Environment-aware Pedestrian Trajectory Prediction for Autonomous Driving

Michael Gerovitch

Mentor: Dr. Igor Gilitchenski (MIT CSAIL)

Abstract. People's safety is a primary concern in autonomous driving. There exist efficient methods for identifying static obstacles. However, the prediction of future trajectories of moving elements, such as pedestrians crossing a street, is a much more challenging problem. A promising direction of research is the use of machine learning algorithms with location bias maps. Our goal was to further explore this idea by training an interchangeable location bias map, a location-specific feature that is added into the middle of a convolutional neural network. For different locations, we used different location bias maps to allow the network to learn from different setting contexts without overfitting to a specific setting. Using pre-annotated video footage of pedestrians moving around in crowded areas, we implemented a pedestrian behavior encoding scheme to generate input and output volumes for the neural network. Using this encoding scheme, we trained our neural network and interchangeable location bias map. Our research demonstrates that the network with an interchangeable location bias map can predict realistic pedestrian trajectories even when trained simultaneously in multiple settings.

1 Introduction

The movement of pedestrians largely depends on their surroundings, whether these are physical barriers in their way (i.e. walls, poles, train tracks, etc.) or other pedestrians surrounding them. These factors shape the behavior of pedestrians. Therefore, in order to be able to predict pedestrian movement, it is crucial to understand the relationships among pedestrians and their environment.

Autonomous vehicles must be able to predict the future trajectories of dynamic agents, such as pedestrians and cyclists, in order to navigate safely, without having to suddenly swerve or stop. Moving vehicles must avoid pedestrians in a constantly-changing scenery. Though there are plenty of works using deep learning to predict pedestrian movement, the use of trainable location-specific features is limited.

As a result, networks often overfit to the location of the training dataset, requiring the use of a completely new network for a different dataset. We suggest using an interchangeable location bias map, which is the only part of the network that is changed when a new dataset is used.

To analyze the effect of the interchangeable location bias map, we use pre-annotated videos from two static vantage points. We use an encoding scheme

that avoids ambiguity by creating displacement volumes [14], making up the input and output of our network.

In summary, the contributions of our work are 1. the creation of a new architecture that uses location specific maps, 2. integration of ETH and Hotel datasets for using in Behavior-CNN, and 3. experimentation with evaluation metrics of the architecture.

2 Related work

Older techniques exist that use non-deep learning algorithms for pedestrian behavior prediction [11–13, 15]. More recently, there has been an abundance of research that uses deep learning to predict pedestrian behavior [1–4, 9].

Pedestrian behavior often depends on the other pedestrians around them. There have been a lot of approaches to take this into account [5, 6, 16].

Most prediction networks are forced to output only one predicted path for a pedestrian. However, a pedestrian may have more than one choice for where to turn. Modeling this uncertainty and multi-modality can yield more successful prediction networks [8].

Though networks that take context into account exist [7], trainable location features have not been fully explored.

One approach explored a location bias map with a behavioral convolutional neural network [14], but this implementation did not utilize an interchangeable feature.

3 Methodology

3.1 Dataset: Displacement Volume

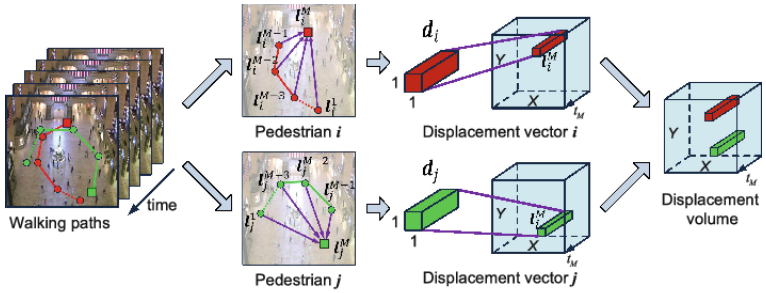
Our two datasets were created from the two annotated video files from the BIWI Walking Pedestrians dataset collected by Stefano Pellegrini and Andreas Ess at ETHZürich [10]. One dataset, Hotel, was collected from a hotel window overlooking the a bus stop on a busy city street. People walked by on the sidewalk, as well as entered and exited buses that occasionally pulled up. The other dataset was collected from the ETHZürich building and overlooked the busy entrance to the building. The annotations included the position of every pedestrian present at each time step.

Though this data differs from that which a car might see because of (1) the top-down perspective and (2) the static surroundings, this data can be used to analyze the validity of a location bias map in general.

