Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry Supplementary Material

Nan Yang[1,2], Rui Wang[1,2], Jörg Stückler[1], and Daniel Cremers[1,2]

[1] Technical University of Munich
[2] Artisense
{yangn,wangr,stueckle,cremers}@in.tum.de

1 Introduction

In this supplementary material, we provide additional technical detail and results for our paper. Firstly, we give details on the architecture of StackNet. In Section 2, we provide additional results of our monocular depth estimation network on the test split [3] of KITTI Raw [6]. We also give depth estimation results on the Cityscapes dataset [2] as well as the Make3D dataset [10] using our model trained on KITTI to demonstrate generalization abilities of our model. In Section 3, we provide more results of DVSO on the KITTI odometry benchmark. Finally, we demonstrate generalization abilities of DVSO trained on KITTI to the Cityscapes *Frankfurt* sequence.

2 Monocular Depth Estimation

2.1 Detailed Architecture of StackNet

The detailed architecture of StackNet is shown in Figure 1. The upper part of Figure 1 describes the architecture of SimpleNet and the lower part describes ResidualNet. The cubes are feature maps and they are independent between SimpleNet and ResidualNet. The colorized cubes indicate feature maps that are concatenated in skip connections from encoder with corresponding maps in the decoder layers. The symbols above the arrows represent the operations on the input layer such as convolution, residual block, max-pooling, up-sampling and activation function. Note that multiple operations are performed from top to bottom. Two kinds of residual blocks with bottleneck design are used in our architecture. One performs an additional convolution on the shortcut due to the feature map expansion or down-sampling with stride-2 convolution, whereas the other block maps the shortcut directly (identity mapping).

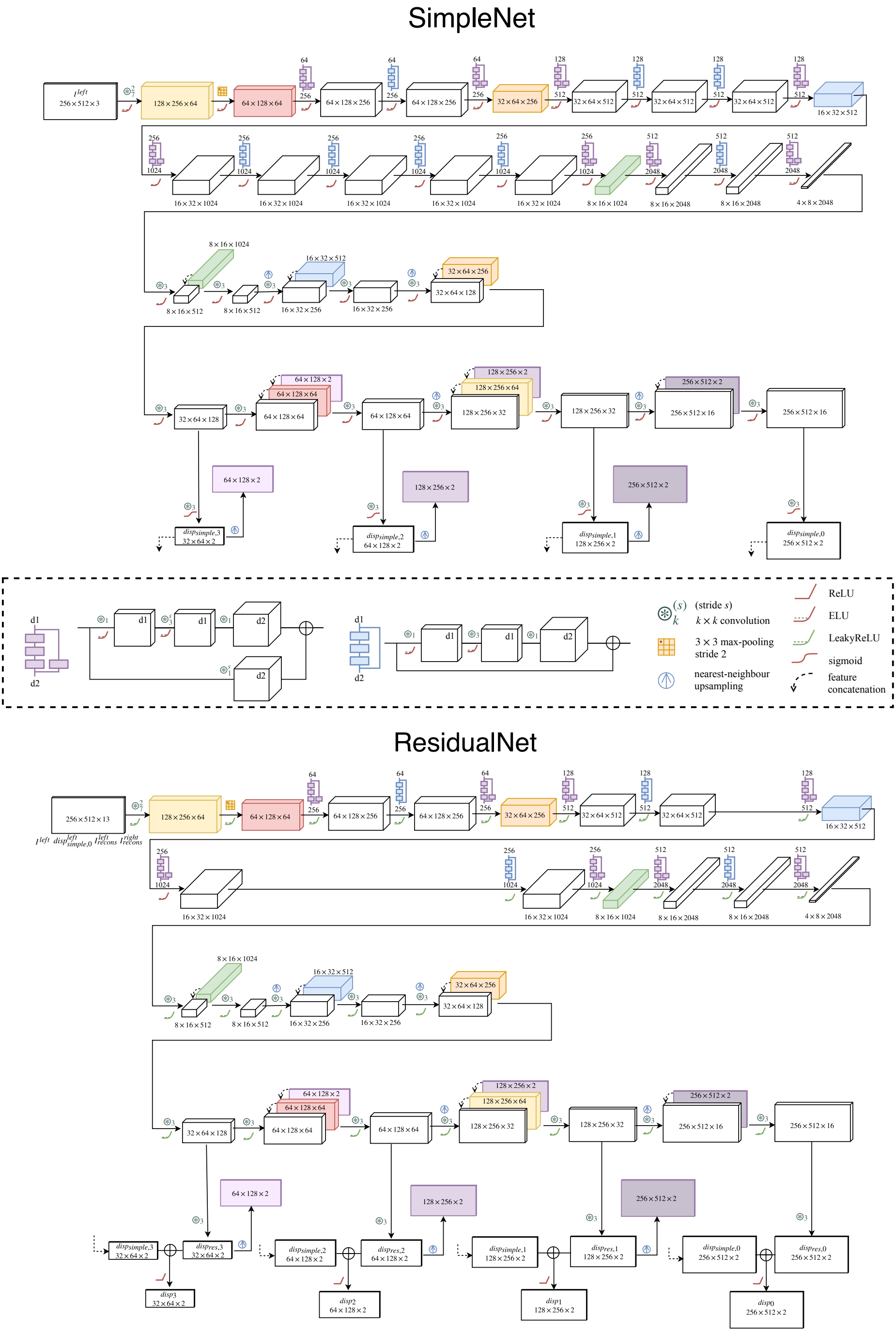


Fig. 1: Detailed network architecture of StackNet.

