Supplementary material for 3DV 2018 paper "Learning monocular depth estimation under unsupervised trinocular assumption"

Matteo Poggi, Fabio Tosi, Stefano Mattoccia
University of Bologna, Department of Computer Science and Engineering
Viale del Risorgimento 2, Bologna, Italy
{m.poggi, fabio.tosi5, stefano.mattoccia}@unibo.it

This document provides additional details and experimental results concerned with 3DV 2018 paper "Learning monocular depth estimation under unsupervised trinocular assumption". The supplementary material is organized as follows: Section 1 reports detailed explanation of the loss functions used at training time, Section 2 describes how we obtain d^c with 3Net and how we post-process it, Section 3 comments additional experiments on the Eigen split [1] assuming as maximum depth 50 meters and Section 4 collects additional qualitative results, Finally, Section 5 reports run time analysis for 3Net and [3].

1. Training losses

In the paper, all loss functions are computed at four scales, ranging from full image resolution to $\frac{1}{8}$. The global loss function is defined as:

$$\mathcal{L}_{total} = \beta_{ap}(\mathcal{L}_{ap}) + \beta_{ds}(\mathcal{L}_{ds}) + \beta_{lcr}(\mathcal{L}_{lcr}) \quad (1)$$

where $\mathcal{L}_{ap}$, $\mathcal{L}_{ds}$ and $\mathcal{L}_{lcr}$ represent, respectively, the appearance, smoothness and consistency terms, while β_{ap}, β_{ds} and β_{lcr} are hyper-parameters. In particular, we set $\beta_{ap} = \beta_{lcr} = 1$ and $\beta_{ds} = 0.1$.

Appearance Loss. It measures the reconstruction error between a warped image and the original one. It is obtained by a weighted sum of a SSIM based score [7] and a L1 distance over pixel intensities.

$$\mathcal{L}_{ap}(I^l, I^r) = \frac{1}{N}\sum_{ij}\alpha\frac{1 - SSIM(I^l_{ij}, \tilde{I^r_{ij}})}{2} + (1-\alpha)||I^l_{ij} - \tilde{I^r_{ij}}|| \quad (2)$$

Smoothness Loss. This term favours the propagation of similar disparity values in low-textured regions, thus enforcing smoothness. It is obtained computing horizontal and vertical gradients on both disparity image and reference image, discouraging disparity smoothness in presence of strong image gradients.

$$\mathcal{L}_{ds}(d, I) = \frac{1}{N}\sum_{ij}|\partial_x d^l_{ij}|e^{-||\partial_x I^l_{ij}||} + |\partial_y d^l_{ij}|e^{-||\partial_y I^l_{ij}||} \quad (3)$$

Left-Right Disparity Consistency Loss. It enforces consistency between reference-to-target and target-to-reference disparity maps. It relies on the L1 distance between reference-to-target map and warped, according to the former, target-to-reference map.

$$\mathcal{L}_{lr}(d^l, d^r) = \frac{1}{N}\sum_{ij}|d^l_{ij} - d^r_{ij+d^l_{ij}}| \quad (4)$$

2. Depth computation and post-processing

For the sake of clarity, we describe in detail how we combine d^{cl} and d^{cr} to obtain the final output map d^c. In [3] the authors obtained d^l and $\hat{d}^l$ by processing, respectively, both I and its horizontally flipped version $\hat{I}$. The two maps were combined as follows:

$$d_{pp} = \omega \cdot d^l + (1 - \omega) \cdot \hat{d}^l \quad (5)$$

with ω obtained as:

$$\omega = \begin{cases} 0 & \text{if} j \leq 0.05 \\ 1 & \text{if} j > 0.95 \\ 0.5 & \text{otherwise} \end{cases} \quad (6)$$

being i, j normalized pixel coordinates.

Following this principle, we combine our d^{cl} and d^{cr} maps as follows:

$$d^c = \omega \cdot d^{cr} + (1 - \omega) \cdot d^{cl} \quad (7)$$

			Proposed method		Lower is better		Higher is better		
Method	Supervision	Train set	Abs Rel	Sq Rel	RMSE	RMSE log	δ <1.25	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al. [9]	Temporal	E	0.201	1.391	5.181	0.264	0.696	0.900	0.966
Mahjourian et al. [5]	Temporal	E	0.155	0.927	4.549	0.231	0.781	0.931	0.975
Zhan et al. [8]	Stereo+Temp.	E	0.135	0.905	4.366	0.225	0.818	0.937	0.973
Godard et al. [3] ResNet50 + pp	Stereo	E	0.1217	0.7630	4.047	0.210	0.847	0.946	0.976
3Net ResNet50 + pp **(ours)**	Stereo	E	**0.1207**	**0.7185**	**3.968**	**0.208**	**0.849**	**0.948**	**0.977**
Zhou et al. [9]	Temporal	CS+E	0.190	1.436	4.975	0.258	0.735	0.915	0.968
Mahjourian et al. [5]	Temporal	CS+E	0.151	0.949	4.383	0.227	0.802	0.935	0.974
Poggi et al. [6] PyD-Net (200)	Stereo	CS+E	0.138	0.937	4.488	0.230	0.815	0.934	0.972
Godard et al. [3] ResNet50 + pp	Stereo	CS+E	0.108	0.657	3.729	0.194	0.873	0.954	**0.979**
3Net ResNet50 + pp **(ours)**	Stereo	CS+E	**0.091**	**0.572**	**3.459**	**0.183**	**0.889**	**0.955**	**0.979**