Figure 1 illustrates the movement of a single pedestrian. The process of creating input volumes from all pedestrians for a given time interval is depicted in Figure 2.

(a) (b) (c)

Fig. 1: Three consecutive frames illustrating progression of pedestrian in blue square.



(a) (b) (c) (d)

Fig. 2: Displacement vectors are created based on the position of each pedestrian at the final frame for a total time M (a-b). They are then placed in a singular volume based on their final position (c-d). Adapted from [14].

First, we re-scale the pedestrian positions so that they fall in the $[0, 1]$ range. The correct position of pedestrian i at time step m, p_i, $[x_i^m, y_i^m]$. We re-scale all the positions so that the new p_i's position follows the following formula:

$$[x_i^m, y_i^m] = [\frac{x_i^m - x_{min}}{x_{max} - x_{min}}, \frac{y_i^m - y_{min}}{y_{max} - y_{min}}] \tag{1}$$

Then, we create each displacement vector based on the displacement of each pedestrian from their position at the final time step, m. The displacement for p_i over M time steps a vector of length $2 \times M$ is:

$$d_i = [x_i^M - x_i^1, y_i^M - y_i^1, x_i^M - x_i^2, y_i^M - y_i^2, \ldots, x_i^M - x_i^M, y_i^M - y_i^M] \tag{2}$$

Next, the displacement vectors are placed into a displacement volume (D with dimensions X by Y by $2M$ initially set to contain only zeros) based on the final positions of the pedestrian. $D(X \times x_i^m, Y \times y_i^m, :)$ is set to $d_i + 1^T$, where 1^2M is a vector of all ones of length $2M$. This inserts the displacement vector into the displacement volume and changes the scale of the values to be in the range $[0, 2]$, making it easier to separate the pedestrians from non-pedestrians.

The above process describes the creation of the input displacement volumes. Output volumes are based on the final position of each pedestrian in the input

volume. Otherwise, the process is the same. Figure 2 demonstrates this methodology.

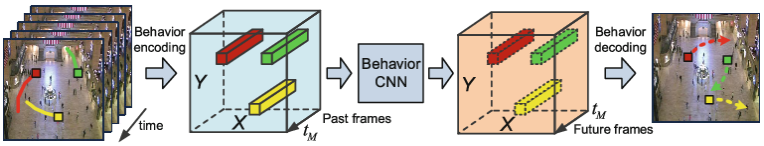


Fig. 3: The input of the network, shown in (a-b) are based on the final position of the pedestrians. The output (c-d) is the volume of future displacements of the pedestrians. It is also based on the final positions in the output. Adapted from [14].

3.2 Network Architecture

Following the network architecture of Figure 3, we create three convolution layers, one element-wise addition layer (location bias map), a max-pooling layer, three more convolution layers, and one deconvolution layer that brings the dimensions back to that of the input volume. ReLU layers are used after the first five convolution layers.

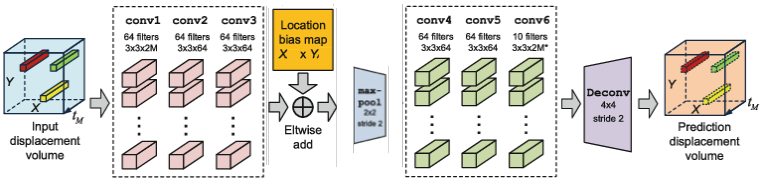


Fig. 4: Convolution layers (b and d) are used around the interchangeable location bias map and max-pool features (c). The deconvolution layer (e) changes the dimensions back to the required size. Adapted from [14].

The convolution layers are used to take into account surrounding pedestrians. The max-pool feature is used to double the receptive field and increase the influence of a pedestrian's neighbors on their behavior.

A padding of zeros is used to keep the spatial size unchanged. This network is expected to learn human walking behavior based on their past trajectory, the immovable objects around them, and surrounding pedestrians.

3.3 Loss Function and Optimizer

The cost function of the training is the Euclidean distance between the predicted and expected displacement volumes only for the location with a pedestrian:

$$L = \frac{1}{N}\frac{1}{M}\sum_{n=1}^{N}\sum_{m=1}^{M}(d_n[2m]^2 - \hat{d}_n[2m]^2) + (d_n[2m+1]^2 - \hat{d}_n[2m+1]^2) \quad (3)$$

L, or loss, is the average of all losses between each pedestrian's predicted and expected displacement vector in a given time frame, where N is the total number of pedestrians, M is the number of time steps, d_i is the ground truth displacement vector for pedestrian i, and $\hat{d}_i$ is the predicted displacement vector for pedestrian i.

