Semantically-Agnostic Unsupervised Monocular Depth Learning in Dynamic Scenes

Hanhan Li[1], Ariel Gordon[1,2], Hang Zhao[3], Vincent Casser[3], and Anelia Angelova[1,2]

[1] Google Research [2]Robotics at Google [3]Waymo LLC

Abstract. We present a method for jointly training the estimation of depth, egomotion, and a dense 3D translation field of objects, suitable for dynamic scenes containing multiple moving objects. Monocular photometric consistency is the sole source of supervision. We show that this apparently heavily-underdetermined problem can be regularized by imposing the following prior knowledge about 3D translation fields: They are sparse, since most of the scene is static, and they tend to be constant through rigid moving objects. We show that this regularization alone is sufficient to train monocular depth prediction models that exceed the accuracy achieved in prior work, including semantically-aware methods.

1 Introduction

Learning to predict depth from monocular video became a well-established technique in the past few years. Remarkably, the required supervision often amounts to only the video itself: Consecutive frames are different veiwpoints of the same scene, and the correct depth and egomotion thus are ones that allow correct reprojection of one frame onto the other.

Dynamic scenes violate the "same scene" assumption above, and result in failure cases in methods that rely on it. Various approaches to addressing this challenge have been proposed. Godard *et al.* propose a method to identify a

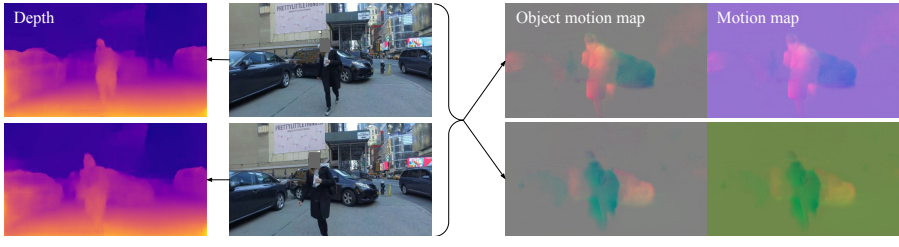

Fig. 1. Depth prediction (for each frame separately) and motion map prediction (for a pair of frames), shown on a training video from YouTube. The total 3D translation map is obtained by adding the learned camera motion vector to the object motion map. Note that the object motion map is mostly zero, and nearly constant throughout a moving object. This is a result of the motion regularizers used.

specific type of object motion, namely where the observing car follows another car at roughly the same speed, so that the followed car appears static in the observing car's camera frame [5]. While this is a very special case of object motion, it is also a very common one, and addressing it resulted in significant improvements in the depth prediction accuracy. However dynamic scenes can exhibit many other common forms of motion, such as pedestrians crossing the road in front of a car.

Other methods [2, 6] utilize semantic information provided by an auxiliary pre-trained segmentation or detection model, to identify moving objects. Having such a pre-trained model as a prerequisite hinders the usefulness of these approaches, and requires knowing in advance the nature of all moving objects that can appear in the scene.

In this work we overcome the limitations of the methods above, by utilizing prior knowledge that is less restrictive than in Ref. [5], and at the same time free of semantics. Given a dense map describing 3D translation of each pixel relative to the scene (henceforth *"translation field"*), we assume that:

- Most pixels belong to the background or to static objects, so their translation value must be zero.
- A moving (rigid) object manifests as a blob of pixels, all moving all the same velocity (and hence having the same translation).

We thus expect the translation field to be *sparse* and *piecewise constant*. A key contribution of this work is a regularization method that casts the translation field into the desired profile. As we show, our method achieves state of the art depth metrics, ***without requiring auxiliary semantic information***, and while allowing more general object motion patterns than objects that move at the same velocity as the camera.

