Region Deformer Networks for Unsupervised Depth Estimation from Unconstrained Monocular Videos

Anonymous ICCV submission

Paper ID ****

Abstract

Learning based depth estimation has advanced a lot recently. However, previous supervised methods are limited by small amount of ground truth data, unsupervised methods for monocular videos are mostly based on the static scene assumption, not performing well on real world scenarios where dynamic objects are often present. To resolve these issues, we propose a generic framework to learn depth from unconstrained monocular videos without ground truth supervision. The key difference from previous unsupervised methods is the proper handling of various forms objects motion, e.g. rigidly moving cars and deformable humans. Specifically, a deformation based motion representation is proposed to model independent object motion on 2D image plane, which makes our framework general enough to be applicable on diverse unconstrained monocular videos. Our method not only achieves state-of-the-art results on standard benchmarks KITTI and Cityscapes, but also shows promising results on publicly available pedestrian tracking datasets and causal videos captured by a hand-held phone camera, validating the effectiveness of our deformation based motion representation.

1. Introduction

Depth sensing plays an important role in 3D scene understanding, for instance, it is crucial for robots to be aware of how far the surrounding objects are away from themselves, which can help robots keep clear of obstacles and adjust future behaviour. Learning based single-image depth estimation has attracted lots of attention recently due to the rapid progress made by Convolutional Neural Networks (CNN). Supervised methods [6, 18, 19, 21] aim at learning a mapping from color image to per-pixel depth by neural networks. However, these works require a large quantity of color-depth pairs, which is challenging to collect, especially in outdoor scenarios.

Unsupervised learning is a promising direction as it gets rid of the dependence on ground truth depth, where the key idea is using warping based image reconstruction loss between adjacent frames to guide the learning process. Several works use stereo images to estimate depth [9, 12]. Although no ground truth is required, the stereo images are still not as common as monocular videos and they need to be carefully synchronized.

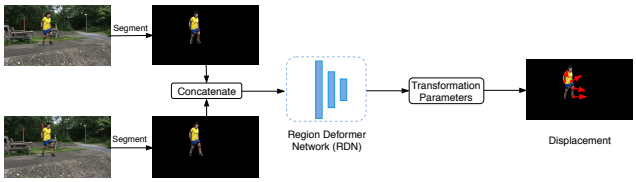

Figure 1: Illustration of our proposed deformation based motion representation. As deformation is applied on region of object, we call it as Region Deformer Network (RDN), composed of several convolution layers. The input is the concatenation of two segmentations from same object in adjacent frames. The object motion is modeled by a function $f_\theta : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, which maps pixel location (x, y) to its displacement $(\Delta x, \Delta y)$ on adjacent frame, the parameters θ of the transformation f are learned by RDN without any predefined correspondences. Details can be found at Sec. 3.2.

In this paper, we focus on unsupervised depth estimation from monocular videos, as videos are more ubiquitous and unlimited. However, many previous works in this line are based on the static scene assumption [23, 29, 33, 35, 37], where only camera motion is considered, leading to inaccurate results for moving objects [11]. Several works try to explicitly model objects motion, either with optical flow [34, 32] or SE(3) transforms [1] by assuming rigid motion of objects. They perform well in outdoor driving scenes, like KITTI [10] and Cityscapes [3] datasets, where moving objects are mainly from rigid cars. However, for real world scenarios, deformable objects are often presented in various forms, e.g. pedestrian and animals. To this end, a deformation based motion representation is proposed to model diverse objects motion between adjacent frames on

2D image plane, making our framework general enough to be applicable on unconstrained monocular videos.

Specifically, the ambiguities of camera motion and independent object motion makes learning harder in dynamic scenes [22]. To disentangle camera motion and independent object motion between adjacent frames, we additionally learn a transformation $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ for every independently moving object to deform itself to adjacent frame, as illustrated in Fig. 1. The main requirement is that f should be complex enough to model rigid and non-rigid objects motion, we choose bicubic function here inspired from powerful expressive ability of bicubic spline interpolation [5]. The transformation parameters are learned by a CNN without any predefined correspondences, which is fully guided by image appearance dissimilarity detailed at Sec. 3.1. An existing Mask R-CNN [13] model is used to segment objects in each frame, note that the segmentation is only needed at training time, which helps our motion representation to learn a better depth estimator.