Adam is the optimizer we use to train the network weights since it adjusts the learning curve as training progresses, speeding up the optimization.

3.4 Training Scheme

The datasets were each divided into three sections: 70% for training, 10% for validation during training, and 20% for evaluation. Next, the corresponding sections between the two datasets were combined. The validation set was used to verify that the network would not overfit to the training data. The evaluation set was used to test the trained network.

The training was split up into two phases: first, we trained just the first three convolution layers, and then, we added the rest of the network, which included the bias map. This by-part training method allowed for faster training of the bias map, as it was added to an already partially trained network.

4 Experimental Setup

4.1 Hyperparameter Tuning

The number of epochs, dimensions of the displacement volumes, and batch sizes were set as tunable hyperparameters.

Epochs. Even with epochs as large as 20 or 30, the model did not overfit. This is likely due to the complex nature of the data; some pedestrians often took turns while others stood in the same spot for a long time. The upper bound of epochs was determined solely by computation time.

Dimensions of volumes. Time frames between three and five were used. This meant the model was predicting a few seconds of movement into the future – similar to average amount of time a driver has to see and react to a pedestrian. However, experiments with larger time frames (10-12) demonstrated that the model could still predict accurately.

Volume height and width were set to 20. Higher values slowed down computation too much and did not yield significantly better results.

Batch size. A batch size of 100 was used, as it yielded the fastest run time.

4.2 Other cost functions

We tried out various cost functions such as Mean Squared Error, which compared the entire predicted and expected displacement volumes, but this did not work

well because the model focused too much on predicting zeros for places without pedestrians.

We also tried a cost function that was based only on the final position of each pedestrian in the expected and predicted displacement vectors, ignoring zeros and any other positions of the pedestrians. However, the network still learned poorly. Acknowledging that this error is very important for realistic assessment of the model, we calculated it on the training data as we went to verify that the network was accurately predicting the final position.

5 Results

The learning curve in Figure 4 demonstrates a successful trial where the training loss is just below the validation loss as both plateau at a value close to zero.

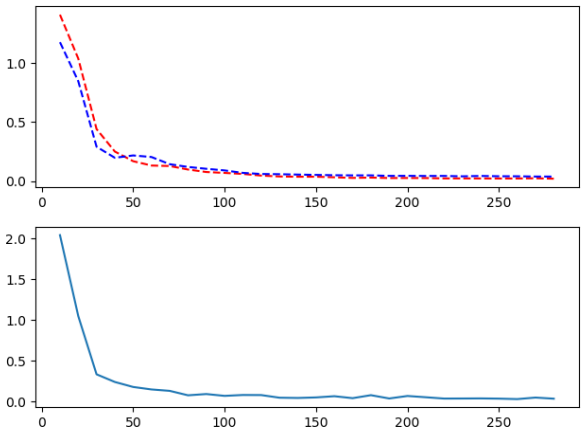


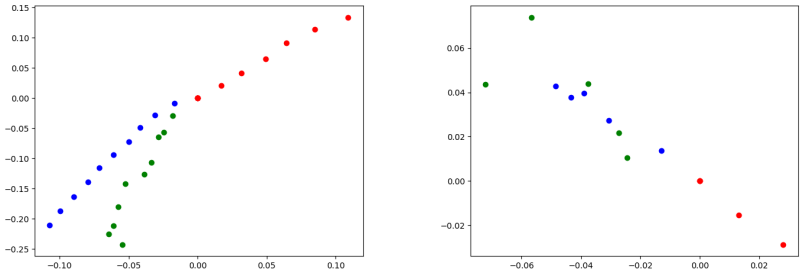
Fig. 5: Learning curve on the top (red: training; blue: validation). Final error on the bottom.

5.1 Graphs

Figures 5 and 6 are visualizations of some of the input (red), expected output (blue), and predicted output (green) vectors from the evaluation datasets. They have been collected from various trials so they have varying hyperparameters. The losses are calculated as the average displacement error described in 3.3.

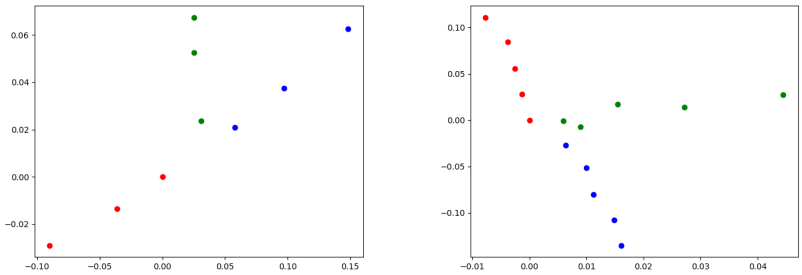
Figure 5 demonstrates that the interchangeable location bias map can accurately predict pedestrian trajectories. This is true for varying time frames.

