2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Visual vehicle tracking through noise and occlusions using crowd-sourced maps

Suraj M S[1,2], Hugo Grimmett[1], Lukáš Platinský[1] and Peter Ondrúška[1]

***Abstract*— We present a location-specific method to visually track the positions of observed vehicles based on large-scale crowd-sourced maps.**

We equipped a large fleet of cars that drive around cities with camera phones mounted on the dashboard, and performed city-scale structure-from-motion to accurately reconstruct the trajectories taken by the vehicles. We show that these data can be used to first create a system enabling high-accuracy localisation, and then to accurately predict the future motion of newly observed cars in the camera view. As a basis for the method we use a recently proposed system [1] for unsupervised motion prediction and extend it to a real-time visual tracking pipeline which can track vehicles through noise and extended occlusions using only a monocular camera.

The system is tested using two large-scale datasets of San Francisco and New York City containing millions of frames. We demonstrate the performance of the system in a variety of traffic, time, and weather conditions. The presented system requires no manual annotation or knowledge of road infrastructure. To our knowledge, this is the first time a perception system based on a large-scale crowd-sourced maps has been evaluated at this scale.

I. Introduction

A fundamental task of robotics perception and planning in dynamic environments is the ability to predict the future evolution of the situation around the robot. For example, a self-driving car needs to know about the positions of other cars and their future motion to plan and avoid collisions.

These predictions are usually based on assumed motion dynamics of the vehicles around the car such as using Kalman Filter. A common disadvantage of these models is that the assumed model can generalise badly to the vast complexity of the real world such as complex intersections or turns. The exhibited motion of vehicles in these situations can often not be reliably predicted using simple motion models, such as linear extrapolation, especially if the prediction horizon is longer than a few seconds. A different approach is to annotate road infrastructure as a semantic map capturing traffic rules. This has the benefit that it can extrapolate the expected motion of a car assuming its motion follow the captured rules. However, this approach requires a large amount of work to constantly annotate and update the map such that it remains reliable despite environmental change.

In this work we propose an alternative approach. Inspired by the impact of large-scale datasets in computer vision [2] we utilise a large amount of crowd-sourced high-quality motion data to drive the motion prediction. We collected it by equipping a large fleet of cars with cameras and performing structure-from-motion at city-scale to accurately reconstruct their trajectories.

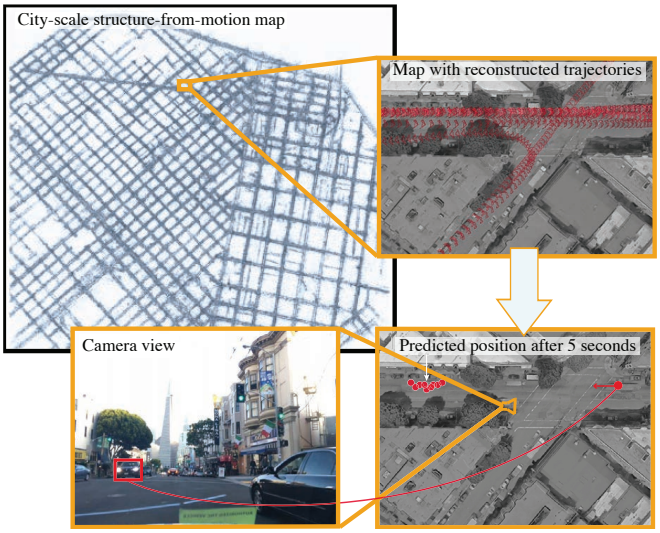


Fig. 1: Our system combines city-scale crowd-sourced map and localisation system to predict motion and track surrounding vehicles.

We use these data to draw samples from the underlying motion distribution for that particular place, and show how these data can be used for predicting the future motion of newly observed cars [1]. It has the benefit of requiring no human annotation while implicitly capturing modelled and unmodelled aspects of the vehicle motion, scaling to large city-scale scenarios and improving with time as the amount of data increases. The proposed system can be used universally as a motion-prediction step in various vehicle-tracking systems for the purpose of vehicle safety and autonomy. We show it can be easily integrated with a camera-equipped vehicle and large-scale, high-accuracy localisation system to create a place-specific 3D tracking pipeline.

We evaluate the method on two city-scale datasets specifically collected for this experiment. We show the system can be used at large scale to drive motion prediction and tracking at large scale in a variety of traffic and environmental conditions. To our knowledge this is the first published approach to be constructed and evaluated at this scale. Specifically, we present:

- A visual pipeline for tracking car positions around the vehicle using a monocular camera, based on crowd-

[1]Blue Vision Labs, London, UK
{mssuraj,hugo,lukas,peter}@bluevisionlabs.com
[2]Georgia Institute of Technology, Atlanta, US
mssuraj@gatech.edu

