Supplementary Materials for "Towards Better Generalization: Joint Depth-Pose Learning without PoseNet"

Wang Zhao Shaohui Liu Yezhi Shu Yong-Jin Liu
Department of Computer Science and Technology, Tsinghua University, Beijing, China
zhao-w19@mails.tsinghua.edu.cn, b1ueber2y@gmail.com,
shuyz19@mails.tsinghua.edu.cn, liuyongjin@tsinghua.edu.cn

This document provides a list of supplemental materials that accompany the main paper.

- **Discussion on Scale-Invariant Design** - We provide more detailed discussion for the scale-invariant design in our system in Section A.
- **Derivation of Triangulation Module** - We include the detailed derivation of differentiable triangulation module in Section B.
- **Details for PoseNet and PoseNet-Flow** - We introduce more details and results about PoseNet and PoseNet-Flow in Section C.
- **Additional Results and Discussion for PoseNet-Flow** - We present additional experiemental results for PoseNet-Flow on visual odometry in Section D.
- **Implementation Details** - We provide more implementation details about network architectures and system hyperparameters in Section E.
- **Additional Comparison on sampled KITTI Odometry dataset** - We show more comparsion results about sampled KITTI Odometry dataset in Section F.
- **Numerical Results of TUM-RGBD dataset** - We report quantitative results for TUM-RGBD dataset in Section G.
- **Additional Visualizations** - In Section H, we provide additional visualizations generated by our system on different datasets.

A. Discussion on Scale-Invariant Design

Given a pair of input images, assume that the fundamental matrix can be accurately recovered from point correspondence and no additional priors exist, the relative translation of the pair should be up to an arbitrary scale. On the other hand, the monocular depth estimation aims to use learned priors from data to directly infer the corresponding depth image. Assume that the intrinsic parameters of the camera are known and fixed, the system can possibly make use of the common priors such as the height of human, the width of the car as well as subtle structural clues to infer the monocular depth, which does not suffer from the scale ambiguity problem.

Most previous works (e.g., [15]) use two separate convolutional neural networks to learn both monocular depth and relative pose, and directly put photometric consistency constraint by using the predicted relative pose to reproject the predicted depth. This makes the assumption that the scale of the predicted relative pose should correspond to the predicted monocular depth, which means that the relative pose estimation is required to not only learn the feature matching and relative pose recovery, but also implicitly learn the scale priors which are exactly the same as the monocular depth estimation is required to learn. This requires the network to firstly infer scale from two input images respectively, and implicitly integrate the predicted scale into the recovered relative pose, making the learning of pose prediction network extremely hard and degrade its generalization capability.

Our method explicitly resolves this problem with two novel designs:

- 1. We use an optical flow network to specifically learn pixelwise matching, then solve the fundamental matrix and recover the relative pose up an arbitrary scale.
- 2. We triangulate the predicted correspondence and explicitly align the predicted depth to the triangulated point clouds to compute the error map.

In this way, the relative pose prediction is not required to implicitly learn the scale priors. This significantly improves the generalization both for training on indoor environments and inference on video sequences with unseen camera egomotion. Note that, the two designs are necessary to be coupled together. Suppose that if the system only employs de-

sign 1 without aligning the depth to the triangulated point clouds, the joint training cannot converge because it is impossible to fit the scale of the depth estimation network to an arbitrary scale of relative pose.

Based on the previous discussion, we can infer that our system is robust under the circumstances where the camera intrinsic parameters are known and fixed. When the camera intrinsic parameters are flexible across different sequences on training and inference, only under the assumption that the monocular depth estimation network can automatically learn the camera calibration from structural clues in the single image can our method still accurately recover the depth image. Otherwise, further system designs on the monocular depth network are required to disentangle the influence of different camera field of view to make the learning problem feasible.