Our main contributions are as follows:

- We propose a deformation based motion representation to model rigid or non-rigid objects motion between adjacent frames on 2D image plane, which makes our framework general enough to be applicable on unconstrained monocular videos without ground truth supervision

- Our framework not only achieves state-of-the-art results on standard benchmarks KITTI and Cityscapes, but also shows promising results on crowded pedestrian datasets and casual videos captured by a hand-held phone camera, validating the effectiveness of our deformation based motion representation

2. Related Works

Learning based depth estimation from single image has attracted considerable attention recently due to the rapid progress made by CNN. Here we review the most related works using deep learning and works learning geometric transformation with CNN.

Supervised Depth Estimation Supervised methods aim at learning a mapping from RGB image to depth map with the strong representation of CNN [6, 21, 31]. Fu *et al*. [8] recast depth estimation as an ordinal regression problem and achieve state-of-the-art results on several benchmarks. However, ground truth depth of real world scenes is hard to acquire, especially in outdoor scenarios, which limits its applicability. Several works try to resolve this limitation by using synthetic data [24, 36] or images from Internet [2, 20], but special care must be taken to generate high quality training data, which can be very time-consuming.

Unsupervised Depth Estimation Unsupervised approach is a promising direction as no ground truth depth is needed. The key idea of unsupervised depth estimation is using image reconstruction loss between adjacent frames to provide self-supervision, which can be accomplished by using carefully synchronized stereo pairs [9, 12] or monocular videos [1, 23, 29, 33, 34, 35, 37]. We focus on depth estimation from monocular videos as videos are more ubiquitous and unlimited than rectified stereo pairs.

Most similar to our approach is [1], however, the differences come from several important aspects. (i) [1] model objects motion on 3D with SE(3) transformation, which is only applicable for rigidly moving objects, like cars in driving scenes. While we use a deformation based representation to model objects motion on 2D image plane, which is general enough to be applicable on diverse real world scenarios. (ii) To handle the common issue that cars moving in front of camera at roughly same speed are often projected into infinite depth in monocular setting, [1] propose to impose object size constraints depending on height of object segmentation mask, which is not applicable for deformable objects, as the actual scale can be varied over time. Also, all of the constraints in [1] are learned by network, which can be tricky to find the right hyper-parameters. We choose to use a simple yet efficient prior proposed by [26] instead, which is more general for diverse scenes and has no parameters to learn.

Learning Geometric Transformation Spatial Transformer Networks (STN) [15] build the first learnable module in the network architecture to handle geometry variation of input data, which is realized by learning a global parametric transformation. Deformable ConvNets [4] further extend STN by learning offsets to regular grid sampling locations in the standard convolution. STN and Deformable ConvNets are both aiming at designing network architectures with geometry invariant for supervised tasks, like classification and segmentation, respectively. Our deformation based motion representation are mostly similar to STN, but we are aiming at learning a transformation for every independent object to model object motion between adjacent frames, and our method is fully unsupervised. WarpNet [16] shares the similar spirit to match images by learning a transformation, but training WarpNet needs the supervision of artificial correspondences, our framework is fully unsupervised and in the context of depth estimation from videos.

3. Method

We propose a generic framework for unsupervised depth estimation from unconstrained monocular videos, where objects motion is explicitly handled by our deformation based motion representation on 2D image plane. We first give an overview of depth estimation from videos, then provide details of our deformation based motion representation.

