Structured Bird's-Eye-View Traffic Scene Understanding from Onboard Images - Supplementary Material

Yigit Baran Can[1] Alexander Liniger[1] Danda Pani Paudel[1] Luc Van Gool[1,2]
[1]Computer Vision Lab, ETH Zurich [2]VISICS, ESAT/PSI, KU Leuven
{yigit.can, alex.liniger, paudel, vangool}@vision.ee.ethz.ch

1. Summary

In the supplementary material we present our data augmentation technique, which uses view synthesis to generate more diverse training examples. We give more insight into our proposed connectivity metric, and finally show more visual and quantitative results.

2. Training

During training of the method, we apply artificial depth-wise motion as a data augmentation. With the flat world assumption, it is possible to calculate the new pixel location of a real world point if the ego vehicle moves β in the depth direction. Let the original pixel row and column coordinates be (m_0, n_0) and the new coordinates be (m_1, n_1). Then $n_0 = (n_1 - d_x)fC/(fC - m_1\beta + d_y\beta) + d_x$ and $m_0 = (m_1 - d_y)fC/(fC - m_1\beta + d_y\beta) + d_x$ where f is the focal length, C is the camera height and (d_x, d_y) are the frame center coordinates. We resample the original image and translate the ground truth (GT) object and centerline points by β.

3. Connectivity metric

The mathematical definition of our connectivity metric and how true positives, false positives and false negatives are defined, is given in the main text. Here we would like to summarize the definition in words. Therefore, let us first give the mathematical definition: Let the estimated binary incidence matrix be E and the GT incidence matrix be I. Let $M(i)$ be the index of the target that the ith estimation is matched to and $S(n)$ be the set of indices of estimations that are matched to target n. A positive entry E_{ij} is a true positive if $(M(i) == M(j)) \mid (I(M(i), M(j)) == 1)$, and a false positive otherwise. A false negative is a positive entry $I_{m,n}$ where $\nexists\ (i,j) : ((i \in S(m)) \ \&\ (j \in S(n)) \ \&\ (E_{i,j} == 1))$.

In words, if two estimated centerlines are associated, there are two possible ways for this association to be true:

- Both estimations are matched with the same target.
- The distinct targets that the two estimated centerlines are matched to are, indeed, associated according to the GT incidence matrix.

A miss, or a false negative, is present if there is a positive entry (m, n) in the GT incidence matrix I and at least one of the following conditions hold:

- No estimation was matched with target m.
- No estimation was matched with target n.
- Among all pairs of estimated centerlines (i, j) where i is matched with target m and j with n, there is no pair whose association estimate is positive.

4. Lane graph results

In the main paper we presented visual results for the lane graph, here we show further examples (see Fig. 1) and explain how we visualized the lane graphs.

4.1. Lane graph merging method

Whenever the network is estimating a full lane graph (this excludes PINET, which does not estimate a graph) we merge the predicted lane graph for visualization. The merging works by post processing the Bezier control points and the incidence matrix estimation in the following way:

- Extract all junction points where at least 2 centerlines meet.
- For all the junctions, get the start point locations of outgoing lines and end points of incoming lines.
- Concatenate all the junction points and take the mean, producing one (x,y) pair for each junction.
- Replace the start points of outgoing and endpoints of incoming lines with their respective junction locations.

Note that, this process does not change the underlying directed graph but it is useful for visualization. It is possible to formulate more advanced post-processing steps, for