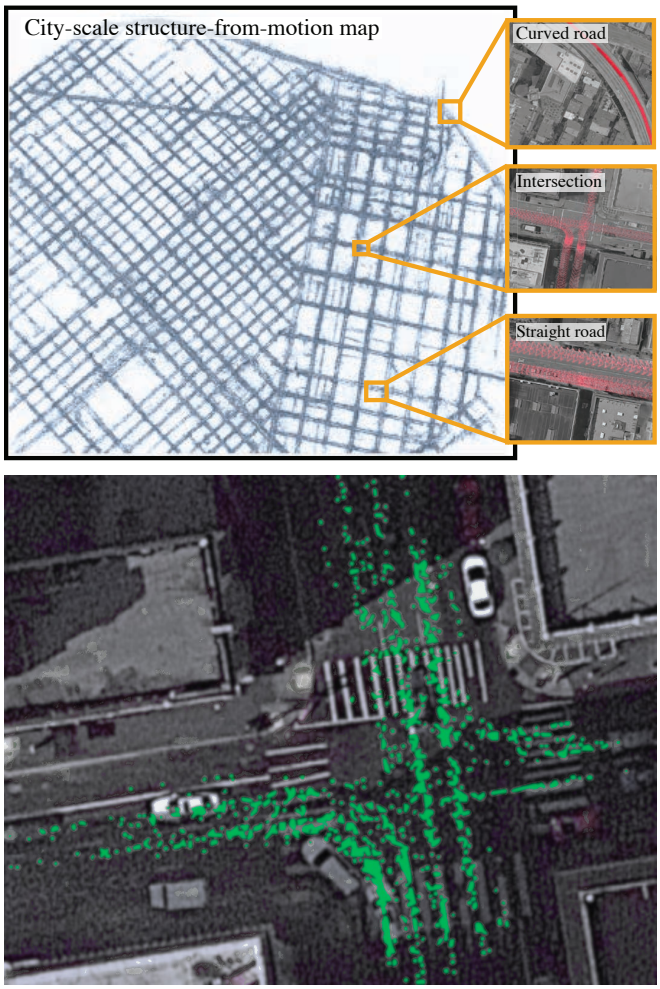


Fig. 2: Examples of the prior trajectories in San Francisco dataset as generated by a fleet of camera-equipped cars and reconstructed by large-scale structure-from-motion. The resolution of the prior on well-travelled intersections allows to accurately distinguish individual lanes and possible turns from the exhibited patterns.

sourced large-scale motion prior

- A comprehensive evaluation on two city-scale datasets in San Francisco and New York containing millions of samples
- A performance evaluation of the proposed visual tracking pipeline and comparison against a baseline model.

The rest of this paper is organised as follows. In the following section we revisit some of the work done in the area of motion prediction, object tracking and on using large-scale datasets. In Section III we summarise a recently proposed work [1] of how an unstructured motion prior can be constructed and used to accurately predict car's position in the future. Then, in Section IV-A we present a method combining this motion prior with a monocular sensor to create a vehicle-tracking pipeline. We evaluate the method in Section V and conclude in Section VI.

II. Related Work

Vehicle trajectory estimation and urban 3D tracking are active areas of study in computer vision and robotics. Various methods have been proposed over years to understand and model vehicle motion dynamics, driver intent and vehicle interactions with the environment and neighboring agents.

Usually, motion prediction involves relying fully or partly on a vehicle dynamics model. The authors of [3] compare and evaluate several motion models for tracking vehicles. These models are usually combined with Kalman filtering [4] or Bayesian filtering [5] for path prediction. However, these approaches are only able to perform predictions in a very short window into the future.

In [6] the authors combine a constant yaw-rate and acceleration model with a maneuver classifier to predict vehicle trajectories. But their methods are restricted to limited scenarios and constrained by the number of maneuvers.

Recently, the focus has shifted to data-driven approaches to learn vehicle dynamics rather than explicitly crafting them. These usually employ dynamic Bayesian networks [7], Gaussian mixture models [8], [9], [10], hidden Markov models [9], neural networks [11], [12], [13] or a combination of these. These achieve better performance than pure vehicle dynamics based approaches. However, they are either trained for specific scenarios like highways or tend to learn a general model that do not utilise environment-specific cues such as traffic pattern in the area, changes in the environment structure, etc.

Our approach, on the other hand, utilises location specific information for accurate predictions. Instead of learning a global model, we rely on the historical vehicle trajectories in the locality to perform on-the-fly prediction. Additionally, our system decouples the prediction system and environment knowledge thereby enabling easy update to environment priors.

In [14] the authors propose the use of kernel density estimation to measure similarities between a test trajectory and the set of all past trajectories in a dataset. Although in spirit they rely on using previous trajectory history, their approach involves learning a prediction model by comparing a query trajectory across the whole space of full-length past trajectories and is therefore not scalable. Instead, we use a simpler similarity measure over individual positions and poses in a local region to perform prediction.

Another related domain is the use of environment cues for 3D tracking. Such methods often rely on 3D scene analysis to augment tracking. In [15] the authors reason about 3D scene layout and object positions at urban intersections while the authors of [16] perform 3D object tracking by enforcing scene geometry and 3D dynamics based constraints. In [17] the ground plane and 3D location priors are used to obtain 3D object detections. However they do not perform 3D tracking and their ground plane assumption fails in real driving scenarios involving up-hill and down-hill slopes. The authors of [18] present a system for 3D tracking from moving vehicles in a modular way similar to ours by using global

trajectory hypotheses generated from a motion model and geometric constraints in the scene. However, their approach relies on a calibrated stereo rig mounted on the car while we rely only on a monocular camera. In addition, they do not use large-scale ground and motion priors like us.