2.2 KITTI

Dataset Schedule and Parameter Tuning The dataset schedule is inspired by [8] where the authors propose that training the network firstly on the relatively easier dataset and then the difficult dataset delivers better performance than the other schedules. The two splits of our training set, $\mathbf{K}_o$ and $\mathbf{K}_r$ also have similar property. The images in $\mathbf{K}_o$ are from KITTI odometry benchmark which contains the scenarios where less objects, e.g., pedestrians and cars, appear, whereas the images in $\mathbf{K}_r$ are from more complicated scenarios. Therefore, we firstly train our network on $\mathbf{K}_o$ and then $\mathbf{K}_r$. As shown in *Table 1* from the main paper, the other dataset schedule delivers worse performance. Furthermore, since only self-supervised training can be applied on $\mathbf{K}_r$, the results of the network contain many outliers in the reflectance and occlusion area. Therefore, we again train the network on $\mathbf{K}_o$ without $\mathcal{L}_U$ using less epochs to smoothen the outliers and increase the precision with supervised learning. For the parameters to be tuned in *Equation (1)* from the main paper, α_u, α_{lr} and α_{smooth} are referred to [7]. For the weight α_s, we tested 1, 5, 10, 20, 50 and 100 on the validation set for which 10 performed the best. In fact, when α_s is set to 10 or 20, the scales of $\mathcal{L}_U$ and $\mathcal{L}_S$ are consistent when the total loss tends to converge. Afterwards, we fixed all the weights except α_{occ}. The values 0.1, 0.01, and 0.001 are tested for α_{occ} and 0.01 showed the best performance. Using a hyper-parameter optimization framework like hyperopt [1] could further improve the parameter tuning.

Ablation Study In Table 1 we give an ablation study which demonstrates the effectiveness of the different loss terms in *Equation (1)* from the main paper.

Further Results In Figure 2, we show more qualitative results of our model on the test split on KITTI Raw proposed by Eigen et al. [3]. In comparison, we also demonstrate the results predicted by the model of Godard et al. [7]. Note that our model is semi-supervisedly trained on KITTI, while the model of Godard et al [7] is self-supervisedly trained on the Cityscapes dataset as well as the KITTI dataset. The ground truth is interpolated for better visualization. In the results shown in Figure 2, our method provides better predictions on thin structures and predicts less shadow effects around object contours than the other approach. On the bottom right, our method is not able to recover the round traffic sign on the upper right part of the image. In Figure 3, we also show the error maps (on sparse LIDAR points) of the images in *Figure 4* from the main paper to compare between [7] and ours.