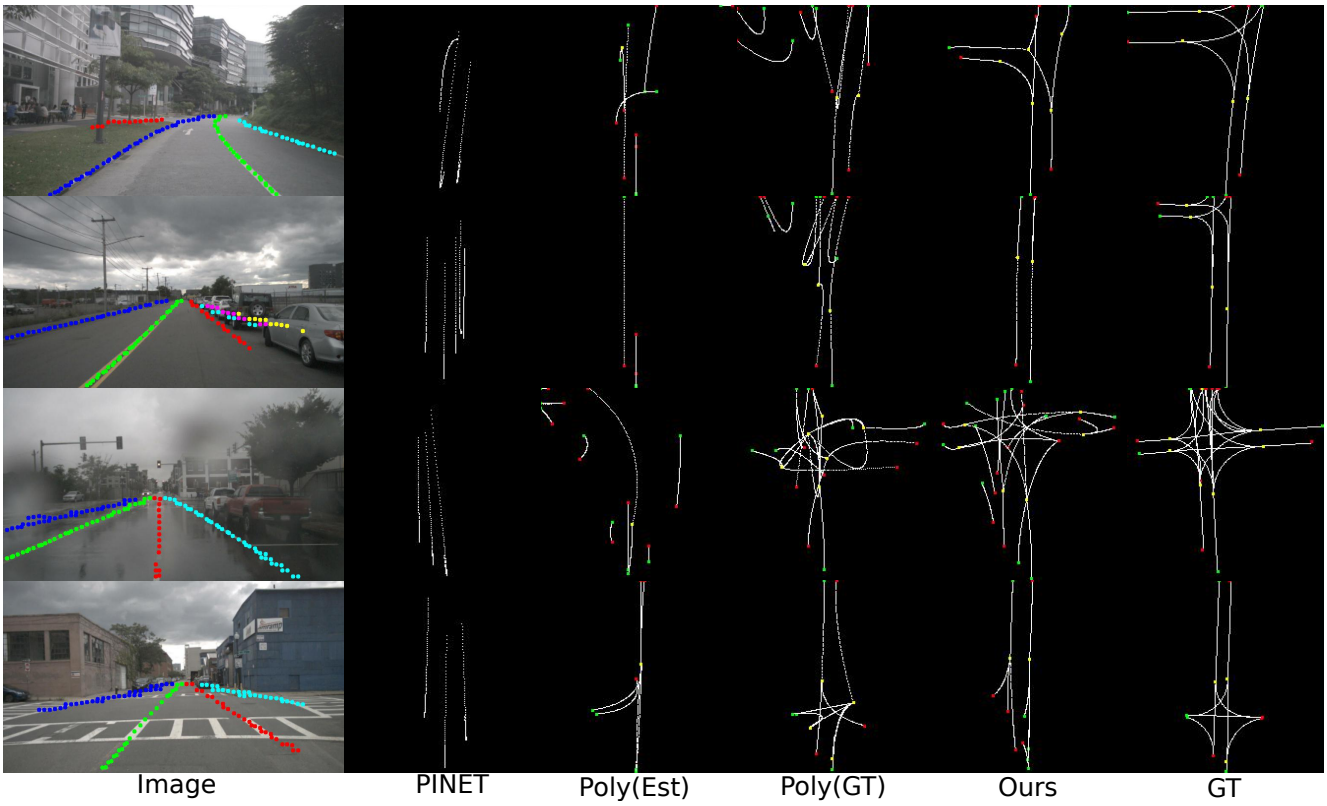

Figure 1. Sample centerline estimates. PINET boundary estimations are shown on the image. Our method produces the best lane graph representation. The detailed statistical results for the scenes in this figure are given.

example to compute the exact junction point locations one could consider the confidence of centerlines. However, this is beyond the scope of this work.

4.2. Visual results

First of all, we would like to emphasize that the results are shown for the whole target region of interest, whether it is occluded or not. We observed that sometimes, the methods can estimate lane graph structure in the occluded regions as well. Moreover, due to difficulty in establishing the occluded regions precisely, we have opted for presenting the results in whole field-of-view. Therefore, the results should be interpreted taking this into consideration. As stated in the main text and shown in Fig. 6, we have complied the statistical results for each method on each given image for each metric. Below, we present an extended version of Fig. 6, which includes four (instead of two) traffic scenes. Additionally to the visual results of the four traffic scenes we give the quantitative results in Tab. 1-4, and discuss the results.

In Scene 1, it can be seen that all methods manage to detect the straight lanes more or less accurately. However, only our method can detect the left turn. Moreover, we see that Poly(GT) produce inaccurate estimations in that region.