B. Derivation of Triangulation Module

We adopt mid-point triangulation method to build an up-to-scale 3D structure from 2D correspondences and relative pose. Mid-point triangulation problem could be easily solved with linear algorithms. The objective function is as follows:

$$x = \arg\min_{x} ' = \arg\min_{x} [d(\mathcal{L}_1; x)]^2 + [d(\mathcal{L}_2; x)]^2 \quad (1)$$

Where $\mathcal{L}_1 = \{p = c_1 + {}_1 n_1 \mid {}_1 \in \mathrm{R}\}$ and $\mathcal{L}_2 = \{p = c_2 + {}_2 n_2 \mid {}_2 \in \mathrm{R}\}$ are two camera rays generated with optical flow correspondence, and d denotes the euclidean distance. $c_i = -R_i^T t_i$ is the ray origin, where $[R; t]$ is the camera extrinsic, and $n_i = R_i^T K^{-1}[x_0; y_0; 1]^T$ is the ray direction, where $[x_0; y_0]$ is the pixel coordinate. The objective function could be written as:

$$'(x; {}_1; {}_2) = \|c_1 + {}_1 n_1 - x\|^2 + \|c_2 + {}_2 n_2 - x\|^2 \quad (2)$$

To minimize $'(x)$, we need $\frac{@'}{@x} = 0$ which easily gives us:

$$x = \frac{(c_1 + {}_1 n_1) + (c_2 + {}_2 n_2)}{2} \quad (3)$$

After substitution of x, the cost function becomes:

$$'(x; {}_1; {}_2) = \frac{1}{2}\|(c_1 + {}_1 n_1) - (c_2 + {}_2 n_2)\|^2 \quad (4)$$

Then we have:

$$\begin{aligned} \frac{@'}{@ {}_1} &= n_1^T ({}_1 n_1 - {}_2 n_2 + c_1 - c_2) = 0 \\ \frac{@'}{@ {}_2} &= n_2^T ({}_2 n_2 - {}_1 n_1 + c_2 - c_1) = 0 \end{aligned} \quad (5)$$

From these two linear equations, the solutions of ${}_1$ and ${}_2$ could be expressed as:

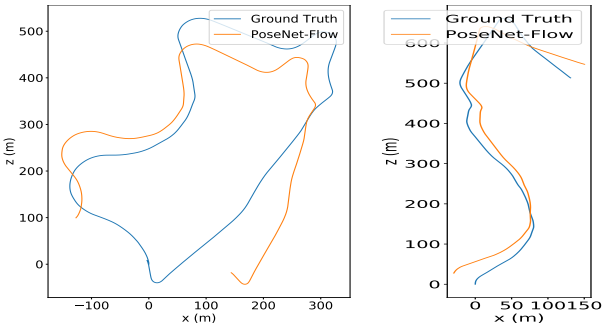

Figure 1. Visual odometry results of PoseNet-Flow method on original sequence 09 and 10.

$$\begin{bmatrix} {}_1 \\ {}_2 \end{bmatrix} = A \begin{bmatrix} \|n_1\|^2 & n_1^T n_2 \\ n_2^T n_1 & \|n_2\|^2 \end{bmatrix} \begin{bmatrix} n_1^T (c_2 - c_1) \\ n_2^T (c_1 - c_2) \end{bmatrix} \quad (6)$$

$$A = \frac{1}{\|n_1\|^2 \|n_2\|^2 - (n_1^T n_2)^2} \quad (7)$$

The triangulation solution x is then computed with Eq. (3). By this way, the triangulation module is naturally differentiable.

C. Details for PoseNet and PoseNet-Flow

We implement two baseline methods, named *PoseNet* and *PoseNet-Flow*, to compare with our method. *PoseNet* system takes image pairs as input, predicts monocular depth and relative pose by depth and pose branch, respectively. The depth branch uses the same network as our system and the pose branch adopts standard PoseNet [4]. Following previous PoseNet-based unsupervised depth pose joint learning methods [15, 1], we utilize photometric loss and depth reprojection loss to train the network. For *PoseNet-Flow* system, we add a flow network to generate optical flow, and feed generated optical flow, rather than RGB image pair, to PoseNet for relative pose estimation. The flow network is the same as that of our system. The depth network and the depth-pose training objectives remain the same as *PoseNet* system. We adopt two-stage training stragegy for *PoseNet-Flow* system. In the first stage we train the optical flow network. Then the flow network is frozen and both the depth and pose networks are joint trained.