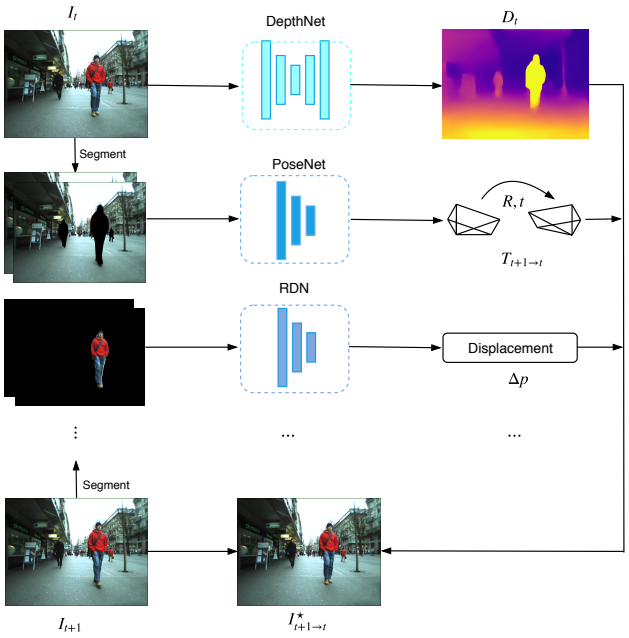

Figure 2: Overview of our proposed framework. Given two adjacent frames I_t and I_{t+1}, we first obtain instance segmentation masks from an existing Mask R-CNN model [14]. A DepthNet is built to output I_t's depth D_t. Camera motion between I_t and I_{t+1} is learned by a PoseNet as $T_{t+1\to t}$, where objects are excluded as its input. We use our proposed RDN (see Sec. 3.2) to model independent object motion Δp in parallel. With $D_t, T_{t+1\to t}, \Delta p$ and I_{t+1}, we can reconstruct frame I_t as $I^{\star}_{t+1\to t}$, the appearance dissimilarity between $I^{\star}_{t+1\to t}$ and I_t provides training signal of our framework.

3.1. Problem Formulation

Given a sequence of video frames $\{I_t\}_{t=1}^N, I_t \in \mathbb{R}^{H\times W\times C}$, where H, W, C denote I_t's height, width and number of channels, respectively. Our goal is to estimate their corresponding depth maps $\{D_t\}_{t=1}^N, D_t \in \mathbb{R}^{H\times W}$, we build a DepthNet to learn the mapping from color image to per-pixel depth map. An additional PoseNet is built to learn the mapping from two adjacent frames I_t, I_{t+1} to their relative camera pose transformation $T_{t+1\to t}$. Given $D_t, I_{t+1}, T_{t+1\to t}$, we can synthetic the reconstructed frame

$$\hat{I}_{t+1\to t} = \mathcal{W}(D_t, I_{t+1}, T_{t+1\to t}) \tag{1}$$

based on the static scene assumption, where $\mathcal{W}$ is a warping function defined below.

For each pixel coordinates $p = (x, y)$ in frame I_t, we can obtain its projected coordinates $\hat{p} = (\hat{x}, \hat{y})$ in frame I_{t+1} with the estimated depth D_t and camera transformation $T_{t+1\to t}$

$$\hat{p} \sim KT_{t+1\to t}D_t(p)K^{-1}h(p), \tag{2}$$

where $h(p) = (x, y, 1)$ denotes the homogeneous coordinates of p, K denotes the camera intrinsic matrix. We use the differentiable bilinear sampling mechanism proposed in [15] to warp frame I_{t+1} and get the reconstructed image $\hat{I}_{t+1\to t}$.

The appearance dissimilarity between reconstructed image $\hat{I}_{t+1\to t}$ and I_t provides training signal for our learning process, which is defined as

$$\ell = \rho(I_t, \hat{I}_{t+1\to t}), \tag{3}$$

where ρ is a dissimilarity measure function, we use a combination of L1 photometric error and Structure Similarity (SSIM) [30]

$$\rho(I, \hat{I}) = \alpha\frac{1 - \text{SSIM}(I, \hat{I})}{2} + (1 - \alpha)\|I - \hat{I}\|_1. \tag{4}$$

To handle occlusion/disocclusion between adjacent frames, per-pixel minimum between previous frame and next frame is used as proposed by [11]

$$L_{\text{ap}} = \min(\rho(I_t, \hat{I}_{t+1\to t}), \rho(I_t, \hat{I}_{t-1\to t})), \tag{5}$$

where $\hat{I}_{t-1\to t}$ is reconstructed frame from I_{t-1}, equivalent to I_{t+1}.

We further impose an image-aware depth smoothness constraint as commonly used by previous works [12, 23, 34]

$$L_{\text{s}} = \sum_{x,y} \|\partial_x D_t\|e^{-\|\partial_x I_t\|} + \|\partial_y D_t\|e^{-\|\partial_y I_t\|}. \tag{6}$$

The total loss function is a combination of L_{ap} and L_{s} applied on 4 scales

$$L = \sum_{k=0}^{3} L_{\text{ap}}^{(k)} + \lambda \cdot \frac{1}{2^k} L_{\text{s}}^{(k)} \tag{7}$$

where λ is a hyper-parameter to balance L_{ap} and L_{s}. We use Eq. 7 as our baseline model, which is the same as [1].

