Every Pixel Counts: Unsupervised Geometry Learning with Holistic 3D Motion Understanding

Zhenheng Yang[1] Peng Wang[2] Yang Wang[2] Wei Xu[3] Ram Nevatia[1]

[1]University of Southern California [2]Baidu Research
[3]National Engineering Laboratory for Deep Learning Technology and Applications

Abstract. Learning to estimate 3D geometry in a single image by watching unlabeled videos via deep convolutional network has made significant process recently. Current state-of-the-art (SOTA) methods, are based on the learning framework of rigid structure-from-motion, where only 3D camera ego motion is modeled for geometry estimation. However, moving objects also exist in many videos, *e.g.* moving cars in a street scene. In this paper, we tackle such motion by additionally incorporating per-pixel 3D object motion into the learning framework, which provides holistic 3D scene flow understanding and helps single image geometry estimation. Specifically, given two consecutive frames from a video, we adopt a motion network to predict their relative 3D camera pose and a segmentation mask distinguishing moving objects and rigid background. An optical flow network is used to estimate dense 2D per-pixel correspondence. A single image depth network predicts depth maps for both images. The four types of information, *i.e.* 2D flow, camera pose, segment mask and depth maps, are integrated into a differentiable holistic 3D motion parser (HMP), where per-pixel 3D motion for rigid background and moving objects are recovered. We design various losses w.r.t. the two types of 3D motions for training the depth and motion networks, yielding further error reduction for estimated geometry. Finally, in order to solve the 3D motion confusion from monocular videos, we combine stereo images into joint training. Experiments on KITTI 2015 dataset show that our estimated geometry, 3D motion and moving object masks, not only are constrained to be consistent, but also significantly outperforms other SOTA algorithms, demonstrating the benefits of our approach.

1 Introduction

Humans are highly competent in recovering 3D scene geometry, *i.e.*per-pixel depths, at a very detailed level. We can also understand both 3D camera ego motion and object motion from visual perception. In practice, 3D perception from images is widely applicable to many real-world platforms such as autonomous driving, augmented reality and robotics. This paper aims at improving both 3D geometry estimation from single image and also dense object motion understanding in videos.

Recently, impressive progress [1,2,3,4] has been made to achieve 3D reconstruction from a single image by training a deep network taking only unlabeled videos or stereo images as input, yielding even better depth estimation results than those of supervised methods [5] in outdoor scenarios. The core idea is to supervise depth estimation by view

synthesis via rigid structure from motion (SfM) [6]. The frame of one view (source) is warped to another (target) based on the predicted depths of target view and relative 3D camera motions, and the photometric errors between the warped frame and target frame is used to supervise the learning. A similar idea also applies when stereo image pairs are available. However, real world video may contain moving objects, which falls out of rigid scene assumption commonly used in these frameworks. As illustrated in Fig. 1, with good camera motion and depth estimation, the synthesized image can still cause significant photometric error near the region of moving object, yielding unnecessary losses that cause unstable learning of the networks. Zhou *et al.* [2] try to avoid such errors by inducing an explanability mask, where both pixels from moving objects and occluded regions from images are eliminated. Vijayanarasimhan *et al.* [7] separately tackle moving objects with a multi-rigid body model by outputting k object masks and k object pivots from the motion network. However, such a system has limitations of maximum object number, and yields even worse geometry estimation results than those from Zhou *et al.* [2] or other systems [4] which do not explicitly model moving objects.

This paper aims for modeling the 3D motion for unsupervised/self-supervised geometry learning. Different from previous approaches, we model moving objects using dense 3D point offsets, *a.k.a.* 3D scene flow, where the occlusion can be explicitly modeled. Thus, with camera motion in our model, every pixel inside the target image is explained and holistically understood in 3D. We illustrate the whole model in Fig. 2. Specifically, given a target image and a source image, we first introduce an unsupervised optical flow network as an auxiliary part which produces two flow maps: from target to source and source to target images. Then, a motion network outputs the relative camera motion and a binary mask representing moving object regions, and a single view depth network outputs depths for both of the images. The four types of information (2D flow, camera pose, segment mask and depth maps) are fused with a holistic motion parser (HMP), where per-pixel 3D motion for rigid background and moving objects are recovered.

Within the HMP, given depth of the target image, camera pose and moving object mask, a 3D motion flow is computed for rigid background. And given the optical flow, depths of the two images, an occlusion aware 3D motion flow of the full image is computed, where the occlusion mask is computed from optical flow following [8]. In principle, subtracting the two 3D flows within rigid regions, *i.e.*without occlusion and outside moving object mask, the error should be zero. Inside moving object mask, the residual is object 3D motion, which should be spatially smooth. We use these two principles to guide additional losses formulation in our learning system, and all the operations inside the parser are differentiable. Thus, the system can be trained end-to-end, which helps the learning of both motion and depth.