D. Additional Results and Discussion for PoseNet-Flow

Table **??** shows the depth estimation results of *PoseNet* and *PoseNet-Flow* in indoor NYUv2 dataset. Due to complex camera motions and large textureless regions, traditional *PoseNet* method fails to generate plausible predictions. *PoseNet-Flow* uses optical flow for pose regression,

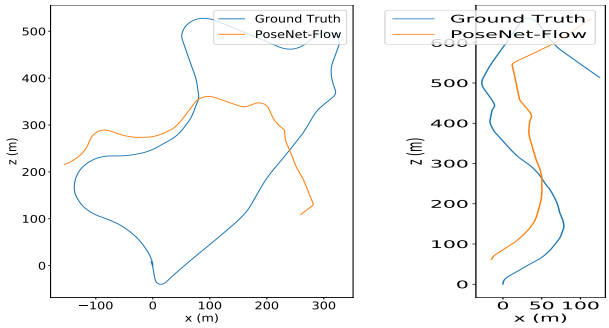

Figure 2. Visual odometry results of PoseNet-Flow method on sampled sequence 09 and 10 with stride 3.

thus improves the interpretability of the system and makes learning problem easier. This is also discussed in [14]. To further explore the capacity of *PoseNet-Flow* system, we conduct experiments on KITTI Odometry dataset. We use two consecutive images as training pairs. Figure 1 and Figure 2 show the results of standard KITTI dataset and sampled KITTI dataset with stride 3. While the *PoseNet-Flow* system could produce feasible results on NYUv2 and standard KITTI dataset, it still tends to fail on unseen ego-motions. This could possibly due to the nature of trained PoseNet that it performs more like image retrieval rather than solving physical constraints and thus works well only on the test data which is similar with training samples. On the contrary, our method works well under all these challenging scenarios, showing much improved robustness and generalization ability.

E. Implementation Details

Here we introduce more details about network architectures and training objectives used in our system.

For depth estimation network, we adopt a same encoder-decoder network with skip connections as proposed in [2]. Specifically, ResNet-18 is used as encoder and DispNet [5, 3] is used as decoder with ELU nonlinearities for all conv layers except output layer, where we use sigmoids and convert the output disparity to depth with $D = 1=(ad+b)$. a and b are set to be 0.1 and 100 to constrain the range of output depth. We only supervise the largest scale of depth output, and replace the nearest upsampling layers in decoder with bilinear upsampling, which makes the training more stable. The depth loss consists of three parts, triangulation depth loss L_d, reprojection loss L_p and edge-aware depth smoothness loss L_s. The detailed descriptions of L_d and L_p are included in the main paper. Given image input I_t and disparity prediction d_t, depth smooth loss L_s is computed as follows:

$$L_s = |@_x d_t^n| \, e^{\,j@_x I_t j} + |@_y d_t^n| \, e^{\,j@_y I_t j} \qquad (8)$$

where $d_t^n = d_t=\overline{d_t}$ is the normalized disparity prediction to

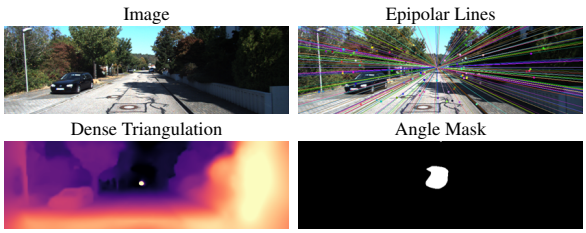

Figure 3. The white area in angle mask means extremely small angles between two rays or negative triangulation depths. Small ray angles and negative depths often happen near epipoles, which are the intersection points of all epipolar lines.

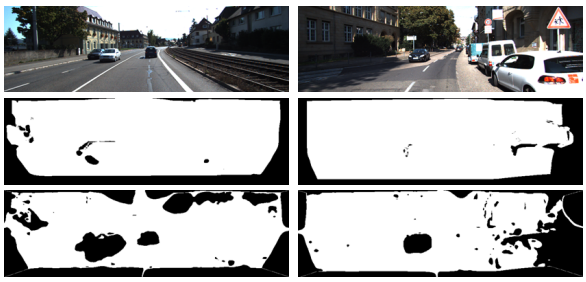

Figure 4. **Top to bottom:** Image, occlusion mask, inlier map. The inlier map is converted to binary mask for better visualization. The occlusion masks and inlier maps could successfully filter out occlusions and non-rigid regions respectively.

avoid depth shrinking, proposed by [9].

