Long-Term On-Board Prediction of Pedestrians in Traffic Scenes

Apratim Bhattacharyya, Mario Fritz, Bernt Schiele
Max Planck Institute for Informatics
Saarland Informatics Campus
Saarbrücken, Germany
{abhattac, mfritz, schiele}@mpi-inf.mpg.de

Abstract: Progress towards advanced systems for assisted or even autonomous driving is leveraging recent advances in recognition and segmentation methods. Yet, we are still facing challenges in particular in bringing reliable driving to inner cities. Anticipation becomes a key element in order to react timely and prevent accidents. In this paper we argue that it is necessary to predict at least 1 second and we thus propose a new model that jointly predicts ego motion and pedestrian motion over such large time horizons. Our experimental results show that it is indeed possible to predict pedestrian movements at the desired time horizons. We also show that both sequence modeling of trajectories as well as our novel method of long term odometry prediction are essential for best performance.

Keywords: Autonomous Driving, Anticipation, Scene Understanding

1 Introduction

While methods for automatic scene understanding have progressed rapidly over the past years, it is just one key ingredient for assisted and autonomous driving. Human capabilities go beyond inference of scene structure and encompass a broader type of scene understanding that also lends itself to anticipating the future.

Anticipation in complex and cluttered inner city traffic situations is key in preventing collisions and maintaining practical safety distances by predicting future trajectories of e.g. pedestrians and cars. Even at conservative and careful driving speeds of $25\text{miles}/\text{hour}$ ($\sim 40\text{km}/\text{hour}$) in residential areas, the distance traveled in 1 second corresponds roughly to the breaking distance. Perfect anticipation on a 1 second time horizon would therefore support safe driving at such speeds. In this paper, we focus on predicting future trajectories from observations on-board vehicles over long-time horizons. We focus on pedestrians due to the particular importance for safety.

Existing works on on-board pedestrian trajectory prediction such as [1, 2] require the 3D positions of pedestrians which are difficult to estimate reliably in unconstrained environments. Moreover, the considered video sequences have linear or no vehicle ego-motion. Other works such as [3, 4, 5] also consider the problem of pedestrian trajectory prediction, but in a social context and focus on stationary or overhead camera videos.

Motion as observed from the vehicle's on-board cameras is determined by two types of motion – vehicle egomotion and pedestrian motion. Our key contribution is to formulate a two-stream model for pedestrian bounding box prediction and odometry (vehicle ego motion). We build on recent success in the field of autonomous driving [6, 7] to predict odometry. However, the focus in [6, 7] was on short time horizons - which is insufficient to compensate for the recording from a mobile platform over long time horizons. Our contributions in detail are: 1. First approach to long-term prediction of pedestrian bounding boxes from a mobile platform. 2. The first method for predicting odometry over long time horizons. 3. Detailed experimental evaluation of alternative architectures illustrating the importance and effectiveness of using a two-stream architecture.

1st Conference on Robot Learning (CoRL 2017), Mountain View, United States.

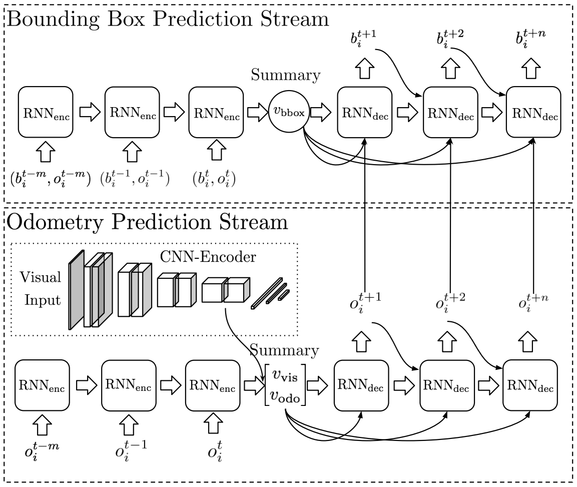

Figure 1: Two stream architecture for prediction of future pedestrian bounding boxes.

