,\eta}$, ξ_0^{z,x_1} and $\xi_0^{z,\varepsilon}$. A random sample from the prior distributions is an initial estimate for the entire model. The initial estimate, together with the observations $y_{1:T}$ from each existing track, can give a posterior distribution on the state of the system through Kalman smoothing: $p(x_{1:T}|y_{1:T})$ is known. If the exact state is known at every time step, it is possible to recover the system noise $\eta_{1:T}$ and observation noise $\varepsilon_{1:T}$ by eq. (15), a direct result from eqs. (1) and (2).

$$\eta_t = B^+(x_{t+1} - Ax_t), \qquad \varepsilon_t = y_t - Cx_t \tag{15}$$

However, as stated, at each time t only the distribution of the state is known, and not the actual state. To circumvent this, Gibbs sampling is used again: sample a random potential state sequence from each track. The sampled state sequence uniquely defines a sampled system noise sequence and an observation noise sequence, which is used to update the distribution of the system noise and observation noise, respectively. Similarly, the initial sample of each sequence is used to update the distribution of the initial state.

A new sample is taken from each distribution to select a system, observation and initial state covariance and system and initial state mean, and the algorithm is repeated.

The algorithm is expected to create a random sample of the distribution of the unknown means and covariances, after the algorithm has gone through a certain "burn-in" period. After the burn-in period, a more robust set of means and covariances can be retrieved by averaging the results of multiple iterations.

ACKNOWLEDGMENT

The research leading to the results of this work has received funding from the European Community's Eighth Framework Program (Horizon2020) under grant agreement n° 634149, the PROSPECT project.

REFERENCES

[1] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Trans. on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, March 2016.

[2] I. Cara and E. de Gelder, "Classification for safety-critical car-cyclist scenarios using machine learning," in *Proc of the IEEE ITSC*. IEEE, 2015, pp. 1995–2000.

[3] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive Bayesian filters: A comparative study," in *Proc. of the GCPR*. Springer Berlin Heidelberg, 2013, pp. 174–183.

[4] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? A study on pedestrian path prediction," *IEEE Trans. on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[5] J. F. P. Kooij, N. Schneider, and D. M. Gavrila, "Analysis of pedestrian dynamics from a vehicle perspective," in *Proc. of the IEEE IV*, 2014, pp. 1445–1450.

[6] B. Völz, H. Mielenz, R. Siegwart, and J. Nieto, "Predicting pedestrian crossing using quantile regression forests," in *Proc. of the IEEE IV*. IEEE, 2016, pp. 426–432.

[7] L. Huang and J. Wu, "Cyclists' path planning behavioral model at unsignalized mixed traffic intersections in china," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 13–19, 2009.

[8] S. Zernetsch, S. Kohnen, M. Goldhammer, K. Doll, and B. Sick, "Trajectory prediction of cyclists using a physical model and an artificial neural network," in *Proc. of the IEEE IV*, June 2016, pp. 833–838.

[9] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila, "A new benchmark for vision-based cyclist detection," in *Proc. of the IEEE IV*, June 2016, pp. 1028–1033.

