Supplementary Material for
Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer

Jun Xie[1] Martin Kiefel[2] Ming-Ting Sun[1] Andreas Geiger[2]
[1]University of Washington [2]MPI for Intelligent Systems Tübingen
{junx,mts}@uw.edu {martin.kiefel,andreas.geiger}@tue.mpg.de

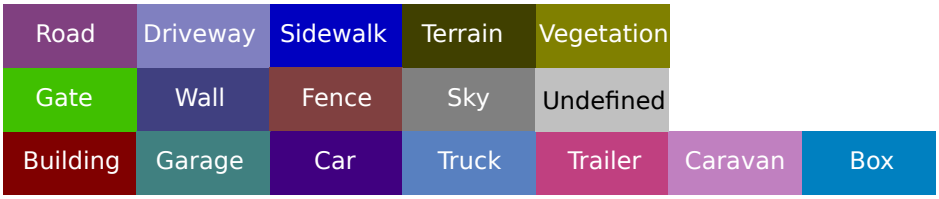
Abstract

This supplementary material provides additional illustrations, visualizations and experiments. We start by showing the color coding and label mapping used for the semantic and instance label results in the paper. Then we provide more details about the 3D fold/curb detection and parameter settings that are used in the paper. Next, we provide additional quantitative and qualitative semi-dense inference results for both semantic and instance segmentation. Finally, we show the ability of our method to annotate 3D point clouds with semantic and instance labels which is a byproduct of our approach.

1. Color Coding

We first illustrate the color coding which we have used for Fig. 1 and Fig. 5 in the main paper in Fig. 1.

(a) **Color Coding of Semantic Labels.**



(b) **Color Coding of Instance Labels.**

Figure 1: **Color Coding.** Illustration of color coding used for Fig. 1 and Fig. 5 in the main paper.

Next, we show the mapping between the semantic and instance labels, the class frequency (computed as the percentage of pixels in the ground truth) and the label definitions of some classes in Table 1. As explained in the main paper, we map similar classes into a single one for semantic evaluation such that each category is represented well. However, note that our method can handle also rare classes which appear less frequently throughout the dataset. Thus for our instance evaluation we evaluate slightly more fine grained and only ignore classes which appear very rarely.

Semantic Labels	**Instance Labels**	**Frequency (%)**	**Definition**
Road	Road	11.55	
Sidewalk	Sidewalk	6.91	
Driveway	Driveway	1.29	
Terrain	Terrain	1.40	Grass, Soil, Stone
Building	Building, Garage	29.04	
Vegetation	Vegetation	25.03	
Car	Car, Truck	9.61	
Trailer	Trailer	0.49	
Caravan	Caravan	0.49	
Box	Box	0.27	Box, Trashbin, Vendingmachine
Wall	Wall	3.63	
Fence	Fence	1.81	
Gate	Gate	0.44	
Sky	Sky	7.80	
Undefined	Undefined	0.26	Motorcycle, Bicycle, Pedestrian, Rider, BigPole, SmallPole, TrafficLight, TrafficSign, Lamp

Table 1: **Mapping between Instance Labels and Semantic Labels.** Frequencies are specified in percentage of pixels.

2. 3D Fold/Curb Detection

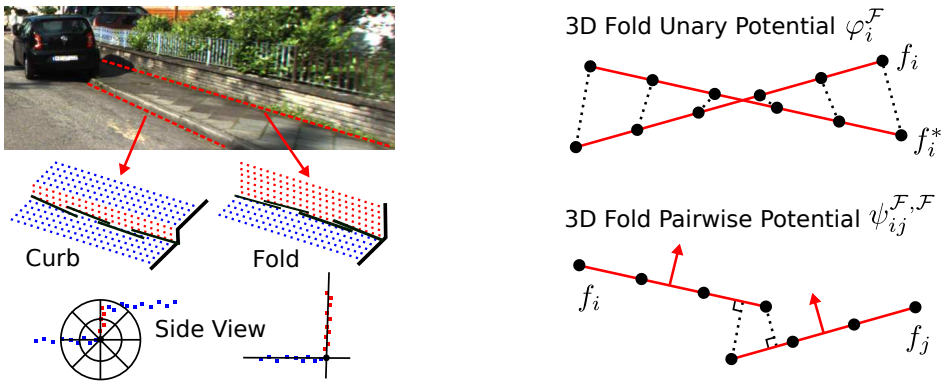
We detect folds and curbs in the 3D point cloud for disambiguating the semantic class at object boundaries. We first extract all relevant object class boundaries by thresholding the gradient over semantic classes in the annotated 3D point cloud (i.e., we sweep a 3D gradient operator over the semantic 3D point cloud). For each boundary point, we fit two perpendicular 3D planes and extract their intersection in terms of a 3D fold (see Fig. 2a, right). The sole exception are boundaries between road and sidewalk for which we detect the bottom part (of the curb) by training an SVM on shape context features [1] (see Fig. 2a, left). Due to the small elevation of the curb and the noise in the 3D data we found this to perform better than 3D plane fitting in terms of separating the objects in 3D.

As the fold detections are noisy, we model the true fold location as a random variable and penalize the deviation of the estimate f from the detection f^* while encouraging continuity/smoothness. We associate a random variable $f_i \in \mathbb{F}$ with each 3D fold or curb $i \in \mathcal{F}$ which specifies the location and orientation of the fold segment in 3D. We discretize the set of possible fold segments for each detection by sampling from a local neighborhood around the parameters of the detection, i.e., we have $\mathbb{F} = \{1, \dots, F\}$, where F is the number of discrete sample points. Each sample is associated with the corresponding fold segment parameters. We formulate a CRF model for optimizing the placement of fold/curb segments with an energy function which encourages smoothness of adjacent segments in 2D:

$$E(\mathbf{f}) = \sum_{i \in \mathcal{F}} \varphi_i^{\mathcal{F}}(f_i) + \sum_{i,j \in \mathcal{F}} \psi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) \tag{1}$$

3D Fold/Curb Unary Potentials: The unary potential for the 3D fold segments and curbs is specified by a quadratic loss on the deviation of the estimated fold f_i from its 3D detection f_i^*:

$$\varphi_i^{\mathcal{F}}(f_i) = w^{\mathcal{F}} \sum_{c \in \mathcal{C}} \|\boldsymbol{\kappa}_i(f_i, c) - \boldsymbol{\kappa}_i(f_i^*, c)\|_2^2 \tag{2}$$



(a) 3D Fold Detection

(b) 3D Fold Potentials

Figure 2: **Illustration of the fold/curb detection in our Model.** (a) Geometric structures such as folds and curbs are detected in the 3D point cloud by fitting planes and training a classifier based on shape context. (b) We model the uncertainty in the folds by introducing an auxiliary random variable f_i with each of them and connect adjacent folds to encourage smoothness.

Here, $\mathcal{C} \subset [0,1]$ is a finite set of 1D control points along the fold segment and $\boldsymbol{\kappa}_i(f_i, c)$ returns the corresponding 3D point. The potential is illustrated in Fig. 2b (top).

3D Fold/Curb Pairwise Potentials: For smoothing the boundaries, we introduce a pairwise term which encourages continuity between neighboring fold segments and curbs

$$\psi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) = \begin{cases} \phi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) & \text{if } (i,j) \in \mathcal{N}_{\mathcal{F}} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where smoothness of neighboring folds is defined via

$$\begin{aligned} \phi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) \quad = \quad & w_1^{\mathcal{F},\mathcal{F}} \left(1 - \frac{|\boldsymbol{\pi}_i(f_i)^T \cdot \boldsymbol{\pi}_j(f_j)|}{\|\boldsymbol{\pi}_i(f_i)^T\|_2 \, \|\boldsymbol{\pi}_j(f_j)\|_2}\right) \\ & + w_2^{\mathcal{F},\mathcal{F}} dist\left(\boldsymbol{\pi}(\boldsymbol{\kappa}_i(f_i, 1)), \boldsymbol{\pi}_j(f_j)\right) \\ & + w_2^{\mathcal{F},\mathcal{F}} dist\left(\boldsymbol{\pi}(\boldsymbol{\kappa}_j(f_j, 0)), \boldsymbol{\pi}_i(f_i)\right) \end{aligned}$$

and $\mathcal{N}_{\mathcal{F}}$ denotes the set of neighboring folds in 3D, i.e., folds for which the endpoint of one fold segment is within a small distance from the startpoint of the next segment. The 3D point $\boldsymbol{\kappa}(\cdot,\cdot)$ is defined as above, $\boldsymbol{\pi}(\cdot)$ projects a point or fold segment from 3D to 2D, and $dist(\cdot,\cdot)$ denotes the shortest distance of a 2D point to a 2D fold segment. We use scaled normals to represent fold segments $\boldsymbol{\pi}_i(f_i)$ in 2D (i.e., $\boldsymbol{\pi}_i(f_i)^T \mathbf{p} = 1$ for all pixels $\mathbf{p} \in \mathbb{R}^2$ on the 2D fold). This potential is illustrated in Fig. 2b (bottom).