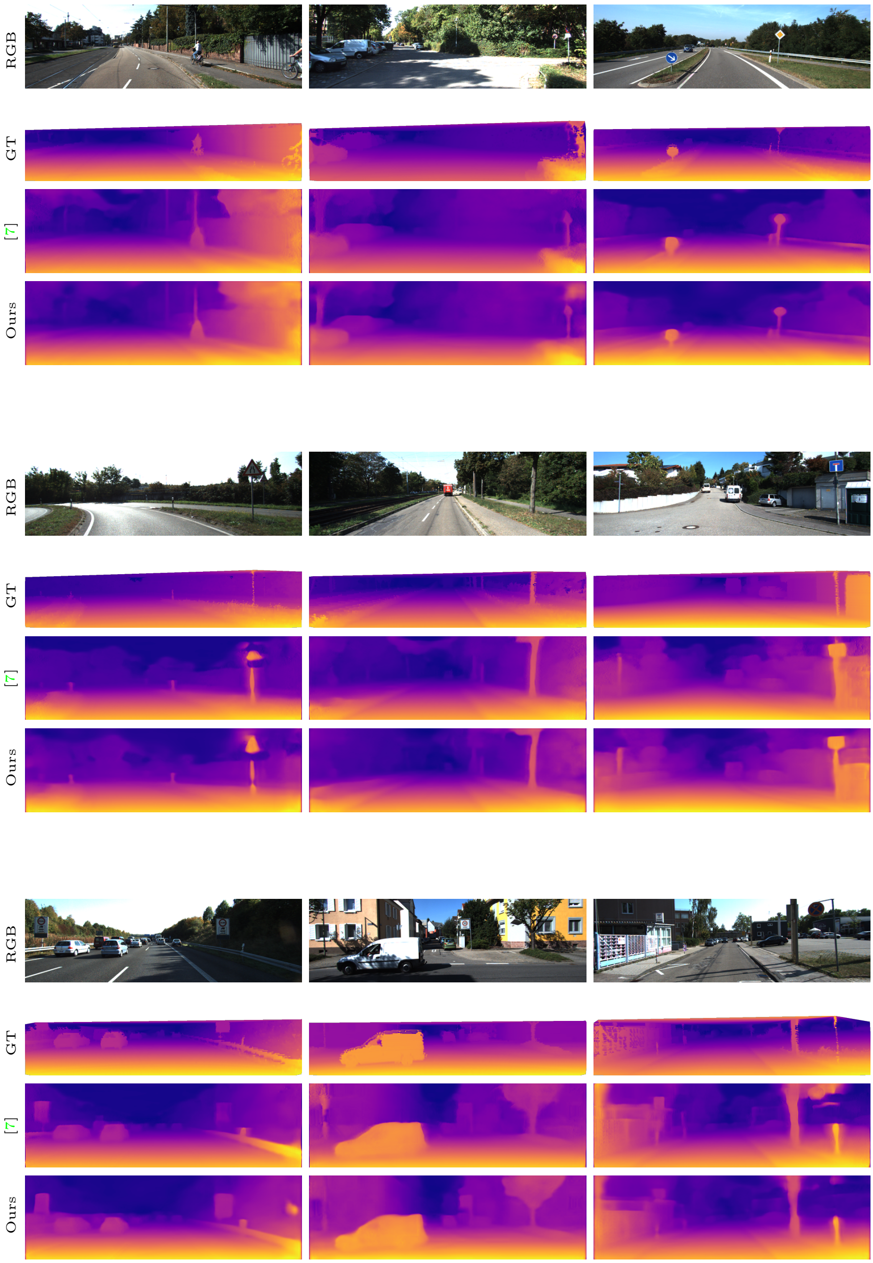


Fig. 2: Qualitative results and comparison on KITTI Raw test set (split of Eigen et al. [3]). Our method provides better predictions on thin structures and predicts less shadow effects around object contours. On the bottom right, our method cannot recover the round traffic sign.

	RMSE	RMSE (log)	ARD	SRD	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Approach	lower is better				higher is better		
$\mathcal{L}_S + \mathcal{L}_{smooth} + \mathcal{L}_{occ}$	4.718	0.194	0.106	0.919	0.869	0.946	0.975
$\mathcal{L}_U + \mathcal{L}_{lr} + \mathcal{L}_S$	4.517	0.189	*0.098*	0.735	0.878	0.950	0.978
$\mathcal{L}_U + \mathcal{L}_{lr} + \mathcal{L}_S + \mathcal{L}_{smooth}$	4.529	0.190	0.100	0.746	0.879	0.950	0.977
$\mathcal{L}_U + \mathcal{L}_{lr} + \mathcal{L}_S + \mathcal{L}_{occ}$	*4.473*	*0.186*	*0.098*	*0.744*	*0.880*	*0.952*	*0.979*
Full	**4.442**	**0.187**	**0.097**	**0.734**	**0.888**	**0.958**	**0.980**

Table 1: Ablation study on the loss terms in *Equation (1)* form the main paper. Surprisingly, we found that the smoothness term $\mathcal{L}_{smooth}$ does not improve the performance unless it is combined with the occlusion term $\mathcal{L}_{occ}$.

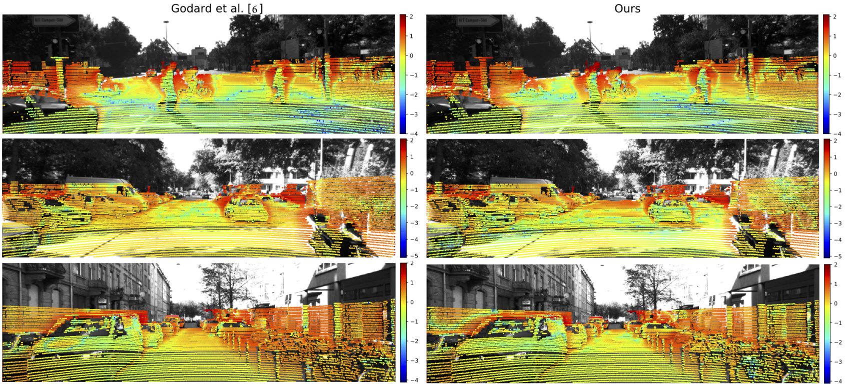


Fig. 3: Error maps for the images in *Figure 4* from the main paper. The values of the errors are rescaled with log_{10} and colorized with 'jet' colormap.

2.3 Generalization to Other Datasets

To show the generalization ability of our approach, we compare the results of our method and other state-of-the-art methods on the Cityscapes and the Make3D datasets in Figure 4. All methods have been trained on the KITTI Dataset. For the Make3D dataset, we show predictions on the original images as well as on a central crop as in Godard et al. [7]. Qualitatively, our method makes similar well predictions than Godard et al. [7] but can recover thin structures better.