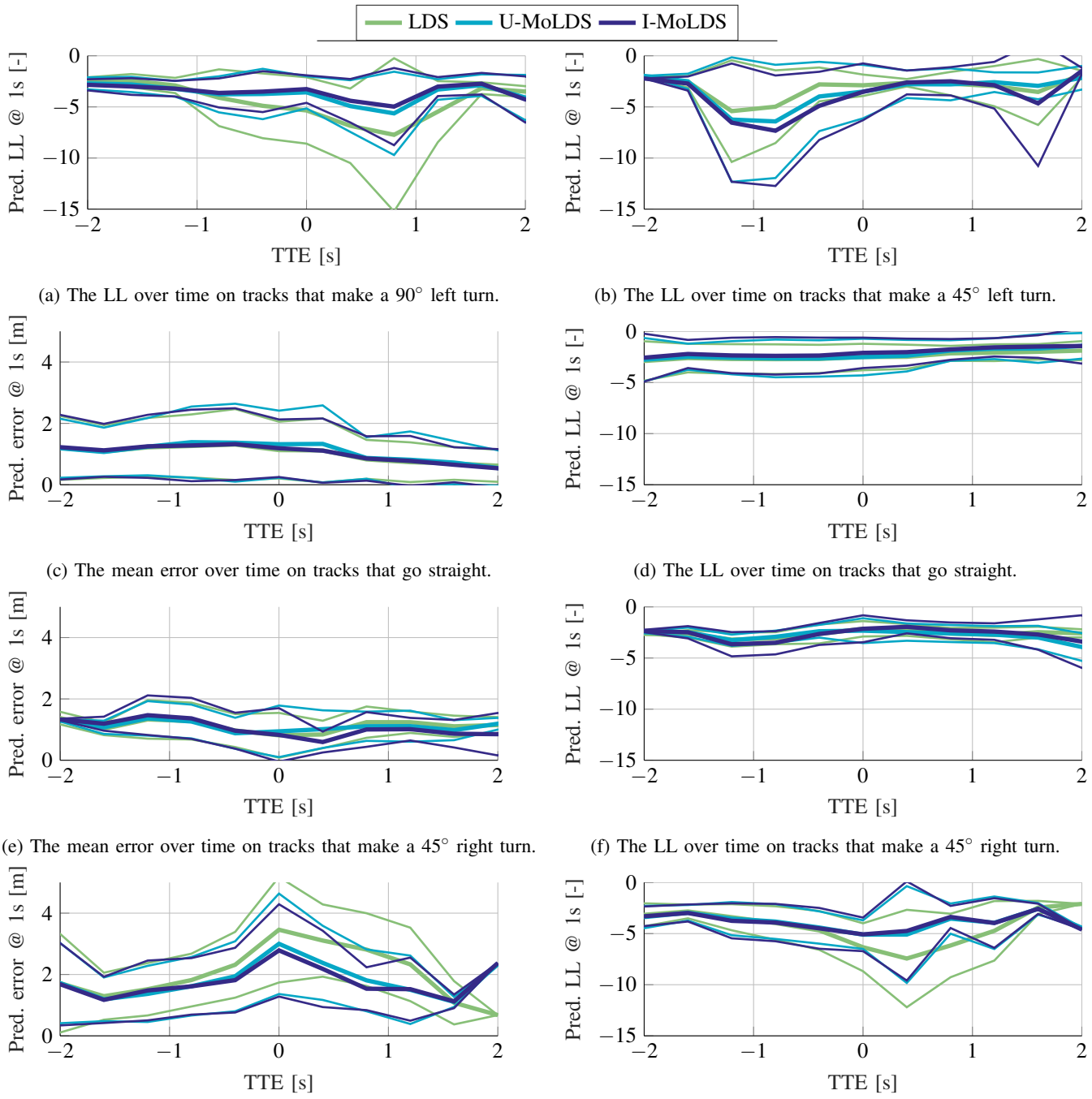
[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[11] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. on PAMI* , vol. 33, no. 11, pp. 2287–2301, 2011.

[12] J. F. P. Kooij, G. Englebienne, and D. M. Gavrila, "Mixture of switching linear dynamics to discover behavior patterns in object tracks," *IEEE Trans. on PAMI* , vol. 38, no. 2, pp. 322–334, 2016.

[13] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, "Surround vehicles trajectory analysis with recurrent neural networks," in *Proc of the IEEE ITSC*, 2016, pp. 2267–2272.

[14] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *Proc. of the ECCV*. Springer International Publishing, 2014, pp. 618–633.



(a) The LL over time on tracks that make a 90° left turn.

(b) The LL over time on tracks that make a 45° left turn.

(c) The mean error over time on tracks that go straight.

(d) The LL over time on tracks that go straight.

(e) The mean error over time on tracks that make a 45° right turn.

(f) The LL over time on tracks that make a 45° right turn.

(g) The mean error over time on tracks that make a 90° right turn.

(h) The LL over time on tracks that make a 90° turn to the right.

[15] A. T. Schulz and R. Stiefelhagen, "A controlled interactive multiple model filter for combined pedestrian intention recognition and path prediction," in *Proc of the IEEE ITSC*. IEEE, 2015, pp. 173–178.

[16] M. Roth, F. Flohr, and D. M. Gavrila, "Driver and pedestrian awareness-based collision risk analysis," in *Proc. of the IEEE IV*, June 2016, pp. 454–459.

[17] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. of the IEEE ICCV*, 2009, pp. 261–268.

[18] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. of the IEEE CVPR* , 2016, pp. 961–971.

[19] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. of the ECCV*. Springer, 2016, pp. 549–565.

[20] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. of the ECCV*. Springer Berlin Heidelberg, 2012, pp. 201–214.

[21] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *IEEE ICRA*. IEEE, 2016, pp. 2543–2549.

[22] E. Rehder and H. Kloeden, "Goal-directed pedestrian prediction," in *Proc. of the IEEE ICCV Workshops*, 2015, pp. 50–58.

[23] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE Trans. on PAMI* , vol. 36, no. 5, pp. 1012–1025, 2014.

[24] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE ICRA*, Shanghai, China, May 9-13 2011.