An RNN-Based IMM Filter Surrogate

Stefan Becker(✉), Ronny Hug, Wolfgang Hübner, and Michael Arens

Fraunhofer Institute for Optronics, System Technologies, and Image Exploitation (IOSB), Gutleuthausstr. 1, 76275 Ettlingen, Germany
stefan.becker@iosb.fraunhofer.de

Abstract. The problem of varying dynamics of tracked objects, such as pedestrians, is traditionally tackled with approaches like the Interacting Multiple Model (IMM) filter using a Bayesian formulation. By following the current trend towards using deep neural networks, in this paper an RNN-based IMM filter surrogate is presented. Similar to an IMM filter solution, the presented RNN-based model assigns a probability value to a performed dynamic and, based on them, puts out a multi-modal distribution over future pedestrian trajectories. The evaluation is done on synthetic data, reflecting prototypical pedestrian maneuvers.

Keywords: Trajectory forecasting · Path prediction · IMM filter · Multiple model filter

1 Introduction

The applications of pedestrian trajectory prediction cover a broad range from autonomous driving, robot navigation, smart video surveillance to object tracking. Traditionally, the task of object motion prediction is done by using a Bayesian formulation in approaches such as the Kalman filter [17], or non-parametric methods, such as particle filters [4]. Driven by the success of recurrent neural networks (RNNs) in modeling temporal dependencies in a variety of sequence processing tasks, such as speech recognition [10,13] and caption generation [12,27], RNNs are increasingly utilized for object motion prediction [2,3,7,15,16]. When relying on traditional approaches, the challenge of varying dynamics over time is commonly addressed with the Interacting Multiple Model (IMM) filter [9]. The IMM filter is a well established approach to elegantly combine a set of candidate models into a single context by weighting each individual model. Each model corresponds to a specific motion pattern and contributes to the final state estimation depending on its current weight. According to the IMM filter solution, in this paper an RNN-based IMM filter surrogate is presented. On the one hand, the presented RNN-based model is able to also provide a confidence value for the performed dynamic and on the other hand can overcome some limitations of the classic IMM filter. The suggested RNN-encoder-decoder model generates the probability distribution over future pedestrian paths conditioned on a dynamic class. The model is based on the work of Deo and Trivedi [11]. For

M. Felsberg et al. (Eds.): SCIA 2019, LNCS 11482, pp. 387–398, 2019.
https://doi.org/10.1007/978-3-030-20205-7_32

the case study of freeway traffic, they used an two branch RNN-encoder-decoder network for vehicle maneuver and trajectory prediction. Since for vehicle applications an on-board lane estimation algorithm is mostly available, a stationary frame of reference, with the origin fixed at the vehicle being predicted, is used in their work. Although this makes the model independent of road curvature and independent of how vehicle tracks are obtained, it can not be applied without adjustments for pedestrian motion prediction. Thus, our RNN-based model infers like classical filters the current position and uses only a single RNN branch for encoding the maneuver class, the filtered position and the trajectory information. In the context of vehicle motion prediction, maneuver or rather dynamic classes can be better defined than for pedestrians. For example by changing or keeping the lane. Due to the dynamic behavior of pedestrians, the maneuver classes are here defined based on the deviation from a straight walking pedestrian. The presented network also extends the maneuver network of Deo and Trivedi [11] with insights from the work of Becker et al. [7] to better adapt to pedestrian motions.

Moreover, this paper aims to highlight some relations between traditional multiple model approaches such as the IMM filter and the suggested RNN-based IMM filter surrogate. By combining the different views on maneuver predictions, this work contributes to an exploration of the connections between both problem formulations. The decoder uses the de-noised position estimate and a context vector, encoding the dynamic classes, to predict future positions. The analysis is done on synthetic data reflecting prototypical scenarios capturing pedestrians maneuvers.

In the following, a brief formalization of the problem and a description of the RNN-based model are provided. The achieved results are presented in Sect. 3. Finally, a conclusion is given in Sect. 4.