For a monocular video, 3D depth and motion are entangled information, and could be confused with a projective camera model [9]. For example, in the projective model, a very far object moving w.r.t. camera is equivalent to a close object keeping relatively still w.r.t. camera. The depth estimation confusion can be caused at regions of moving object. We tackle this by also embedding the stereo image pair into the monocular learning framework when it is available. In our case, through holistic 3D understanding, we find the joint training yields much better results than solely training on stereo pairs

or monocular videos individually. Finally, as shown in Fig. 1, our model successfully explains the optical flow to 3D motion by jointly estimating depths, understanding camera pose and separating moving objects within an unsupervised manner, where nearly all the photometric error is handled through the training process. Our learned geometry is more accurate and the learning process is more stable.

We conduct extensive experiments over the public KITTI 2015 [10] dataset, and evaluate our results in multiple aspects including depth estimation, 3D scene flow estimation and moving object segmentation. As elaborated in Sec. 4, our approach significantly outperforms other SOTA methods on all tasks.

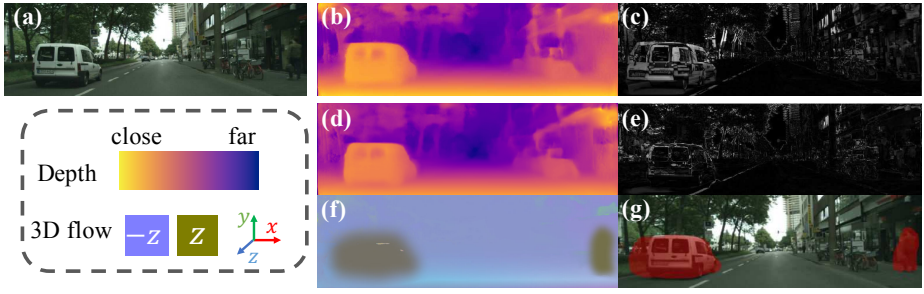

Fig. 1: With good depth estimation (b), there is still obvious reconstruction error around moving object (c). With joint training of 3D flow and depth, our framework generates depth result (d) and camera motion that causes less reconstruction error (e), and also consistent 3D scene flow (f) and moving object segmentation (g) results.

2 Related Work

Estimating single view depth and predicting 3D motion from images have long been center problems for computer vision. Here we summarize the most related works in several aspects without enumerating them all due to space limitation.

Structure from motion and single view geometry. Geometric based methods estimate 3D from a given video with feature matching or patch matching, such as PatchMatch Stereo [11], SfM [6], SLAM [12,13] and DTAM [14], which could be effective and efficient in many cases. When there are dynamic motions inside a monocular video, usually there is scale-confusion for each non-rigid movement, thus regularization through low-rank [15], orthographic camera [16], rigidity [17] or fixed number of moving objects [18] are necessary in order to obtain an unique solution. However, those methods assume 2D matching are reliable, which can fail at where there is low texture, or drastic change of visual perspective *etc.*. More importantly, those methods can not extend to single view reconstruction.

Traditionally, specific rules are necessary for single view geometry, such as computing vanishing point [19], following rules of BRDF [20,21], or extract the scene layout with major plane and box representations [22,23] *etc.*. These methods can only obtain sparse geometry representations, and some of them require certain assumptions (*e.g.* Lambertian, Manhattan world).

Supervised depth estimation with CNN. Deep neural networks (DCN) developed in recent years provide stronger feature representation. Dense geometry, i.e., pixel-wise depth and normal maps, can be readily estimated from a single image [24,25,26,27] and trained in an end-to-end manner. The learned CNN model shows significant improvement compared to other methods based on hand-crafted features [28,29,30]. Others tried to improve the estimation further by appending a conditional random field (CRF) [31,32,33,34]. However, all these methods require densely labeled ground truths, which are expensive to obtain in natural environments.

Unsupervised single image depth estimation. Most recently, lots of CNN based methods are proposed to do single view geometry estimation with supervision from stereo images or videos, yielding impressive results. Some of them are relying on stereo image pairs [35,36,1], by warping one image to another given known stereo baseline. Some others are relying on monocular videos [2,37,38,3,39,40,4] by incorporating 3D camera pose estimation from a motion network. However, as discussed in Sec. 1, most of these models only consider a rigid scene, where moving objects are omitted. Vijayanarasimhan *et al.* [7] model rigid moving objects with k motion masks, while the estimated depths are negatively effected comparing to the one without object modeling [2]. Yin *et al.* [40] model the non-rigid motion by introducing a 2D flow net, which helps the depth estimation. Different from those approaches, we propose to recover a dense 3D motion into the joint training of depth and motion networks, in which the two information are mutually beneficial, yielding better results for both depth and motion estimation.