To the best of our knowledge, this is the first work that proposes the use of large-scale environment priors for urban vehicle tracking in city-scale.

III. Large-Scale Motion Prior

In this section we summarise the method of [1] which predicts a car's future position in 3D space given its currently observed position, using a motion prior G in the area. This method is then used as a component in the novel full visual-tracking pipeline described in the next section. Formally, given the car's observed state $s_0 = (x_0, r_0, v_0)$ consisting of position $x_0 \in R^3$, rotation $r_0 \in SO(3)$ and velocity $v_0 \in R^3$ we aim to predict its state after $t > 0$ seconds: $p(s_t|s_0, G)$.

The considered motion prior consists of a large set of highly-accurately localised individual trajectories of the crowd-sourcing fleet through the area

$$G = \{G^1, G^2, \ldots, G^N\}, \tag{1}$$

where trajectory $G^i = \{s_1^i, s_2^i, \ldots, s_m^i\}$ is a sequence of observed position, rotation and velocity of the car at regular intervals $t = \{1, 2, 3, \ldots\}$.

This prior can be automatically extracted by performing a large-scale structure-from motion using pictures captured by the vehicle phone camera mounted on the dashboard. This is the sole source of input and we don't use any explicit annotation about the environment or traffic rules. Examples of the car trajectories are displayed in Figure 2. As shown, the resolution of the prior allows to distinguish individual lanes and possible turns directly from the exhibited patterns.

Algorithm 1 Motion prior sampling

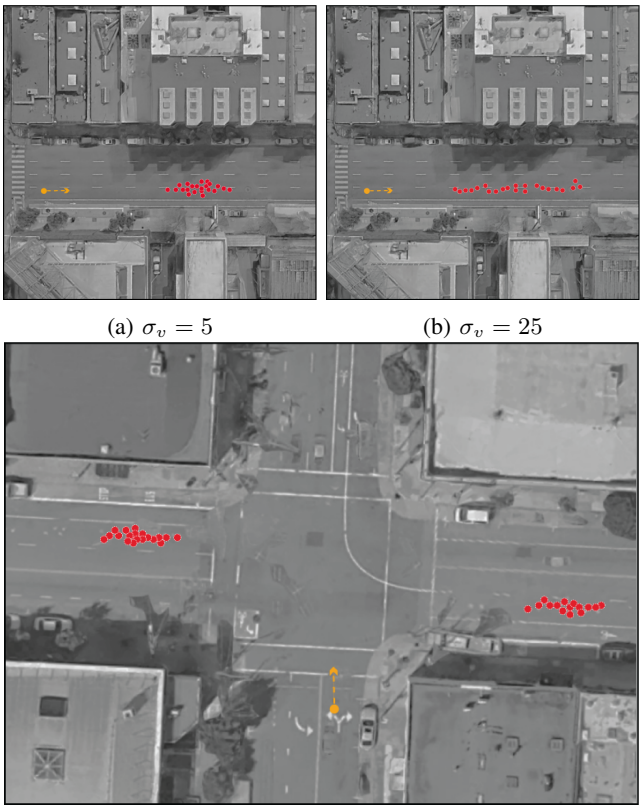
Input

s_0: initial state (position, rotation, velocity)
t: time horizon
G: motion prior

Output

$\hat{s}_{1:N}$: samples of the predicted future car state

Compute prior state relevance
$Z \leftarrow \sum_{i,j} K(s_i^j, s_0)$
$\mu_i^j \leftarrow \frac{1}{Z} K(s_i^j, s_0)$
Sample future state
for $k = 1, 2, 3, \ldots, N$ **do**
Sample prior state according to relevance
    $s_j^i \leftarrow$ *MultinomialSample*(G, μ)
Sample future state
    $\hat{s}_k \leftarrow s_{j+t}^i + \epsilon$
end for
return $\hat{s}_{1:N}$



(a) $\sigma_v = 5$ (b) $\sigma_v = 25$

Fig. 3: Vehicle motion predictions at intersections. The orange icon represents the query position, pose and velocity at time t. The red dots represent the distribution of predicted samples at $t + 5$. Note that the road ahead is a one-way route in opposite direction. Our prior implicitly capture this information without any manual annotation.

To predict the future position of a vehicle at time t we assume a hypothesis that the car is following the same trajectory pattern as one of cars in the past at the same location. Specifically, for each prior pose s_j^i we assume a hypothesis the vehicle is going to follow the same motion pattern starting at that pose. Under this assumption the pose of the car in the future is going to be

$$s_t = s_{j+t}^i + \epsilon, \tag{2}$$

where s_{j+t}^i is the observed pose after time t and ϵ is a random noise modelling the fact the trajectory can slightly differ.

The distribution of the future pose is then a weighted sum of these individual distributions

$$p(s_t|s_0, G) = \frac{1}{Z} \sum K(s_j^i, s_0) p(s_t|s_{j+t}^i, \epsilon), \tag{3}$$

where Z is a normalisation factor

$$Z = \sum K(s_j^i, s_0), \tag{4}$$

and $K(s_j^i, s_0)$ measures the similarity of the prior pose to the current pose expressing the likelihood it can indeed follow the exhibited motion pattern.

