This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Pedestrian and Ego-vehicle Trajectory Prediction from Monocular Camera

Lukáš Neumann
Visual Recognition Group
Faculty of Electrical Engineering
Czech Technical University in Prague
neumalu1@fel.cvut.cz

Andrea Vedaldi
Visual Geometry Group
Department of Engineering Science
University of Oxford
vedaldi@robots.ox.ac.uk

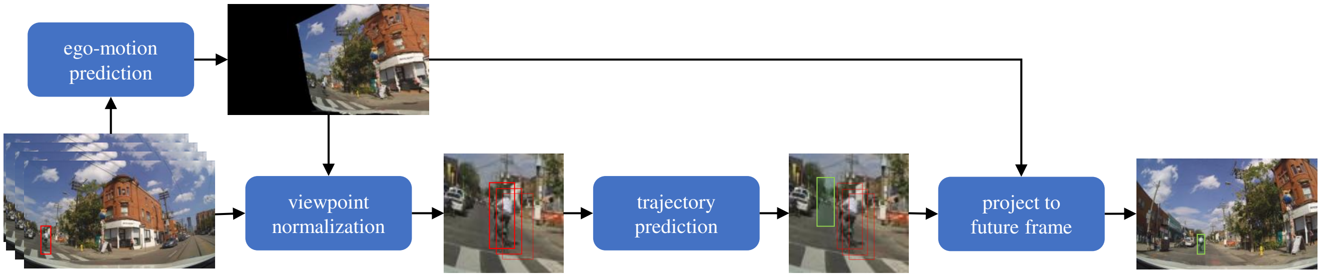

Figure 1: By inferring current and predicting future ego-motion, we can "subtract" vehicle movement to create a normalized view of the pedestrian as if it was captured by a static camera, allowing to observe and predict the intrinsic pedestrian trajectory.

Abstract

Predicting future pedestrian trajectory is a crucial component of autonomous driving systems, as recognizing critical situations based only on current pedestrian position may come too late for any meaningful corrective action (e.g. breaking) to take place. In this paper, we propose a new method to predict future position of pedestrians, with respect to a predicted future position of the ego-vehicle, thus giving a assistive/autonomous driving system sufficient time to respond. The method explicitly disentangles actual movement of pedestrians in real world from the ego-motion of the vehicle, using a future pose prediction network trained in self-supervised fashion, which allows the method to observe and predict the intrinsic pedestrian motion in a normalised view, that captures the same real-world location across multiple frames.

The method is evaluated on two public datasets, where it achieves state-of-the-art results in pedestrian trajectory prediction from an on-board camera.

1. Introduction

Predicting the future behavior of objects in images and videos is of considerable importance in applications, especially in areas such as robotics or automotive systems. For example, predicting future trajectory of pedestrians is a crucial component of autonomous and assistive driving systems, as recognizing critical situations only based on the current pedestrian position (*e.g.* a child at the margin of the road) may come too late for any meaningful corrective action (*e.g.* breaking) to be effective. For any such prediction algorithm to be widely adopted, however, it is crucial that its sensing hardware requirements are as low as possible. This is the reason why methods based on a single camera mounted on the vehicle — also known as *first-person view monocular* methods — are raising a lot of attention [2, 5, 10, 11, 26, 28].

Pedestrian trajectory prediction has been extensively studied in a static or a bird-eye view camera setup [1, 12, 18, 23], but these methods typically fail in dynamic scenes captured by an on-board camera due to constantly changing camera viewpoint, occlusions and other scene dynamics. Moreover, these methods typically rely on discovering pedestrian motion patterns to infer future trajectory, which is not possible in the context of on-board videos, where the movement of the car, not the pedestrian movement, is the main observed effect: even a pedestrian who stands still appears in different image position in every frame, creating an apparent motion in the 2D pixel space.

In this paper, we thus propose a method[1] that explicitly

[1] The source code is available at https://gitlab.com/lukeN86/pedFutureTracking

disentangles the two sources of motion — the actual movement of the pedestrian in real world (*e.g.* walking, running) and the ego-motion of the vehicle, as it drives around. The key observation is that the motion of a pedestrian only affects a specific part of the image, whereas the motion of the car (ego-motion) affects the appearance of the whole scene. Using a self-supervised training paradigm, we train a ego-motion prediction network, which infers the ego-motion of the vehicle where the camera is mounted. This network is trained in a similar manner to current self-supervised monocular depth estimator systems [10,11,30], but with two differences: we are as much interested in recovering egomotion as in recovering depth; and we predict the egomotion deep into the future. In this manner, we can "subtract" the predicted motion of the vehicle and observe and predict the intrinsic pedestrian motion in a normalised view, which captures the same real-world location across multiple frames.

The view normalization then allows us to use a very simple model to predict the intrinsic motion of pedestrians and yet achieve state-of-the-art results on two public datasets, suggesting that indeed properly disentangling ego-motion is a crucial component in these systems. Compared to previous designs that used complex predictors such as LSTMs, our predictor is much simpler. Furthermore, our method does not require additional annotations compare to these baselines as it learns to interpret the vehicle's egomotion in an unsupervised fashion.