2 RNN-Based IMM Filter Surrogate

The goal is to devise a model that can successfully predict future paths of pedestrians and represent alternating pedestrian dynamics, e.g. dynamics that can transition from a straight walking to a turning maneuver or stopping. Here, trajectory prediction is formally stated as the problem of predicting the future trajectories of a pedestrian, conditioned on its track history. Given an input sequence $\mathcal{Z} = \{(x^t, y^t) \in \mathbb{R}^2 | t = 1, \ldots, t_{obs}\}$ of T_{obs} consecutive observed pedestrian positions $\boldsymbol{z}^t = (x^t, y^t)$ at time t along a trajectory, the task is to generate a multi-modal prediction for the next T_{pred} positions $\{\boldsymbol{x}^{t+1}, \boldsymbol{x}^{t+2}, \ldots, \boldsymbol{x}^{t+T_{pred}}\}$ and to filter the current position $\boldsymbol{x}^t = (x^t, y^t)$. One insight from the work Becker et al. [7] is that motion continuity is easier to express in offsets or velocities, because it takes considerably more modeling effort to represent all possible conditioning positions. In order to exploit scene-specific knowledge for trajectory prediction, additional use of the position information is required. When sufficient training samples from a particular scene are available, Hug et al. [15] showed that RNN-based trajectory prediction models are able to capture spatially dependent behavior changes only from motion data. However, here the offsets are additionally used for conditioning the network $\mathcal{Z} = \{(x^t, y^t, \delta_x^t, \delta_y^t) \in \mathbb{R}^4 | t = 2, \ldots, t_{obs}\}$. Apart from the smaller modeling effort to represent conditioned offsets, the shift to offsets helps to prevent undefined states due to a limited data range [7] and it is easier to make better generalizations across datasets. Since, we analyze the model capabilities on synthetic data reflecting prototypical pedestrian maneuvers for a fixed scenario, the amount of training samples is not restricted. Thus, in order to localize in the reference system position information is used to estimate the true position. The future trajectory is denoted with $\mathcal{Y} = \{(x^t, y^t) \in \mathbb{R}^2 | t = t_{obs} + 1, \ldots, t_{pred}\}$. The model estimates the conditional distribution $P(\mathcal{Y}, \boldsymbol{x}^t | \mathcal{Z})$. In order to identify specific dynamics under M desired maneuver classes (e.g. turning maneuvers, stopping and straight walking), this term can be given by:

$$P(\mathcal{Y}, \boldsymbol{x}^t | \mathcal{Z}) = \sum_{i=1}^{M} P_{\Theta}(\mathcal{Y}, \boldsymbol{x}^t | m_i, \mathcal{Z}) P(m_i | \mathcal{Z}) \tag{1}$$

Here, $\Theta = \{\Theta^{t_{obs}+1}, \ldots, \Theta^{t_{pred}}\}$ are the parameters of a L component Gaussian mixture model $\Theta^t = (\boldsymbol{\mu}_l^t, \Sigma_l^t, w_l^t)_{l=1,\ldots,L}$. By adding the maneuver context in form of the posterior mode probability, $P(m_i | \mathcal{Z}) \mathrel{\hat{=}} \alpha_i$ the analogy to the classic IMM filter becomes apparent. For an IMM filter, the mode probability is used to calculate the mixing probabilities to combine the set of chosen candidate models into a merged estimate. The time behavior of the basic filter set is modeled as a homogeneous (time invariant) Markov chain with a fixed transition probability matrix (TPM) $m_{ij} \mathrel{\hat{=}} P(m_i^t | m_j^{t-1})$. Under the assumption that M models describe the variation of the dynamics, the posterior density of the IMM filter can be written as follows:

$$P(\boldsymbol{x}^t | \mathcal{Z}) = \sum_{i=1}^{M} P_{\Theta_{IMM}}(\boldsymbol{x}^t | m_i, \mathcal{Z}) P(m_i | \mathcal{Z}) \tag{2}$$

Here, $P_{\Theta_{IMM}}(\boldsymbol{x}^t | m_i, \mathcal{Z})$ is in the context of an IMM filter a Gaussian distribution and $P(m_i | \mathcal{Z}) \mathrel{\hat{=}} \alpha_i$ is the posterior mode probability for the IMM filter. As mentioned above, the transition between different dynamics is modeled as a first order Markov chain for an IMM filter. The law of total probability allows to compute new mode probabilities based on the transition probabilities. Given the current mode probabilities and transition probabilities, the mixing probabilities $\alpha_{i|j}$ for the mixing step of the IMM filter can be calculated. For each model M_i and M_j, they are calculated as $\alpha_{i|j}^{t-1} = \frac{1}{\bar{c}_j} m_{ij} \alpha_i^{t-1}$ with a normalization factor $\bar{c}_j = \sum_{i=1}^{M} m_{ij} \alpha_i^{t-1}$. Then, in the prediction stage, each filter is applied independently using the calculated mixed initial condition. Subsequently, the model probabilities are adapted according to the likelihood of each filter.