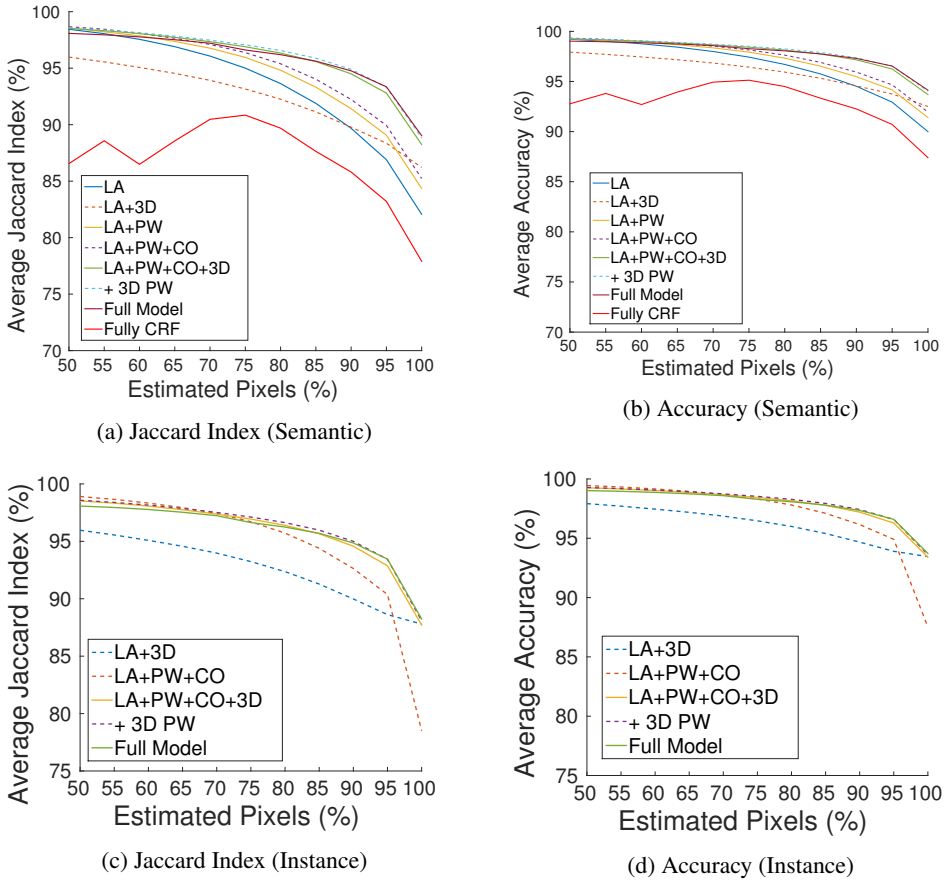
Inference: Eq. 1 corresponds to a non-loopy pairwise CRF as folds are connected in chains, e.g., along the sidewalk-road boundary. We obtain a global minimizer of the corresponding Gibbs energy via belief propagation. The parameters of the model have been set empirically to yield smooth results.

3. Parameter Estimation

The log-linear weights in our model are trained end-to-end as described in the main paper. The number of these parameters is too large to specify all of them here, but we will provide our code and the trained models on acceptance of this paper. In contrast to the log-linear weights, the kernel width parameters are more difficult to learn using empirical risk minimization. Thus, we obtain these parameters by coordinate descent on the validation set. In particular, we obtained $\theta_1^{\mathcal{P},\mathcal{P}} = 3$, $\theta_2^{\mathcal{P},\mathcal{P}} = 43$, $\theta_3^{\mathcal{P},\mathcal{P}} = 9$, $\theta_1^{\mathcal{L},\mathcal{L}} = 0.05$ and $\theta_2^{\mathcal{L},\mathcal{L}} = 1.0$ which we fixed throughout all our experiments. Overall we found that our model is not very sensitive to the exact setting of these parameters.

4. Additional Semi-Dense Inference Results

In this section, we show additional quantitative semi-dense inference results. In particular, we show the average Jaccard Index and the average accuracy for semantic segmentation as well as instance segmentation when estimating only a fraction of the pixels, elected according to the label uncertainty/entropy at each pixel as described in the main paper. We also include the results for the "Fully Conn. CRF" label transfer baseline, for which we estimate uncertainty the same way as for our methods. For all other baselines, uncertainty estimates are not directly accessible.



(a) Jaccard Index (Semantic)

(b) Accuracy (Semantic)

(c) Jaccard Index (Instance)

(d) Accuracy (Instance)

Figure 3: **Performance wrt. Estimated Pixels.** This figure shows the average Jaccard Index (a, c) and the average accuracy (b, d) for semantic segmentation (top, including the "Fully Conn. CRF" baseline) and instance segmentation (bottom) when estimating only a fraction of the pixels which is selected according to the uncertainty/entropy in our predictions.

Method	Road	Park	Sdwlk	Terr	Bldg	Vegt	Car	Trler	Carvn	Gate	Wall	Fence	Box	Sky	JI	Acc
LA+3D	94.5	74.7	83.5	73.4	80.7	84.5	86.3	90.8	90.9	66.3	74.7	75.6	63.1	81.9	83.5	91.0
LA+PW+CO	92.8	70.3	79.8	73.9	64.9	84.6	82.2	90.7	87.1	51.7	67.8	66.6	24.7	88.0	78.4	87.4
LA+PW+CO+3D	94.6	78.4	84.2	78.4	86.3	87.6	90.8	93.0	93.3	70.9	77.6	79.4	68.6	91.1	87.5	93.3
+ 3D PW	95.1	80.6	85.3	79.3	86.4	87.9	91.5	93.0	93.6	73.6	78.1	79.0	70.4	90.7	87.9	93.5
Full Model	95.7	80.6	86.9	79.2	86.4	87.9	91.5	93.1	93.6	73.6	78.5	79.1	70.5	90.7	88.1	93.6
Full Model (90%)	97.4	88.9	90.8	88.2	92.1	92.7	95.8	94.7	97.6	80.0	85.9	87.3	76.1	92.7	92.8	96.2
Full Model (80%)	98.6	92.4	93.3	91.9	94.0	94.6	97.2	95.5	98.5	83.0	89.4	91.3	79.1	93.9	94.7	97.3
Full Model (70%)	**99.0**	**94.1**	**94.3**	**93.6**	**94.7**	**95.7**	**97.7**	**96.0**	**99.0**	**84.3**	**90.9**	**93.0**	**80.9**	**94.9**	**95.7**	**97.8**

Table 2: **Ablation Study on Instance Segmentation Task** including the semi-dense instance inference results (bottom).

5. Semantic Segmentation

We also compare our method with two state-of-the-art semantic segmentation approaches which require a moderate number of annotated data: The fully connected CRF by Krähenbühl et al. [4] as well as the robust high order potentials (i.e., ALE library) by Kohli, Ladicky et al. [3]. In addition, we compare the Jaccard index of the particular "Car" category with [2], which automatically performs car labeling given weak supervisory signals similar to us. We perform 2-fold cross validation on 100 densely labeled images. The results are shown in Table 3. From the results, our approach lowers errors consistently across almost all categories. Thus, we expect our annotations to be useful also for training models of larger capacity.