This likelihood is modelled as the sum of similarities of individual factors

$$K(s_j^i, s_0) = \exp\{-\frac{\|x_j^i - x_0\|^2}{\sigma_x^2} - \frac{\|r_j^i - r_0\|^2}{\sigma_r^2} - \frac{\|v_j^i - v_0\|^2}{\sigma_v^2}\} \tag{5}$$

where $\|x_j^i - x_0\|^2$ is the euclidean distance of the sample in the 3D space, $\|r_j^i - r_0\|^2$ is the relative heading angle difference and $\|v_j^i - v_0\|^2$ is the difference in the linear speed. The parameters σ_x, σ_r and σ_v model relevance of individual components.

We evaluate the probability density function $p(s_t|s_0, G)$ explicitly and use an efficient sampling procedure detailed in Algorithm 1.

In Figure 3 we show some samples drawn from the distribution S_t. The sampling follows the previously observed trajectories of prior motion in the area while parameters σ model relevance of the individual components of the state. Small σ_v results in predictions matching the current velocity value, while large σ_v results in predictions sampled using all previously observed initial velocities. This could be useful if the initial velocity is uncertain or not known (such as when the car is observed for the first time).

The following section shows how this method can be effectively used in a pipeline tracking poses of cars surrounding a camera-equipped vehicle.

IV. Visual Tracking Pipeline

In this section we incorporate the single-shot motion prediction system from the previous section into a novel fully-fledged pipeline that continuously tracks the positions of nearby cars around a camera-equipped vehicle. This can be used as a situation-awareness module in planning algorithms to predict and react to the motion of the other traffic participants. We combine a high-accuracy localisation subsystem, a convolutional neural network-based car detector, and the proposed motion prediction subsystem.

Although the presented method specifically uses monocular cameras, it should be straightforward to generalise the method to other hardware configurations using LIDAR, radar or stereo cameras as well. We choose to showcase this configuration as it is, arguably, not only the most prevalent and cost-effective hardware platform, but also the most difficult for implementation due to the missing depth perception of LIDARs or stereo cameras. We show that using the motion prior alleviates this problem and helps to predict the correct motion with excellent results.

A. Pipeline overview

The input to the system is a live stream of images $I_1, I_2, I_3, ...$ captured at regular intervals Δt by a camera mounted on a moving vehicle. The algorithm processes this stream iteratively frame by frame and in each step produces a set of 3D positions and velocities of visible vehicles $s_t^1, s_t^2, ..., s_t^n$ and their 2D observations $c_t^1, c_t^2, ..., c_t^n$.

First, for each new image I_t we determine its exact pose $q_t \in SE(3)$ in the 3D space. Although large-scale visual localisation is a challenging topic, as discussed in Section V this can be done very efficiently by performing a feature-based visual localisation using the same structure-from-motion 3D map constructed to extract the prior motion. This guarantees that the captured image pose is accurately aligned with respect to the prior motion samples in the area necessary for 2D-3D association described later.

Second, each image is processed by a convolutional neural network to produce a list of vehicle observations $c_t^1, c_t^2, ..., c_t^n$ in the form of 2D bounding boxes and a confidence distribution over the object categories. For our implementation, we use a standard Faster-RCNN object detector and consider only cars detected above a certain threshold.

Next, each observation can be a part of an existing track (seen before at time $t' > t - T$ where T is the tracking window) or a new track. For each c_t^i and $c_{t'}^j$ we consider a hypothesis that we observed the same vehicle and consider its previous position $s_{t'}^j$ and likely motion induced by prior G. Similarly, we consider another hypothesis that we observed a new vehicle. Details of this step are described below.

Finally, for each c_t^i we choose the most likely candidate hypothesis and the associated estimated pose s_t^1. The entire algorithm is summarised in Algorithm 2.

B. Frame-to-frame association

As described in Section IV-A, for each detected car c_t^i we consider a hypothesis p^{ij} that it is an observation of the same previously detected vehicle $c_{t'}^j$. For each such hypothesis we compute the most probable 3D pose and velocity supporting this hypothesis s_t^{ij}:

$$s_t^{ij} = \underset{s_t^i}{\operatorname{argmax}}\, p(s_t^i, c_t^i | c_{t'}^j, s_{t'}^j, q_t, G) \tag{6}$$

This probability can be factorised as

$$p(s_t, c_t | c_{t'}, s_{t'}, q_t, G) \propto p(c_t|c_{t'}) p(s_t|c_t, q_t) p(s_t|s_{t'}, G), \tag{7}$$

where

- $p(c_t|c_{t'})$ is the similarity in visual appearance,
- $p(s_t|c_t)$ is the consistency of the observed vehicle in the 2D image and its position in 3D space, and

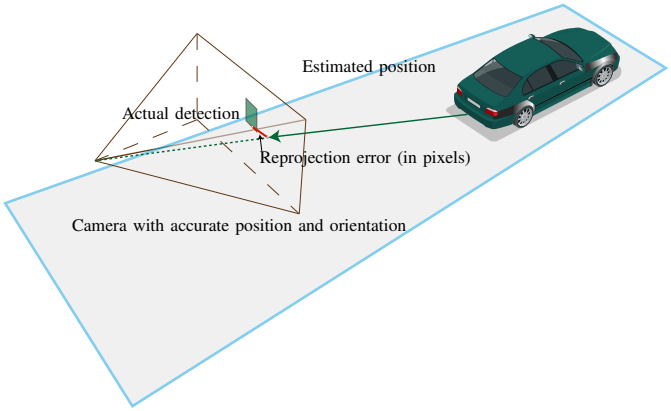


Fig. 4: The re-projection term models the consistency between 3D pose and 2D detection.