RNN-IMM: Whereas an explicit modeling of the switching behavior and the object dynamics of the IMM filter stands in contrast to an implicit dynamic encoding of an RNN-based approach. In order to provide an IMM filter surrogate, the proposed model also estimates mode probabilities and filters or rather de-noises the current position based on noisy observations $\mathcal{Z}$. By writing the conditional distribution $P(\mathcal{Y}, \boldsymbol{x}|\mathcal{Z})$ of the RNN-based approach in form of Eq. 1, the desired estimates can be inferred from the hidden states of the RNN $\boldsymbol{h}$. This formulations does not require to set the parameters of the TPM matrix manually, which is commonly done based on the mean sojourn time (the mean time an object stays in a motion type [5,24]) or as stated in the work of Bar-Shalom [5], an ad-hoc approach to fill the diagonals with values close to one. For the proposed RNN-based IMM filter surrogate (RNN-IMM), the basic architecture is a recurrent encoder-decoder model. The encoder takes the frame by frame input sequence $\mathcal{Z}$. The hidden state vector of the encoder is updated at each time step based on the previous hidden state and the current observation. The generated internal representation is used to predict mode probabilities $\boldsymbol{\alpha}^t$ at the current time step and $\boldsymbol{x}^t$. With embedding of the current observations, the encoder can be defined as follows:

$$
\begin{aligned}
\boldsymbol{e}^t_{encoder} &= \mathrm{EMB}(\boldsymbol{z}^t; W_{ee}) \\
\boldsymbol{h}^t_{encoder} &= \mathrm{RNN}(\boldsymbol{h}^{t-1}_{encoder}, \boldsymbol{e}^t_{encoder}; W_{encoder}) \\
\hat{\boldsymbol{x}}^t, \boldsymbol{\alpha}^t_{logits} &= \mathrm{MLP}(\boldsymbol{h}^t_{encoder}; W_{en}) \\
\hat{\boldsymbol{\alpha}}^t &= \frac{\exp\left(\boldsymbol{\alpha}^t_{logits}\right)}{\sum_{j=1}^{M} \exp\left(\alpha^t_{logits,j}\right)}
\end{aligned}
$$

Here, RNN$(\cdot)$ is the recurrent network, $\boldsymbol{h}$ the hidden state of the RNN, MLP$(\cdot)$ the multilayer perceptron, and EMB$(\cdot)$ an embedding layer. W represents the weights and biases of the MLP, EMB or respectively RNN. The final state of the encoder can be expected to encode information about the track histories. For generating a trajectory distribution over dynamic modes, the encoder hidden state is appended to a one-hot encoded vector corresponding to specific maneuvers and the filtered current position. Instead of only filtering the position, the encoder could also be used to parametrize a mixture density output layer (MDL). The decoder of the model can be defined as follows:

$$
\begin{aligned}
\boldsymbol{h}^t_{decoder} &= \mathrm{RNN}(\boldsymbol{h}^{t-1}_{decoder}[\boldsymbol{h}^t_{encoder}], \hat{\boldsymbol{x}}^t, \boldsymbol{\alpha}^t; W_{decoder}) \\
\hat{\mathcal{Y}} = \{(\hat{\boldsymbol{\mu}}^t_l + \hat{\boldsymbol{x}}^{t_{obs}}, \hat{\Sigma}^t_l, \hat{w}^t_l) | t = t_{obs}+1, \ldots, t_{pred}\} &= \mathrm{MLP}(\boldsymbol{h}^t_{decoder}; W_{de})
\end{aligned}
$$

The decoder is used to parametrize a mixture density output layer (MDL) or rather Θ directly for several positions in the future (one distribution for every time step). Nevertheless, the overall RNN-IMM uses the trajectory prediction and dynamic classification jointly, the loss function for training is split into three parts.