Figure 6 suggests that a multi-modal approach should be considered. The path predicted is realistic, but it is not correct. However, we cannot know the certainty with which the network picked this path over the correct path with implementing multi-modality.



(a) L = 0.0008 (b) L = 0.0003

Fig. 6: Fairly accurate and realist predictions.



(a) L = 0.0085 (b) L = 0.0102

Fig. 7: Realistic, but not accurate predictions.

6 Conclusions and future work

We will continue to investigate the effectiveness of the interchangeable location bias map. Though the results thus far are promising, it is still not clear how large an advantage it provides for autonomous driving, as the location constantly changes.

Additionally, the location bias map can be turned into its own layer in the network or it can be concatenated with the previous layer instead of the current element-wise addition. This could increase its influence over the network, which would help us better understand its effect on performance.

Furthermore, since the destination of a pedestrian is more import than their path, a loss function that more significantly emphasizes final position error would likely help the network learn better.

Finally, cars must be able to avoid not only pedestrians, but other drivers, cyclists, buses, and scooterists. Our work can be extended to all of these dynamic elements.

7 Acknowledgements

This research was motivated and supported by Dr. Igor Gilitschenski, CSAIL MIT, and the MIT PRIMES program, all of whom provided resources that made this research possible.

References

1. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. Trans. Sys. Man Cyber Part C **38**(2), 156–172 (Mar 2008). https://doi.org/10.1109/TSMCC.2007.913919, https://doi.org/10.1109/TSMCC.2007.913919
2. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR **abs/1406.1078** (2014), http://arxiv.org/abs/1406.1078
3. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR **abs/1412.3555** (2014), http://arxiv.org/abs/1412.3555
4. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. CoRR **abs/1411.4389** (2014), http://arxiv.org/abs/1411.4389
5. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. CoRR **abs/1803.10892** (2018), http://arxiv.org/abs/1803.10892
6. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Physical Review E **51**(5), 4282–4286 (May 1995). https://doi.org/10.1103/physreve.51.4282, http://dx.doi.org/10.1103/PhysRevE.51.4282
7. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H.S., Chandraker, M.K.: DESIRE: distant future prediction in dynamic scenes with interacting agents. CoRR **abs/1704.04394** (2017), http://arxiv.org/abs/1704.04394
8. Makansi, O., Ilg, E., Çiçek, Ö., Brox, T.: Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. CoRR **abs/1906.03631** (2019), http://arxiv.org/abs/1906.03631
9. Park, H., Hwang, J., Niu, Y., Shi, J.: Egocentric future localization. In: Proceedings - 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016. pp. 4697–4705. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society (12 2016). https://doi.org/10.1109/CVPR.2016.508
10. Pellegrini, S., Ess, A., Schindler, K., Gool, L.V.: You'll never walk alone: Modeling social behavior for multi-target tracking. 2009 IEEE 12th International Conference on Computer Vision pp. 261–268 (2009)
11. Tomasi, C., Kanade, T.: Detection and tracking of point features. Tech. rep., International Journal of Computer Vision (1991)
12. Williams, C.K.I.: Prediction with gaussian processes: From linear regression to linear prediction and beyond. In: Learning and Inference in Graphical Models. pp. 599–621. Kluwer (1997)

13. Xu, L.Q., Landabaso, J.L., Lei, B.: Segmentation and tracking of multiple moving objects for intelligent video analysis. BT Technology Journal **22**(3), 140–150 (Jul 2004). https://doi.org/10.1023/B:BTTJ.0000047128.53316.f2, https://doi.org/10.1023/B:BTTJ.0000047128.53316.f2
14. Yi, S., Li, H., Wang, X.: Pedestrian behavior understanding and prediction with deep neural networks. In: ECCV (2016)
15. Ziebart, B.D., Maas, A., Bagnell, J.A., Dey, A.K.: Maximum entropy inverse reinforcement learning. In: Proc. AAAI. pp. 1433–1438 (2008)
16. Zou, H., Su, H., Song, S., Zhu, J.: Understanding human behaviors in crowds by imitating the decision-making process. CoRR **abs/1801.08391** (2018), http://arxiv.org/abs/1801.08391