As for the practical impact, the resulting method requires solely on a monocular camera, which expands its possible applications, because it can be used either as a standalone method in a current-generation vehicles which do not have any advanced LiDAR sensors, or it can be used as a redundancy system in the new generation of autonomous cars. Because of the pedestrian view normalization, the method could also be incorporated into more sophisticated trajectory prediction methods that work with stationary camera.

To summarise, we make three key contributions in this paper: (i) we introduce a new self-supervised framework for ego-vehicle movement prediction, (ii) we use the latter to disentangle the motion of the vehicle from the intrinsic motion of pedestrians, allowing to predict pedestrian trajectories from a normalised sequence of patches observing the same part of the scene regardless of the variable viewpoint and (iii) we show that, when the pedestrian viewpoint is normalised in this manner, a simple linear model for trajectory prediction outperforms the traditional LSTM sequence output used in literature.

2. Related Work

Monocular Depth Self-supervised training has been widely exploited in the depth estimation literature [10,11,30]. The idea of using view synthetics with a depth and a 6-DoF pose network to supervise the training was first introduced by Zhou et al. [30], and was further improved in [3–5,10,19]. Most notably, Godard et al. [11] improved the depth estimation accuracy by modelling occlusions, stationary objects and by calculating the loss across multiple scales.

Our method in contrast is not focused on depth estimation, and rather than calculating pose change between two subsequent and already observed frames as in [11], our method predicts poses for unseen future frames, using a novel single encoder-multiple decoder heads architecture.

Trajectory Prediction from Static Camera Many models consider pedestrian trajectory prediction as a 2D problem, where pedestrians are observed from a static camera facing downwards (birds-eye view). These models focus on human to human interactions, (static) obstacles modelling and are able to jointly reason about the whole observed scene. One of the most prominent models in this category is Social LSTM [1], which introduces a new pooling layer which jointly combines all trajectories and interaction to form a single prediction for the scene. Other methods use GANs [12, 18, 23] to better model social interactions and further improve prediction accuracy.

The main drawback of these methods is the requirement for a static camera viewpoint, which typically captures a single static scene where pedestrians move. This assumption obviously does not hold for on-board cameras, where the scene is changing constantly and where the main source of change in pedestrian bounding box position is the ego-vehicle, not the pedestrian itself.

Trajectory Prediction from On-board Camera Fang and López [7] used a human pose detection CNN to predict a binary flag encoding whether a pedestrian intents to cross to cross the road or not.

Bhattacharyya et al. [2] used a RNN for odometry prediction (speed + steering angle), followed by a separate RNN for pedestrian trajectory prediction. Same as our method, it uses last available visual information to predict future movement of the vehicle, but it does not exploit any of the known geometric relationships of a moving camera, their model only predicts two scalars whereas we predict a complete 6-DoF pose, and thanks to the explicit geometrical semantics of our predictor we are able to also meaningfully extract and use image features of the pedestrian itself. Last but not least, their method requires annotations for future movement prediction, whereas our egomotion prediction is trained self-supervised.

Rasouli et al. [20] expand this further by adding a third RNN for pedestrian intention, which takes a pedestrian image patch and predicts pedestrian's intention. This allows them to exploit visual feature of individual pedestrians, but each pedestrian needs to be first hand-labelled as going/not going to cross. The model is also unable to capture actual pedestrian movement in the scene, because each patch is

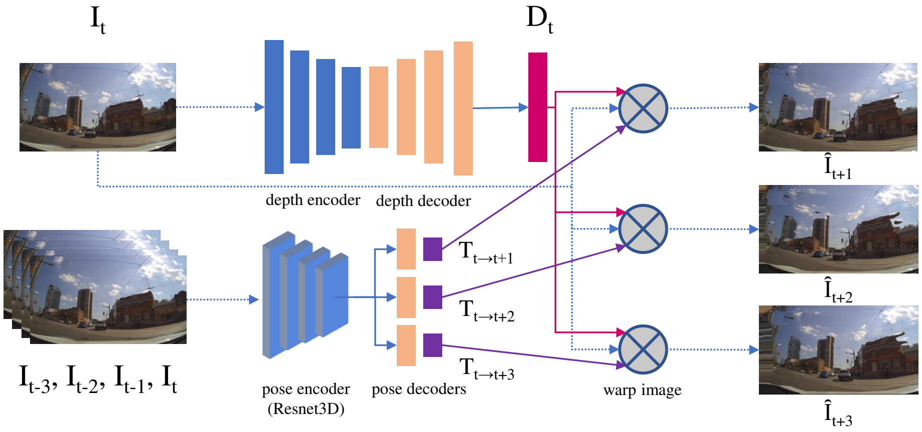

Figure 2: Ego-vehicle motion prediction. Having observed images frames up to time t, the network predicts future vehicle pose transformations $T_{t\to t+1}, T_{t\to t+2}, \ldots, T_{t\to t+P}$ with respect to the current frame t.

centered around current pedestrian bounding box in every frame.