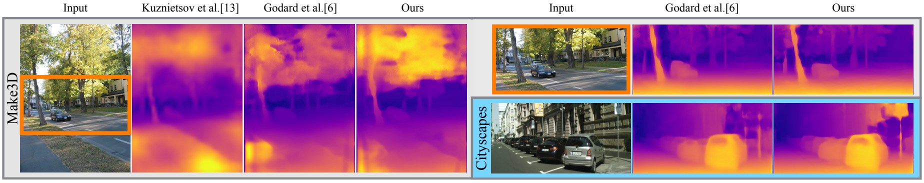


Fig. 4: Generalization results on Cityscapes and Make3D. On Make3D we show results for original images and the central crop as in [7]. On both datasets our approach appears to capture the contours of objects well.

In Figure 5, we give further qualitative results for generalization on the Cityscapes dataset. We also compare our method to Godard et al. [7]. Note that again both models are trained on KITTI only. From the results in Figure 5 it can be observed that our model better predicts the contours of objects like the traffic signs. The last row shows failure cases where both methods are not able to accurately recover the depth of the van, the train and some pedestrians.

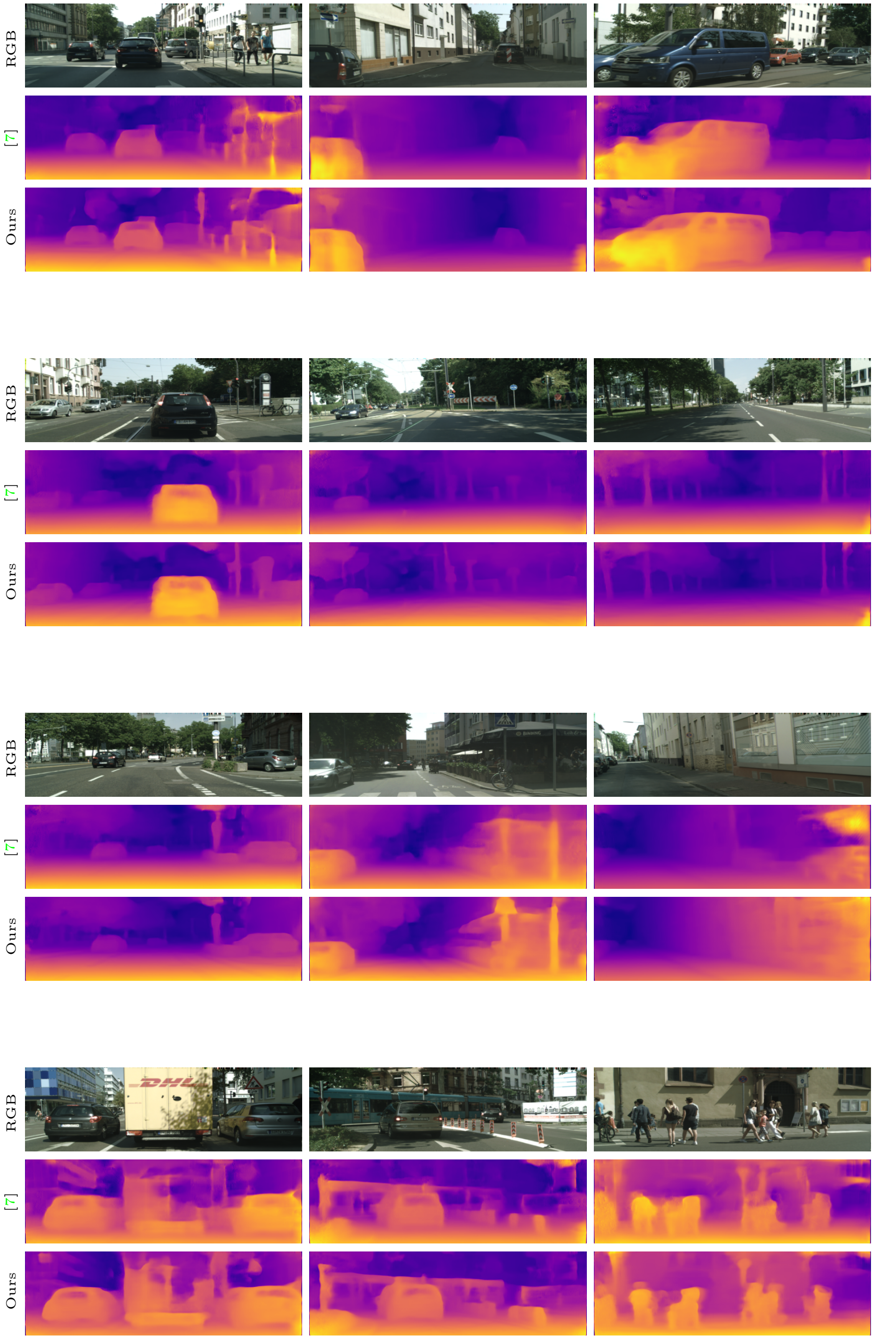


Fig. 5: Qualitative results demonstrating generalization to Cityscapes of models trained on KITTI. Our model better predicts the contours of objects like the traffic signs. The last row shows failure cases where both methods cannot accurately recover the depth of some objects.