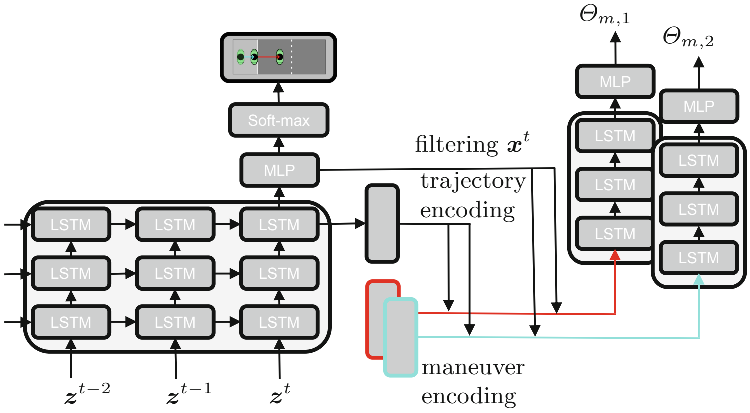

Fig. 1. Visualization of the RNN-based IMM filter surrogate (RNN-encoder-decoder network) for jointly predicting specific dynamic probabilities and corresponding future distributions of trajectory positions. The encoder predicts the dynamic probabilities and the filtered position for the current time step. The decoder uses the context vector and the position estimate to predict future pedestrian locations.

Dynamic classification is trained to mimimize the sum of cross-entropy losses of the different M motion model classes:

$$\mathcal{L}(\mathcal{Z})_{maneuver} = -\sum_{j=1}^{M} \alpha_{j,GT}^t \log(\hat{\alpha}_j^t) \tag{3}$$

Additionally, the encoder is trained by minimizing the filtering loss $\mathcal{L}(\mathcal{Z})_{filter}$ in form of the mean squared error to the ground truth current pedestrian locations. In case the encoder should generate the parameter of a mixture of Gaussian or single Gaussian distribution, the negative log likelihood for the ground truth pedestrian locations can be minimized. Finally, the complete encoder-decoder is trained by minimizing the negative log likelihood for the ground truth future pedestrian locations conditioned under the performed maneuver class. The context vector is appended with the ground truth values of the dynamic model or maneuver classes for each training trajectory. This results in the following loss function:

$$\begin{aligned} \mathcal{L}(\mathcal{Z})_{pred} &= -\log(P_\Theta(\hat{\mathcal{Y}}|m_{GT}, \mathcal{Z})P(m_{GT}|\mathcal{Z})) \\ \mathcal{L}(\mathcal{Z})_{pred} &= \sum_{t=t_{obs}+1}^{t_{pred}} -\log(\sum_{l=1}^{L} \hat{w}_l^t \mathcal{N}(\boldsymbol{x}^t|\hat{\boldsymbol{\mu}}_l^t + \boldsymbol{x}^{t_{obs}}, \hat{\Sigma}_l^t; m_{GT})) \end{aligned} \tag{4}$$

The overall architecture is visualized in Fig. 1. The context vector combines the encoding of the track history with the encoding of the alternating dynamic classes. Together with the filtered position, it is used as input for the decoder.