For flow estimation network, we adopt the PWCNet [8] as backbone for predicting forward and backward optical flow of an image pair. We utilize the backward warping method proposed in [10] to explicitly handle occlusions. Generated occlusion masks are not only used as a better supervision for the optical flow, but also for sampling reliable pixel matches when solving relative pose and triangulation. Optical flow is predicted and supervised at three different scales. Following [12, 16], we use a combination of L1 loss, SSIM loss [11] and flow smoothness loss for flow supervision. Therefore, the total flow loss L_f is expressed as:

$$L_f = (1- \;)\|I_a - I_b\| + \frac{}{2}(1-SSIM(I_a; I_b)) + \;\; L_{fs} \qquad (9)$$

where L_{fs} is the flow smoothness loss which has a similar formulation as Eq. (8). and are set to be 0.85 and 0.1 respectively.

For relative pose estimation, we recover it by solving fundamental matrix. Specifically, we first compute optical flow forward-backward distance map D_{fb} by flow warping. Then forward-backward score map M_s is generated as $M_s = 1=(0.1 + D_{fb})$. Together, $M_o * M_s$ is used for sampling accurate correspondences from dense flow. We sample the top 20% correspondences according to score map and then randomly sample 6k matches. We perform this sampling strategy, rather than directly top sampling, to discourage spatial accumulation of sampled matches. Then

Methods	Seq. 09		Seq. 10	
	t_{err} (%)	r_{err} (=100m)	t_{err} (%)	r_{err} (=100m)
ORB-SLAM2^{y} [6]	11.12	**0.33**	2.97	0.36
ORB-SLAM2 [6]	**2.37**	0.40	**2.97**	**0.36**
Zhou *et al.* [15]	24.75	7.79	25.09	11.39
Depth-VO-Feat [13]	20.54	6.33	16.81	7.59
CC [7]	24.49	6.58	19.49	10.13
SC-SfMLearner [1]	33.35	8.21	27.21	14.04
Ours	**7.02**	**0.45**	**4.94**	**0.64**

Table 1. Visual odometry results on sampled sequence 09 and 10 with stride 2. The average translation and rotation errors are reported. ORB-SLAM2† indicates disablement of loop closure.

Methods	Seq. 09		Seq. 10	
	t_{err} (%)	r_{err} (=100m)	t_{err} (%)	r_{err} (=100m)
ORB-SLAM2 [6]	X	X	X	X
Zhou *et al.* [15]	61.24	18.32	38.94	19.62
Depth-VO-Feat [13]	42.33	11.88	25.83	11.58
CC [7]	51.45	14.39	34.97	17.09
SC-SfMLearner [1]	59.32	17.91	42.25	21.04
Ours	**7.72**	**1.14**	**17.30**	**5.94**

Table 2. Visual odometry results on sampled sequence 09 and 10 with stride 4. The average translation and rotation errors are reported.

we run the normalized 8-point algorithm in RANSAC loop to solve fundamental matrix. The RANSAC inlier threshold and desirable confidence are set to be 0.1 and 0.99 respectively. After solving fundamental matrix, we decompose it into $[R; t]$ and further triangulate matches for all four $[R; t]$ solutions. We choose the one which has the most triangulated points in front of both cameras as final relative pose. An inlier score map M_r is generated from fundamental matrix to mask out non-rigid regions such as moving objects and bad matches. See examples in Figure 4. Specifically, we compute the distance from one pixel to its corresponding epipolar line, resulting in distance map D_{epi}. The inlier score map is computed as $M_r = (D_{epi} < 0.5) = (1.0 + D_{epi})$. Again we perform top score sampling and random sampling from $M_r * M_s * M_o$ to acquire 6k matches. We filter out the matches which have extremely small ray angles or have invalid reprojection. To be specific, given two camera rays $\mathcal{L}_1 = \{p = c_1 + {}_1n_1 \mid {}_1 \in \mathrm{R}\}$ and $\mathcal{L}_2 = \{p = c_2 + {}_2n_2 \mid {}_2 \in \mathrm{R}\}$, where c_i is the ray origin and n_i is the ray direction, we could have $v = c_2 + \langle c_1 - c_2; n_2 \rangle n_2 - c_1$. Then the cosine value of angle between v and n_1 is computed. We filter out the regions where the cosine value is smaller than 0.001. See an example in Figure 3. After filtering, matches are further triangulated to 3D structure, and then used for scale alignment and supervision of depth prediction.