Yao et al. [27] use a multi-stream RNN to combine pedestrian location, vehicle motion and optical flow estimate. The ego-motion is however only represented as 2D rotation and translation in the pixel space, which we show is suboptimal (see section 4.4), and their model does not incorporate visual features of the observed agents (pedestrians). This work was further extended in [26], where conditional variational autoencoder is added to predict multi-modal future trajectories.

3. Method

We first describe our method to learn future egomotion from observation of the past using self-supervised learning (section 3.1). We then apply that to normalizing pedestrian detections and thus facilitate predicting their future trajectories (section 3.2).

3.1. Self-Supervised Vehicle Motion Prediction

Our ego-vehicle prediction network is inspired by the self-supervised depth estimation literature [8, 10, 11]. Given two sequential frames I_{t-1} and $I_t \in \mathbb{R}^{3\times H\times W}$, a deep network is trained to produce a dense depth map $D_t \in \mathbb{R}_+^{H\times W}$ and a pose transformation estimate $T_{t-1\to t} \in SE(3)$, by minimising appearance loss between the original second frame I_t and the synthesised version of the second frame, which is warped from the first frame I_{t-1}, using the inferred D_t and $T_{t-1\to t}$ values. This form of training does not require any ground truth information (apart from camera intrinsics $K \in \mathbb{R}^{3\times 3}$, which we assume to be known), as the training signal originates in self-supervision from the observed sequence of two subsequent frames.

In our method, we expand this paradigm to make a network predict future ego-vehicle poses (see fig. 2). Having observed several frames up to time t, the pose network is tasked to predict a transformation $T_{t\to t+f}$, which transforms the current vehicle position at the time t to the expected position in the future frame $t+f$ (in our case, the network predicts future poses up to $f = 45$ frames into the future).

More formally, a pose encoder network

$$\Phi : \mathbb{R}^{(L+1)\times 3\times H\times W} \to \mathbb{R}^D$$

takes a sequence of past images $I_{t-L}, \ldots, I_{t-1}, I_t$ and generates shared features for pose decoder networks $\Psi_f : \mathbb{R}^D \to SO(3)$, each predicting a transformation $T_{t\to t+f}$ for a specific future frame f (in this way, the pose encoder Φ allows to share most of the learned parameters between individual predictions). The networks are trained by minimizing the photometric loss $\mathcal{L}_p$ between observed frame I_{t+f} and the synthesised image $\hat{I}_{t+f}$:

$$\hat{I}_{t+f} = \mathcal{W}(I_t; D_t, T_{t\to t+f}, K), \tag{1}$$

$$T_{t\to t+f} = \Psi_f(\Phi(I_{t-L}, \ldots, I_{t-1}, I_t)), \tag{2}$$

$$D_t = \Xi(I_t), \tag{3}$$

$$\mathcal{L}_p(I_{t+f}, \hat{I}_{t+f}) = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathcal{P}(I_{t+f}^{ij}, \hat{I}_{t+f}^{ij}), \tag{4}$$

where Ξ is the monocular depth prediction network [11], $\mathcal{W}$ is the image warping operation [14] and $\mathcal{P}$ is the photometric loss as the per-pixel sum of L^1 and SSIM difference [25, 29].

The model above is thus similar to current self-supervised monocular depth estimation networks, but there are two key differences. The first one is that, in monocular depth prediction the predicted egomotion is just a by-product and is discarded, whereas for us this is just as important as depth. The second and more fundamental one is that the information