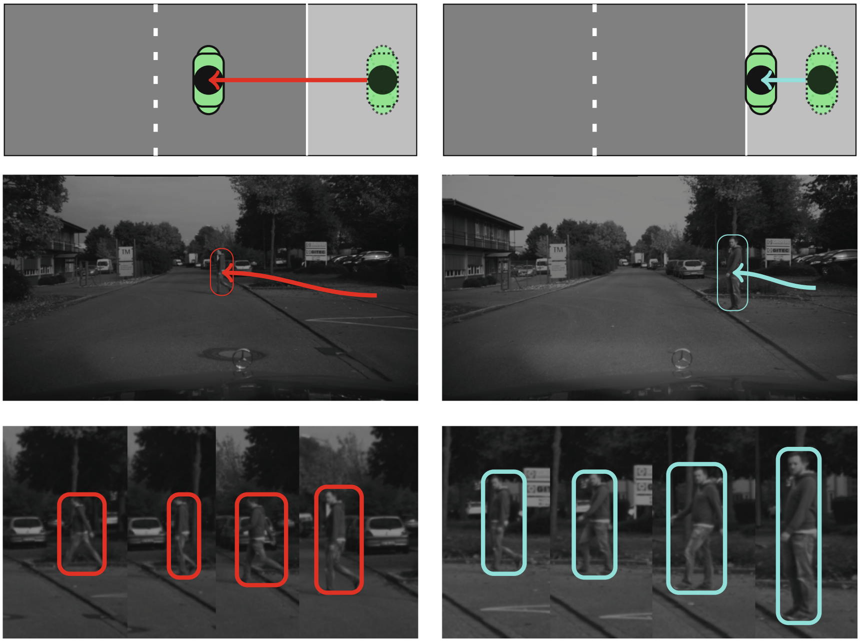

Fig. 2. Illustration of typical pedestrian motions. The above images depict the two chosen maneuver classes of straight walking or rather crossing and stopping. The images on the left show a person crossing the street. The images on the right show a person changing from walking to standing at the curbside of the street. In particular changing from straight walking to stopping [24].

3 Data Generation and Evaluation

This section consists of a brief evaluation of the proposed RNN-IMM. The evaluation is concerned with verifying the overall viability of the approach in maneuver situations. For initial results, a synthetic test condition is used in order to gain insight into the model behavior in different typical pedestrian motion types. A prototypical maneuver performed by a pedestrian, which has important implications for the field of intelligent vehicles and video surveillance is a stopping or deceleration maneuver.

Data Generation and Reference Methods: For the first mentioned context of intelligent vehicles, Schneider et al. [24] performed a comparative study on recursive Bayesian filters for pedestrian path prediction at short time horizons (below 2 s). They applied different filters on typical pedestrian motion types. Although, the comparison was done on the Daimler path prediction dataset, we evaluate on synthetic data but make use of the provided real data to capture a similar condition. Firstly, the Daimler path prediction dataset provides only a maximum amount of 23 sequences for single motion types. As mentioned before, in order to avoid problems such as a limited number of training samples and to gain some insights into a controlled setup, synthetic data is used. Secondly, the location information is biased in the dataset. Since recursive Bayesian filters make in their standard formulation no use of the spatial context of a scene, this does not harm their mutual comparison. However, RNN-based prediction networks are able to capture spatially dependent behavior changes [15], thus a fair comparison is difficult to achieve. The evaluation on the Daimler dataset is done in an ego-motion compensated reference system. The frame rate of the camera system inside the recording vehicle is 16 fps and it is taken over accordingly for our experiments. The pedestrians change their behavior abruptly. Therefore, the sensible time horizons are short. Here, 8 (0.5 s) consecutive positions are observed, before predicting the next 8 (0.5 s), 12 (0.75 s) and 16 (1 s).

For generating synthetic trajectories of a basic maneuvering pedestrian, random agents are sampled from a Gaussian distribution according to a preferred pedestrian walking speed [26] ($\mathcal{N}(1, 38\,\mathrm{m}, 0.37\,\mathrm{m})$) from the distribution of starting positions of the corresponding Daimler dataset sequences. During a single trajectory simulation the agents can perform a stopping maneuver or cross the street. Figure 2 illustrates such maneuvers with example images from the Daimler dataset [24]. For mapping the pedestrian detections to a vehicle-motion compensated ground plane, Schneider et al. used on-board sensors for velocity and yaw rate and a stereo camera system to compute the median disparity. Due to the non-linear observation model based on a perceptive camera model, an inevitable linearized extension for the Kalman and IMM filter observation models are required. Here, the observation uncertainty of the position sensor is assumed to be Gaussian distributed $r^t \sim \mathcal{N}(0, 0.01\,\mathrm{m})$ in the compensated reference system. Thus, the standard formulation of the Bayesian filters are well suited for this task. For the stopping maneuver or rather the event of deceleration till standing, a mean sojourn time of 1 s with a standard deviation of 0.1 s is used. As long as a person moves in a straight line at a reasonably constant speed, their dynamics can be captured with a Kalman filter using a constant velocity model. During the maneuver, the relation to one fixed process model describing the dynamics fails due to an additional deceleration. Similar to Schneider et al. [24] or Kooij et al. [21], the reference IMM filter is set up by combining two basic models, in particular, the constant velocity (CV) and the constant acceleration (CA) model. For avoiding side effects due to independent motions in different directions, see for example [6], only the crossing direction, from the vehicle perspective, the lateral motion is considered. Following the aforementioned explanations, the IMM-RNN is compared to an IMM filter with two motion models (CV, CA), a Kalman filter with a single CV model, a Kalman filter with a single CA model, and as baseline to a linear interpolation. Also correspondingly to Schneider et al., the process noise q is determined by $Q(t) = Q_0(t)q$, where $q \in \{\sigma_{CV}, \sigma_{CA}\}$ are spectral densities (continuous time variances) of the process noise, describing the changes in velocity or respectively in acceleration over a sampling period Δt (CV: $\sqrt{Q_{22}} = \sqrt{\Delta t \cdot q}$; CA: $\sqrt{Q_{33}} = \sqrt{\Delta t \cdot q}$, see for example [23]). Based on this process noise model, the optimal process noise parameters for the different chosen filters (IMM filter (CV, CA), Kalman filter CV, CA) on the Daimler dataset are for the two IMM filter models $\sigma_{IMM,CV} = 0.70, \sigma_{IMM,CA} = 0.80$ and for the single Kalman filters $\sigma_{CV} = 0.77$ and $\sigma_{CA} = 0.44$ [24]. These parameters are consistent with the suggested practical setting in Bar-Shalom [5] and the chosen sojourn time for the simulation.