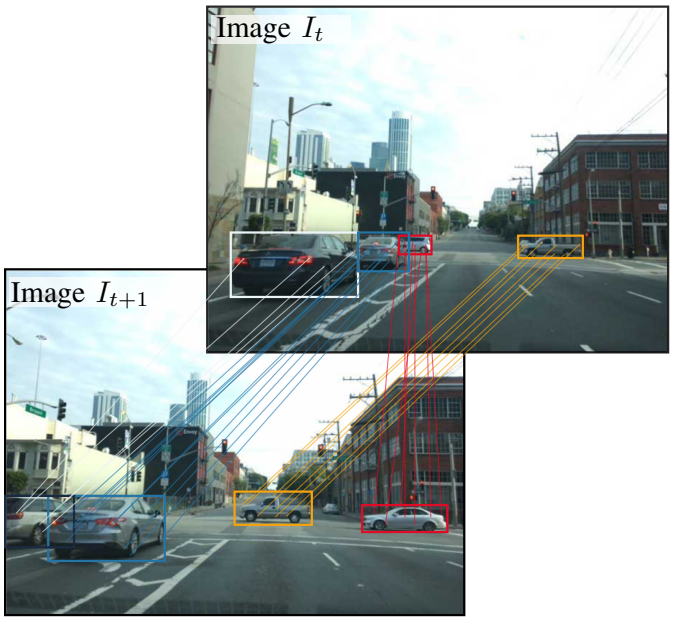


Fig. 5: An example of the visual consistency factor used in the pipeline. We extract and match ORB features between the images and use the ratio of shared features between the bounding boxes of detected objects to determine their correspondence. This alone does not provide satisfactory results (see the car in red and orange boxes) but together with the motion prior leads to a correct solution.

- $p(s_t|s_{t'}, G)$ is the likelihood of the exhibited motion as described in (3).

Intuitively, a solution which satisfies the appearance model but violates the prior motion model will have a low probability and similarly for the opposite case while a good solution satisfies all of the models.

The consistency of the visual appearance $p(c_t|c_{t'})$ is modelled by the number of visually matching features on both detected cars. We first extract ORB features for both images $I_t, I_{t'}$ and match the descriptors between the frames [19]. The probability is then calculated as the ratio of shared features between c_t and $c_t t'$:

$$p(c_t|c_{t'}) = \frac{f^{i,j}}{f^i}. \tag{8}$$

The entire concept is illustrated in Figure 5.

To ensure that the estimated 3D position of the car corresponds to its 2D detection we use a simple re-projection constraint illustrated in Figure 4:

$$p(s_t|c_t) = \mathcal{N}(\pi(x_t, p_t), \sigma_c), \tag{9}$$

where $\pi(x_t, p_t)$ is the projected position of the 3D point x_t into the camera image I_t located at position p_t. This is illustrated in Figure 4.

As both $p(s_t|c_t)$ and $p(s_t|s_{t'}, G)$ are continuous and differentiable, the maximisation of (6) can be performed using a classical Gauss-Newton optimisation method.

Finally, the probability that the observation belongs to a new vehicle is modelled by a constant p_0.

Algorithm 2 Tracking Pipeline

Input

$I_{1,2,...,N}$: camera images
G: motion prior
T: tracking window length

Output

$C_{1:N}$: detected cars in 2D images
$S_{1:N}$: 3D pose and velocity of detected cars

$K_0 \leftarrow 0$
for $t = 1, 2, 3, ..., N$ **do**
  Estimate 6dof pose of the image.
  $q_t \leftarrow$ *localise*(I_t)
  Detect 2D bounding boxes of the visible cars.
  $c_t^{1:K_t} \leftarrow$ *detectVehicles*(I_t)
  for $i = 1, 2, ..., K_t$ **do**
    Init solution with a new observation track
    $s_t^i \leftarrow \text{argmax}_s\, p(s|c_t^i, q_t, G)$
    Try to associate with any previously observed car
    for $t' = t-1, t-2, ..., t-T$ **do**
      for $j = 1, 2, ..., K_{t'}$ **do**
        $s_t^{ij} \leftarrow \text{argmax}_s\, p(s, c_t^i|c_{t'}^j, s_{t'}^j, q_t, G)$
        if $p(s_t^{ij}) > p(s_t^i)$ **then**
          $s_t^i \leftarrow s_t^{ij}$
        end if
      end for
    end for
  end for
end for

V. Experiments

In this section we evaluate the performance of the outlined visual tracking pipeline. First we describe the used dataset and proceed with the evaluation of the method and the influence of the employed crowd-sourced motion prior.