F. Additional Comparison on sampled KITTI Odometry dataset

To better demonstrate the robustness of our system, we provide additional comparison on sampled KITTI Odometry dataset. The test sequences 09 and 10 are sampled

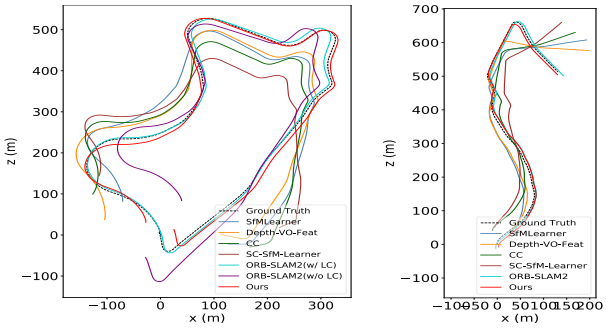

Figure 5. Visual odometry results on sampled sequence 09 and 10 with stride 2.

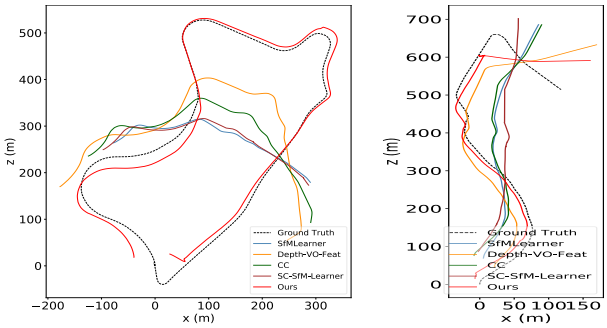

Figure 6. Visual odometry results on sampled sequence 09 and 10 with stride 4.

Sequences	fr3/cabinet	fr2/desk	fr3/str_ntex_far	fr3/str_tex_far
PoseNet	1.45	1.51	0.32	0.38
ORB-SLAM2 [6]	X	0.006	X	0.009
Ours	1.09	0.52	0.24	0.14

Table 3. Results for selected sequences on TUM-RGBD dataset. We report the absolute translational RMSE in meter.

with stride 2 and 4, and we run the PoseNet-based learning systems and ORB-SLAM2 on these sampled sequences without additional training. Table 1 and 2 summarize the results of sampling with stride 2 and 4 respectively. Trajectories results are shown in Figure 5 and 6. Again our system shows improved robustness and generalization ability compared to our baselines. However, when the camera moves extremely fast, such as sampling with stride 4 or more, the optical flow estimation becomes bottleneck and the performance degrades due to inaccurate correspondences.

G. Numerical Results of TUM-RGBD dataset

In Table 3, we report the quantitative results of TUM-RGBD dataset. Our methods could produce reasonable trajectories under challenging scenarios while PoseNet baseline fails to generalize. ORB-SLAM2 relies on sparse ORB features to establish correspondences, and it suffers on large textureless regions (fr3/cabinet, fr3/str_ntex_far). However, ORB-SLAM2 works much better than ours when the scene contains rich textures (fr2/desk, fr3/str_tex_far). Our system could be further improved with better optical flow estima-

tion and combination with back-end optimization. TUM-RGBD and NYUv2 are both indoor datasets and share some similar data distributions. We trained our method and PoseNet on TUM-RGBD dataset and directly tested on the NYUv2 dataset to demonstrate the transfer ability of trained model. Experimental results show that our model achieves better transfer performance (AbsRel 0.276) than PoseNet baseline (AbsRel 0.324). However, this transfer ability is still limited and has large room for improvement in the future.

H. Additional Visualizations

We provide more qualitative results on KITTI and NYUv2 dataset in Figure 7 and Figure 8.