Table 1. Evaluation on the KITTI dataset [2] using the split of Eigen et al. [1], with maximum depth set to 50m. Results concerned with state-of-the-art techniques for unsupervised monocular depth estimation leveraging video sequences (Temporal), binocular stereo pairs (Stereo) and both cues (Stereo+Temp.).

Running two forwards, we can post-process both intermediate maps and

$$d^{c}_{pp} = \omega \cdot d^{cr}_{pp} + (1 - \omega) \cdot d^{cl}_{pp} \tag{8}$$

being d^{cr}_{pp} and d^{cl}_{pp} obtained as:

$$d^{cr}_{pp} = \omega \cdot d^{cr} + (1 - \omega) \cdot \hat{d^{cr}} \tag{9}$$

$$d^{cl}_{pp} = \omega \cdot \hat{d^{cl}} + (1 - \omega) \cdot d^{cl} \tag{10}$$

3. Depth estimation: additional experiments with 50m cap

We report additional experimental results on the Eigen split [1], evaluating depth maps up to a maximum distance of 50 meters as reported in some recent works [3, 8, 5, 9]. Table 1 contains a comparison between all previous works reporting this experiment as well and our best model, i.e. 3Net ResNet50 + pp. This further evaluation confirms, once again, the superiority of our technique with respect to all competitors.

4. View synthesis and multi-baseline stereo

Finally, deploying 3Net ResNet50 + pp trained on CS+E, we provide additional qualitative results for depth-from-mono estimation and view synthesis. Figure 1 and 2 reports six examples taken from the evaluation set of the Eigen split [1]. In particular, we show in the leftmost column the generated left view (a), the single input image fed to our network (b) and the generated right view (c). In the mid column, the three output maps of 3Net, respectively, d^{cl} (d), d^{c} (e) and d^{cr} (f). Finally, in the rightmost column, we report disparity maps obtained processing with SGM [4] the three stereo pairs obtainable with 3Net from the three views (one real, two synthetic) depicted in the leftmost column. In particular, the disparity maps computed by SGM are concerned with three stereo pairs: left-to-center (g), center-to-right (h) and left-to-right (i). It is worth to note that the left-to-right stereo pair (i) is made of two completely novel views synthesized by our network. The other two stereo pairs contain the input image and a a novel image synthesized by 3Net.

	256×512		384×1280	
	1×	2×	1×	2×
[3] ResNet50	0.57s	1.10s	1.98s	3.92s
3Net ResNet50	0.80s	1.55s	2.95s	5.87s

Table 2. Run time comparison between Godard et al. [3] and 3Net running single and double forward on a CPU Intel Core i7-7700K.

Observing (a), (b) and (c) we can easily notice three different view points: the two *virtual* cameras are located at the left and right side of the *real* camera (i.e., the central one). The three maps in the middle column clearly show artifacts occurring near depth discontinuities and occlusions in (d) and (f) and how they are greatly dampen in the final output of our network (e). Finally, we can perceive how (g) and (i) share the same reference image (synthetic left) and how they compute different disparity values according to different baselines, *narrow* and *wide*, made available by the three-view *virtual* rig enabled by 3Net.

A video showing the performance of 3Net on the KITTI sequence *2011_10_03_drive_0047_sync* [2] not part of the Eigen split imagery used for training is available at this link: https://www.youtube.com/watch?v=uMA5YWJME4M.
Finally, the source code is available at this link: https://github.com/mattpoggi/3net

5. Runtime analysis

In this section, we briefly compare the runtime of 3Net compared to the models by Godard et al. [3]. On high-end GPUs (e.g., Titan X Pascal), the difference between the two models either running single or double forward is negligible, taking between 0.09 and 0.11 seconds both. Nevertheless, in case of applications deploying different architectures the margin rises.