A. Datasets

To evaluate the method we collected a data set of 10M images captured by a fleet of 50 drivers using camera-equipped mobile phones capturing imagery data at regular 3Hz intervals in Downtown San Francisco and New York City. Examples from this data set are displayed in Figure 6. In total, this data set covers almost 1000 hours of driving at different time, weather, and traffic conditions experienced by the fleet during its operation. Next, we performed a large-scale structure-from-motion reconstruction to recover both the 3D map of the city and accurate 3D trajectories taken by the individual drivers. The resulting trajectories capture the motion pattern exhibited by the drivers in the fleet precisely localised in 3D space. This map is also used to visually localise position of the camera when the system is used in prediction mode to provide the pose of the camera necessary for 2D-3D association as described earlier.

For evaluating the visual tracking pipeline we randomly select subset of 1000 manually annotated image pairs lo-

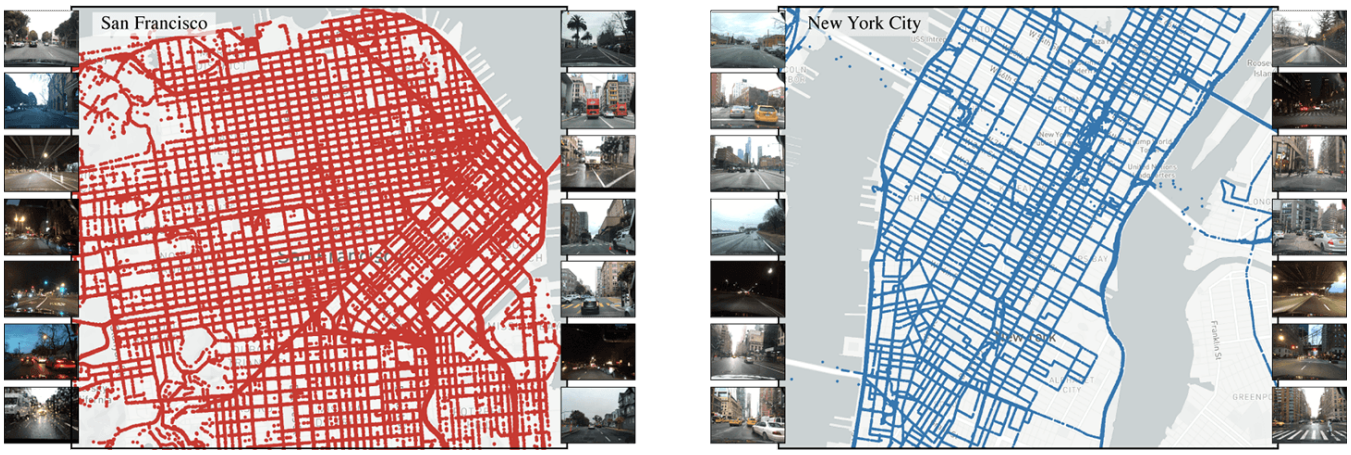


Fig. 6: The datasets used to evaluate the method. We have collected over 10M images in San Francisco and New York using dashcam-mounted mobile phones. These images were used to perform large-scale structure-from-motion to reconstruct accurate vehicle trajectories in the city over a period of several weeks.

calised in the map with ground-truth vehicle detection and their frame-to-frame association.

B. Motion model

We consider three different models for predicting a car's future motion $p(s_t|s_0, G)$:

- No model (vision only)
- Proposed crowd-sourced model from Section III
- Simple linear model.

As the linear model we consider a simple noisy linear model commonly used in various Kalman-filter-based methods defined as

$$v_t \sim v_0 + \epsilon_v, \quad (10)$$

$$r_t \sim r_0 + \epsilon_r, \quad (11)$$

$$x_t \sim x_0 + r_t * v_t + \epsilon_x, \quad (12)$$

and fit the noise constants $\epsilon_v, \epsilon_r, \epsilon_x$ to minimise the negative log-likelihood $-logp(s_t|s_0, G)$ of the data set.

As further discussed in [1] the motion-driven model significantly outperforms simpler linear model. The results are summarised in Table I, and Figure 8. While the linear model prediction error grows with the prediction horizon, our model's error is much slower to grow.

This makes the method suitable for predicting motion in long occlusions. A typical scenario of when using motion prior leads to better results is shown in Figure 7. While the linear model predicts a motion leading to collision the prior-based method predicts previously experienced curved motion.

Data	Model	1	2	3	4	5
All	Motion prior	28.74	30.56	32.80	37.55	41.48
All	Linear model	5.30	24.21	116.19	302.90	383.74
Intersection	Motion prior	27.61	32.67	33.56	40.34	44.34
Intersection	Linear model	5.23	25.12	144.90	355.35	441.22

TABLE I: The average negative log likelihood $-logp(s_t|s_0, G)$ for different prediction horizons (in seconds) of crowd-sourced motion prior [1]. As the prediction horizon increases the predictive power decreases. The prior-based method, however, degrades more gracefully than the linear-motion method.