2 On-board pedestrian prediction

We begin by formally describing the problem. A bounding box corresponding to the i^{th} pedestrian observed on-board a vehicle at time step t can be described by the top-left and bottom-right pixel coordinates: $b_i^t = \{(x_{tl}, y_{tl}), (x_{br}, y_{br})\}$. We want to predict the future bounding box sequence B_{future} (where $\mid \text{B}_{\text{past}} \mid = m$) of the pedestrian. We condition our predictions on the past bounding box sequence B_{past}, the past odometry sequence O_{past} and the corresponding future odometry sequence O_{future} of the vehicle. The future odometry sequence O_{future} is predicted conditioned on the past odometry sequence O_{past} and on-board visual observation. Odometry sequences consists of the speed s^t and steering angle d^t of the vehicle, that is, $o^t = (s^t, d^t)$ at time-step t.

$$\text{B}_{\text{future}} = [b_i^{t+1}, ..., b_i^{t+n}] \;\mid\; \big\{\text{B}_{\text{past}} = [b_i^{t-m}, ..., b_i^t],\; \text{O}_{\text{past}} = [o^{t-m}, ..., o^t],\; \text{O}_{\text{future}} = [o^{t+1}, ..., o^{t+n}]\big\}$$

The prediction of future bounding box and odometry sequences constitute a multivariate sequence to sequence learning task. RNN encoder-decoder architectures are suitable and popular [8, 9]. They consist of an encoder RNN (RNN_{enc}) that reads in the input sequence and produces a summary vector v. This summary vector is used by the decoder RNN (RNN_{dec}) for prediction. We propose a two-stream RNN model that consists of two specialist RNN encoder-decoder pairs (streams); one for (conditioned) bounding box prediction and another for odometry prediction (see Figure 1). We use LSTMs as RNN encoders and decoders. We begin by describing the odometry prediction stream.

Odometry prediction. The past odometry O_{past} is input to the encoder RNN in the odometry prediction stream which embeds it into a summary vector v_{odo}. The past odometry of the vehicle gives a strong cue about the future short term (∼100ms) odometry, the visual observation on-board the vehicle can help in the longer term prediction of odometry; e.g. bends in the road, obstacles etc. Similar to [7, 6] we employ a convolutional neural network (CNN-encoder) to embed the visual information provided by the last observed frame; a visual summary vector v_{vis}.

As we are interested in long-term (multi-step versus single-step in [7, 6]) odometry prediction; we use a more complex CNN compared to [6] and during training we learn the parameters from scratch (unlike [7]). Our CNN-encoder has 10 convolutional layers with *ReLU* non-linearities. We use a fixed, small filter size of 3x3 pixels. We use max-pooling after every two layers. After max-pooling we double the number of convolutional filters; we use {32,64,128,256,512} convolutional filters. The convolutional layers are followed by three fully connected layers with 1024, 256 and 128 neurons and *ReLU* non-linearities. The output of the last fully connected layer is the visual summary v_{vis}. We use a dropout layer (20% probability) before the last fully connected layer to obtain a robust embedding.

The decoder RNN in the odometry prediction stream takes as input the summaries v_{odo}, v_{vis} and predicts the future odometry sequence O_{future}.

Two Stream Bounding Box and Odometry Prediction. The RNN encoder in the bounding box prediction stream takes as input the concatenated bounding box sequence B_{past} and the corresponding vehicle odometry sequence O_{past} and produces the summary v_{bbox}. The RNN decoder predicts the

				$\|\mathrm{B}_{\mathrm{past}}\|$	
Method	Odometry	2	4	6	8
Kalman Filter	x	x	1938	1289	1098
RNN	x	906	754	712	689
RNN-CL	x	899	692	663	650
RNN-CL	Ground-truth	742	417	353	349

Table 1: Bounding box prediction error (MSE) with varying $|\mathrm{B}_{past}|$

Method	MSE
Social LSTM [5]	1514
Our RNN	695
Our BBox centers	648

Table 2: Bounding box center prediction error (MSE).

			$\|\mathrm{B}_{\mathrm{past}}\|,\|\mathrm{O}_{\mathrm{past}}\|$	
Method	Streams	Visual	4	8
Kalman Filter	x	None	1938	1098
RNN	One	None	621	555
RNN	Two	None	593	536
RNN	Two	RGB	565	516

Table 3: Evaluation of the two stream model (MSE) against a joint model (Figure 1).

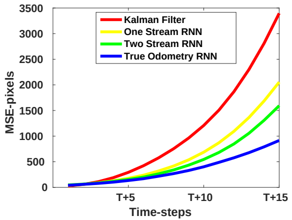

Table 4: MSE per time-step of the models in Table 1 row 1, 4, 5 and Table 3 row 4.

future bounding box sequence $\mathrm{B}_{\mathrm{future}}$ conditioned on the summary of vector v_{bbox} of the RNN encoder and the prediction of the RNN decoder of the odometry prediction stream for that time-step.