2 Method

Given a pair adjacent video frames (I_a and I_b), a network predicts a depth map $D(u, v)$ at the original resolution from a each frame separately. The two depth maps are concatenated with I_a and I_b in the channel dimension and are fed into a motion prediction network. The latter predicts a 3D translation map $T_{obj}(u, v)$, where (u, v) are image-space coordinates, at the original resolution for the moving objects and a 6D motion vector $M_{ego} = [R_{ego}, T_{ego}]$ for the camera, where R_{ego} is a 3D rotation matrix and T_{ego} is a 3D translation vector. The object motion relative to the camera is defined as rotation R_{ego} followed by a translation $T(u, v) = T_{obj}(u, v) + T_{ego}$. We apply a number of regularization losses on the predictions.

Motion Regularization The regularization $L_{\text{reg,mot}}$ on the motion map $T_{obj}(u, v)$ consists of the ***group smoothness loss*** L_{g1} and the $L_{1/2}$ ***sparsity loss***:

$$L_{reg,mot} = L_{g1}[T(u, v)] + \lambda L_{1/2}[T(u, v)] \tag{1}$$

where λ is a balancing coefficient. L_{g1} minimizes changes in the moving areas, encouraging $T(u,v)$ to be nearly constant throughout a moving object:

$$L_{g1}[T(u,v)] = \sum_{i\in\{x,y,z\}} \iint \sqrt{\left(\partial_u T_i(u,v)\right)^2 + \left(\partial_v T_i(u,v)\right)^2}\,\mathrm{d}u\mathrm{d}v$$

The $L_{1/2}$ sparsity loss on $T(u,v)$ is defined as:

$$L_{1/2}[T(u,v)] = 2 \sum_{i\in\{x,y,z\}} \langle|T_i|\rangle \iint \sqrt{|T_i(u,v)|\,/\langle|T_i|\rangle + 1}\,\mathrm{d}u\mathrm{d}v \qquad (2)$$

where $\langle|T_i|\rangle$ is the average of $|T_i(u,v)|$ over the (u,v) space. The coefficients are designed in this way so that the regularization is self-normalizing. In addition, it asymptotes to L_1 for small $T(u,v)$, and its strength becomes weaker for larger $T(u,v)$. We visualize its behavior in the appendix. Overall, the $L_{1/2}$ regularization is better at inducing sparsity than the L_1 regularization.

3 Experiments

We evaluated our method on three datasets: Cityscapes [3], Waymo Open Dataset [8], and KITTI [4]. Cityscapes has many dynamic scenes, with multiple vehicles and pedestrians moving through them. Indeed, our method was able to improve prior benchmarks on Cityscapes, even methods that use semantic queues from auxiliary models (by most metrics; see Table 1). Lastly, we trained the model on YouTube videos showing street footage from a camera held by a walking person. Qualitative results of the latter are shown in Fig. 1 and in the Supplementary Material.

Method	Uses semantics?	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Struct2Depth [2]	Yes	0.145	1.737	7.28	0.205	0.813	0.942	0.978
Gordon [6]	Yes	0.128	**0.959**	**5.23**	0.212	0.845	0.947	0.976
Pilzer [7]	**No**	0.440	6.04	5.44	0.398	0.730	0.887	0.944
Ours	**No**	**0.119**	1.29	6.98	**0.190**	**0.846**	**0.952**	**0.982**

Table 1. Depth estimation test error for models trained and evaluated on Cityscapes using the standard split. The depth cutoff is always 80m. Our model uses a resolution of 416×128 for input/output. For the red metrics, lower is better; for the green metrics, higher is better. The evaluation uses the code and methodology from Struct2Depth [2].

The Waymo Open Dataset is currently one of the largest and most diverse publicly released autonomous driving datasets. Its scenes are not only dynamic but also comprise nighttime driving and diverse weather conditions. Unaware of previously published monocular depth estimation benchmarks on this dataset, we compare our method to benchmarks we obtained from running public code of prior methods. As Table 2 shows, our method outperforms the latter.

KITTI is the most popular benchmark, albeit being poor in dynamic scenes, which makes it a less than ideal test bed for out method. As Table 3 shows, our method is on par with state of the art methods.