3 Monocular Visual Odometry

3.1 KITTI

In Figures 6 to 8 , we show all the estimated trajectories by DVSO in the training set (sequences 00-10) of the KITTI odometry benchmark compared with other state-of-the-art monocular [4,9] and stereo [5,11] visual odometry methods. We also show the full evaluation results on the test set in Figure 9 (accuracy plots) and Figure 10 (trajectory plots)[1].

On the training set sequences, we observe that both monocular ORB-SLAM2 and DSO suffer from strong scale drift, while DVSO barely drifts in scale (see Figures 6 to 8). DVSO furthermore achieves comparable results to stereo methods despite using only monocular images.

Quantitatively, on the test set sequences, DVSO shows lower translational error with higher driving speed in average (see Figure 9). We suspect two reasons: 1) Depth initialization of points becomes more difficult for geometry-based methods with high driving speed due to the relatively low frame rate of the KITTI dataset. 2) The highway scenes contain only few image regions with stable features for tracking forward driving motion. This also makes depth initialization challenging for geometry-based methods.

3.2 Generalization to Cityscapes

We demonstrate generalization capabilities of DVSO on the Cityscapes *Frankfurt* sequence from frame 0 to 32000, which covers around 10 kilometers path in a dynamic urban environment. Estimated trajectories for different length fragments are shown in Figure 11. Since the camera parameters between KITTI and Cityscapes dataset are different, we Sim(3)-align the estimated trajectories to the GPS ground truth for qualitative comparison. As can be seen, DVSO works well in frames 0-20000. Afterwards, the drift becomes larger.

3.3 Runtime

For measuring and evaluating the computational complexity of the optimization procedure, we ran DVSO, monocular DSO and Stereo DSO on KITTI 00 for 5 times and measured the average execution time of the total optimization procedure. The same parameters are applied for the three methods, i.e., 7 active keyframes, 2000 active points and max. 6 iterations for Gauss-Newton. With resolution 512×256, monocular DSO took 24.66ms, DVSO 35.73ms and Stereo DSO 43.18ms on CPU. The inference step of the deep network of DVSO required about 40ms on GPU. As a consequence, the overall processing frame-rate of DVSO remains roughly the same when parallelizing CPU- and GPU processing.

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

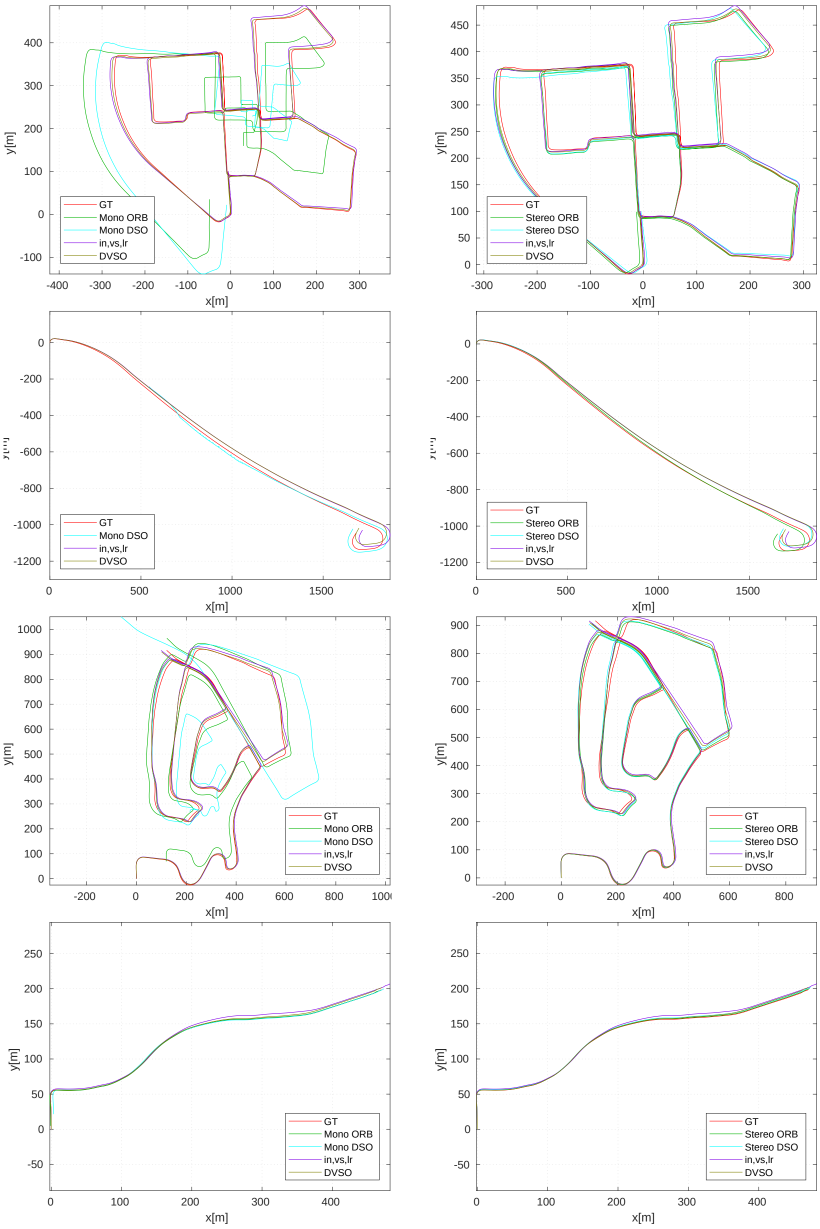


Fig. 6: Results of DVSO and state-of-the-art monocular and stereo methods on KITTI odometry seq. 00-03. Mono ORB-SLAM2 and DSO suffer from strong scale drift, while DVSO barely drifts in scale. DVSO (monocular) achieves comparable results to stereo methods. Note that for sequence 01, the result of Mono ORB is not shown due to its very large scale drift.