References

[1] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, pages 35–45, 2019. 2, 4

[2] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *CVPR*, pages 3828–3838, 2019. 3

[3] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 270–279, 2017. 3

[4] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, pages 2938–2946, 2015. 2

[5] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016. 3

[6] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 4

[7] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, pages 12240–12249, 2019. 4

[8] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 3

[9] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, pages 2022–2030, 2018. 3

[10] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, pages 4884–4893, 2018. 3

[11] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 3

[12] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018. 3

[13] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*, pages 340–349, 2018. 4

[14] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Moving indoor: Unsupervised video depth learning in challenging environments. In *ICCV*, pages 8618–8627, 2019. 3

[15] 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. 1, 2, 4

[16] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, pages 36–53, 2018. 3

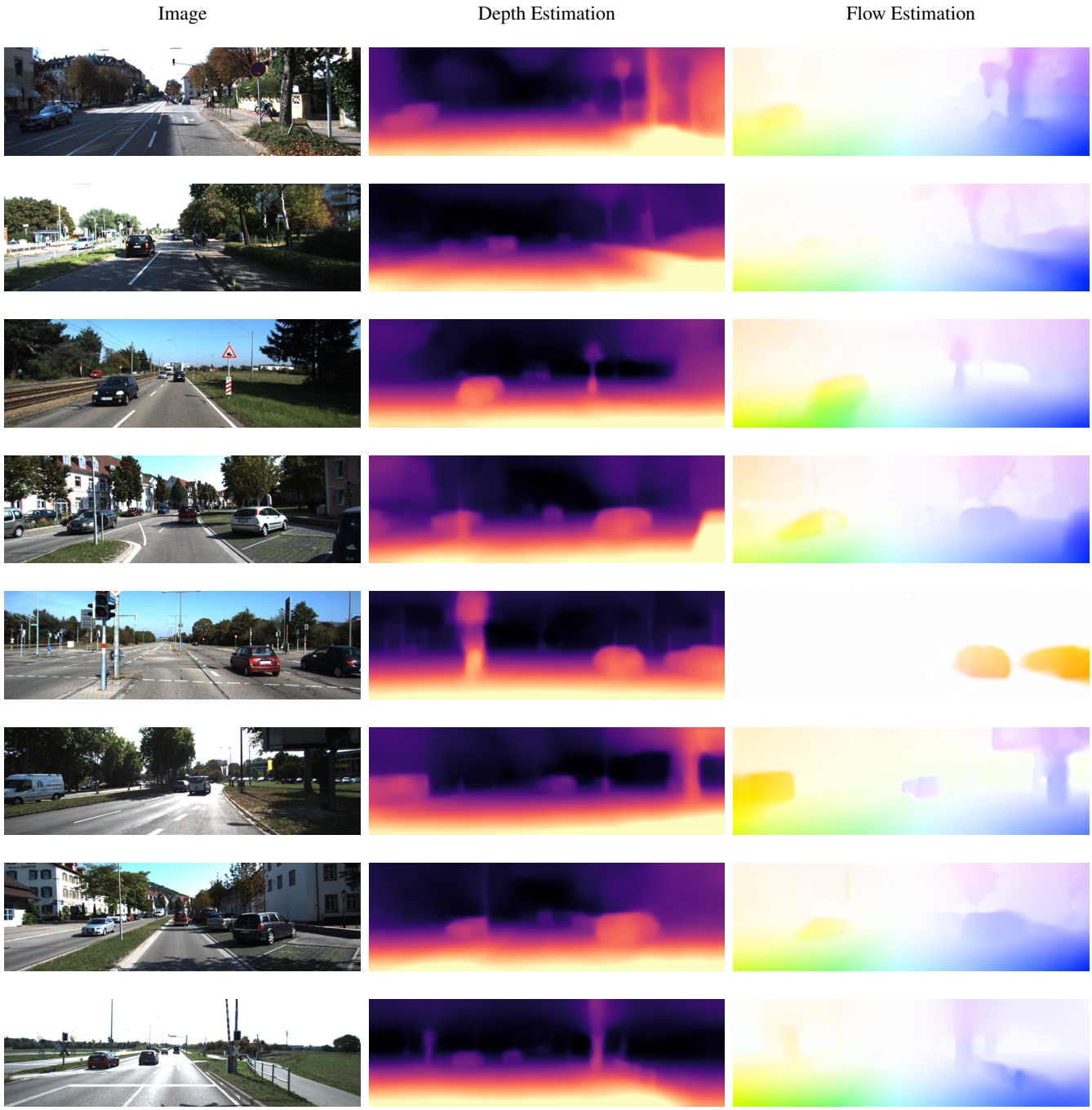

Figure 7. Visualization for KITTI depth and flow estimation.