3.2. Region Deformer Networks

For moving objects, the aforementioned formulation doesn't hold as motion between adjacent frames includes independent object motion besides camera motion, i.e. the correspondence $\hat{p}$ in Eq. 2 is not accurate as it only models camera motion. We propose to explicitly model independent object motion on 2D image plane. Specifically, our goal is to learn a displacement $\Delta p = (\Delta x, \Delta y)$ for every pixel belonging to moving objects, then the accurate correspondence for p can be found by $p^{\star} = \hat{p} + \Delta p$. This process is accomplished by learning a function $f : \mathbb{R}^2 \to \mathbb{R}^2$ for every independent object to map (x, y) to its displacement $(\Delta x, \Delta y)$, which shares the similar idea of deformation on 2D image plane [27], but we are deforming in object level

and no predefined correspondence is needed. We choose bicubic function as f here inspired from powerful expressive ability of bicubic spline interpolation [5]. The transformation parameters are learned by our proposed Region Deformer Network (RDN), which is fully guided by image appearance dissimilarity in Eq. 4.

Our proposed RDN is illustrated in Fig.1. Given two adjacent frames I_t and I_{t+1}, an existing instance segmentation model [13] is used to segment objects within each frame, M_t^i and M_{t+1}^i denote i-th object binary segmentation masks in frame I_t and I_{t+1}, respectively. We first compute the reconstructed image $\hat{I}_{t+1\to t}$ and mask $\hat{M}_{t+1\to t}^i$ by camera motion using Eq. 1, which eliminates camera motion between adjacent frames. The input of RDN is the concatenation of objects $O_t^i = I_t \odot M_t^i$ and $\hat{O}_{t+1\to t}^i = \hat{I}_{t+1\to t} \odot \hat{M}_{t+1\to t}^i$ in I_t and $\hat{I}_{t+1\to t}$, respectively, where $\odot$ denotes element-wise multiplication, which makes only i-th object is visible for every independent RDN. When multiple objects exist in a single frame, multiple RDN are used in parallel to model every independent object motion. The output of RDN is the parameters of transformation, by applying the transformation for every independent object, we can get the displacement for every pixel belonging to moving objects.

Now we can define our new warping function $\mathcal{W}^\star$ as follows. If a pixel p belongs to static background, we compute its correspondence by Eq. 2. If p belongs to moving objects, its correspondence is found by $p^\star = \hat{p} + \Delta p$, where $\hat{p}$ is from camera motion using Eq. 2, Δp is from object motion modeled by our RDN. In general, for every pixel p in frame I_t, we can get its correspondence by

$$p^\star = \hat{p} + M_t(p) \cdot \Delta p, \tag{8}$$

where $M_t = \sum_{i=1}^{S} M_t^i$ is the binary instance segmentation mask for I_t, $M_t(p)$ is 1 if p belongs to moving objects, otherwise 0, M_t^i is the binary segmentation mask for every independent object O_t^i, S is the total number of objects in I_t. By modeling independent object motion with RDN, we can get more accurately reconstructed image

$$I_{t+1\to t}^\star = \mathcal{W}^\star(D_t, I_{t+1}, T_{t+1\to t}, f_1, f_2, \cdots, f_S), \tag{9}$$

where $\{f_i\}_{i=1}^S$ are the independent transformations learned by our RDN. This process is illustrated in Fig. 2. Equivalently, we can get the reconstructed image $I_{t-1\to t}^\star$.

3.3. Object Depth Prior

Depth estimation from monocular videos has an issue that objects moving with camera in the same lane at the roughly same speed are often projected to infinite depth, as this shows very little appearance change, resulting in low reprojection error [11]. [1] propose to impose object size constraints by additionally learning actual scales of object. These constraints are internally based on the assumption that object scales are fixed, however, in real world scenarios, deformable objects are often presented in various forms, like pedestrian and animals, which are not applicable for these constrains. Furthermore, as the actual scales are also learned during training process, it can be tricky to find the right hyper-parameters. Instead, we use a simple yet efficient prior proposed by [26]: objects are supported by their surrounding environment, which is often true in most real world scenes. This prior can be expressed as requiring the depths of moving objects to be smaller or equal to their horizontal neighbors. However, overlapping objects may exist in real world, violating the aforementioned depth prior, we propose a soft constraint instead formulated as

$$L_{\text{prior}} = \max(d_{\text{obj}} - d_{\text{neigh}} - \delta, 0), \tag{10}$$