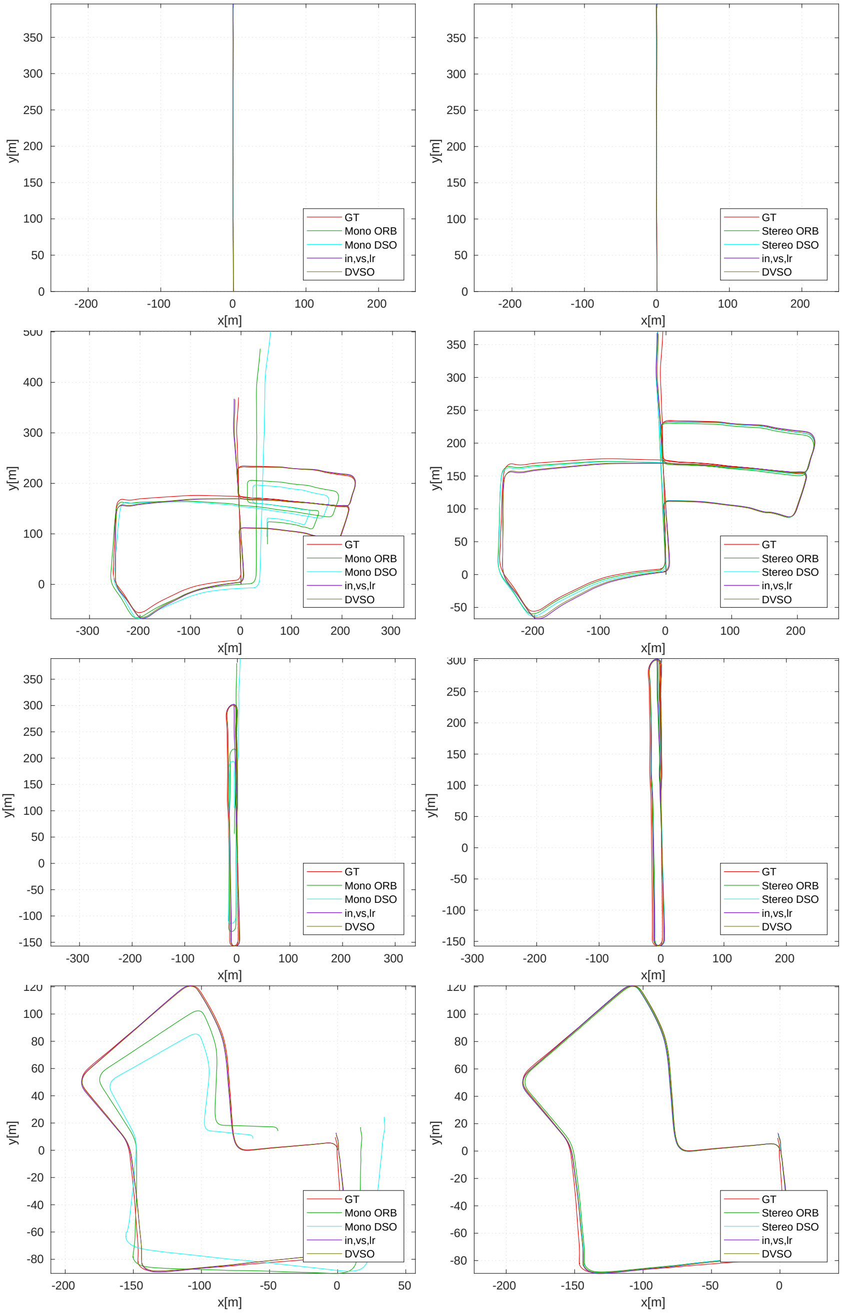


Fig. 7: Results of DVSO and state-of-the-art monocular and stereo methods on KITTI odometry seq. 04-07. Mono ORB-SLAM2 and DSO suffer from strong scale drift, while DVSO barely drifts in scale. DVSO (monocular) achieves comparable results to stereo methods.