As mentioned above, a definition of maneuver classes for pedestrians is harder to establish than for vehicles. Hence, the main interest is here to detect the deviation from a standard behavior, and whether the pedestrian is in a *normal* mode. A set of deviation in velocity, deceleration, along with the tangential ground truth trajectory is used to assign a maneuver label to a time step of a single trajectory. Thus, the RNN-IMM and IMM filter have a similar basic dynamic model set description. As the distribution over the trajectories for the RNN-IMM is captured with a Gaussian mixture model, the maneuver description for a single model can still be multi-modal. Since the IMM filter predicts a multi-modal distribution in form of a combination of the uni-modal model specific prediction, in the presented results the RNN-IMM is set to also only predict conditioned on a single maneuver class a uni-modal Gaussian distribution.

Implementation Details: The model has been implemented using *Tensorflow* [1] and is trained for 2000 epochs using ADAM optimizer [19] with a decreasing learning rate, starting from 0.01 with a learning rate decay of 0.95 and a delay factor of 1/10. During the learning rate adaption, the number of epochs is multiplied by the delay factor. For the experiments, the RNN variant Long Short-Term Memory (LSTM) [14] is used.

Results and Analysis: In Fig. 3, predictions for two different preformed motion types are depicted for 8 future positions weighted by the predicted maneuver probability. In the shown images the positions are normalized to start at the origin. The resulting multi-modal prediction is visualized as a heatmap. On the left, it can be seen that for a crossing sequence with straight walking the RNN-IMM mainly uses the corresponding straight walking model. On the right, where the deceleration started, the straight walking probability is visibly lower and the predicted distribution maximum is very close to the last observation. For the quantitative evaluation, 1000 noisy trajectories have been synthetically generated, where 80% are used for training and 20% for the comparison to the recursive Bayesian filters. The results are summarized in Table 1.

The performance is compared with the final displacement error (FDE) (see for example [22]) of the lateral motion (from the vehicle perspective) for three different time horizons, in particular 8 steps (0.5 s), 12 steps (0.75 s) and 16 steps (1 s). These results show that the presented RNN-IMM is able to faster capture the change in dynamic for the synthetically generated data. In terms of the single motion models (CV vs. CA), one can observe the benefits for the CA in capturing the deceleration. The IMM filter combines both and shows an improvement. Hence, the aim of this paper is more on highlighting the relation between traditional multiple model approaches and the suggested RNN-based