Method	M-Pre	M-Rec	Detect	C-Pre	C-Rec	C-IOU
PINET	49	50	20	-	-	-
Poly(Est)	37.9	33.2	60.0	0.0	0.0	0.0
Ours	60.0	53.4	60	75.0	60.0	50.0
Poly(GT)	54.5	53.5	70.0	66.7	44.4	36.4

Table 1. Scene 1 Results

Yet, because our method estimates the turn in a slightly wrong distance, both our method and Poly(GT) suffer similarly in the precision-recall metric. The proposed connectivity metric, however, clearly favors our estimation which is also backed by visual inspection.

Method	M-Pre	M-Rec	Detect	C-Pre	C-Rec	C-IOU
PINET	38	39	20	-	-	-
Poly(Est)	70.4	62.5	20.0	0.0	0.0	0.0
Ours	84.5	77.7	50.0	60.0	33.3	27.3
Poly(GT)	53.4	66.9	70.0	50.0	25.0	20.0

Table 2. Scene 2 Results

In Scene 2, Poly(GT) produces many false lines that are matched with straight road segments. This causes a decrease in precision-recall. Our method misses the left turn completely, which decreases the detection score, but it faithfully represent the straight lanes. It should be noted that

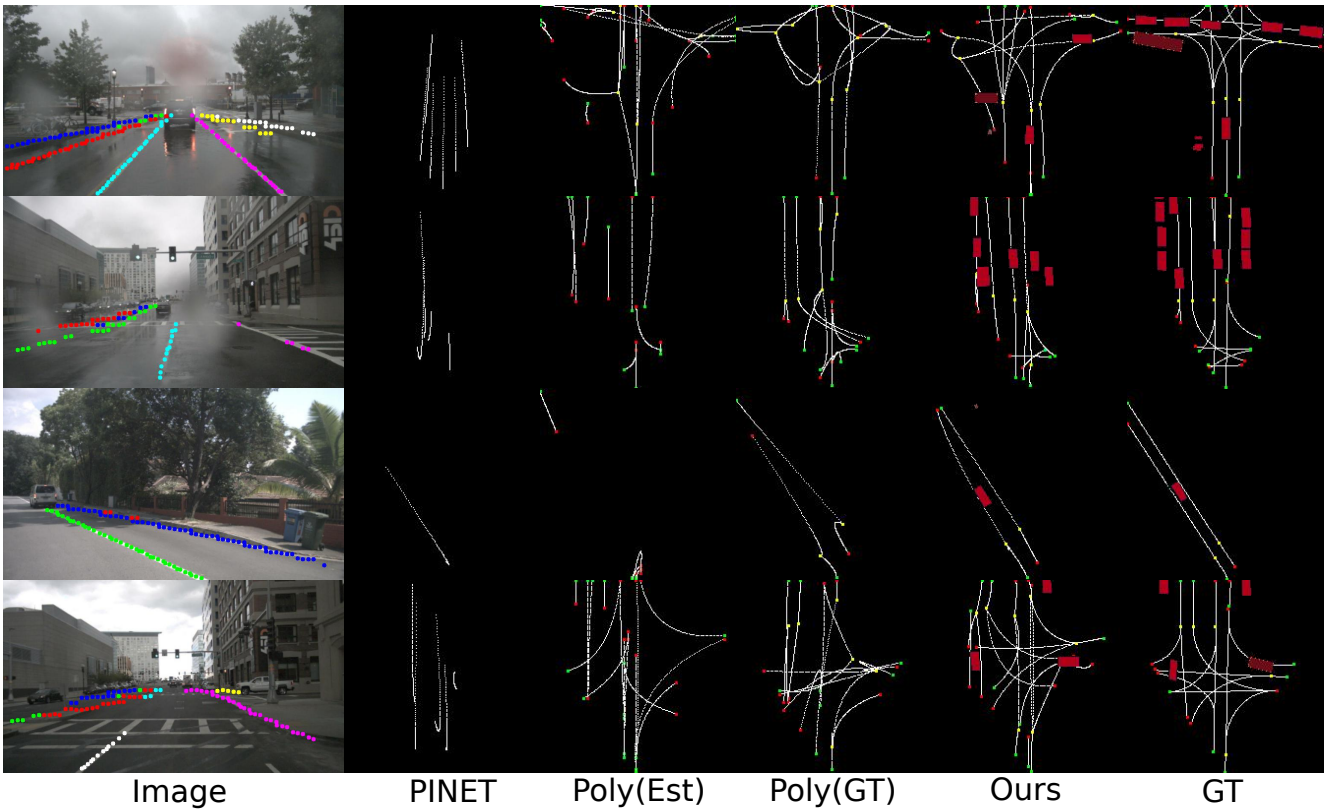

Figure 2. Some additional examples with object estimations also shown for our method. It can be seen that our method produces the best graphs.

Poly(Est) performs better than Poly(GT) in precision-recall in this image. Note that PINET can not handle the parked cars and generates several centerlines in the carpark area.

Method	M-Pre	M-Rec	Detect	C-Pre	C-Rec	C-IOU
PINET	64	75	8	-	-	-
Poly(Est)	32.9	30.7	26.1	100	4.4	4.4
Ours	48.4	47.6	74.0	82.6	67.9	59.4
Poly(GT)	62.2	62.4	65.0	48.1	48.1	31.7

Table 3. Scene 3 Results

Scene 3 shows a complicated road network and the left and right turns are barely visible. However, our method manages to produce a good estimate. The small differences in the exact location of the lines results, however, in lower precision-recall than Poly(GT). PINET only finds the straight lines as expected while Poly(Est) detects some true initial points that are the beginnings of the turns but the Polygon-RNN head fails to produce the lanes. Again, we see that the connectivity metric demonstrates the superiority of the proposed method.