Model training We first train the odometry stream of our two-stream model independently. We then train the bounding box prediction stream keeping the the parameters of the odometry prediction stream fixed. We use mean square error for training, using the ADAM optimizer [10]. We fix the LSTM hidden and cell state vector sizes to 128. For sequences in the training set longer than $|\mathrm{B}_{\mathrm{past}}| + |\mathrm{B}_{\mathrm{future}}|$ ($|\mathrm{O}_{\mathrm{past}} + \mathrm{O}_{\mathrm{future}}|$ respectively) we use a sliding window to convert to multiple sequences. Moreover, as the sequences in the training set can be of varying lengths, we use a curriculum learning (CL) approach. We fix the length of the conditioning sequence $|\mathrm{B}_{\mathrm{past}}|$, $|\mathrm{O}_{\mathrm{past}}|$ and train for increasing longer time horizons $|\mathrm{B}_{\mathrm{future}}|$, $|\mathrm{O}_{\mathrm{future}}|$ (initializing the model parameters with those for shorter horizons). This allows us to train on sequences shorter than $|\mathrm{B}_{\mathrm{future}}|$, $|\mathrm{O}_{\mathrm{future}}|$ of the final model and leads to faster convergence.

3 Experiments

Dataset and evaluation metric. We use the Cityscapes dataset [11] which contains 2975 training and 1525 test high definition (recorded on-board a vehicle) video sequences of length 1.8 seconds (30 frames), with associated odometry information. The video resolution is 2048x1024 pixels. Pedestrian bounding box tracks were obtained using the tracking by detection method of [12]. Detections were obtained using the Faster R-CNN based method of [13]. We use as evaluation metric the mean squared error (MSE) averaged across all time-steps and also report plots per time-step.

Evaluation of bounding box prediction. We first evaluate the bounding box prediction stream of our two stream model, without conditioning it on predicted odometry. We predict 15 time-steps into the future and report the results in Table 1. We compare its performance with a linear Kalman filter. We evaluate the effectiveness of curriculum learning (CL) and as an oracle case compare against a version in which the RNN encoder can see the past odometry (concatenated with $\mathrm{B}_{\mathrm{past}}$) and the RNN decoder can see the true future odometry. We also vary the length of the conditioning sequence $|\mathrm{B}_{\mathrm{past}}|$. In Table 1, we see that our RNN encoder-decoder model (2nd row) outperforms the linear Kalman filter (1st row). This shows that many bounding box sequences have a non-linear trajectory and therefore cannot be modelled by a Kalman filter. In order to ensure comparability, training/test sets are kept constant across varying $|\mathrm{B}_{\mathrm{past}}|$). We see that increasing the length of the conditioning sequence improves the model performance saturating at $|\mathrm{B}_{past}| = 8$. Curriculum learning (CL) improves performance (2nd to 3rd row). Furthermore, in oracle study (4th row) our RNN model performs significantly better. This shows that knowledge odometry is crucial for good performance.

Comparison with Social LSTM [5]. We compare our RNN encoder-decoder model with the vanilla LSTM model of Alahi et al. [5] (with 128 neurons) that predicts trajectories independently in Table 2. Both models are trained to predict sequences of bounding box centers. Our RNN model (without CL) performs better as it is more robust to mistakes during recursive prediction. The model of [5]