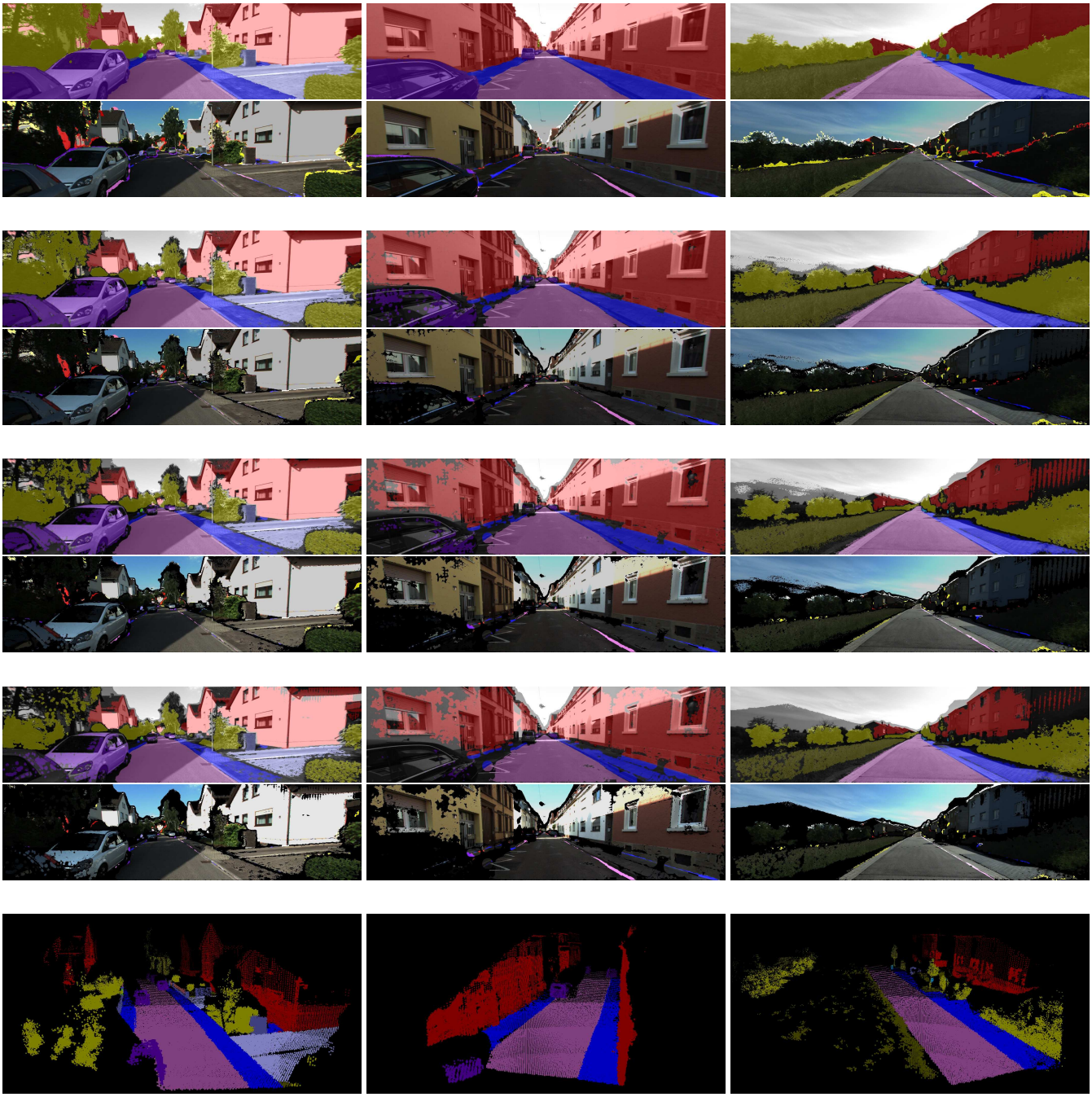
Method	Road	Park	Sdwlk	Terr	Bldg	Vegt	Car	Trler	Carvn	Gate	Wall	Fence	Box	Sky	JI	Acc
Fully Conn. CRF [4]	82.9	18.3	52.7	0.9	77.6	75.2	69.8	0.3	1.6	1.5	20.7	19.2	0.6	75.0	68.6	81.1
ALE [3]	89.9	31.5	66.2	5.5	82.4	79.9	76.0	6.1	12.4	6.4	33.9	44.3	2.4	83.1	75.5	85.4
Beat Mturkers [2]	-	-	-	-	-	-	82.9	-	-	-	-	-	-	-	-	-
Our Model	**95.4**	**80.1**	**87.1**	**80.0**	**90.6**	**87.0**	**91.2**	**91.3**	**93.9**	**72.6**	**78.4**	**78.6**	**69.4**	**90.8**	**89.0**	**94.1**

Table 3: **Semantic Segmentation.** This table shows the Jaccard Index (JI) with respect to each class, the average Jaccard Index (Avg. JI) and the average accuracy (Avg. Acc) of our method and two state-of-the-art semantic segmentation methods.

6. Additional Qualitative Inference Results

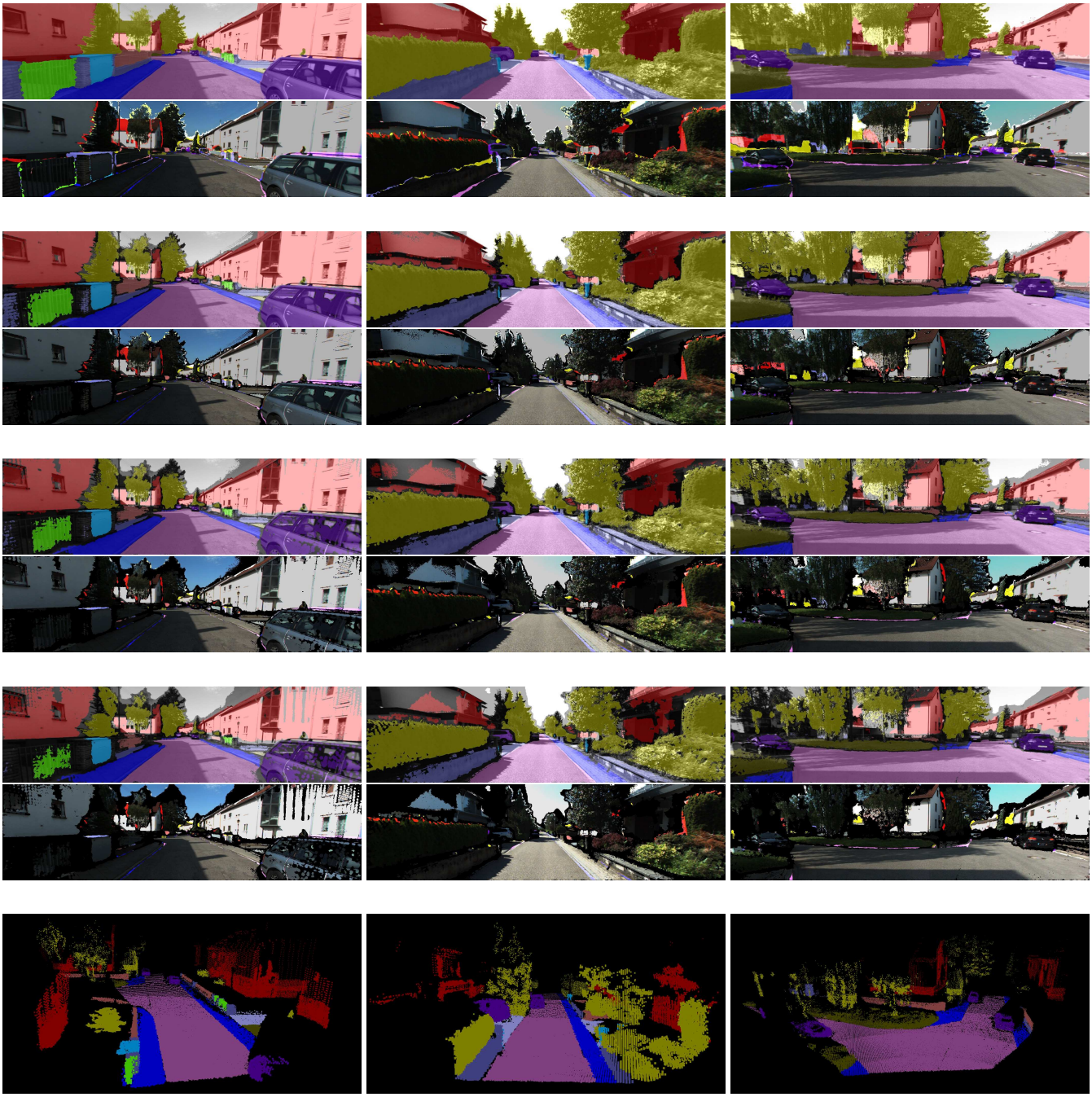
6.1. Semi-Dense Semantic Inference Results

In this section, we show several semantic inference results qualitatively for different estimation densities.