where d_{obj} is the mean depth of an independent object, d_{neigh} is the mean depth of its horizontal neighbors in a small range, δ is a small positive number to handle exceptions violating our depth prior. The main idea of enforcing this prior is that estimating depth from monocular videos is inherently ill-posed, even with accurate 2D image correspondences, like the aforementioned issue of infinite depth. Our insight here is using this prior to prevent the degenerated cases of infinite depth, then refine the results by our proposed RDN in Sec. 3.2. Note that this prior can be satisfied in most times, then Eq. 10 actually has no use (loss is 0 when satisfied), we only rely on RDN to estimate objects depth accurately.

Incorporating the proposed RDN and object depth prior, our final loss function can be expressed as

$$L^\star = \sum_{k=0}^{3} L_{\text{ap}}^{\star(k)} + \lambda \cdot \frac{1}{2^k} L_s^{(k)} + \mu \cdot L_{\text{prior}}^{(k)}, \tag{11}$$

where μ is a hyper-parameter. The only difference between $L_{\text{ap}}^{(k)}$ and $L_{\text{ap}}^{\star(k)}$ is that we replace $I_{t+1\to t}, I_{t-1\to t}$ in Eq. 1 with $I_{t+1\to t}^\star, I_{t-1\to t}^\star$. We denote Eq. 11 as our motion model.

4. Experiments

4.1. Datasets

We conduct extensive experiments on several datasets across diverse scenes, not only standard benchmarks KITTI and Cityscapes, but also publicly available pedestrian tracking datasets and causal videos captured by a hand-held phone camera. The details about each dataset are as follows:

KITTI. The KITTI dataset [10] is a popular benchmark for scene understanding in outdoor driving scenario. Only the monocular video sequences are used for training, no ground truth depth is needed. We follow [37] to pre-process

Table 1: Unsupervised monocular depth estimation results on Eigen test split of KITTI raw dataset. We use K and C to denote models trained on KITTI dataset and Cityscapes dataset, respectively. Abs Rel, Sq Rel, RMSE and RMSE log are error metrics, lower is better. $\delta < 1.25$, $\delta < 1.25^2$ and $\delta < 1.25^3$ are accuracy metrics, higher is better. The best performance is highlighted as bold.

Method	Dataset	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou *et al*. [37]	K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Yang *et al*. [33]	K	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian *et al*. [23]	K	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Yin *et al*. [34]	K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Wang *et al*. [29]	K	0.151	1.257	5.583	0.228	0.810	0.936	0.974
Zou *et al*. [38]	K	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Casser *et al*. [1]	K	0.1412	1.0258	**5.2905**	**0.2153**	0.8160	**0.9452**	**0.9791**
Ours (Baseline)	K	0.1436	1.1732	5.4468	0.2188	0.8202	0.9428	0.9762
Ours (Motion)	K	**0.1378**	**1.0233**	5.3590	0.2160	**0.8225**	0.9429	0.9761
Casser *et al*. [1]	C	0.1876	1.3541	6.3166	0.2641	0.7135	0.9046	0.9667
Ours (Baseline)	C	0.1929	1.6131	6.4098	0.2680	0.7189	0.9071	0.9641
Ours (Motion)	C	**0.1816**	**1.3160**	**6.1484**	**0.2573**	**0.7263**	**0.9124**	**0.9677**

the dataset, and randomly split the processed images as training and validation set, resulting in about 40K images for training and 4K images for validation. The performance are evaluated on Eigen split [6] using standard evaluation protocol.

Cityscapes. The Cityscapes dataset [3] is a similar outdoor driving dataset as KITTI, but with more moving objects. We do the same data pre-processing as [1], resulting in 38,675 images for training. We use this dataset for training and evaluation is done on Eigen split [6].

Pedestrian Tracking Dataset. To validate that our motion representation is general enough to model deformable objects, we collect videos from a publicly available pedestrian tracking dataset [7], which was recorded on crowded pedestrian zone. This dataset is very challenging as large human deformations are frequently exist. 9,369 images are used for training in this dataset.

3DPW. We further evaluate our framework on a recently proposed 3DPW [28] dataset, which is taken by a handheld phone camera used for 3D human pose estimation in the wild. This dataset is more challenging as heading drift, cluttered background and occlusions are exist in the videos. We use 2,648 images for training in this dataset, the promising results validate that our framework can be applicable on unconstrained videos.

