DN Learner: Unsupervised Learning of Depth and Surface Normal from Video

Aman Raj
A53247556
amraj@eng.ucsd.edu

Menghe Zhang
A53208258
mez071@eng.ucsd.edu

Sarah Wang
A53276846
sawang@eng.ucsd.edu

Tingwei Yu
A53281022
t3yu@eng.ucsd.edu

Abstract

Learning to reconstruct depths and camera pose from video sequences in an unsupervised approach is attracting significant attention since the work of Zhou et al.[19]. Our work extends their work by jointly estimating depth, camera pose and surface normal in an end-to-end framework. We verify that deep learning networks can benefit from explicit geometric cues and constraints provided by instance and semantic-level segmentations. In addition, we enforce the consistency between depth and normal to yield more robust geometrical predictions. Evaluations are performed on the KITTI dataset[8] and ablation studies regarding different geometric constraints are also conducted to compare their effectiveness.

1. Introduction

Inferring camera pose, depth and surface normal of a scene at a detailed level is crucial for 3D scene understanding. The techniques to recover these information from monocular video sequences can be widely applied in various real-world applications. For instance, in robot navigation, this enables the robot to avoid obstacles and travel through unseen areas. Such ability also facilitates 3D reconstruction of the environment and can be applied in the field of augmented reality.

Recently, there has been large progress in unsupervised pose and depth estimation using monocular cameras. Works such as [19] has significantly reduced human effort to obtain large quantities of ground truth color-depth image pairs for training, while also yielding comparable results to supervised approaches. The key concept is to use view synthesis as supervision by warping a frame in a sequence to the view of its consecutive frames. The work of Yang *et al.*[16] refines the estimated depth by leveraging the constraint between surface normal and depth and image gradients to represent edges where depth discontinuities can occur. Casser *et al.*[1] takes additional instance-level segmentation along with RGB images as input and predicts poses for the static background and each dynamic object separately.

In our work, we study the effects of posing different geometric constraints and additional input cues in the form of semantic segmentation to improve geometry predictions. First, we investigate if edge-awareness leads to better depth predictions. Second, we show that by augmenting the input of network with semantic information generates sharper depth results as it provides notion of object boundaries and geometrical consistencies. Lastly, we explore two different methodologies to learn normal. We examine the effect of a weaker depth-normal constraint from [16] which forces the normal predictions to be perpendicular to the tangent object plane. Furthermore, we examine patch-based photometric consistency constraint motivated from [7] that forces the normal prediction within a patch to be consistent. Evaluation on the KITTI dataset [8] demonstrates the effectiveness of different approaches.

2. Related work

Warping-based view synthesis. View synthesis aims to create novel views of a specific subject from images from different view points. Classic paradigm explicitly reconstruct the accurate 3D model of the scene and composite the novel view from the input images [12][5][20]. An alternative approach is to synthesize images without explicitly estimating the 3D geometry of the scene. For instance, Mahajan *et al.* [14] proposed to move the gradients in the input images along a specific path to reconstruct a novel view. Shechtman *et al.* [14] proposed a patch-based optimization framework to reconstruct new views. The end-to-end learning based framework DeepStereo [6] uses two towers to predict depth and color and fuses them together to reconstruct the scene.

Image segmentation. Semantic-level segmentation is a fundamental task in computer vision, where a semantic label is assigned to every pixel. In recent years, deep convolutional neural networks that rely on hand-crafted features showed remarkable improvements on different segmentation benchmark tasks. Among these, the state-of-the-art work DeepLabv3+ [2] uses an atrous spatial pyramid pooling decoder module which refines

the segmentation results along object boundaries. In our work, we use DeepLabV3+ model trained on Cityscapes [3] to generate semantics for KITTI. For instance level segmentation we use state-of-the-art Mask-RCNN model [9] trained on COCO dataset [10]. We argue that with these additional geometric cues, the network could generate more promising depth predictions with finer details.