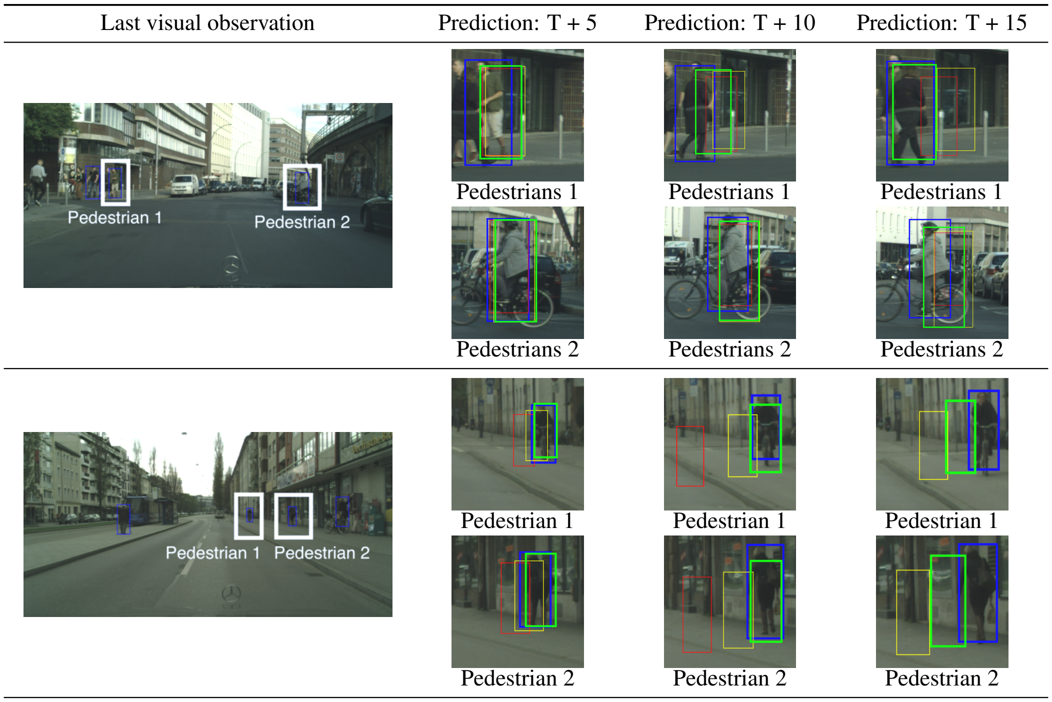

Figure 2: Visual examples of prediction of pedestrian bounding boxes by models in Table 3. Color codes: Blue: Ground-truth, Red: Kalman Filter, Yellow: Our one-stream RNN, Green: Our two-stream RNN with visual input. Top: case with vehicle deceleration, Bottom: case with turning vehicle. In these cases with non-linear vehicle ego-motion, our two-stream RNN with visual input outperforms other methods. Video results in supplementary material.

observes true past pedestrian coordinates during training. However, during prediction it observes its own predictions causing errors to be propagated though multiple steps of prediction. Furthermore, we compare both methods to the centers obtained from the predictions of our single-stream RNN (second row of Table 1). The results show that we can improve upon bounding box center prediction by predicting bounding boxes.

Evaluation of our two-stream model. We perform an ablation study of our two-steam model in order to evaluate different aspects of our architecture: we compare to i) a joint single-stream RNN encoder-decoder model where the encoder observes the concatenated past bounding box and velocity sequence $\{\mathrm{B}_{\mathrm{past}}, \mathrm{O}_{\mathrm{past}}\}$ and the decoder jointly predicts the future odometry and bounding box sequence $\{\mathrm{B}_{\mathrm{future}}, \mathrm{O}_{\mathrm{future}}\}$ (with double the learn-able LSTM parameters for fair comparison); ii) a model where the odometry specialist RNN has no CNN-encoder (odometry specialist decoder RNN sees only v_{odo}). We evaluate the models and report the results in Table 3, Table 4 and Figure 2. The results show that jointly predicting odometry with pedestrian bounding boxes (3rd row) significantly improves performance (2nd row). The superior performance of our two-stream model over a joint model validates our modelling approach. Overall, our two-stream model with the CNN-encoder (and odometry specialist RNN decoder seeing both $\{v_{odo}, v_{vis}\}$) performs best. This is due to the superior performance of the odometry prediction stream with the CNN-encoder. Better odometry prediction is leveraged for better bounding box prediction (4th row in Table 3).

4 Conclusion

We highlight the importance of anticipation for practical and safe driving inner city traffic scenes. We contribute to this important research direction the first model for long term prediction of pedestrians from on-board observations. We show predictions over long time horizons. Key to our success is the first long term prediction of egomotion by predicting odometry into the future. We evaluate and compare several different architecture choices and arrive at a two-stream encoder-decoder model with separate losses on odometry prediction and pedestrian prediction.

References

[1] C. G. Keller, C. Hermes, and D. M. Gavrila. Will the pedestrian cross? probabilistic path prediction based on learned motion features. In *Joint Pattern Recognition Symposium*, pages 386–395. Springer, 2011.

[2] E. Rehder and H. Kloeden. Goal-directed pedestrian prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015.

[3] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *CVPR*, 2011.

[4] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, 2016.

[5] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.

[6] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[7] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. *arXiv preprint arXiv:1612.01079*, 2016.

[8] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[12] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Multi-person tracking by multicut and deep matching. In *ECCV*, 2016.

[13] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. *arXiv preprint arXiv:1702.05693*, 2017.