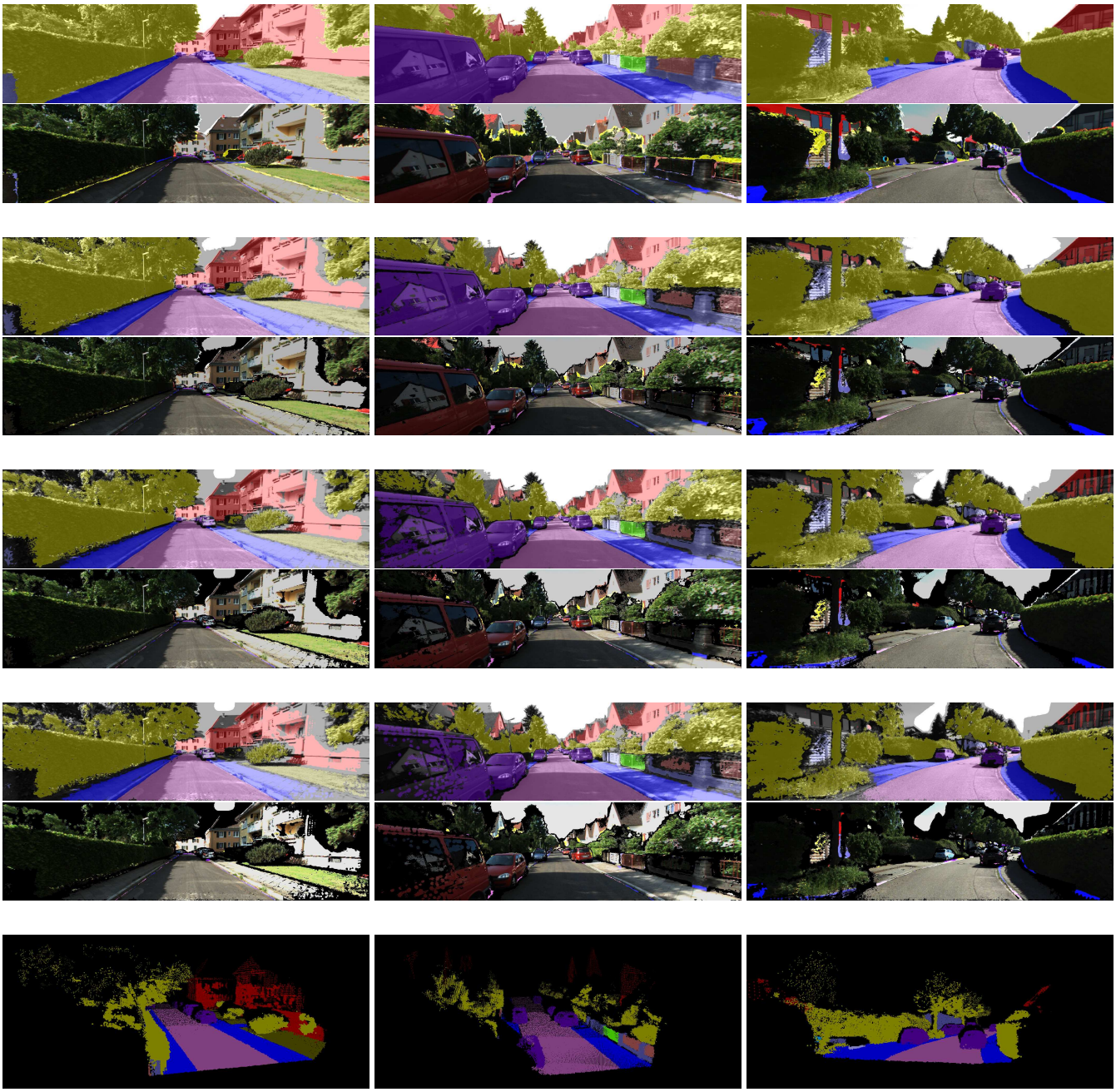
(a) Scene 1 (b) Scene 2 (c) Scene 3

Figure 4: **Qualitative Semi-Dense Semantic Results.** Each subfigure shows from top-to-bottom: the input image with inferred semantic segmentation and the errors with respect to 2D ground truth annotation where colors indicate the groundtruth label. The second, third and fourth row of subfigures show the results at 90%, 80% and 70% density, respectively. The last row shows the corresponding semantic 3D point cloud.



(a) Scene 4 (b) Scene 5 (c) Scene 6

Figure 5: **Qualitative Semi-Dense Semantic Results.** Each subfigure shows from top-to-bottom: the input image with inferred semantic segmentation and the errors with respect to 2D ground truth annotation where colors indicate the groundtruth label. The second, third and fourth row of subfigures show the results at 90%, 80% and 70% density, respectively. The last row shows the corresponding semantic 3D point cloud.