3. Method

The approaches we mention here are used for jointly estimating depths, surface normal and ego-motion. In particular our experiments are focused on improving depth and normal. The core idea is to perform inverse warping from the target view to source view with awareness of the underlying 3D geometry of the scene.

3.1. Algorithm Baseline

The baseline for our algorithms come from Zhou *et al.* [19] having separate models for depth and pose predictions.
DepthNet The architecture of DepthNet is taken from [13] which uses an encoder followed by a decoder with skip connections and multi-scale side outputs. Other than the final output layer, which has a sigmoid function applied to enforce the predictions in an reasonable range, the other convolution layers are are followed by ReLU activations.
PoseNet We adopted the PoseNet architecture proposed in [19]. The input of the network is the target-source frame pairs, and the output is the 6D camera pose from each target frame to the source frame. All convolution layers are followed by ReLU activations except for the final output layer, where no non-linear activation is applied.
View synthesis as supervision Following the notation used in [19], given a pair of images consisting of a target frame I_t and a source frame I_s, we can estimate the depth map $\hat{D}_t$ of the target view and the 6-DOF transformation $\hat{T}_{t\to s}$ from I_t to I_s. Thus, for any pixel p_t in the target frame I_t, its corresponding pixel coordinate in the source frame p_s can be found through perspective projection.

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t \tag{1}$$

With this correspondence, a synthesized target view $\hat{I}_s$ can be generated from I_s through bilinear interpolation. By using the target frame as our reference, the photometric loss can be formulated as:

$$L_{vs} = \sum_s \sum_p |I_t(p) - \hat{I}_s(p)| \tag{2}$$

by summing over all pixels p and source images s. $\hat{I}_s$ is the source view I_s warped to the target coordinate frame. Now we will explain experiments in subsequent subsections.

3.2. Edge-aware depth estimation

In this section, we illustrate two ways to parameterize geometric edges. First we compute the gradient map of target image I_t and synthesized target image $\hat{I}_s$, and require the two gradient images to match as described in [17].

$$L_g = \sum_{s=1}^{S} \sum_{x_t} M_s(x_t)|| \bigtriangledown I_t(x_t) - \bigtriangledown \hat{I}_s(x_t)||_1 \tag{3}$$

Here, $M_s(x_t)$ is the explainablitiy mask as in [19]. Since image gradient is robust under lighting condition it consistently encodes the discontinuities in predicted depth. Another way to parameterize edges is to jointly learn the edge map E_t for the target image from semantic mask which is done by directly predicting E_t using a decoder network similar to DepthNet.
Regularization of depth In order to produce smooth depth results, a smooth loss is added for regularization. This loss encourages the estimated depth to be locally similar when no significant image gradient exists, that is

$$L_s = \sum_{pt} \sum_{d\in x,y} || \bigtriangledown_d^2 D_t(p_t)||e^{-\alpha|\bigtriangledown_d I(p_t)|} \tag{4}$$

We use the weight $\lambda_s = 0.5 = l$ for this loss term along with common training setup as outlined in Section 4.2.
Photometric pixel loss In addition to the reconstruction loss and smooth loss, Structural Similarity(SSIM)[15] loss is applied to be part of pixel loss, which is basically window-based pixel comparing, specificly,

$$L_{pixel} = \alpha_{pixel}\frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{5}$$

where μ_x and μ_y are averages of $xandy$,while $\sigma_x and \sigma_y$ is the variance.σ_{xy} is the covariance. L is the dynamic range of the pixel-values. $c_1 = (k_1L)^2$,$c_2 = (k_2L)^2$ to stabilize the division with week denominator. We set $k_1 = 0.01 and k_2 = 0.03$. Our best result from these experiments come from using loss terms of Equation 4 and 5 as shown in Table 1.