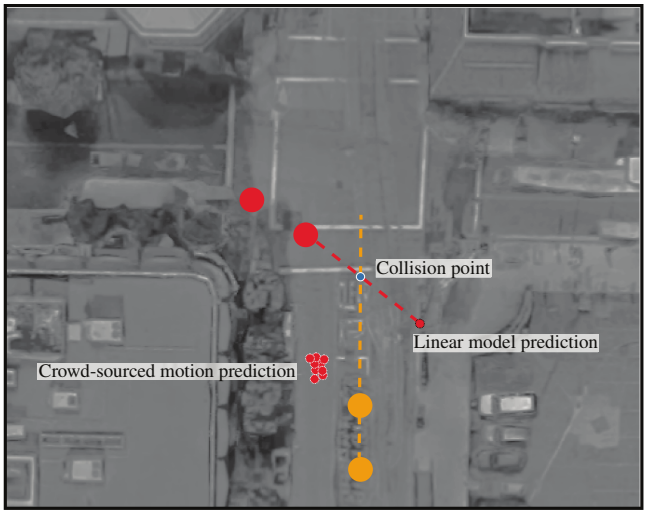


Fig. 7: A comparison of our model to the linear model. The big red circles indicate the position of an oncoming car at t and $t+1$. Under the linear model, it is expected to collide with our vehicle (orange). Our model corrects for this error using the motion prior.

Figure 10 shows one of the failure modes of the crowd-sourced prior. It depicts a segment of the map with insufficient priors to account for all possible trajectories, thus resulting in incorrect predictions. This is a typical performance when not enough data from the area was collected and improves itself as all possible motions are exhibited over time.

C. Visual tracking accuracy

The performance of the proposed visual tracking pipeline is largely defined by the accuracy of the frame-to-frame

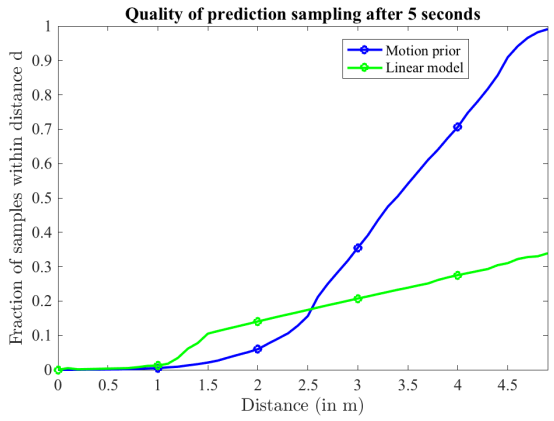


Fig. 8: The average fraction of the predictive distribution within distance d between crowd sourced prior [1] and a linear model. On average, a random sample from the distribution based on using the prior motion has a higher chance to be near the true position as in the case of a linear model.

association subroutine which is performed at every step of the algorithm as defined by (6). Specifically, we first evaluated a single-shot prediction: given one detection c_0 of the car we measure the ability to correctly predict its future position and associate it with a new observation c_t of the same car at time t. We measure the following properties:

1) Re-projection error of the predicted pose before association [in pixels],
2) Error of the predicted pose before association [in meters] computed as the difference between the predicted and the projected pose from the observation at time t,
3) Success rate of the association.

We measure these quantities for different values of t to simulate the effect of occlusions and/or noisy detections.

We also compare the resulting association rate with a pure visual approach when no prediction pose is available and the association is driven purely based on the visual similarity term in Equation 8.

The results are summarised in Table II. It captures the performance of the system on initialisation. Using the motion priors helps to reduce the error rate and outperform a purely visual approach universally over the entire prediction horizon. Figure 9 shows a typical scenario where the motion prior reduces incorrect associations. Due to an insufficient number of visible observations the system cannot distinguish between two visually similar cars. Using the motion priors our method penalises incorrect association since the likelihood of cars moving in that direction at that particular region of the map is very low.

Finally, we were interested in evaluating the situation when two correct observations are provided and thus the velocity of the vehicle can be estimated. This is a typical scenario when the vehicle is being reliably tracked. We

method	metric	1	2	3	4	5
Prior motion	re-projection error	19.19	23.66	35.25	39.76	52.4
Prior motion	pose error estimate	1.49	1.54	1.65	1.75	1.91
Prior motion	association rate	0.86	0.77	0.69	0.61	0.54
Vision-only	association rate	0.81	0.68	0.60	0.54	0.48

TABLE II: The prediction pose error and association rate of a car with unknown velocity at different prediction horizons and its comparison to a vision-only approach.

method	metric	1	2	3	4	5
Prior motion	re-projection error	2.49	3.71	5.69	7.35	10.86
Linear motion	re-projection error	14.71	27.68	48.11	63.82	87.6
Prior motion	pose error estimate	0.49	0.59	0.70	0.94	1.06
Linear motion	pose error estimate	1.42	1.58	1.89	2.43	3.43
Prior motion	association success	0.97	0.92	0.83	0.76	0.70
Linear motion	association success	0.84	0.73	0.67	0.61	0.54

TABLE III: Performance of motion prior vs linear model in predicting a car with known velocity.

compared this case to a linear model described earlier. The results of this are summarised in Table III. The performance is greatly improved because the knowledge of the velocity allows to select the prior motion pattern more accurately.

VI. Conclusion

In this work we presented a method for visually tracking motion of nearby vehicles using a camera system. This system uses a crowd-sourced data-driven non-parametric approach to predict the motion of visible vehicles.

We show that such an approach requires no form of annotation and is easy to scale to city sized data. We perform evaluations to show the effectiveness of our motion prediction method as a stand-alone technique and in combination with a monocular camera-based tracking pipeline.