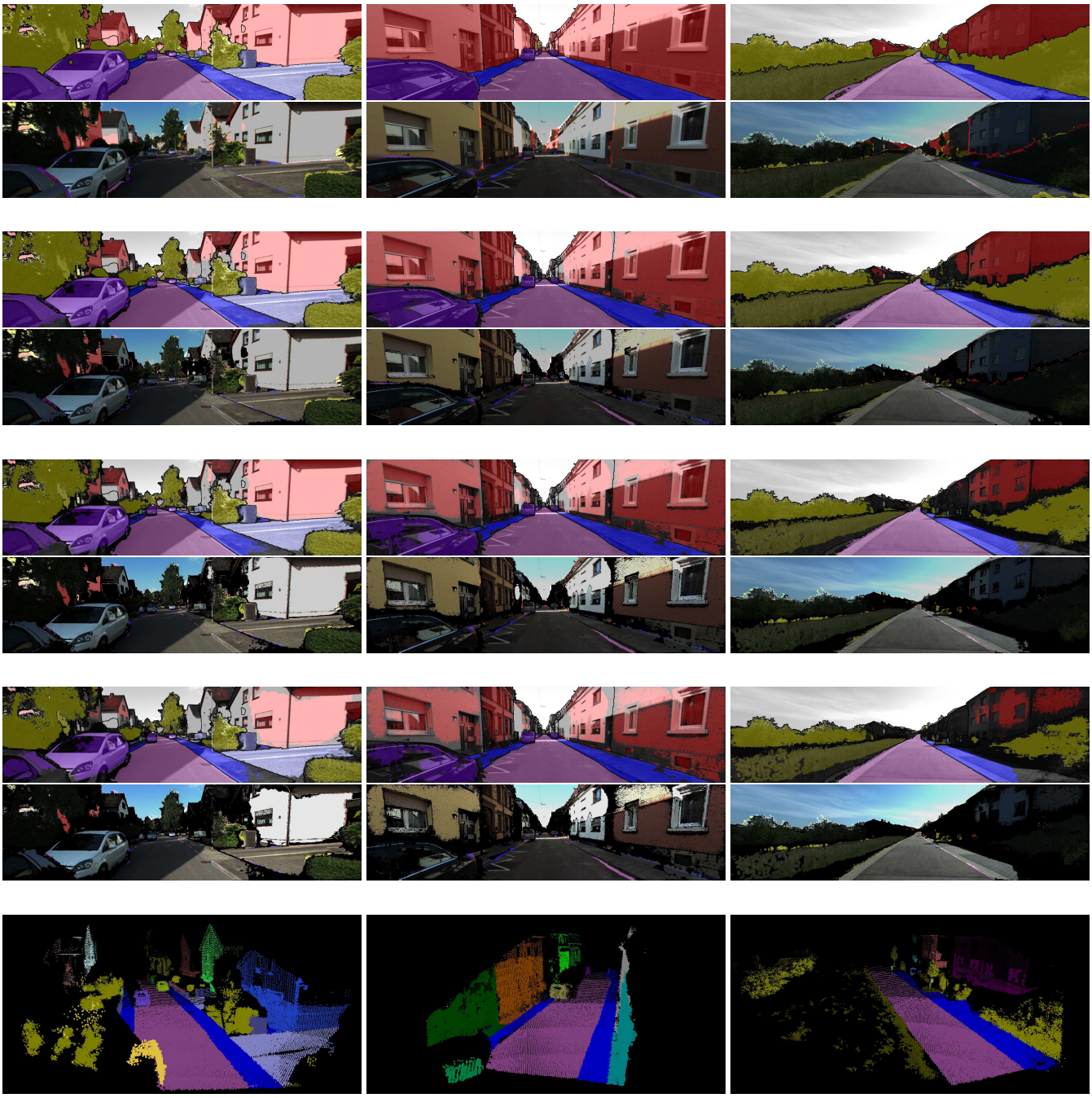


(a) Scene 7 (b) Scene 8 (c) Scene 9

Figure 6: **Qualitative Semi-Dense Semantic Results.** Each subfigure shows from top-to-bottom: the input image with inferred semantic segmentation and the errors with respect to 2D ground truth annotation where colors indicate the groundtruth label. The second, third and fourth row of subfigures show the results at 90%, 80% and 70% density, respectively. The last row shows the corresponding semantic 3D point cloud.

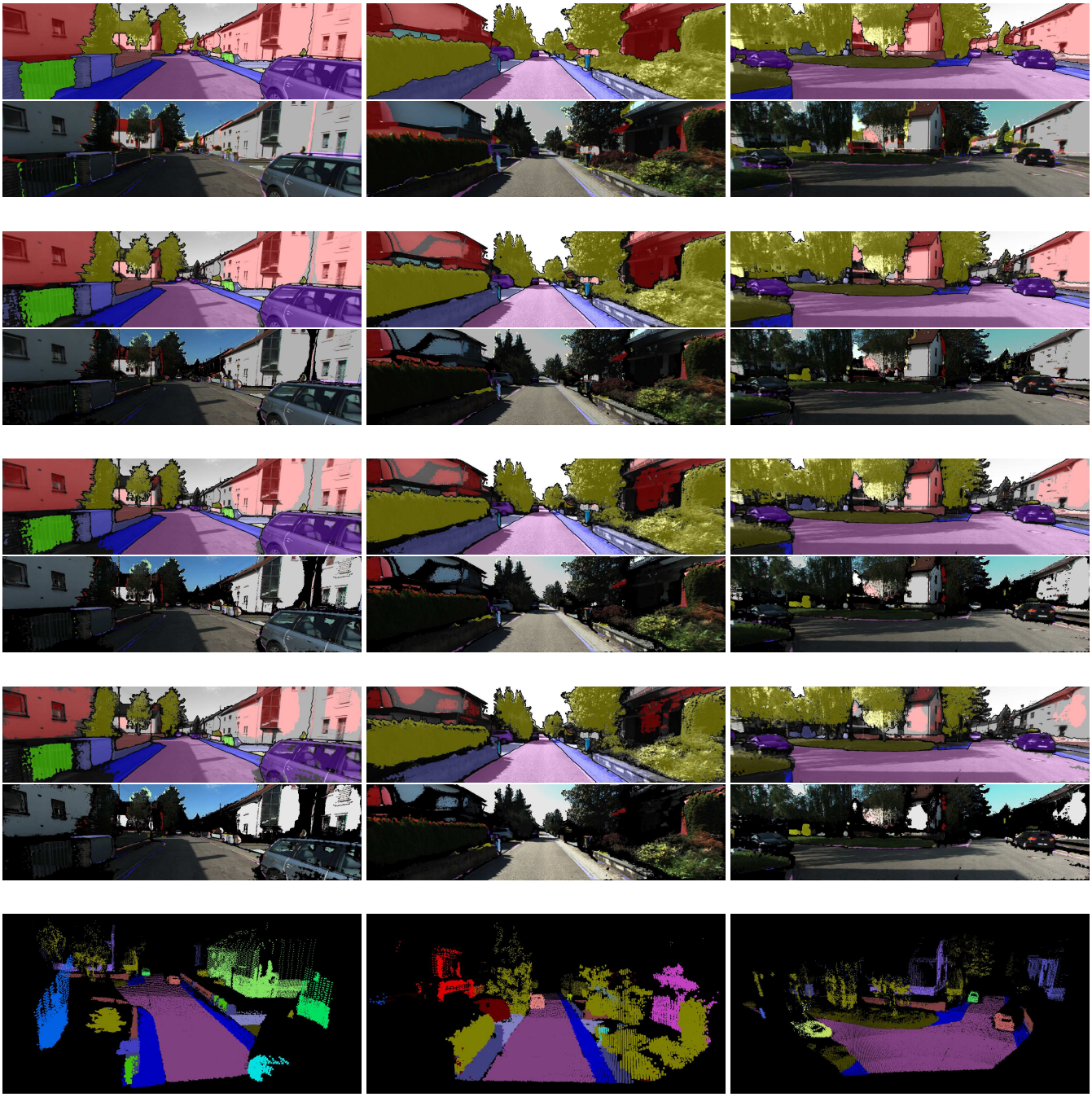
6.2. Semi-Dense Instance Inference Results

In this section, we show several instance inference results qualitatively for different estimation densities.



(a) Scene 1 (b) Scene 2 (c) Scene 3

Figure 7: **Qualitative Semi-Dense Instance Results.** Each subfigure shows from top-to-bottom: the input image with inferred instance segmentation and the errors with respect to 2D ground truth annotation where colors indicate the groundtruth semantic label. The second, third and fourth row of subfigures show the results at 90%, 80% and 70% density, respectively. The last row shows the corresponding instance 3D point cloud (random colors).



(a) Scene 4 (b) Scene 5 (c) Scene 6

Figure 8: **Qualitative Semi-Dense Instance Results.** Each subfigure shows from top-to-bottom: the input image with inferred instance segmentation and the errors with respect to 2D ground truth annotation where colors indicate the groundtruth semantic label. The second, third and fourth row of subfigures show the results at 90%, 80% and 70% density, respectively. The last row shows the corresponding instance 3D point cloud (random colors).

7. Semantic and Instance Results in 3D

In this section, we show accumulated semantic and instance point clouds which are inferred as a byproduct by our model.