3.3. Depth estimation using object motion model with semantic priors

Adapting the work of [1] which explicitly models the 3D motion of dynamic objects along with ego-motion of camera, we extend this work further by integrating semantic information of scene to improve depth prediction. Intuitively, semantic information encodes spatial constraints which can explain depth and normal discontinuities. Our DepthNet takes as input RGB sequence along with corresponding semantic segmentation(pixel-wise), encoded in one-hot encoding format. The ego-motion network PoseNet ϕ_E is

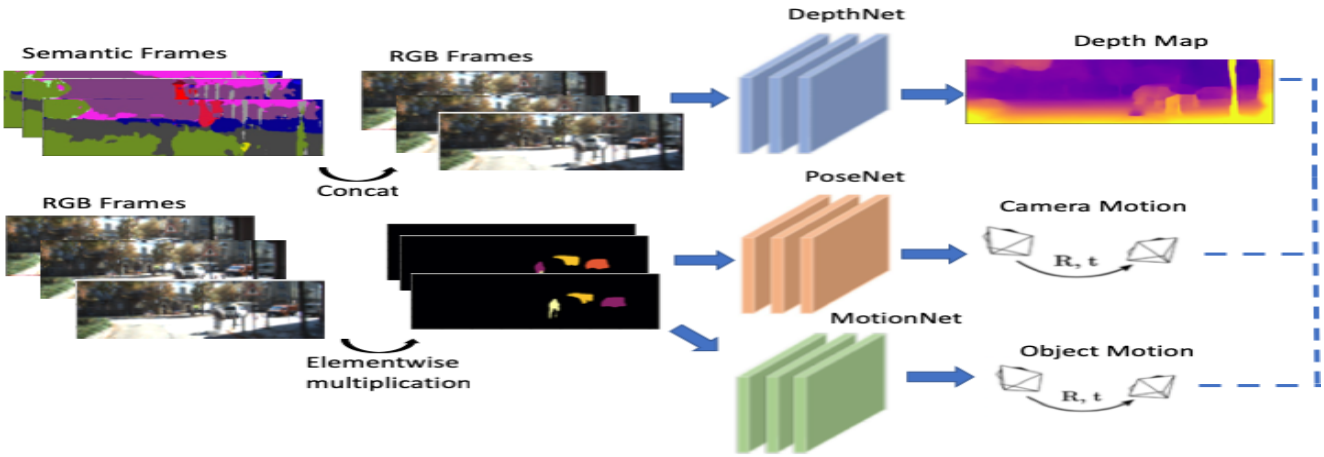

Figure 1: Our Unsupervised architecture for Subsection 3.3 which contains DepthNet, PoseNet and MotionNet

used to predict ego-motion of static background and object motion model ϕ_M is used to predict motion of individual dynamic object. ϕ_E and ϕ_M share the same architecture but different inputs.

We define instance-aligned segmentation masks (S_1, S_2, S_3) for corresponding sequence (I_1, I_2, I_3). In order to compute ego-motion of static scene only, we define static scene binary mask (O_1, O_2, O_3) where O_i is binary complement of S_i. The static binary mask is applied to all images in sequence by element-wise multiplication to mask out dynamic objects before feeding the sequence to ego-motion model:

$$V = O_1 \odot O_2 \odot O_3$$
$$E_{1\to2}, E_{2\to3} = \phi_E(I_1 \odot V, I_2 \odot V, I_3 \odot V) \quad (6)$$

To compute object motion model, we first apply ego-motion estimates to get warped RGB sequences $(\hat{I}_{1\to2}, I_2, \hat{I}_{3\to2})$ and instance-aligned segmentation sequences $(\hat{S}_{1\to2}, S_2, \hat{S}_{3\to2})$. Now, for every object instance in image, the object motion estimate $M^{(i)}$ of the *i*-th object is computed as:

$$M^{(i)}_{1\to2}, M^{(i)}_{2\to3} = \phi_M(\hat{I}_{1\to2} \odot \psi_i(\hat{S}_{1\to2}),$$
$$I_2 \odot \psi_i(S_2), \hat{I}_{3\to2} \odot \psi_i(\hat{S}_{3\to2})) \quad (7)$$