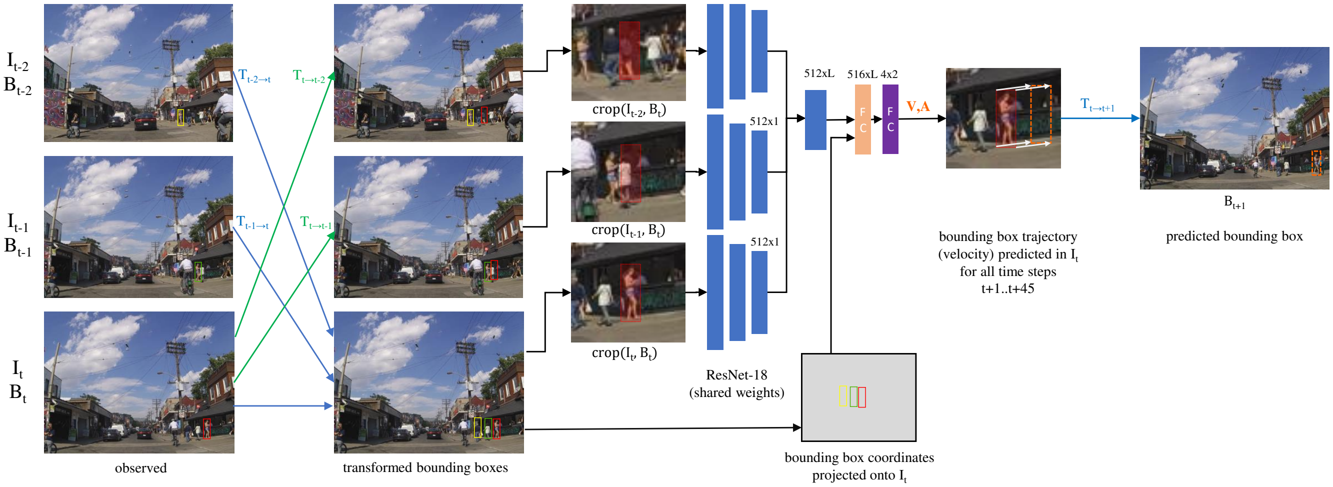

Figure 3: Pedestrian view normalization and trajectory prediction. The trajectory prediction operates in 2D space, where the observed pedestrian patches are aligned and normalized to capture the same location in real world, irrespective of pedestrian position in previous frames. This allows our method to capture and predict actual pedestrian movement, as if the scene was captured by a static camera (where in fact the camera is mounted on a moving car).

flow in the model is different. In particular, in eq. (2) the pose encoder takes as input frames in the range $t - L, \dots, t$, but outputs the camera poses for times $t + 1, \dots, t + f$, thereby predicting the future. In self-supervised monocular depth prediction, the network instead outputs the poses for the same frames passed to the pose predictor, as there is no need to predict the future at all. The factorization of eq. (2) in intermediate features and time-specific decoder heads is also new and is required in order to predict the poses for many different frames in an efficient manner.

Implenetation details. As pose encoder Φ, we use ResNet3D [24] with 18 layers, pretrained on the Kinetics-400 dataset [15]. For each pose prediction head Ψ_f, we then use a separate decoder, which is a simple stack of four 3×3 convolutions and ReLUs followed by an average pooling layer for the final prediction.

3.2. Normalization and Trajectory Prediction

The second component of our method is the trajectory predictor for individual agents, in our case pedestrians. The trajectory predictors operates in a 2D space, where the observed images as well as observed bounding boxes have been normalised using the predicted egomotion (viewpoint transformation) in the manner discussed above. The prediction of future pedestrian trajectories is also made within the same normalised space.

Intuitively, the viewpoint normalization ensures that the bounding box of a pedestrian, as detected in the past pr predicted in the future, does not move as long as the pedestrian is standing still in the real world. For this, the system must compensate for the fact that the car is moving and turning, so that in the original image data the pedestrian bounding box appears to move significantly, including getting bigger as the ego-vehicle drives towards the pedestrian. Equally, if the pedestrian is indeed moving in the world space, the viewpoint normalization ensures that such sequence is observed as if it was captured from a stationary camera, so that the motion becomes apparent (see fig. 3).

Moving boxes. Let $B_a$be a 2D bounding box tightly enclosing a pedestrian in a view a. Furthermore, let $T_{a\to b} \in SO(3)$ be the camera view transformation from view a to view b. Under the assumption that the object enclosed in B_a is static in the 3D world, we can find an approximate location B_a^b for the bounding box in view b as follows.

To this end, we express B_a as the matrix of its four corners:

$$B_a = \begin{bmatrix} i_{\min} & i_{\max} & i_{\max} & i_{\min} \\ j_{\min} & j_{\min} & j_{\max} & j_{\max} \end{bmatrix} \in \mathbb{R}^{2\times 4}.$$

If we denote by $\operatorname{supp} B_a$ the set of indices contained in the box, we can assign to it a depth value by averaging the corresponding depth predictions:

$$d(B_a) = \frac{1}{|\operatorname{supp} B_a|} \sum_{ij \in \operatorname{supp} B_a} [D_a]_{ij}$$

Furthermore, let $\dot{B}_a$ be the matrix obtained by adding a row of ones to B_a, thus expressing the vertices in homogeneous coordinates. We define the transformed box as:

$$B_a^b = T_{a\to b}(B_a) \quad \text{such that} \tag{5}$$

$$\dot{B}_a^b \propto K T_{a\to b} d(B_a) K^{-1} \dot{B}_a \tag{6}$$

Intuitively, this process amounts to pretending that the object (pedestrian) is planar, fronto-parallel, and situated at a distance $d(B_a)$ from the observer in view a. The viewpoint change then reduces to applying a certain homography to this box. Importantly, eq. (5) is a differentiable operator. Note that the resulting 'box' B_a^b is not necessarily axes-aligned any more; however, in our application this is approximately the case. When needed, we define the operator $\operatorname{encl} B_a^b$ to denote the tightest axis-aligned bounding box containing the shape thus obtained.

Predicting the future. We are now ready to explain how our future trajectory predictor network Γ works. At its core, the model assumes that the motion of the box can be described, in the normalised space, by a linear model. We thus write the future box $\hat{B}_{t+p}$ as:

$$\hat{B}_{t+p} = T_{t\to t+p}\left(B_t + pV + \frac{p^2}{2}A\right), \quad p = 1, \dots, P \tag{7}$$

Here $V, A \in \mathbb{R}^4$ are the parameters of a basic accelerated motion model for the box vertices in the normalised space. The goal of the network Γ is thus to output these two vectors.