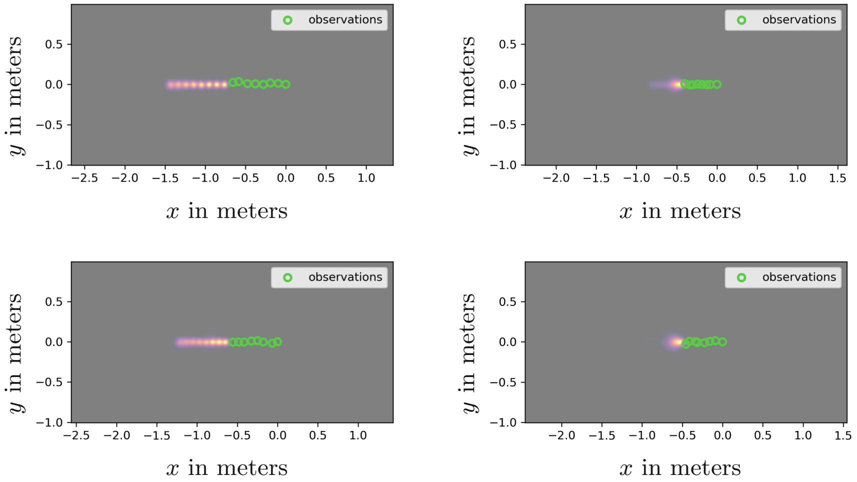

Fig. 3. Visualization of the predicted multi-modal distributions of future position as heatmap. (Left) Density plots for crossing or rather straight walking examples. (Right) Density plots for stopping examples in which the maximum of the predicted distribution is visible close to the last observation.

IMM filter surrogate, it should be mentioned that RNN-based approaches are designed to receive input data for every time step, whereas Bayesian filters are well suited for handling missing observations. Especially with such a short initialization time, this can be crucial. One argument towards a learning based RNN-IMM is that we only choose the maneuver definition based on deviation of standard straight walking. The engineering task of finding the best model set up for IMM filters and their extensions can lead to an improved behavior (see for example Keller et al. [18]) in specific maneuver situations, but is also very tedious to find a good setting. It should also be mentioned that recent work like the approaches of Kooij et al. [20] show options how to further improve the prediction performance by including scene context and using more cues than pedestrian point kinematics (e.g. head orientation, gaze, body tilt, articulated body information).

In summary, the presented RNN-IMM is able to also provide a confidence value $P(m_i|\mathcal{Z}) \triangleq \alpha_i$ for the performed dynamic, but avoids modeling the dynamic transitions with a fixed transition probability matrix $P(m_i^t|m_j^{t-1})$. Similar to the provided mode probabilities of IMM filters, this can be used for further processing steps or rather applications (see for example [8,25]). Further, instead of choosing the basic filter set, the prediction model is learned. In case there exists some well known model for describing the standard dynamic of the desired target, only deviations from the known dynamic can be used to define additional maneuver classes. This study on synthetically generated data shows, that by exploiting the connections between different views on maneuver prediction some perspectives on overcoming respective limitations can be gained.

Table 1. Results for the comparison between the proposed RNN-IMM and an IMM filter with two motion models (CV, CA), a Kalman filter with a single CV model, a Kalman filter with a single CA model, and using linear interpolation on the simulated maneuver situations. The prediction is done for 8, 12, and 16 time steps conditioned on 8 observations for a frame rate of 16 fps.

	8/8		8/12		8/16	
Approach	FDE [m]	σ_{FDE} [m]	FDE [m]	σ_{FDE} [m]	FDE [m]	σ_{FDE} [m]
RNN-IMM	0.0309	0.0404	0.0427	0.0817	0.0517	0.0941
IMM filter (CV,CA)	0.0674	0.0602	0.1188	0.1255	0.1862	0.1915
Kalman filter (CA)	0.0796	0.0638	0.1575	0.1137	0.2386	0.1696
Kalman filter (CV)	0.1578	0.1601	0.2890	0.2965	0.4701	0.4700
Linear interpolation	0.1587	0.1610	0.2903	0.2978	0.4724	0.4718

4 Conclusion

In this paper, an RNN-encoder-decoder model, which can be interpreted as an IMM filter surrogate, has been presented. The RNN-IMM is able to jointly predict specific motion probabilities and corresponding distributions of future pedestrian trajectory. The model capabilities were shown on synthetic data that were reflecting typical pedestrian maneuvers. By conditioning on specific dynamic models or rather deviation from standard behavior, the model makes it possible to generate additional information in terms of an assigned maneuver probability similar to an IMM filter, but reduces the amount of explicit modeling of filter parameters (e.g. the dynamic transitions matrix). Thus, the presented RNN-IMM helps to reduce the amount of hard-coded engineering of traditional multiple model filter such as the IMM filter.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). Software: https://www.tensorflow.org/
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 961–971. IEEE (2016)
3. Alahi, A., et al.: Learning to predict human behaviour in crowded scenes. In: Group and Crowd Behavior for Computer Vision. Elsevier (2017)
4. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. Trans. Sig. Process. **50**(2), 174–188 (2002). https://doi.org/10.1109/78.978374
5. Bar-Shalom, Y., Kirubarajan, T., Li, X.R.: Estimation with Applications to Tracking and Navigation. Wiley, New York (2002)
6. Becker, S., Hübner, W., Arens, M.: State estimation for tracking in imagespace with a de- and re-coupled IMM filter. Multimedia Tools Appl. **77**(15), 20207–20226 (2018). https://doi.org/10.1007/s11042-017-5324-3