In particular, Table 2 compares the execution times of the

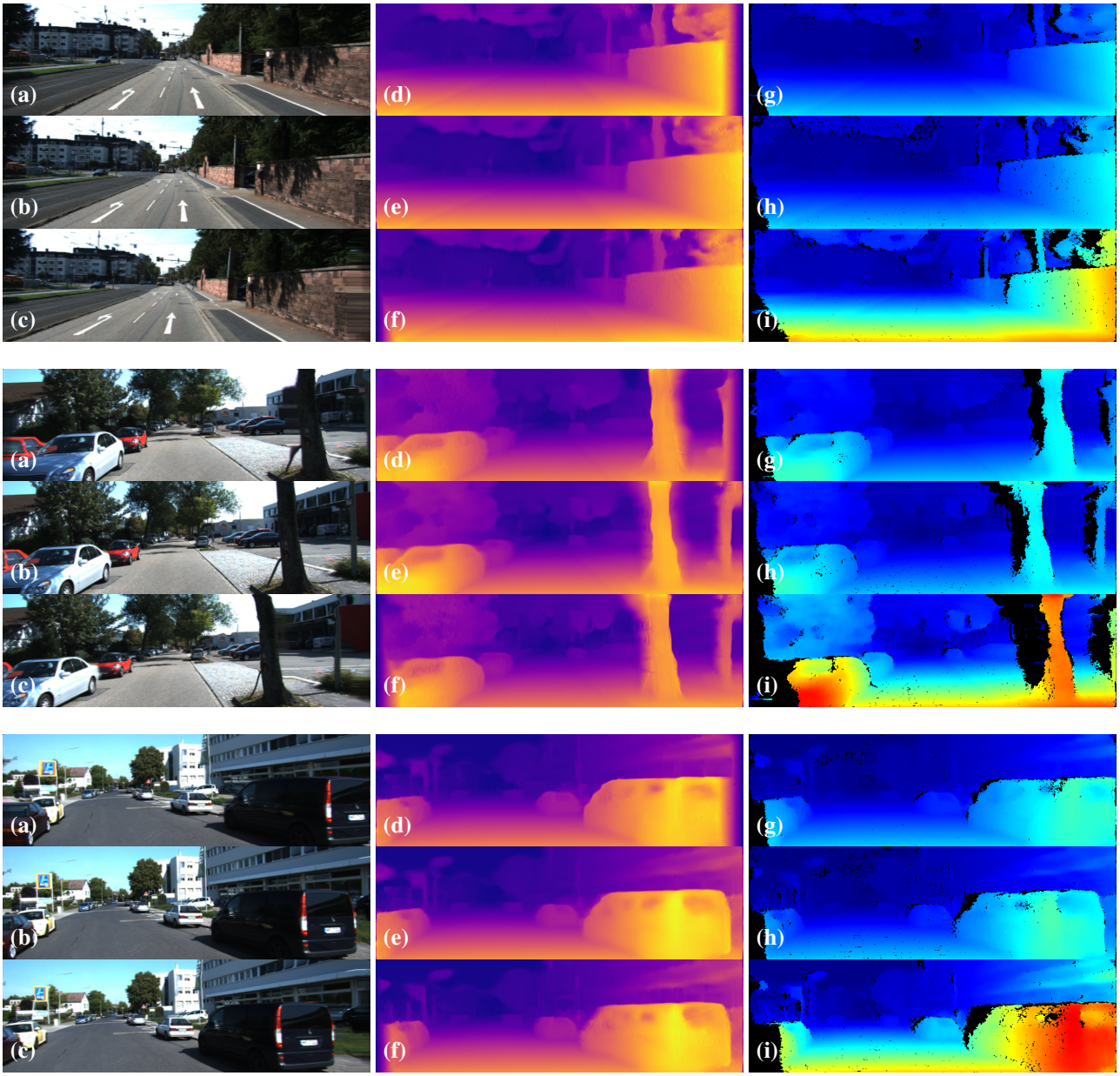

Figure 1. Qualitative evaluation of 3Net. In the leftmost column, we show (always from top to bottom) *synthetic* left (a), *real* central (b) and *synthetic* right (c) view. In the middle column, d^{cl} (d), d^{c} (e) and d^{cr} (f) depth maps computed by our network processing the input image. In the rightmost column, disparity maps obtained by the SGM algorithm [4] processing respectively, left-center (g), center-right (h) and left-right (i) stereo pair.

considered models using ResNet50 encoder on a CPU Intel Core i7-7700K. Times are averaged on the entire Eigen split testing set. We report numbers at 256×512 resolution (i.e., the dimensions used by [3] at inference time), as well as at full KITTI resolution, to stress how the difference between them increases with the image size. We can see how the second encoder in 3Net adds about 50% overhead, while $2\times$ forwards usually doubles it. However, by recalling results reported in the main paper (Table 2, last 3 rows on bottom), 3Net ResNet50 running a single forward is more accurate and faster than [3] ResNet50 running two forwards.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