In order to do so, the network Γ observers normalised image crops together with corresponding normalised bounding boxes. Specifically, consider time $t - \ell$ in the past. First, we take the position of the pedestrian at time $t - \ell$, as captured by box $B_{t-\ell}$, and projecting to the "present" view at time t by computing $B_{t-\ell}^t$. Second, we take the present location of the pedestrian B_t and crop and rescale the past image $I_{t-\ell}$ at the corresponding location by computing $\text{crop}(I_{t-\ell}, \operatorname{encl} B_t^{t-\ell})$. These two operations result in a pair

$$\mathcal{W}_\ell = (B_{t-\ell}^t, \text{crop}(I_{t-\ell}, \operatorname{encl} B_t^{t-\ell}))$$

which simulates observing the pedestrian and corresponding bounding box at time $t - \ell$ from the current viewpoint at time t (*i.e.* as if the camera did not move).

The network Γ takes as input an entire sequence of such observations:

$$(V, A) = \Gamma\Big(\mathcal{W}_{t-L}, \dots \mathcal{W}_{t-1}, \mathcal{W}_t,\Big) \tag{8}$$

Implementation details. The network Γ is a simple ResNet-18 network [13] where image patches are individually processed to generate 512 features per image, which are then concatenated to form a $512 \times (L + 1)$ matrix, where L is the length of observation (number of past observed images). The matrix is, together with observed bounding box coordinates, flattened to a single $(512 + 4) \times (L + 1)$ vector and fed through two fully-connected layers to create the final prediction $(V, A) \in \mathbb{R}^{4\times 2}$.

The network is trained in a fully-supervised fashion, using standard L^2 loss between predicted $\hat{B}_{t+i}$ and ground truth B_{t+i} bounding box positions

$$\mathcal{L}_b(B, \hat{B}) = \sum_{i=1}^{P} \left\| B_{t+i} - \hat{B}_{t+i} \right\|^2 \tag{9}$$

where the predicted positions are obtained as described before. We note that the ResNet-18 subnetwork is as it is standard practice pretrained on the ImageNet dataset [6] and its weights are shared across the sequence.

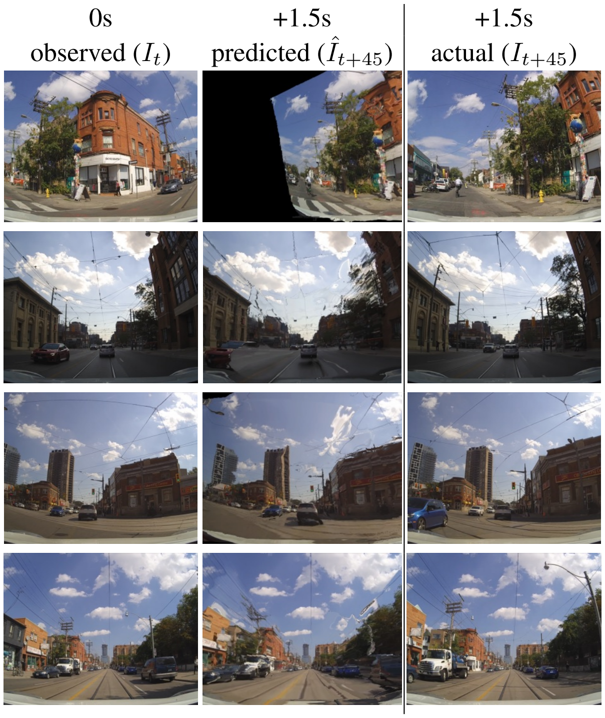

Figure 4: Ego-vehicle motion prediction samples from the PIE dataset. The network is able to predict turning (rotation) as well as forward motion (translation) up to 1.5seconds into the future, using purely visual input from a monocular camera

4. Experiments

In this section, our method is evaluated. We first describe the implementation and training details, followed by evaluation on two public datasets and ablation experiments.

4.1. Training

The ego-vehicle pose network (denoted Φ and Ψ_f in Section 3.1) was trained on the training subset of the respective dataset for 20 epochs, using Adam [17] optimiser with the learning rate of 10^{-4}. All frames are first resized to 640×352 pixels.

As it is standard in the literature [20, 26], the pose network observes previous 0.5 seconds and predicts position 1.5 seconds into the future. Given the frame

Method	MSE			MSE$_C$	MSE$_{CF}$
	0.5s	1.0s	1.5s	1.5s	1.5s
Linear [20]	123	477	1365	950	3983
LSTM [20]	172	330	911	837	3352
B-LSTM [2]	101	296	855	811	3259
PIETraj [20]	58	200	636	596	2477
FOL-X [27]	47	183	584	546	2303
MSPM [16]	57	182	565	526	2191
BiTraP-D [26]	**41**	161	511	481	1949
ours	42	**153**	**453**	**418**	**1683**

Table 1: Trajectory (bounding box) prediction error on the PIE dataset. MSE is the squared error of bounding boxes corners, MSE$_C$ and MSE$_{CF}$ is the error of bounding box center averaged over the whole sequence, respectively in the last frame.