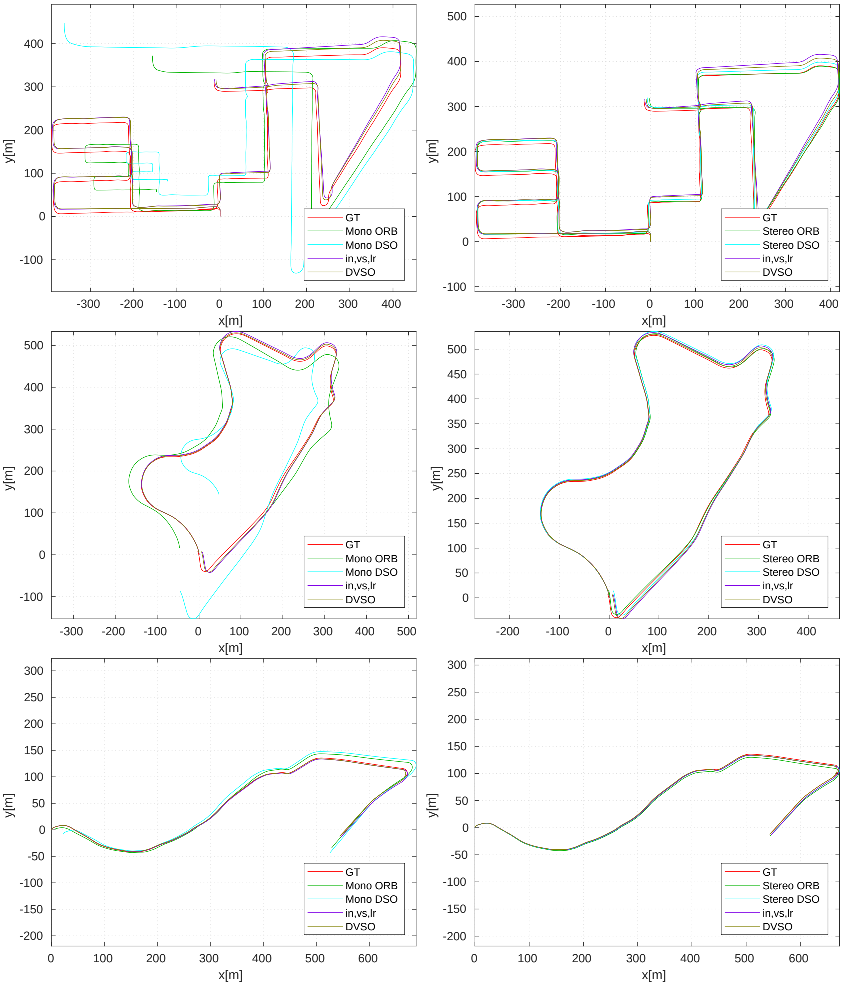


Fig. 8: Results of DVSO and state-of-the-art monocular and stereo methods on KITTI odometry seq. 08-10. Mono ORB-SLAM2 and DSO suffer from strong scale drift, while DVSO barely drifts in scale. DVSO (monocular) achieves comparable results to stereo methods.

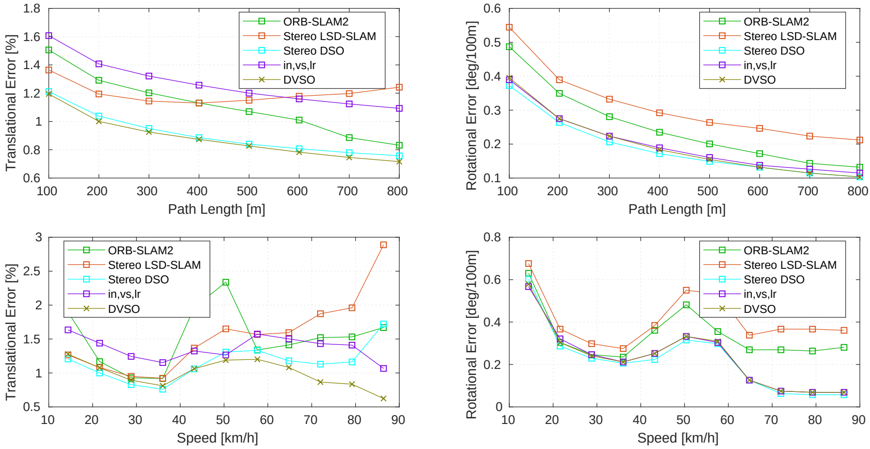


Fig. 9: Evaluation results of DVSO and other state-of-the-art methods on the KITTI odometry test set. Top: translational and rotational errors wrt. driving intervals. Bottom: translational and rotational errors wrt. driving speed. DVSO shows lower translational error with higher driving speed.