4.2. Implementation Details

We implement our framework in TensorFlow. When training on KITTI and Cityscapes datasets, the input images are resized as 128×416. The loss weights are set as $\lambda = 0.04, \mu = 0.5, \alpha = 0.15$. For dataset specific hyperparameters and more details about the pedestrian tracking and 3DPW dataset, please refer to supplemental materials. We set batch size as 4 and use Adam [17] to optimize our network with $\beta_1 = 0.9, \beta_2 = 0.99$. When training our baseline model, the learning rate is set as 2×10^{-4}. Our motion model is trained with learning rate of 2×10^{-5} and the network weights are initialized from our pre-trained baseline model. We use the same DepthNet and PoseNet architectures as [1], our RDN use the same architecture as PoseNet except the last output layer. We will make our code and pre-trained models publicly available to the community.

4.3. Evaluation Metrics

We evaluate our method with several standard metrics reported in [6], including 4 error metrics (Abs Rel, Sq Rel, RMSE, RMSE log) and 3 accuracy metrics ($\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$).

4.4. Results

The KITTI and Cityscapes Dataset

We report our depth estimation results on standard Eigen test split [6] of KITTI raw dataset in Tab. 1. All methods are evaluated on KITTI dataset, no matter they are trained on KITTI or Cityscapes dataset. We achieve comparable results with [1] when training on KITTI dataset, which is specially designed for outdoor driving scenes. When training on more dynamic dataset Cityscapes, our performance is consistently better than [1]. More evidence can be seen from the qualitative comparisons in Fig. 3. On the other hand, comparing our motion model with baseline, the results are consistently better no matter training on KITTI or Cityscapes dataset. To further validate the effectiveness of our proposed motion representation, we evaluate on the specific moving objects on Eigen test split using segmentation masks, results are shown in Tab. 2, the performance of our motion model is consistently better than baseline, demonstrating that our motion representation can model objects