rate of both datasets is 30fps, this would imply observing 15 previous frames and predicting position for future 45 frames. This would be computationally quite expensive, so the pose encoder network Φ only has 4 frames $I_{t-14}, I_{t-10}, I_{t-5}, I_t$ as its input. Equally, predicting future 45 frames would in theory require 45 pose decoder heads Ψ_f, but again for computational reasons we only use 6 heads - 4 heads $\Psi_{10}, \Psi_{20}, \Psi_{30}, \Psi_{45}$ to predict future pose transformations $T_{t\to t+10}, T_{t\to t+20}, T_{t\to t+30}, T_{t\to t+45}$ and 2 heads Ψ_{-14}, Ψ_{-7} to infer observed vehicle movement in the past frames $T_{t-14\to t}, T_{t-7\to t}$. During inference, the missing transformations are then approximated using simple linear interpolation from the two neighbouring transformations. The transformations in the opposite direction are then obtained using matrix (pseudo-)inverse, i.e. $T_{t+i\to t} = \left(T_{t\to t+i}\right)^{-1}$.

The pedestrian trajectory prediction network Γ (see Section 3.2) then observes a full sequence of 15 bounding box positions, alongside with again 4 image patches normalised to 128 × 128 pixels from the same image indices as above, and it outputs a sequence of 45 predicted bounding box positions. The network was trained again on the respective training subset for 60 epochs, using Adam optimiser with the learning rate of 10^{-3}.

The monocular depth prediction Ξ is the off-the-shelf Monodepth2 `mono_640x192` model [11] downloaded from the authors website[2] (the authors used KITTI dataset [9] to train their model).

4.2. PIE dataset

The Pedestrian Intention Estimation (PIE) dataset [20] is a large-scale first-person view driving dataset consisting of 911k frames split 50/40/10 between train/test/validation subsets, where in total 293k frames are annotated with 1.8k pedestrians. Since our ego-vehicle pose prediction network does not require any annotations, we use the full training subset for training the pose network. For the pedestrian trajectory prediction network, we then only used the annotated frames. The method was then evaluated on the PIE test subset, which contains 719 pedestrians in 330k frames.

The method is evaluated (see table 1) using the standard metrics [2,20] of Mean Square Error of bounding box corners (MSE) 0.5 second, 1 second and 1.5 second in the future, mean square error of bounding box center over the whole sequence (MSE$_C$) and the mean square of the bounding box center in the last frame (MSE$_{CF}$).

Our method outperforms existing methods by a significant margin, which increases as frames more in the future are considered. We hypothesise that this is because our method is able to better capture and anticipate future ego-vehicle motion, as i) our model is explicitly optimised to capture vehicle motion ii) the model can benefit from significantly more training data, as it can also exploit unannotated frames. The only exception is the prediction 0.5 second in the future, where BiTrap-D [26] performs slightly better - this is likely because the ego-motion within 0.5 second is not very significant, so the sophisticated pedestrian trajectory prediction model of BiTrap-D actually prevails our simple linear model.

The ego-vehicle pose network can not only predict forward motion, which is the most common case, but also turning to some degree (see Figure 4). We note that the network does not use any additional sensors, and therefore all predictions are based only on the observed image sequence and its dynamics.

Method	MSE			MSE$_C$	MSE$_{CF}$
	0.5s	1.0s	1.5s	1.5s	1.5s
Linear [20]	233	857	2303	1565	6111
LSTM [20]	289	569	1558	1473	5766
B-LSTM [2]	159	539	1535	1447	5615
PIETraj [20]	110	399	1248	1183	4780
FOL-X [27]	147	484	1374	1290	4924
BiTraP-D [26]	**93**	378	1206	1105	4565
ours	97	**373**	**1158**	**1042**	**4471**

Table 2: Trajectory (bounding box) prediction error on the JAAD dataset.

4.3. JAAD dataset

The Joint Attention for Autonomous Driving (JAAD) dataset [21, 22] is a driving dataset consisting of smaller discontinous video chunks consisting of 200-400 frames each, totalling into 82k frames containing 2.8k annotated pedestrians. We again used all frames from the training subset to train the pose prediction network, and pedestrian annotations to train the pedestrian trajectory prediction network.

Our method again outperforms existing methods (see table 2), yet the margin is smaller – this is chiefly because the