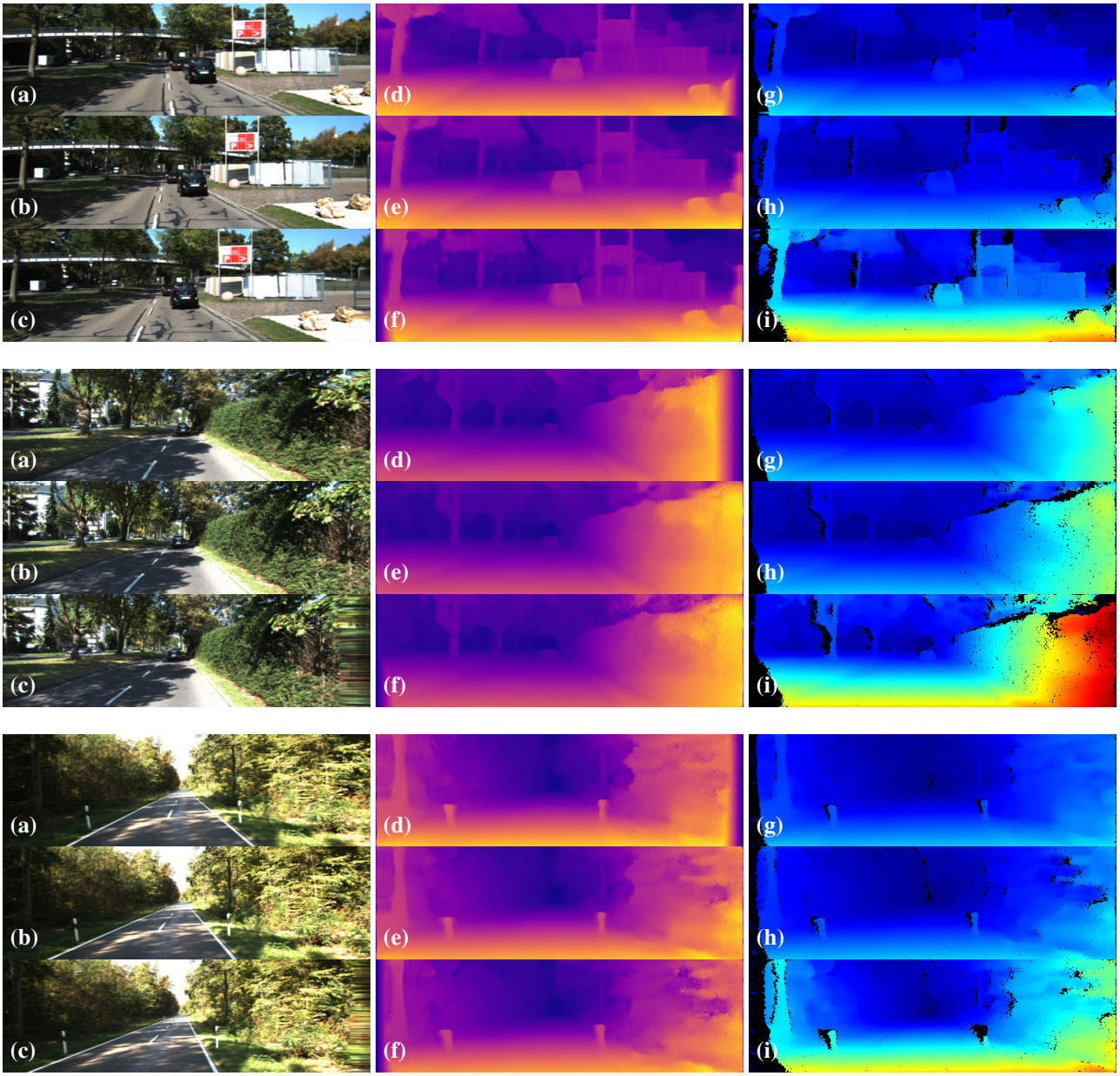

Figure 2. Qualitative evaluation of 3Net. In the leftmost column, we show (always from top to bottom) *synthetic* left (a), *real* central (b) and *synthetic* right (c) view. In the middle column, d^{cl} (d), d^{c} (e) and d^{cr} (f) depth maps computed by our network processing the input image. In the rightmost column, disparity maps obtained with SGM algorithm [4] processing respectively, left-center (g), center-right (h) and left-right (i) stereo pair.

References

[1] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 1, 2

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2

[3] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 1, 2, 3

[4] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005. 2, 3, 4

[5] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *The IEEE Conference on Com-*

puter Vision and Pattern Recognition (CVPR), 2018. 2

[6] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *IEEE/JRS Conference on Intelligent Robots and Systems (IROS)*, 2018. 2

[7] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, Apr. 2004. 1

[8] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[9] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 2