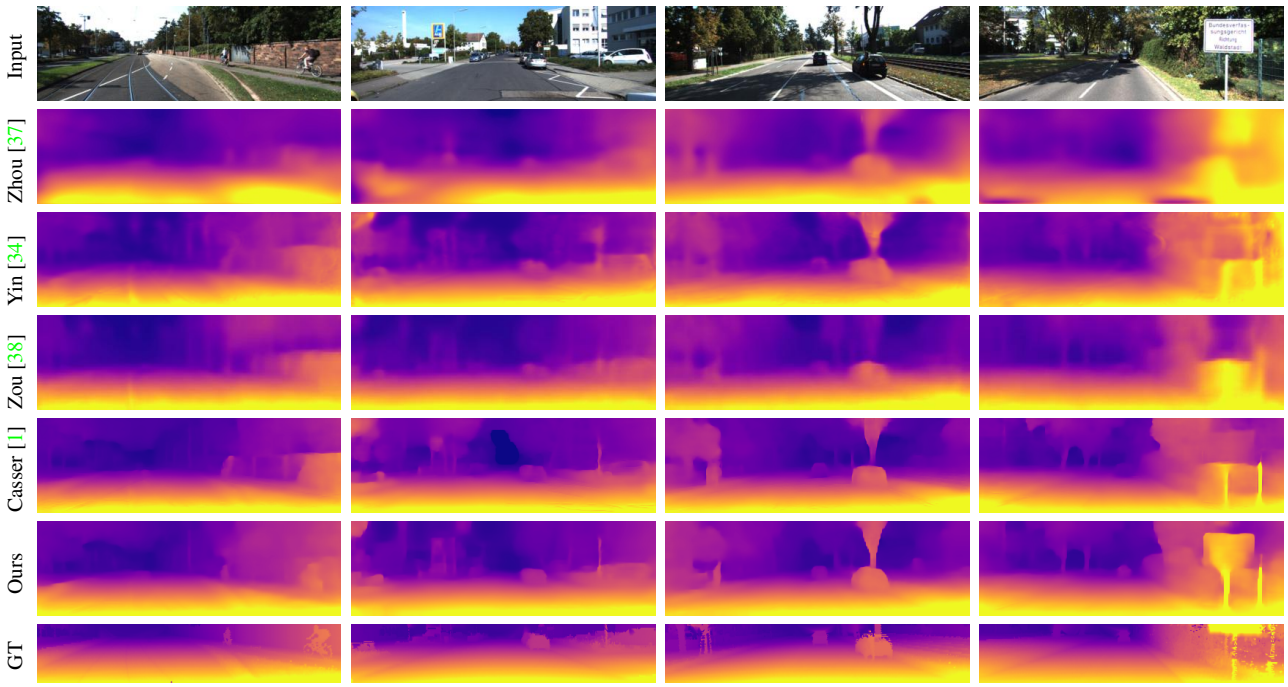

Figure 3: Qualitative comparisons on Eigen test split. GT denotes ground truth depth, which is interpolated for visualization purpose, and the upper regions are cropped as they are not available. Compared with other algorithms, our method can better capture scene structures and moving objects, like cars and riding people.

Table 2: Depth estimation of specific objects on Eigen test split, realized by using the segmentation masks obtained from an existing Mask R-CNN model [13]. Our motion model is consistently better on rigidly moving cars and deformable people, demonstrating the effectiveness of our proposed motion representation.

Method (Objects)	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline (Cars)	0.2503	4.5965	8.5980	0.3217	0.6751	0.8491	0.9157
Motion (Cars)	0.1896	2.1050	7.4392	0.2979	0.7149	0.8764	0.9335
Baseline (People)	0.1860	1.9269	6.2676	0.2462	0.7432	0.9100	0.9634
Motion (People)	0.1772	1.7350	6.1226	0.2419	0.7501	0.9123	0.9645

motion quite well on KITTI and Cityscapes datasets.

The Pedestrian Tracking Dataset

TODO: results on image border are not very well due to texture-less regions (road), will be further improved later.

The 3DPW Dataset

TODO: the camera motion of this dataset is more complex than outdoor driving scenes, some problems still exist, will try to fix later.

Generalization to Unseen Datasets

TODO: maybe train a single model on mixed datasets later, see how it generalize to unseen datasets.

5. Conclusion

We have presented a deformation based motion representation to model various forms of objects motion in diverse scenes, the promising results on several datasets validate the effectiveness of our proposed motion representation. However, depth estimation in dynamic scenes is still a challenging problem that remains unsolved, future work would be incorporating more domain knowledge, like non-rigid structure from motion [25], into the learning process. On the other hand, same as previous works on learning depth from monocular videos, we assume the camera intrinsic parameters are given, which prevents fully use of unlimited Internet videos with unknown camera calibration. In the future, it would be interesting to see how to address this problem, which can enable us to build a truly unconstrained model not only across diverse scenes, but also across different cameras.

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. 1, 2, 3, 4, 5, 6

[2] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016. 2

[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. 1, 5

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017. 2

[5] C. De Boor. Bicubic spline interpolation. *Journal of mathematics and physics*, 41(1-4):212–218, 1962. 2, 4

[6] 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, 5

[7] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust multiperson tracking from a mobile platform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1831–1846, 2009. 5

[8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 2

[9] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 1, 2

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 1, 4

[11] C. Godard, O. Mac Aodha, and G. Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018. 1, 3, 4

[12] 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

[13] 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. 2, 4, 6

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[15] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2, 3

[16] A. Kanazawa, D. W. Jacobs, and M. Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2016. 2

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 1

[19] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015. 1

[20] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2

[21] 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. 1, 2

[22] Z. Lv, K. Kim, A. Troccoli, D. Sun, J. M. Rehg, and J. Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. *arXiv preprint arXiv:1804.04259*, 2018. 2

[23] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018. 1, 2, 3, 5

[24] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, pages 1–19, 2018. 2

[25] V. Rabaud and S. Belongie. Re-thinking non-rigid structure from motion. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 6

[26] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4058–4066, 2016. 2, 4

[27] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *ACM transactions on graphics (TOG)*, volume 25, pages 533–540. ACM, 2006. 3

[28] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 601–617, 2018. 5

[29] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018. 1, 2, 5

[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to

structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3

[31] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016. 2

[32] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding. *arXiv preprint arXiv:1806.10556*, 2018. 1

[33] 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. 1, 2, 5

[34] 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. 1, 2, 3, 5, 6

[35] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 1, 2

[36] C. Zheng, T.-J. Cham, and J. Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 2

[37] 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. 1, 2, 4, 5, 6

[38] Y. Zou, Z. Luo, and J.-B. Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*, pages 38–55. Springer, 2018. 5, 6