[2] https://github.com/nianticlabs/monodepth2

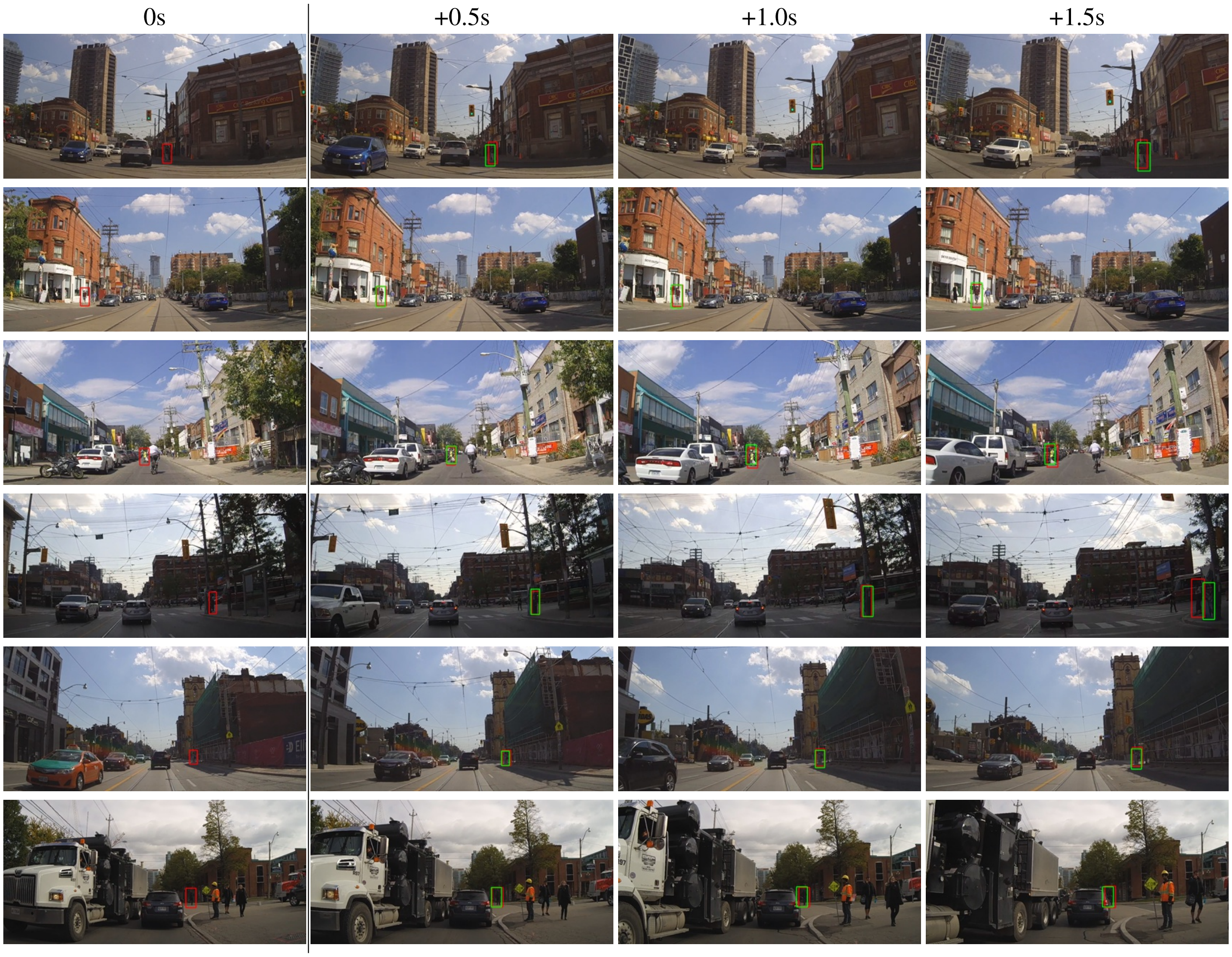

Figure 5: Pedestrian trajectory prediction qualitative samples on the PIE dataset. Ground truth position in red, predictions in green. Best viewed in colour.

JAAD dataset contains shorter but more varying sequences, so the pose prediction network would likely need more training data to better capture the variety. Additionally, 3 different cameras were used to capture the data, but only one camera has available calibration information, so we had to assume the intristics is the same for all cameras.

4.4. Ablation Experiments

Pedestrian Image. We first ablate the impact of using the normalised pedestrian image features in the pedestrian trajectory module (see table 3). We observe that even with just the bounding box sequence as input, the method outperforms some of the recent previous methods [20], even when using identical output encoding (LSTM). When the pedestrian image is added as additional 512 features to the LSTM, the error drops significantly, outperforming all previous methods. When the LSTM is then switched to our linear model, the accuracy is improved even further.

Additional inputs. We also experiment with additional inputs of information - vehicle speed from odometry until time t, speed for the predicted future segment, and a manually annotated intent flag capturing whether pedestrian is about to cross the road or not. For simplicity, we use inputs directly from the ground truth, however as it was shown in [2, 20], future speed and pedestrian intent can be also predicted with reasonably high accuracy. We show that using all three additional inputs does not compensate the loss of pedestrian image input, which suggests that pedestrian image indeed contains crucial information for future trajectory prediction and that the ego-motion prediction model is able to compensate vehicle movement so the speed information becomes less relevant. When additional inputs and pedestrian image are combined, the error drops even further, when compared to our method - but this may be partially due to the fact