where $M^{(i)}_{1\to2}, M^{(i)}_{1\to2} \in \mathbb{R}^6$ and $\psi_i(S_1)$ returns binary mask only for object *i* in S_1. Now, corresponding to these motion estimates, inverse warping is done to move the objects according to predicted motions. This is done by first warping the full image using motion estimate and then masking out the corresponding object. For example, $\hat{I}^{(i)}_{1\to2}$ is obtained by using motion estimate $M^{(i)}_{1\to2}$. Final warped image $\hat{I}^{(F)}_{1\to2}$ is a combination of warped static background and individual warping of each dynamic object given by:

$$\hat{I}^{(F)}_{1\to2} = \hat{I}_{1\to2} \odot V + \sum_{i=1}^{N} \hat{I}^{(i)}_{1\to2} \odot \psi_i(S_2) \quad (8)$$

Equivalent for $\hat{I}^{(F)}_{3\to2}$ can be found using above Equation 8. We also impose object size constraint as in [1], to make sure the network is learning reasonable depth matching object motion estimates. The architecture overview of this method can be found in Figure 1. We obtained our best depth prediction result using this approach which is mentioned in Table 1. We also show some qualitative results of this method in Figure 6.

3.4. Depth-normal consistency constraints

For surface normal estimation, we build NormalNet by modifying the architecture of DepthNet such that it shares the same encoder of DepthNet while having a separate decoder to predict surface normal. While training, we jointly train the decoder branches of both the networks.

3.4.1 Depth and normal orthogonality constraint.

To train NormalNet, we applied the depth-normal orthogonality constraint proposed in [17] to our network. This could enforce the predicted surface normal to be perpendicular to its tangent plane. For each pixel x_i in the target view, we compute sum of the absolute dot products between the estimated normal and the vectors pointing from the pixel to it's 8 neighbors as shown in Figure 2. The loss term can be given by:

$$L_{orth} = \sum_{i\in t} \sum_{j\in Nei(i)} ||[\phi(x_j) - \phi(x_i)]\hat{N}_t(x_i)||_1$$
$$where\ \phi(x) = D_t(x)K^{-1}x \quad (9)$$

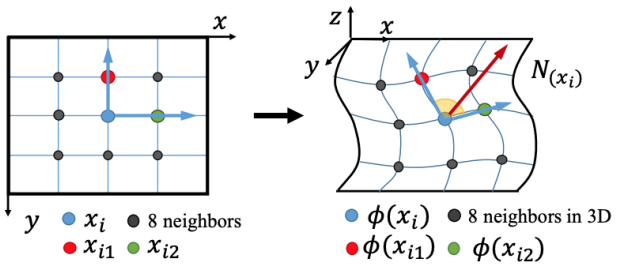

Figure 2: Depth and normal orthogonality constraint. To enforce the predicted surface to be perpendicular to the tangent surface, we compute the dot product between the estimated normal and the 8 vectors pointing from the center pixel to the neighboring pixels.

where $Nei(x_i)$ denotes the set of 8 neighboring pixels of x_i, K the camera intrinsic matrix and $\hat{N}(x_i)$ and $\hat{D}(x_i)$ the estimated surface normal and depth of the corresponding pixel, respectively. We define $\phi(x_i)$ as the back-projected 3D point of pixel x_i in the 3D space, thus $\phi(x_j) - \phi(x_i)$ is the vector pointing from the center pixel to the neighboring pixels in the world coordinates. Since we don't have direct supervision for surface normal, training with L_{orth} would result into the network always predicting zero for the elements in $\hat{N}$. Therefore, an extra regularization term is added in the the loss function to encourage the predicted surface normal to have a unit length. Our best result using this method comes with semantic as additional input. We show the results of the same in Table 1. Some qualitative result of this method is shown in Figure 5a.