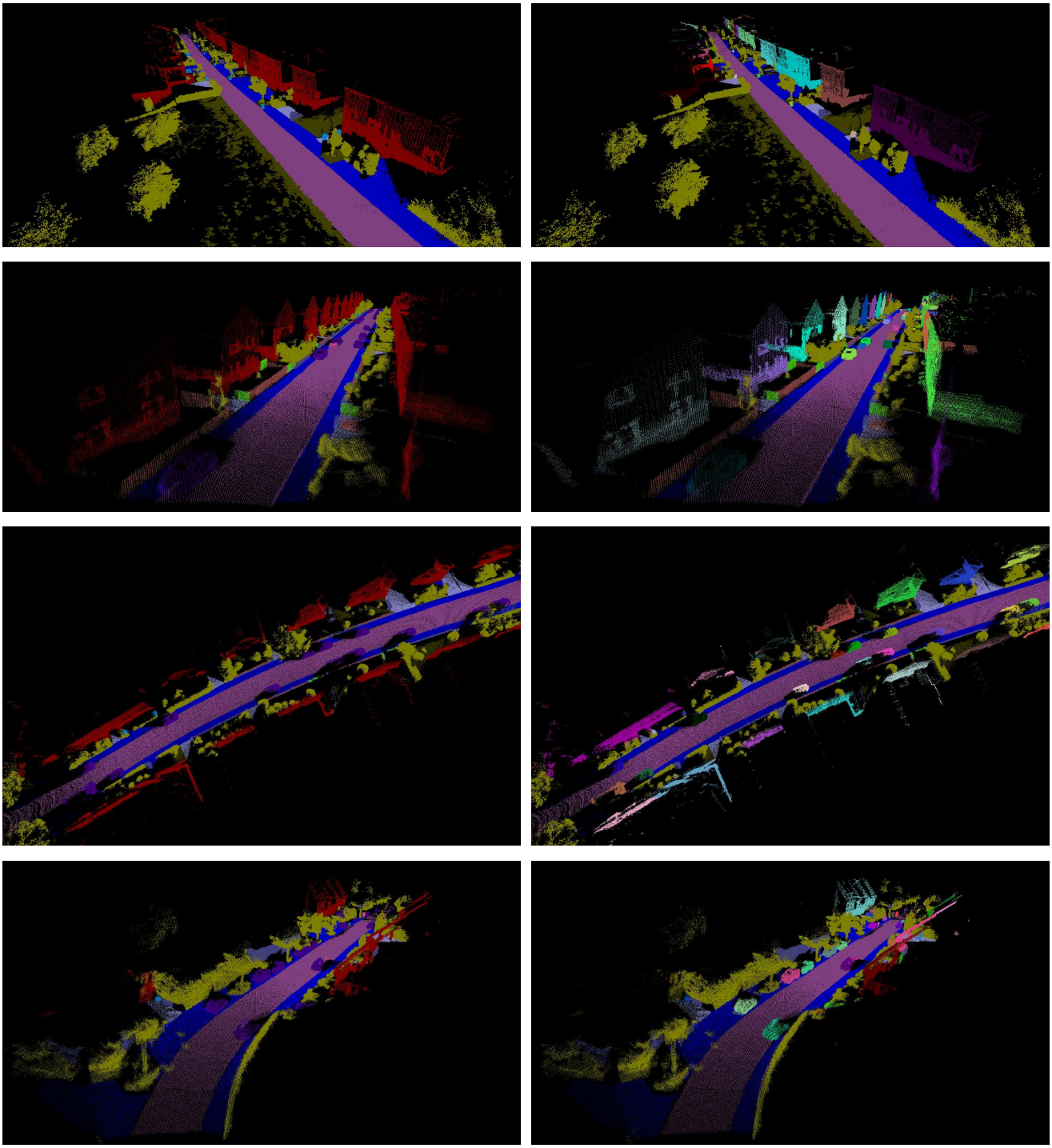
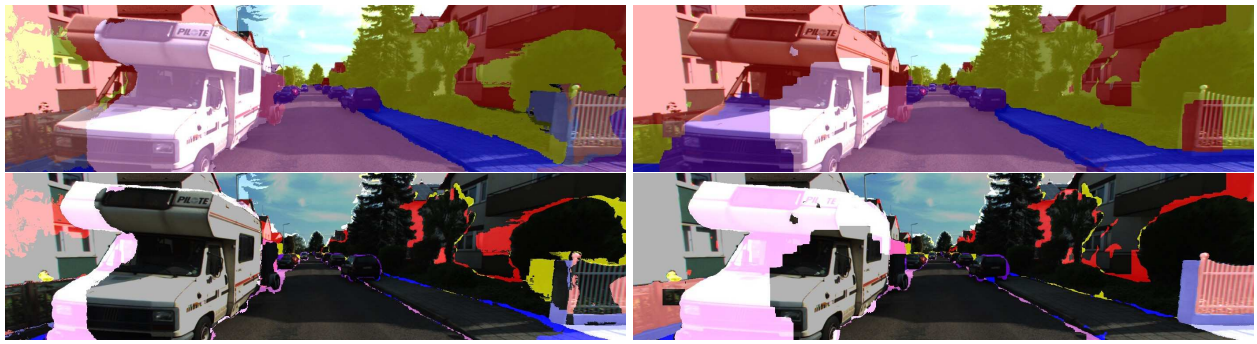


Figure 9: **Inferred 3D Point Clouds.** Left: Semantic results. Right: Instance results (random colors).

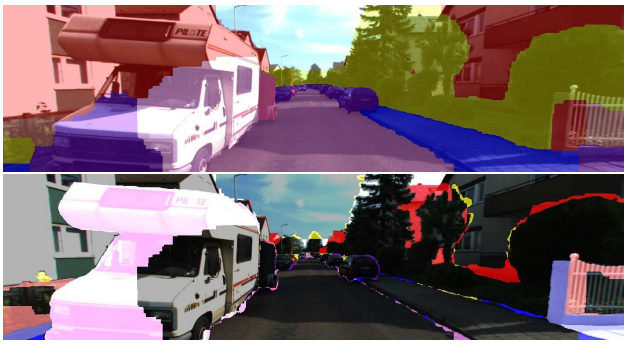
8. Qualitative Comparison with Baselines

Here, we compare our method qualitatively to several 2D-to-2D and 3D-to-2D label transfer baselines. Note how the 2D-to-2D label transfer baselines fail in the presence of strong occlusions and large displacements.

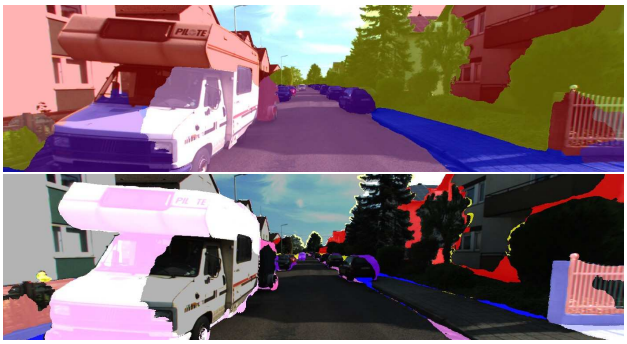


(a) Label Prop. [6]

(b) Sparse Track. + GC [5]

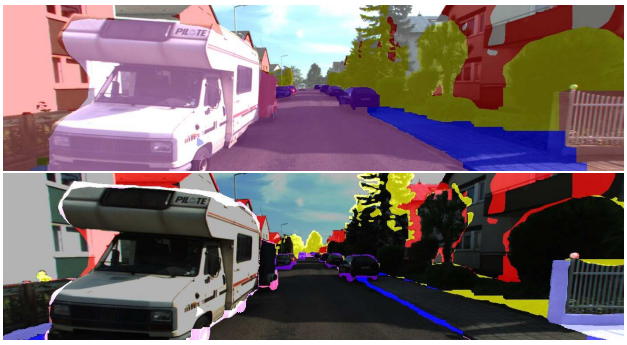


(c) 3D Prop. + GC

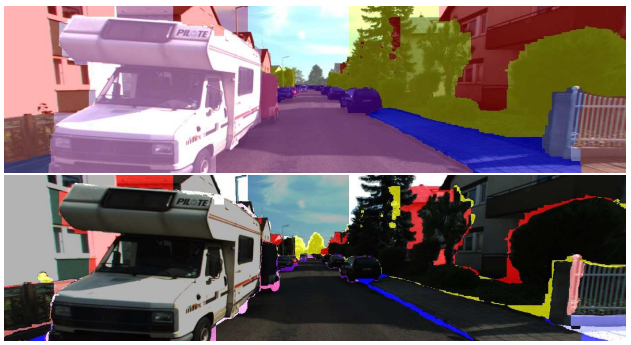


(d) Fully Conn. CRF [4]