There are many avenues to extend our system. In the

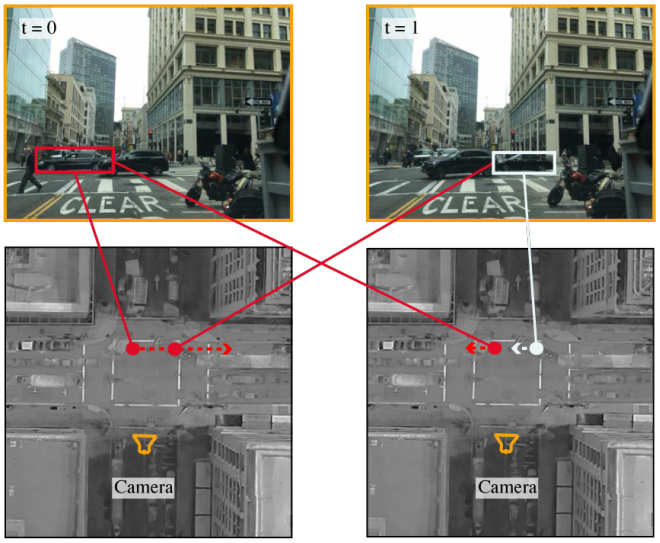


Fig. 9: The effect of the prior on the data association. An incorrect association and velocity estimate without using a motion prior based on a vision-only approach [left]. Correctly separated tracks and velocity estimate with presented vision + motion prior [right].

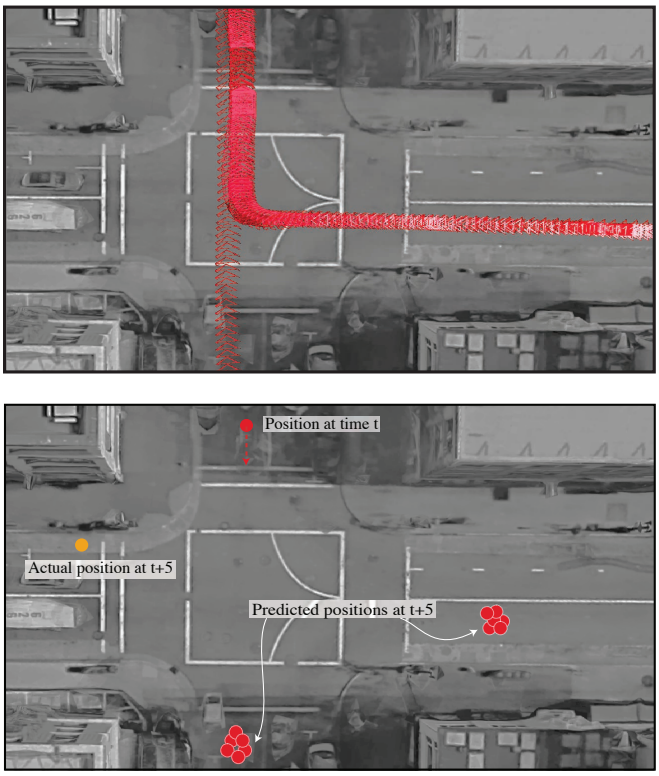


Fig. 10: Example failure case caused by insufficient prior data. The observed prior trajectories in the area [top] resulting into incorrect prediction for a car to turn right due to the lack of observed right turns [bottom].

future, we would like to integrate our system with other sensors, such as LIDAR, radar or stereo-camera. We would also like to extend the method to learn to predict the behaviour of other traffic participants, particularly pedestrians and explore how the interaction between different agents can be effectively taken into account.

REFERENCES

[1] S. M S, H. Grimmett, L. Platinskỳ, and P. Ondrúška, "Predicting trajectories of vehicles using large-scale motion priors," in *2018 IEEE Intelligent Vehicles Symposium (IV) Changshu, Suzhou, China*, 2018.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[3] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Information Fusion, 2008 11th International Conference on*. IEEE, 2008, pp. 1–6.

[4] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*. IEEE, 2009, pp. 417–422.

[5] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. Dietmayer, "Imm object tracking for high dynamic driving maneuvers," in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 825–830.

[6] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 4363–4369.

[7] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[8] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 141–146.

[9] S. Sivaraman, B. Morris, and M. Trivedi, "Learning multi-lane trajectories using vehicle-based vision," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2070–2076.

[10] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A Unified Framework for Maneuver Classification and Motion Prediction," *ArXiv e-prints*, Jan. 2018.

[11] P. Ondruska and I. Posner, "Deep tracking: Seeing beyond seeing using recurrent neural networks," *arXiv preprint arXiv:1602.00991*, 2016.

[12] B. Kim, C. M. Kang, S. H. Lee, H. Chae, J. Kim, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," *arXiv preprint arXiv:1704.07049*, 2017.

[13] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, "Surround vehicles trajectory analysis with recurrent neural networks," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 2267–2272.

[14] V. Akbarzadeh, C. Gagné, and M. Parizeau, "Kernel density estimation for target trajectory prediction," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 3449–3456.

[15] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.

[16] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele, "Monocular visual scene understanding: Understanding multi-object traffic scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 4, pp. 882–897, 2013.

[17] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.

[18] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3d scene analysis from a moving vehicle," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[19] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.