3.4.2 Patch-based depth and normal constraint.

Through experiments we observed that by only enforcing the depth and normal orthogonality correlation could not provide our network enough geometry cues to reconstruct the surface normal(Figure 5a). Thus, we applied the patched-based depth and normal constraint to our network based on previous work of [7]. For each pixel in the target image, we reconstructed a $\mu \times \mu$ patch p in the 3D space with the center calculated by back-projecting the pixel (Figure 3). The orientation of the patch is given by the estimated surface normal and we assign one of its edges to be parallel to the x-axis of the target frame. Through re-projecting each patch back in to the target image I_t as well as the source image I_s, we sample the intensities of the patch in both views through bilinear interpolation to acquire I_{t,p_i} and I_{s,p_i}. The loss term L_{patch} is then the sum of $L1$ difference of the intensities between each pair of patches. Some qualitative result of this method is shown in Figure 5b.

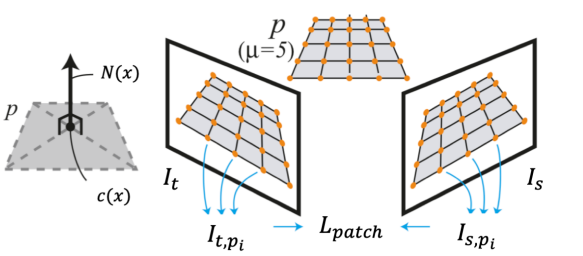

Figure 3: Patched-based depth and normal constraint. For each pixel, we reconstruct a patch in the 3D space. The center of the patch $c(x)$ is obtained by back-projecting the pixel with the estimated depth, intrinsic matrix and extrinsic matrix. The orientation of the patch is given by estimated normal $N(x)$.

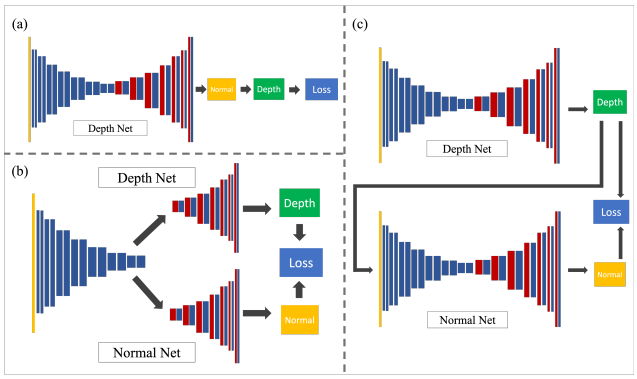

Figure 4: Network architectures for different experiments. (a) For the edge-aware experiments, the network architecture is based on (include related work), where we compute the normal directly from the predicted depth. (b)While evaluating the patch based photo-consistency constraints, the normal and depth net shares a single encoder and uses different decoders to separately predict depth. (c)To train the normal network with only the depth and normal orthogonality constraint, we froze the depth network and individually trained the normal network with an additional estimated depth input to get reasonable results.

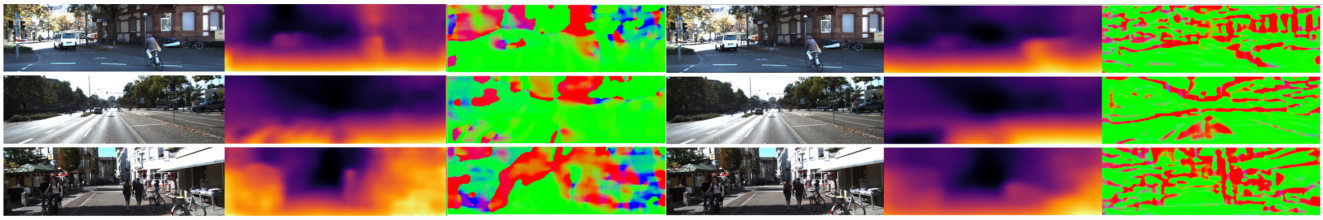

Figure 5: (a) Experiment results on KITTI with only the depth and normal constraint mentioned in Subsection 3.4.1 (b) Patch-based experiment results on KITTI mentioned in Subsection 3.4.2