In Scene 4, most of the crossroads is not visible. PINET mistakenly estimates 3 lanes (4 lane boundaries) while the rightmost one is actually a carpark area. Poly(Est) detects the lines but estimates them to be close to each other and in the same direction. Poly(GT) estimate suffers from faulty association where the rightmost lane is distorted due to a recalculated junction point location. Our method produces the initial part of the turns but fails to estimate the whole crossroads. In this scene Poly(Est) produces the best results except for the detection score. This is due to the fact that is it misses the whole right part of the image, but produces reasonable estimates of the left part.

Method	M-Pre	M-Rec	Detect	C-Pre	C-Rec	C-IOU
PINET	39	46	30	-	-	-
Poly(Est)	80.2	81.7	30	100	60.0	60.0
Ours	54.9	52.7	60.0	100	66.7	66.7
Poly(GT)	64.4	69.6	90	55.6	83.3	50.0

Table 4. Scene 4 Results

4.3. Additional visual results

In Fig 2, we present some additional results with object estimations included.

In the first scene, the GT shows many cars travelling in the horizontal direction but upon inspection of the image, we observe that part of image is not clear. Thus our method's object estimates are reasonable. The road network estimate of our method is vastly superior to all other base-

lines including Poly(GT).

In the second image, our method produces all 4 lanes faithfully. Poly(GT) also produces decent estimates. Our method missed the cars on the right side of the scene, but we see that the rain drop is making that region non-visible.

Scene 3 provides a relatively easy task for all methods. PINET, unsurprisingly, produces an accurate estimate but Poly(Est) failed to produce the lanes. While Poly(GT) manages to somewhat estimate the direction of the lanes, it produces faulty structures in the bottom part of the FOV. Our method produces very accurate estimations for both centerlines and the car.

The last scene demonstrates a very complex crossroads scene. All 4 directions of the crossroads are visible in the image. This results in Poly(Est) producing good estimates, capturing the essence of the lane graph. The Poly(GT) estimate is denser than Poly(Est) and covers a larger area of the true lane graph but it fails to handle the junctions properly. Our method produces a better lane graph estimate but fails especially in the left side of the image. Our object estimates are accurate, with the exception of the truck which is labelled as car.