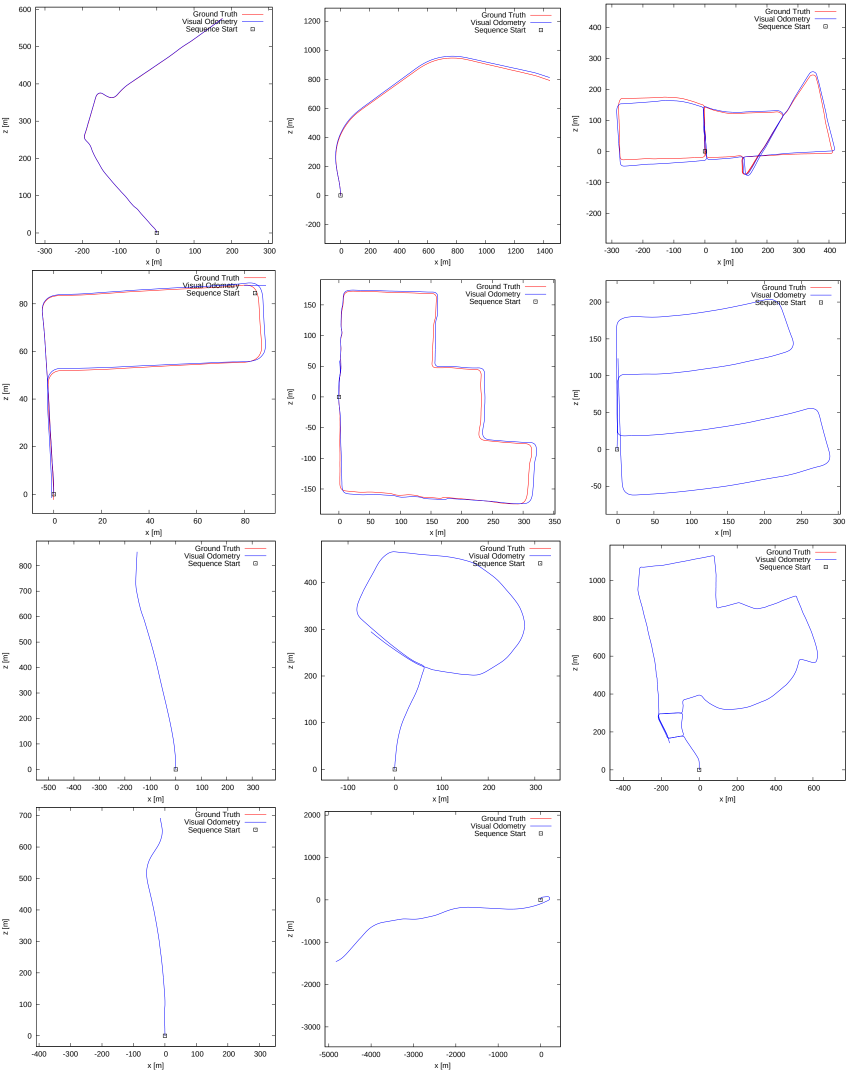
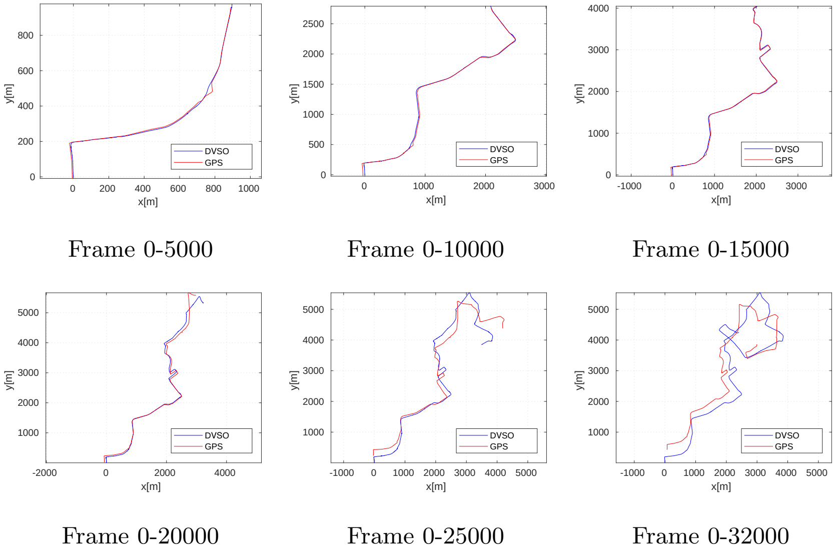


Fig. 10: Results of DVSO on KITTI odometry seq. 11-21. Note that for seq. 11-15, we downloaded the evaluation results from the KITTI website. Seq. 16-21 are not provided on the website and ground-truth is not available.



Frame 0-5000

Frame 0-10000

Frame 0-15000

Frame 0-20000

Frame 0-25000

Frame 0-32000

Fig. 11: Generalization result of DVSO on the Cityscapes *Frankfurt* sequence. The estimated trajectories are Sim(3) aligned to GPS ground truth. DVSO works well in frames 0-20000, while the drift becomes larger afterwards.

References

1. Bergstra, J., Yamins, D., Cox, D.D.: Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In: Proceedings of the 12th Python in Science Conference. pp. 13–20. Citeseer (2013) 3
2. 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. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 1
3. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. pp. 2366–2374 (2014) 1, 3, 4
4. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence (2017) 8
5. Engel, J., Stückler, J., Cremers, D.: Large-scale direct slam with stereo cameras. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 1935–1942. IEEE (2015) 8
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 1
7. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. arXiv preprint arXiv:1609.03677 (2016) 3, 4, 6, 7
8. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2 (2017) 3
9. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017) 8
10. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Advances in neural information processing systems. pp. 1161–1168 (2006) 1
11. Wang, R., Schwörer, M., Cremers, D.: Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In: International Conference on Computer Vision (ICCV). Venice, Italy (October 2017) 8