Method	Uses samantics?	Abs Rel	Sq Rel	RMSE	RMSE log
Open-source code from [2]	Yes	0.180	1.782	8.583	0.244
Open-source code from [6]	Yes	0.168	1.738	7.947	0.230
Ours	No	**0.162**	**1.711**	**7.833**	**0.223**

Table 2. Performance on the Waymo Open Dataset. Even though our approach doesn't require masks, it outperforms prior work.

Method	Uses samantics?	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Struct2Depth [2]	Yes	0.141	1.026	5.291	0.2153	0.8160	0.9452	0.9791
Gordon [6]	Yes	**0.127**	1.33	6.96	**0.195**	**0.830**	**0.947**	0.981
Bian [1]	**No**	0.137	1.089	5.439	0.217	0.830	0.942	0.975
Godard [5]	**No**	**0.128**	1.087	5.171	0.204	0.855	0.953	**0.978**
Ours	**No**	0.130	**0.950**	**5.138**	0.209	0.843	0.948	**0.978**

Table 3. Depth estimation test error, for models trained and evaluated on KITTI using the Eigen Split. The depth cutoff is always 80m. Our model uses a resolution of 416×128 for input/output.

References

1. Bian, J.W., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.M., Reid, I.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. arXiv preprint arXiv:1908.10553 (2019)
2. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8001–8008 (2019)
3. 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)
4. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
5. Godard, C., Aodha, O.M., Firman, M., Brostow, G.: Digging into self-supervised monocular depth estimation. ICCV (2019)
6. Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
7. Pilzer, A., Xu, D., Puscas, M.M., Ricci, E., Sebe, N.: Unsupervised adversarial depth estimation using cycled generative networks. 3DV (2018)
8. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset (2019)

Supplemental Material

In Figures 3 and 4, we present depth and 3D motion learned with our unsupervised approach on a collection of YouTube videos and scenes from the Cityscapes dataset. In many of them, there are vehicles and people moving around. The collection of YouTube videos were recorded with hand-held monocular cameras by people walking around in diverse environments, and the camera intrinsics were unknown to us. Figure 5 provides additional visualizations of the learned 3D motion maps on the Waymo Open Dataset. Figure 6 contains an additional visualization of our motion module.

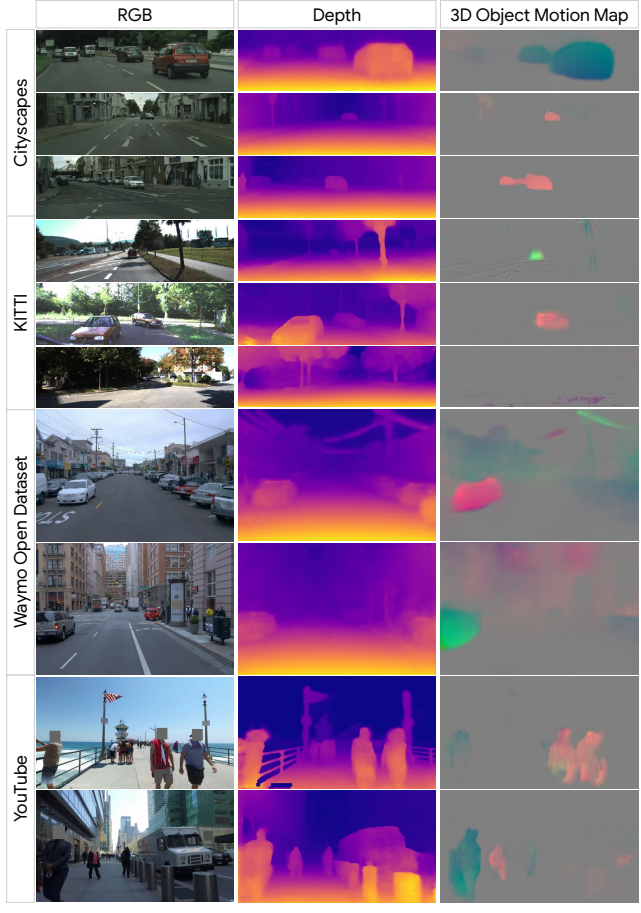

Fig. 2. Qualitative results of our unsupervised monocular depth and 3D object motion map learning in dynamic scenes across all datasets: Cityscapes, KITTI, Waymo Open Dataset and YouTube.

