Future Semantic Segmentation Leveraging 3D Information

Suhani Vora[1], Reza Mahjourian[1,2], Soeren Pirk[1], and Anelia Angelova[1]

[1] Google Brain, Mountain View CA 94043, USA.
[2] University of Texas at Austin
{svora,rezama,pirk,anelia}@google.com

Abstract. Predicting the future to anticipate the outcome of events and actions is a critical attribute of autonomous agents. In this work, we address the task of predicting future frame segmentation from a stream of monocular video by leveraging the 3D structure of the scene. Our framework is based on learnable sub-modules capable of predicting pixel-wise scene semantic labels, depth, and camera ego-motion of adjacent frames. Ultimately, we observe that leveraging 3D structure in the model facilitates successful positioning of objects in the 3D scene, achieving state of the art accuracy in future semantic segmentation.

Keywords: Depth, Egomotion, Semantic Segmentation, 3D Transformation

1 Introduction

Previous attempts at forecasting future video scene have centered around the task of predicting the RGB video frames [6, 9]. Prediction in RGB pixel space is difficult because of inherent ambiguities, and is unnecessary for many tasks. Alternatively, future semantic segmentation [7] aims to predict pixel-wise semantic labels of future frames given a stream of past video frames. Prior work on future semantic segmentation [7, 5] focuses on future prediction by directly mapping the inputs to the future frame in an unstructured manner. In this paper, we propose a novel, and complementary to prior work, approach for future segmentation, by introducing 3D scene and motion information and by proposing algorithms for learning each component, such as future ego-motion (Figure 1).

2 Related Work

Future frame segmentation was introduced in a video scene parsing method by Luc et al.[7], in which the primary task was pixel-wise semantic labeling of future frames by learning and using features, which enforce temporal consistency. This was accomplished with a network that was trained to directly predict the next 2D future frame segmentation from a series of four preceding frames. Luc et al. [7] further introduce an autoregressive convolutional neural network that is

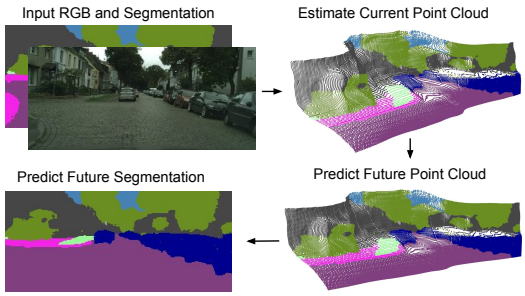

Fig. 1. High-level overview of our method. The input RGB frame plus a segmentation mask predicted from it are used to construct a 3D point cloud (via a learnable depth estimation). This point cloud is then transformed according to a prediction for future motion. Projecting the predicted point cloud into 2D space produces a segmentation prediction for the future without access to the future RGB frame or motion data

trained to iteratively generate multiple future frames. In our work we instead break down the problem into learnable structured sub-modules to construct a 3D segmentation prediction and use a projection to produce the 2D segmentation prediction. Jin et al [5] propose leveraging temporal consistency to enhance the segmentation task itself; future frames were used jointly with past frames, to produce the highest quality segmentation possible. Unlike this work we focus on producing future frame segmentation masks without employing future frame information.

3 3D Future Semantic Segmentation

To predict the segmentation of a future frame $(t + 3j)$ for $j \in \{1, 2, 3, 4, 5\}$ from the current frame (t), we propose to reconstruct the scene in 3D using learnable sub-modules as described below. An overview of the method is shown in Figure 2.

3.1 Baseline Segmentation Model

We train a baseline segmentation model based on DeepLab ResNet-101 [1, 4], or employ a pre-trained Xception-65 model [2]. Both models produce a segmentation map S_t^{ij} with a 19 dimension one-hot class label vector (corresponding to Cityscapes classes), at each pixel i, j in the frame, from the starting frame X_t.

3.2 3D Segmentation Transformation

To cast the segmentation prediction task as a 3D problem, we learn to predict static scene depth and inter-scene ego-motion using the unsupervised method of [8] on the Cityscapes dataset.

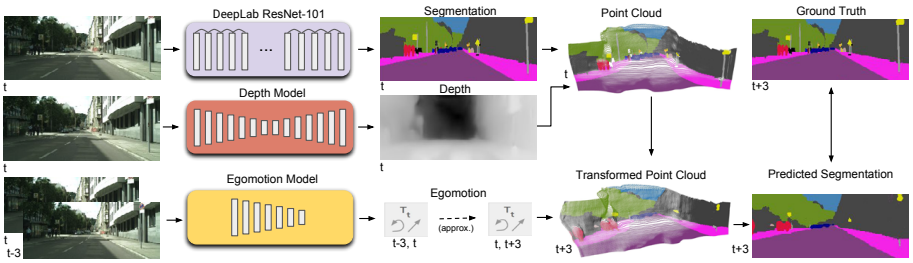

Fig. 2. An overview of our method: to predict the segmentation of a future frame $(t+3)$, we first compute the segmentation of the current frame (t) and estimate its depth (which is also learned). Together this allows us to generate a segmented 3D point cloud of the current frame (t). Additionally, we use the current frame (t) and its preceding frames to estimate the ego-motion of the future frames $(t+3)$, $(t+6)$, etc. We then use the predicted ego-motion to transform the segmented point cloud from frame (t) to generate the segmentation of the future frame $(t+3)$, and apply iterative transformations to produce subsequent frames $(t+6)$, $(t+9)$, etc.