7. Becker, S., Hug, R., Hübner, W., Arens, M.: RED: a simple but effective baseline predictor for the *TrajNet* benchmark. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11131, pp. 138–153. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11015-4_13
8. Becker, S., Münch, D., Kieritz, H., Hübner, W., Arens, M.: Detecting abandoned objects using interacting multiple models. In: Proceedings of SPIE, Optics and Photonics for Counterterrorism, Crime Fighting, and Defence, vol. 9652 (2015)
9. Blom, H., Bar-Shalom, Y.: The interacting multiple model algorithm for systems with Markovian switching coefficients. Trans. Autom. Control **33**(8), 780–783 (1988). https://doi.org/10.1109/9.1299
10. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., Bengio, Y.: A recurrent latent variable model for sequential data. In: Advances in Neural Information Processing Systems (NIPS) (2015)
11. Deo, N., Trivedi, M.M.: Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs. In: Intelligent Vehicles Symposium (IV), pp. 1179–1184. IEEE (2018). https://doi.org/10.1109/IVS.2018.8500493
12. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2015)
13. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649 (2013). https://doi.org/10.1109/ICASSP.2013.6638947
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
15. Hug, R., Becker, S., Hübner, W., Arens, M.: On the reliability of LSTM-MDL models for pedestrian trajectory prediction. In: International Workshop on Representations, Analysis and Recognition of Shape and Motion from Imaging Data (RFMI), Savoie, France (2017)
16. Hug, R., Becker, S., Hübner, W., Arens, M.: Particle-based pedestrian path prediction using LSTM-MDL models. In: International Conference on Intelligent Transportation Systems (ITSC), pp. 2684–2691 (2018). https://doi.org/10.1109/ITSC.2018.8569478
17. Kalman, R.E.: A new approach to linear filtering and prediction problems. ASME J. Basic Eng. **82**, 35–45 (1960)
18. Keller, C.G., Hermes, C., Gavrila, D.M.: Will the pedestrian cross? Probabilistic path prediction based on learned motion features. In: Mester, R., Felsberg, M. (eds.) Pattern Recogn., pp. 386–395. Springer, Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-23123-0_39
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference for Learning Representations (ICLR) (2015)
20. Kooij, J.F.P., Flohr, F., Pool, E.A.I., Gavrila, D.M.: Context-based path prediction for targets with switching dynamics. Int. J. Comput. Vis. (2018). https://doi.org/10.1007/s11263-018-1104-4
21. Kooij, J.F.P., Schneider, N., Flohr, F., Gavrila, D.M.: Context-based pedestrian path prediction. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 618–633. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_40
22. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: International Conference on Computer Vision (ICCV). pp. 261–268. IEEE (2009). https://doi.org/10.1109/ICCV.2009.5459260

23. Särkkä, S.: Bayesian Filtering and Smoothing. Institute of Mathematical Statistics Textbooks, Cambridge University Press, Cambridge (2013). https://doi.org/10.1017/CBO9781139344203
24. Schneider, N., Gavrila, D.M.: Pedestrian path prediction with recursive Bayesian filters: a comparative study. In: Weickert, J., Hein, M., Schiele, B. (eds.) GCPR 2013. LNCS, vol. 8142, pp. 174–183. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40602-7_18
25. Stierlin, S., Dietmayer, K.: Scale change and TTC filter for longitudinal vehicle control based on monocular video. In: International Conference on Intelligent Transportation Systems (ITSC), pp. 528–533 (2012). https://doi.org/10.1109/ITSC.2012.6338681
26. Teknom, K.: Microscopic pedestrian flow characteristics: development of an image processing data collection and simulation model. Ph.D. thesis, Tohoku University (2002)
27. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning (ICML), vol. 37, pp. 2048–2057. PMLR, Lille (2015)