Method	Supervised	Error-related metrics				Accuracy-related metrics		
		Abs Rel	Sq Rel	RSME	RSME log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al.* [4] Fine	Depth	0.203	1.548	6.307	0.282	0.702	0.890	0.957
Liu *et al.* [11]	Depth	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou *et al.* [19](updated)	No	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Yang *et al.* [16]	No	0.162	1.352	6.276	0.252	0.783	0.921	0.969
Yin *et al.* [18]	No	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Casser *et al.* [1](C)	No	0.146	1.147	5.386	0.218	0.810	0.942	0.978
Casser *et al.* [1](R)	No	**0.108**	**0.825**	**4.750**	**0.186**	**0.873**	**0.957**	**0.982**
Ours(sub section 3.4.1)	No	0.173	2.095	6.297	0.257	0.777	0.923	0.923
Ours(sub section 3.2)	No	0.206	2.860	6.969	0.278	0.734	0.901	0.956
Ours(sub section 3.3)	No	**0.140**	**1.059**	**5.374**	**0.215**	**0.816**	**0.942**	**0.978**

Table 1: Monocular depth results on the KITTI 2015 [8] by the split of Eigen *et al.* [4]. Result of Casser *et al.* (C) is obtained using their provided checkpoint. Casser *et al.* (R) is taken from their paper that has additional refinement during test time. Ours(Subsection 3.3) based on work of [1] doesn't use refinement but beats result of provided checkpoint.

4. Experiments

4.1. Datasets and metrics

We conduct experiments on depth and normal estimation. The performances are evaluated on the KITTI dataset [8]. For depth evaluation, we adopt metrics used in Zhou *et al.* [19], details are shown in Table 2.

4.2. Training

Similar to [19], we implemented the system using TensorFlow framework along with photo-consistency loss L_{vs}. For all the experiments, the common setup is: Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate of 0.0002 and mini-batch size of 4. All the experiments are performed with image sequences of size 3. We resize the images to 128×416 during training. We use the eigen split of KITTI for training and evaluation. Additional experimental setups for different methods are outlined below.

Edge-aware depth experiments. Architecture for approach outlined in Subsection 3.2 is shown in Figure 4a. We obtain the surface normal computed using the estimated depth and calculate the loss as mentioned in [17]. The best result that we got was with regularization loss L_s set to 0.5 and L_{pixel} set to 0.3 with RGB as input. The result of same is shown in Table 1.

Evaluation on the depth and normal orthogonality constraint. We set the weight of L_{orth} to 0.001. Yet, the results aren't very promising as can be seen in Table 1. The network can not generate meaningful surface normal and also impaired the depth predictions. Thus, we independently train the NormalNet along with the pre-trained DepthNet to obtain more meaningful results shown in Figure 4c. As can be seen in Figure 5, we can roughly determine the road of the scene but geometric details are coarse.

Evaluation on the patch-based photometric constraint. We evaluate the patch-based photometric constraint by adding the L_{patch} term in our loss function and setting its weight to 0.1. We implement the network architecture of this experiment as shown in Figure 4b. In Figure 5 we show a qualitative result of this approach. As can been seen, the estimated normal could roughly capture the contours of the scene; however, there are still some spurious blobs in the image.

Threshold: $\%of y_i, s.t. max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$	RMSE(linear): $\sqrt{\frac{1}{\|T\|}\sum_y \|\|y_i - y_i^*\|\|^2}$
Abs Relative difference: $\frac{1}{\|T\|}\sum_{y \in T} \|\|y - y^*\|\|/y^*$	RMSE(log): $\sqrt{\frac{1}{\|T\|}\sum_{y \in T} \|\|logy_i - logy_i^*\|\|^2}$
Squared Relative difference: $\frac{1}{\|T\|}\sum_{y \in T} \|\|y - y^*\|\|^2/y^*$	