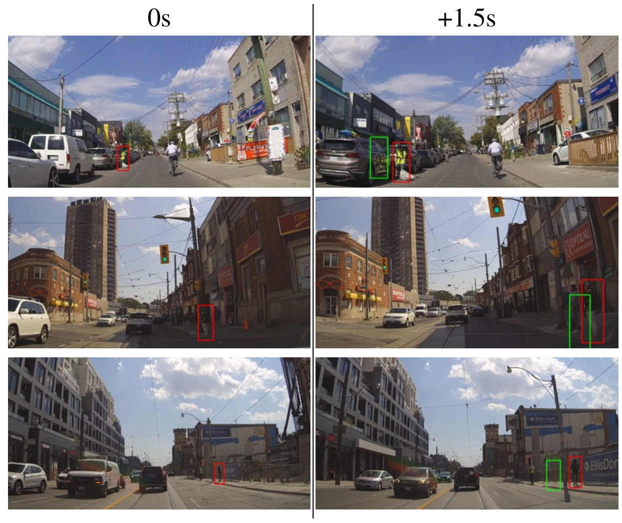

Figure 6: Failure modes on the PIE dataset. The predictor is confused by the surrounding context, expecting the pedestrian to enter the vehicle (top), bumps on the road unexpectedly causing the camera to tilt (middle), repeated structures causing incorrect forward motion estimate (bottom).

Bounding box	Pedestrian image	+ Past Speed	+ Future Speed	+ Crossing intention	ego-motion warping		MSE
		input				output	1.5s
✓	x	x	x	x	3D	LSTM	577
✓	✓	x	x	x	3D	LSTM	465
✓	x	✓	✓	✓	3D	LSTM	471
✓	✓	x	x	x	2D	LSTM	557
✓	✓	**x**	**x**	**x**	**3D**	**linear**	**453**
✓	✓	✓	✓	✓	3D	linear	439

Table 3: Ablation experiments of different input channels, ego-motion model and output format. Trajectory (bounding box) prediction error on the PIE dataset. (The configuration used in experiments denoted in bold)

we use ground truth speed and intention data rather than predictions in this ablation.

LSTM vs Linear model. Next, we compare our trajectory prediction model from eq. (7) with the standard LSTM model [20], which directly outputs a matrix of 45 × 4 bounding box coordinates, one row per time stamp. Our model yields better accuracy than the LSTM, likely because it has less degrees of freedom and it is therefore more robust. On the other hand, motion patterns of pedestrians captured in the data are not very complex, as people are typically walking in one direction, so a more complex model might be needed if more complex motion or interactions were to be modelled.

2D projection. Last but not least, we compare the impact of using 3D ego-motion prediction, compared to using a simple 2D transformation in pixel space, where the pose network predicts only a 2D translation and scale change. In this experiment, the depth network is not needed, as all transformations are directly in the pixel space, but as it can be seen in table 3, there is quite significant drop in accuracy.

5. Conclusion

In this paper, we proposed a method that explicitly disentangles actual movement of the pedestrians in real world from the ego-motion of the vehicle as it drives around. Using a self-supervised training paradigm, we trained an ego-motion prediction network, which infers the ego-motion of the vehicle and allows the method to observe and predict the intrinsic pedestrian motion in a normalised view, which captures the same real-world location across multiple frames.

The method was evaluated on two public datasets, where it achieved state-of-the-art results in pedestrian trajectory prediction from an on-board camera, whilst being conceptually and computationally simpler than the previous methods.

Acknowledgement

Lukas was supported by OP VVV funded project CZ.02.1.01/0.0/0.0/16019/0000765 “Research Center for Informatics” and by Continental Corporation. We are also grateful for compute resources made available by the RCI and VGG clusters.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016. 1, 2

[2] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4194–4202, 2018. 1, 2, 6, 7

[3] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019. 2

[4] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *CVPR Workshops*, pages 0–0, 2019. 2

[5] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monoc-

ular video: Connecting flow, depth, and camera. *ICCV*, pages 7062–7071, 2019. 1, 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[7] Zhijie Fang and Antonio M López. Is the pedestrian going to cross? answering by 2d pose estimation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1271–1276. IEEE, 2018. 2

[8] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, pages 740–756. Springer, 2016. 3

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 6

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1, 2, 3

[11] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019. 1, 2, 3, 6

[12] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. 1, 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc., 2015. 3

[15] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[16] Kyungdo Kim, Yoon Kyung Lee, Hyemin Ahn, Sowon Hahn, and Songhwai Oh. Pedestrian intention prediction for autonomous driving using a multiple stakeholder perspective model. 6

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks. In *Advances in Neural Information Processing Systems*, pages 137–146, 2019. 1, 2

[19] Robert McCraith, Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Monocular depth estimation with self-supervised instance adaptation, 2020. 2

[20] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6262–6271, 2019. 2, 5, 6, 7, 8

[21] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–213, 2017. 6

[22] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. It's not all about size: On the role of data properties in pedestrian detection. In *ECCVW*, 2018. 6

[23] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 1, 2

[24] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 4

[25] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 3

[26] Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du. BiTraP: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. *arXiv preprint arXiv:2007.14558*, 2020. 1, 3, 5, 6

[27] Yu Yao, Mingze Xu, Chiho Choi, David J Crandall, Ella M Atkins, and Behzad Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9711–9717. IEEE, 2019. 3, 6

[28] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018. 1

[29] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016. 3

[30] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 1851–1858, 2017. 2