Fig. 3. This figure shows the learned object motion maps (right column) and disparity maps (middle column) for RGB frames (left column) in a collection of YouTube videos taken with moving cameras. The last two examples show static scenes, where the object motion maps are mostly near zero.

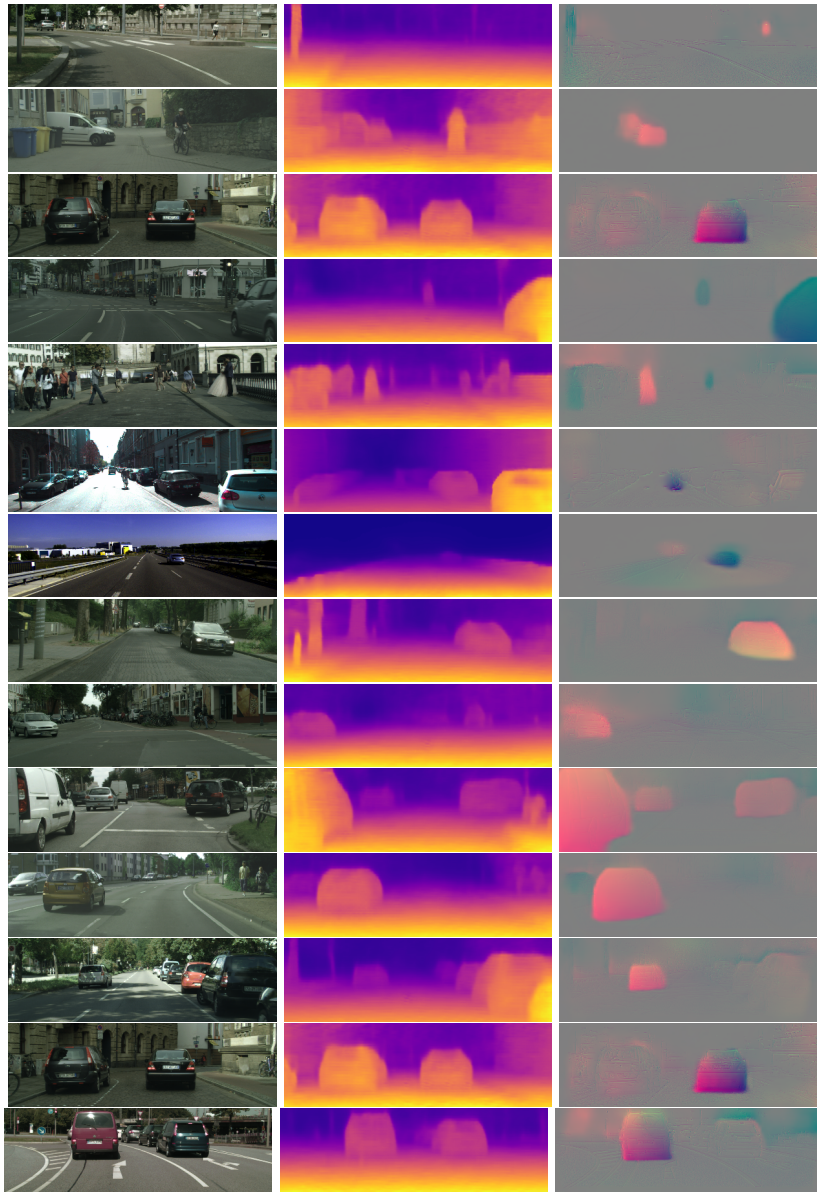

Fig. 4. This figure shows the learned object motion maps (right column) and disparity maps (middle column) for RGB frames (left column) in the Cityscapes dataset.