(e) 3D Primitives + GC



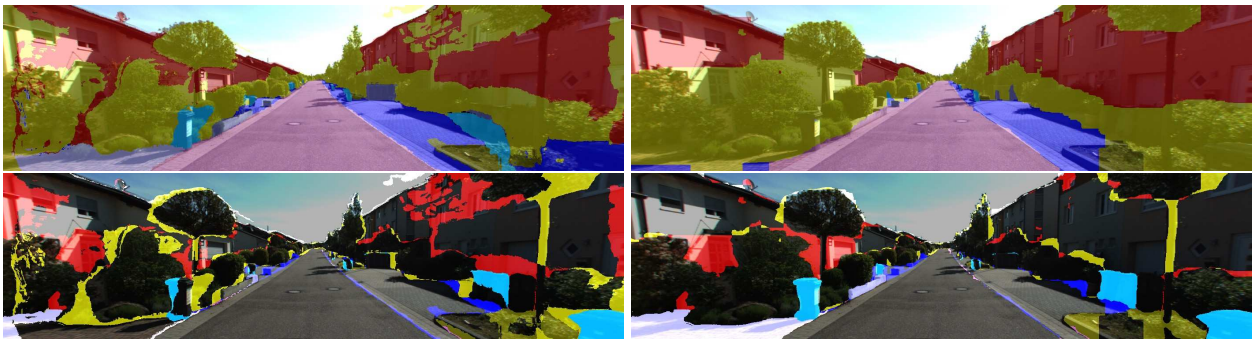
(f) 3D Mesh + GC



(g) 3D Points + GC

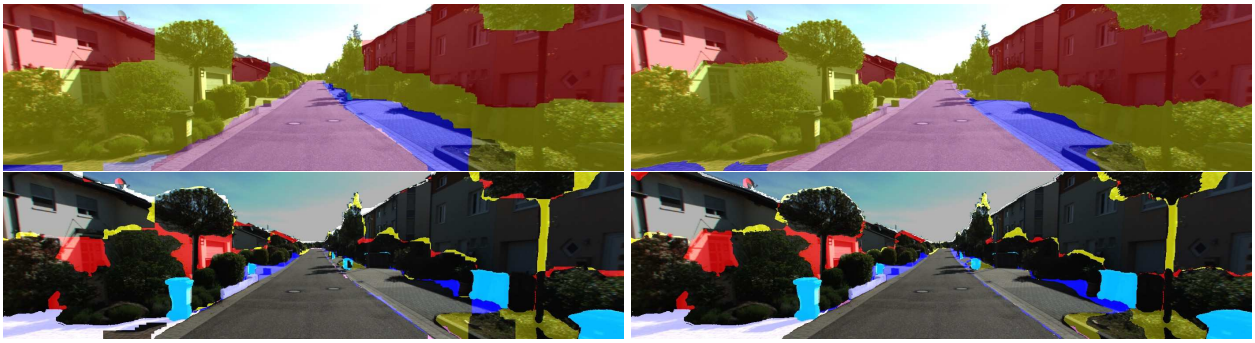
(h) Proposed Method

Figure 10: **Comparison to Baselines.** Each subfigure shows from top-to-bottom: the input image with inferred semantic segmentation and the errors with respect to 2D ground truth annotation where colors indicate ground truth labels.



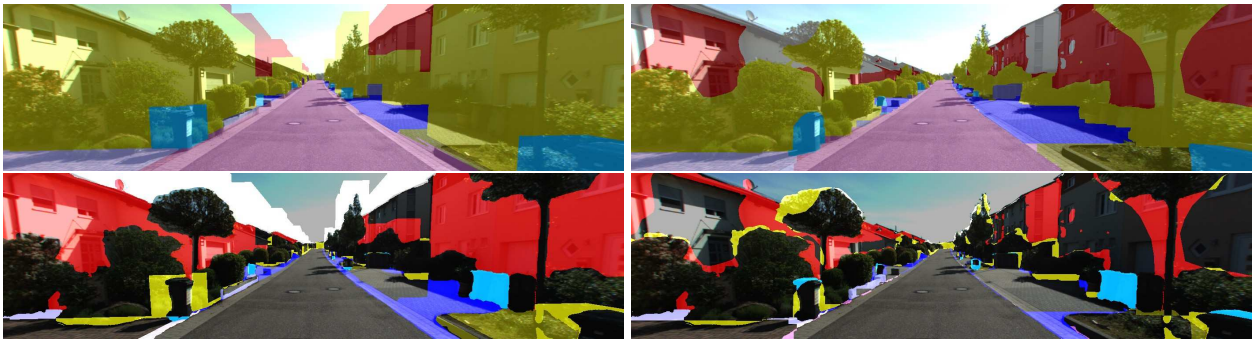
(a) Label Prop. [6]

(b) Sparse Track. + GC [5]



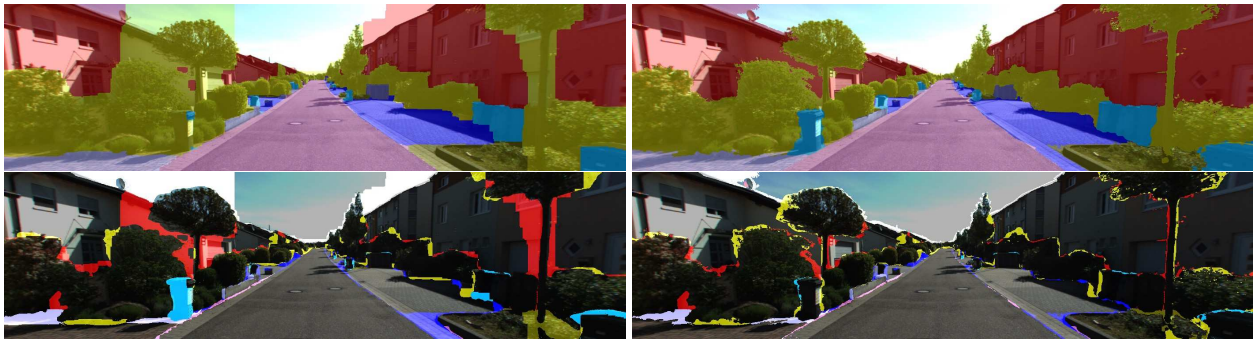
(c) 3D Prop. + GC

(d) Fully Conn. CRF [4]



(e) 3D Primitives + GC

(f) 3D Mesh + GC

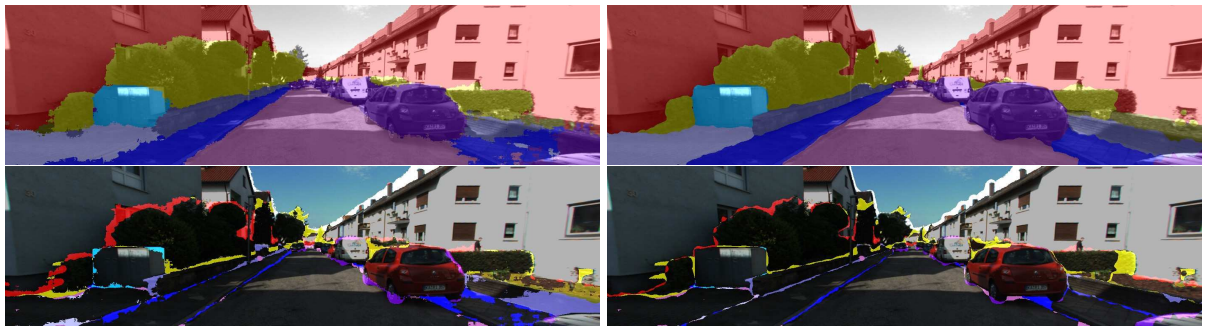


(g) 3D Points + GC

(h) Proposed Method

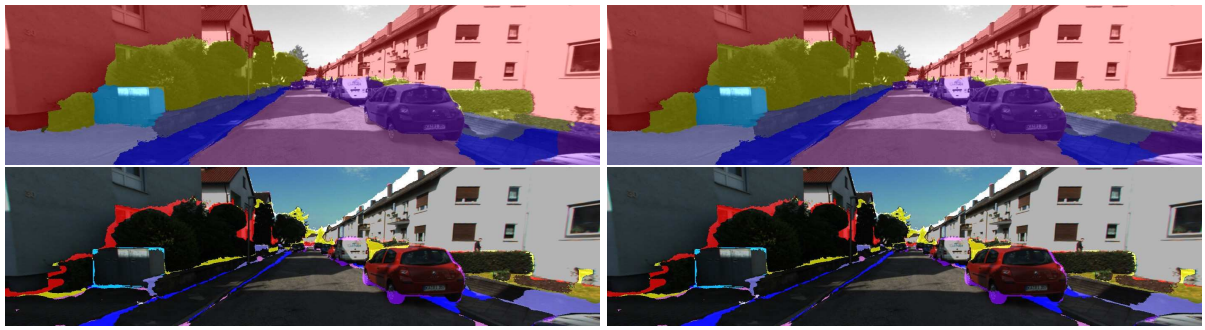
Figure 11: **Comparison to Baselines.** Each subfigure shows from top-to-bottom: the input image with inferred semantic segmentation and the errors with respect to 2D ground truth annotation where colors indicate ground truth labels.