Given a current RGB frame X_t, we apply the trained depth model to produce an estimated depth map for the current frame, D_t. The camera intrinsics are then used to obtain a structured 3D point cloud Q_t from the estimated depth map D_t. We attach to each coordinate i, j in Q_t^{ij} the one-hot segmentation class vector at the same coordinates in S_t^{ij}.

In parallel, given X_t, and X_{t-3}, the frame three time steps in the past, we apply the trained ego-motion model to estimate the motion between frames $T_{t-3\to t}$ and copy this ego-motion as an estimate of future ego-motion $\tilde{T}_{t-3\to t}$. Ego-motion is treated as an SE3 transform represented by a 6D vector describing the camera's movement as translation in x, y, z and rotation described by pitch, yaw, and roll.

The estimated future ego-motion $\tilde{T}_{t-3\to t}$ is applied as a transformation on the current point cloud Q_t^{ij}, to generate a predicted point cloud for the future $\hat{Q}_{t+3}^{ij}$. We then project the 3D point cloud prediction to 2D space via a forward warp using previously attached segmentation vectors, to construct a future frame segmentation prediction $\hat{S}_{t+3}^{\hat{i}\hat{j}}$. To predict more than one transformation step into the future, the above-mentioned ego-motion transformation is applied iteratively.

3.3 Learning Future Ego-Motion

In this section we describe our proposed algorithm which learns to predict the future ego-motion vectors as a function of the prior trajectory. Using our estimate of ego-motion from previous frames, e.g. $T_{t-3j-3\longrightarrow t-3j}$, for $j \in \{0, 1, 2\}$, we can produce a prediction for future motion as:

$$\hat{T}_{t\to t+3} = \mu(T_{t-9\to t-6}, T_{t-6\to t-3}, T_{t-3\to t}), \tag{1}$$

where μ is a nonlinear function, to be learned from the observed motion sequences. We use past given frames, X_{t-9}, X_{t-6}, X_{t-3}, and X_t to generate

the three past network-estimated[3] ego-motion vectors, $T_{t-9 \longrightarrow t-6}$, $T_{t-6 \longrightarrow t-3}$, $T_{t-3 \rightarrow t}$. To learn the unknown nonlinear function μ, we train a three-layer recurrent LSTM; each layer consists of a basic LSTM cell with six units, corresponding to the 6D ego-motion vector. The output from the last layer is then considered the future ego-motion, $\hat{T}_{t \rightarrow t+3}$ (which can be fed back to a final frozen network to estimate $\hat{T}_{t+3 \rightarrow t+6}$, so on.) The l_1 loss between the originally estimated ego-motion $T_{t \rightarrow t+3}$ and the RNN-predicted ego-motion $\hat{T}_{t \rightarrow t+3}$ is used for training:

$$\mathcal{L}_{\ell_1}(\hat{T}_{t \rightarrow t+3}, T_{t \rightarrow t+3}) = \sum_{i=1}^{6} \| T_{t \rightarrow t+3_i} - \hat{T}_{t \rightarrow t+3_i} \| \tag{2}$$

3.4 Inpainting Projected Segmentation

Since some parts of the future scene may have been occluded or out of the field of view in the starting frame, transforming and projecting segmentation maps will yield undesirable missing pixels. To address this problem, an inpainting step is applied in the segmentation space after every motion transformation by replacing every missing pixel with the most frequent segmentation class from its surrounding pixels.

4 Experiments

4.1 Dataset and Evaluation Metric

We use the Cityscapes dataset [3] which contains video sequences of city streets as captured by driving a car. Each video stream is 1.8 seconds long, and contains 30 frames with a ground truth segmentation mask available for the 20th frame. The primary evaluation metric employed is the average IOU (intersection over union) mean, across the full validation dataset of 500 sequences.

4.2 3D Transformation Model and Inpainting

The results of the main pipeline are reported in Table 1 starting from frame 5 i.e. $5 \rightarrow 20$ up to $17 \rightarrow 20$. Results are reported for the baseline segmentation model, the initial ego-motion (copy based) transformation method, and application of inpainting. For the ego-motion 'copy' method, we use the learned estimate of camera motion from $t-3$ to t, to approximate future motion as $\hat{T}_{t \rightarrow t+3} = T_{t-3 \rightarrow t}$ for all future transformations. While this approach is rough, it can be applied with very few available prior frames. As such, we evaluate predictions as far as 15 frames in advance (prior work [7] only reports results for 3 and 9 future frames). The motion transformation model and inpainting method provide significant successive improvements over the static baseline segmentation model. Inpainting most notably improves accuracy for longer-term predictions since the more transformations are done, more missing values need to be filled.

[3] The ego-motion vectors are previously learned in an unsupervised way from the monocular video and are not provided as ground truth in the data.