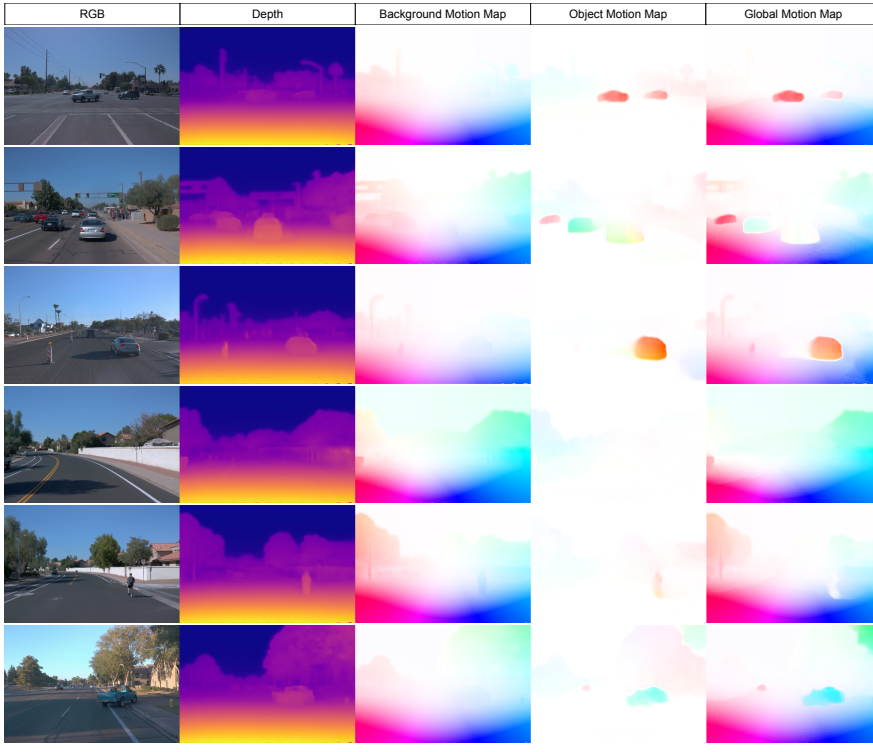

Fig. 5. This figure shows RGB images, disparity maps, and the 2D-appearance flows projected from 3D motion maps on the Waymo Open Dataset. Here, we colorize based on flow direction with intensity corresponding to flow magnitude. Using our depth and background motion estimate, we can derive 2D appearance flow of static parts of the scene (middle). We can use the same procedure to visualize our object motion field and their global composite (right), respectively.

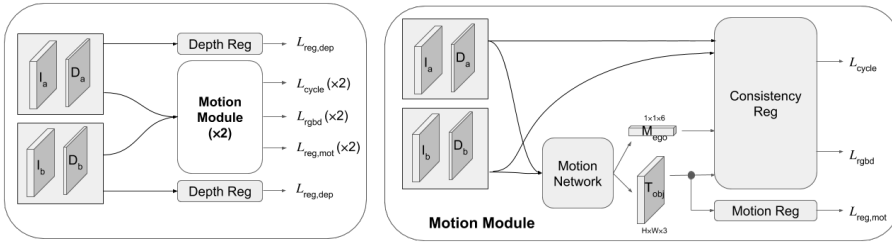

Fig. 6. Overall training architecture. A depth network is independently applied on two adjacent frame images, I_a and I_b, to produce the depth maps, D_a and D_b. The depth maps together with the two original images are fed into the motion module, whose details are shown on the right hand side. This module is applied twice, reverting the places of the first and second images, i.e., the input image I_a is switched with I_b, and the input depth D_a is switched with D_b. A composite of regularization losses is imposed on the network predictions. At inference time, only the depth network is used. Given two input images, the 3D motion map can also be obtained at inference.