9. Qualitative Comparison of Ablation Study



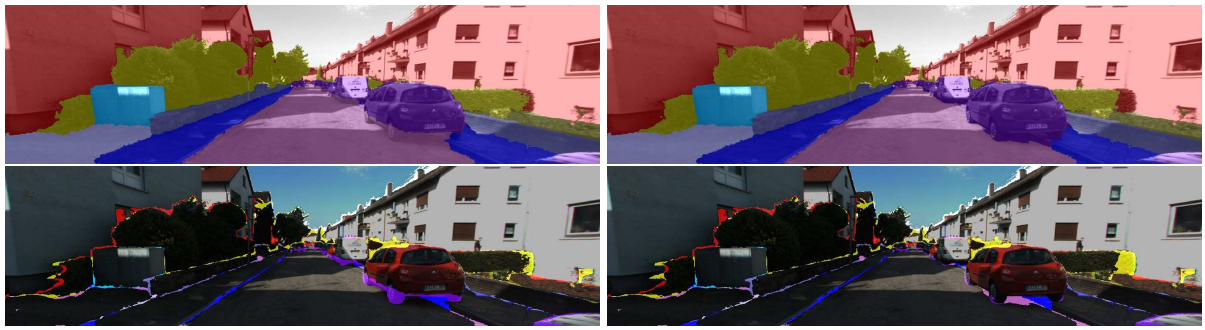
(a) LA

(b) LA + 3D



(c) LA + PW

(d) LA + PW + CO



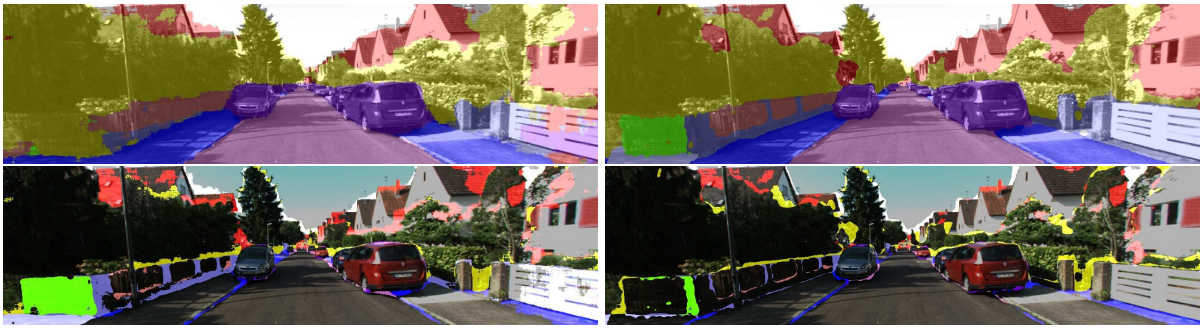
(e) LA + PW + CO + 3D

(f) LA + PW + CO + 3D + 3D PW

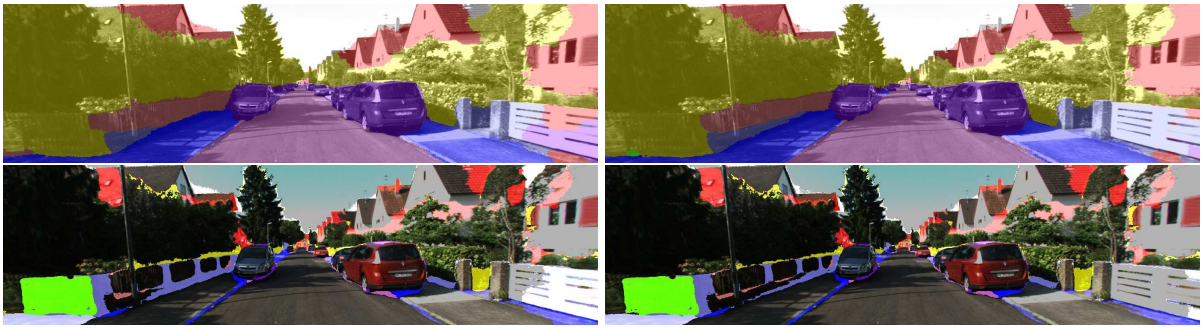
(g) Full Model

(h) Projected 3D points (top) and Line segments (bottom)

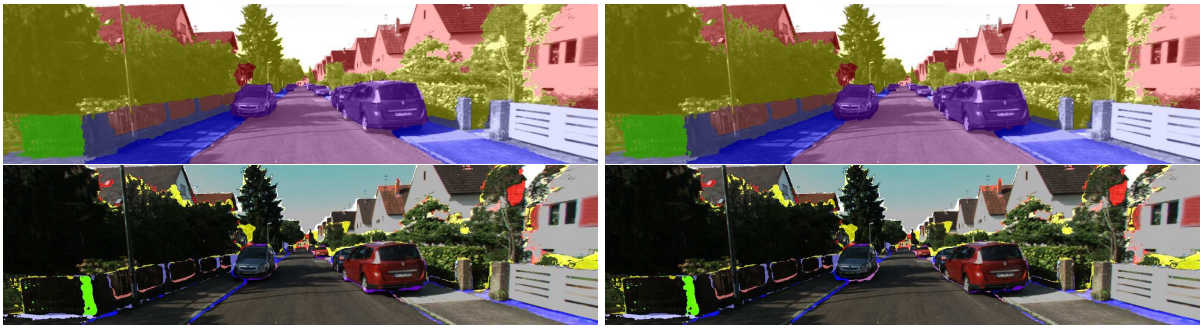
Figure 12: **Qualitative Results for Ablation Study.** Each subfigure (except the last one) shows from top-to-bottom: the input image with inferred semantic segmentation, and the errors with respect to 2D ground truth annotation where colors indicate ground truth labels. The last subfigure shows the projected 3D points (top) and detected line segments in green (bottom). Abbreviations are defined the same as in the paper.



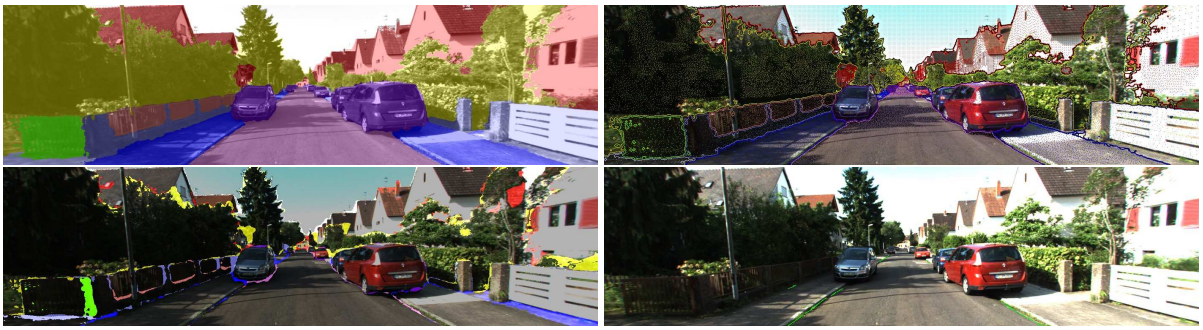
(a) LA (b) LA + 3D



(c) LA + PW (d) LA + PW + CO



(e) LA + PW + CO + 3D (f) LA + PW + CO + 3D + 3D PW



(g) Full Model (h) Projected 3D points (top) and Line segments (bottom)

Figure 13: **Qualitative Results for Ablation Study.** Each subfigure (except the last one) shows from top-to-bottom: the input image with inferred semantic segmentation, and the errors with respect to 2D ground truth annotation where colors indicate ground truth labels. The last subfigure shows the projected 3D points (top) and detected line segments in green (bottom). Abbreviations are defined the same as in the paper.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002. 2

[2] L.-C. Chen, S. Fidler, A. L. Yuille, and R. Urtasun. Beat the mturkers: Automatic image labeling from weak 3d supervision. In *CVPR*, 2014. 5

[3] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009. 5

[4] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NIPS*, 2011. 5, 12, 13

[5] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 12, 13

[6] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *ECCV*, 2012. 12, 13