Table 2: Depth evaluation metrics

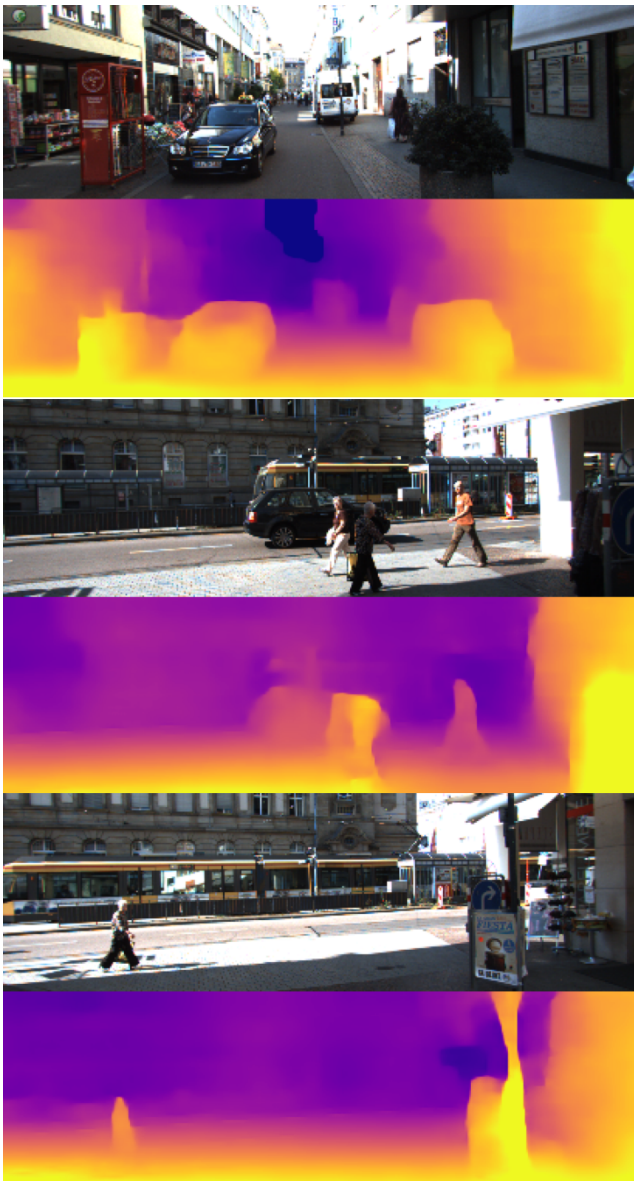

Figure 6: Depth-result using our overall best model using the method from Subsection 3.3.

5. Conclusion and Future Work

We tried several approaches to estimate better depth and normal predictions by offering geometric cues and constraints. In our experiments, we observed that both instance-level and semantic information generate more desirable depth results. Our best depth estimation approach uses instance-level segmentation to model dynamic object motion along with semantic priors(Subsection 3.3). Also, we observed that by restraining the network with the depth and normal orthogonality and patch-based constraint, we can also estimate the surface normal of the scene, which is beneficial for future applications.

In the future, we plan to experiment the refinement methodology as explained in [1] to further improve the depth prediction of our best approach. Currently, since our NormalNet is incapable of generating high quality normal maps as compared to non-learning based approaches, we will continue to improve our network with different geometric constraints and architectural designs. Lastly, we would like to show scalability of our approach on different dataset like Cityscape and Make3D.

References

[1] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. *arXiv preprint arXiv:1811.06152*, 2018.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] 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.

[5] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005.

[6] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. *CoRR*, abs/1506.06825, 2015.

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Com-*

puter Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 3354–3361. IEEE, 2012.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, 2016.

[12] P. D. T. Malik, P. E. Debevec, and C. J. Taylor. " modeling and rendering architecture from photographs: A hybrid geometry and image-based approach. In *Proc. ACM SIGGraph*, volume 96, pages 11–20, 1996.

[13] N. Mayer, E. Ilg, P. Husser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, June 2016.

[14] E. Shechtman, A. Rav-Acha, M. Irani, and S. Seitz. Regenerative morphing. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 615–622. IEEE, 2010.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[16] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 225–234, 2018.

[17] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *arXiv preprint arXiv:1711.03665*, 2017.

[18] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2018.

[19] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[20] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. A. J. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, 2004.