[4] Model normally uses frame $t-6$, ego-motion estimate may be inaccurate here.

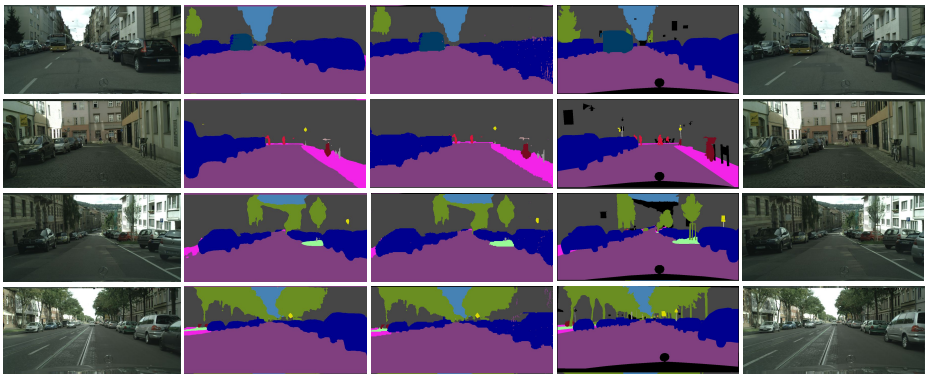

Fig. 3. Results of the 3D future segmentation model. From left to right: RGB input frame (top two rows 11, bottom rows 17), corresponding segmentation, predicted segmentation for 20, the ground truth segmentation of 20, and the RGB image of 20

Table 1. IOU mean results for future segmentation predictions for 3 - 15 frames into the future (i.e. 0.18 to 0.9 seconds). The future ego-motion model is a copy of prior ego-motion, as a temporary proxy

Input Frame	Target Frame (GT)	# Motion Transforms	Segmentation Copy (Baseline)	Motion Transform (Ours)	Motion Transform with Inpainting (Ours)
5	20	5	**0.3126**	0.2840	0.3059[4]
8	20	4	0.3391	0.3341	**0.3531**
11	20	3	0.3683	0.3854	**0.3993**
14	20	2	0.4095	0.4434	**0.4501**
17	20	1	0.4910	0.5335	**0.5353**

4.3 Application of Future Ego-Motion Learning

In this section we apply the learning of the future ego-motion trajectory, as described in Section 3.3, together with the main model. More specifically in the experimental setup we use three prior ego-motion vectors as input to future ego-motion predictions. This is in accordance with prior work which use 4 prior frames as input. With that setup however, only mid- to short-term predictions, namely $11 \rightarrow 20$ and $17 \rightarrow 20$ can be evaluated. Table 2 shows the results for mid- to short-term predictions with learning of future ego-motion. As seen, learning the future trajectory has a positive effect by improving the future scene segmentation estimates further than the 3D transformation and inpainting model alone, and examples of the full method are depicted in Figure 3.

4.4 Enhanced baseline segmentation

We replace our weak baseline segmentation model (IOU 0.6298) with a more powerful Xception-65 model (IOU 0.800). As seen in Table 3, medium and short term predictions are improved, achieving SOA accuracy at short term. We compare to previous SOA method by Luc et al. [7] but not to Jin et al.[5] as this work did not report numerical results for their predictive parsing network alone.

Table 2. Application of Learned Future Ego-Motion: IOU mean results for future segmentation predictions, when the future ego-motion vectors are learned from the history of prior trajectory

Method	$11 \rightarrow 20$	$17 \rightarrow 20$
Baseline	0.3683	0.4910
Motion Transform	0.3913	0.5335
Motion Transform + Inpainting	0.3993	0.5353
Motion Transform + Inpainting + Learning Future Ego-Motion	**0.4120**	**0.5370**

Table 3. Application of Enhanced Segmentation: IOU for full method with Xception

Method	$11 \rightarrow 20$	$17 \rightarrow 20$
Baseline	0.3956	0.5461
Motion Transform + Inpainting + Learning Future Ego-Motion	0.4540	**0.6147**
Luc et al.	**0.4780**	0.5940

5 Conclusions

We introduce a novel method for producing future semantic segmentation, by employing the 3D information of the scene. We incorporate learnable sub-modules for scene depth, ego-motion, and for future ego-motion prediction. We demonstrate improvements of future frame prediction accuracy, and future frame segmentation results up to 0.9 seconds away from the starting frame.

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE TPAMI **40**(4), 834–848 (Apr 2018)
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. CoRR **abs/1610.02357** (2016)
3. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. CVPR (2016)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
5. Jin, X., Li, X., Xiao, H., Shen, X., Lin, Z., Yang, J., Chen, Y., Dong, J., Liu, L., Jie, Z., Feng, J., Yan, S.: Video scene parsing with predictive feature learning. In: ICCV. pp. 5581–5589 (2017)
6. Kalchbrenner, N., van den Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., Kavukcuoglu, K.: Video pixel networks. CoRR (2016)
7. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: ICCV. pp. 648–657 (2017)
8. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. CVPR (2018)
9. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. ICLR (2016)