3D Scene flow estimation. Estimating 3D scene flow [41] is a task of finding per-pixel dense flow in 3D given a pair of images, which evaluates both the depth and optical flow quality. Existing algorithms estimate depth from stereo images [42,43], or the given image pairs [17] with rigid constraint. And for estimation optical flow, they are trying to decompose the scene to piece-wise moving planes in order to finding correspondence with large displacement [44,45]. Most recently, Behl *et al.* [43] adopt semantic object instance segmentation and supervised optical flow from DispNet [46] to solve large displacement of objects, yielding the best results on KITTI dataset. Impressively, in our case, based on single image depth estimation and unsupervised learning pipeline for optical flow, we are able to achieve comparable results with the SOTA algorithms. This demonstrates the effectiveness of our approach.

Segment moving objects. Finally, since our algorithm decomposes static background and moving objects, we are also related to segmentation of moving object from a given video. Current contemporary SOTA methods are dependent on supervision from human labels by adopting CNN image features [47,48] or RNN temporal modeling [49]. For video segmentation without supervision, saliency estimation based on 2D optical flow is often used to discover and track the objects [50,51,52], and a long trajectory [53,54] of the moving objects needs to be considered. However, salient object is not necessary to be the moving object in our case. Moreover, we perform segmentation using only two consecutive images with awareness of 3D motion, which has not been considered in previous approaches.

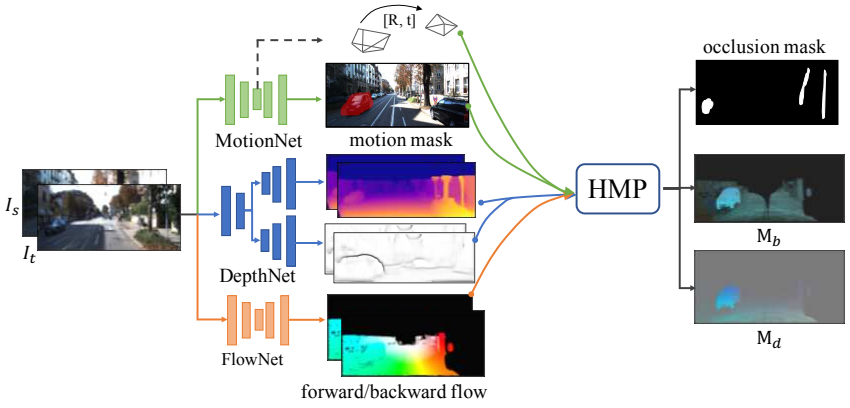

Fig. 2: Pipeline of our framework. Given a pair of consecutive frames, *i.e.*target image I_t and source image I_s, a FlowNet is used to predict optical flow $\mathbf{F}$ from I_t to I_s. Notice here FlowNet is not the one in [55]. A MotionNet predicts their relative camera pose $\mathbf{T}_{t\to s}$ and a mask for moving objects $\mathbf{S}$. A single view DepthNet estimates their depths $\mathbf{D}_t$ and $\mathbf{D}_s$ independently. All the informations are put into our Holistic 3D Motion Parser (HMP), which produce an occlusion mask, 3D motion maps for rigid background $\mathbf{M}_s$ and dynamic objects $\mathbf{M}_d$. Finally, we apply corresponding loss over each of them.

3 Geometry Learning via Holistic 3D Motion Understanding

As discussed in Sec. 1, a major drawback of previous approach [2,4] is ignorance of moving object. In the following, we will discuss the holistic understanding following the rule of geometry (Sec. 3.1). Then, we elaborate how we combine stereo and monocular images with aware of 3D motion, and the losses used to train our depth networks.

3.1 Scene geometry with 3D motion understanding

Given a target view image I_t and a source view image I_s, suppose their corresponding depth maps are $\mathbf{D}_t, \mathbf{D}_s$, their relative camera transformation is $T_{t\to s} = [\mathbf{R}|\mathbf{t}] \in \mathcal{SE}(3)$ from I_t to I_s, and a per-pixel 3D motion map of dynamic moving objects $\mathbf{M}_d$ relative to the world. For a pixel p_t in I_t, the corresponding pixel p_s in I_s can be found through perspective projection, *i.e.*$p_s \sim \pi(p_t)$,

$$h(p_s) = \mathbf{V}(p_t)\frac{\mathbf{K}}{\mathbf{D}(p_s)}[\mathbf{T}_{t\to s}\mathbf{D}(p_t)\mathbf{K}^{-1}h(p_t) + \mathbf{M}_d(p_t)], \tag{1}$$

where $\mathbf{D}(p_t)$ is the depth value of the target view at image coordinate p_t, and $\mathbf{K}$ is the intrinsic parameters of the camera, $h(p_t)$ is the homogeneous coordinate of p_t. $\mathbf{V}(p_t)$ is a visibility mask which is 1 when p_t is also visible in I_s, and 0 if p_t is occluded or flies out of image. In this way, every pixel in I_t is explained geometrically in our model, yielding a holistic 3D understanding. Then given the corresponding p_t and p_s, commonly, one may synthesize a target image $\hat{I}_t$ and compute the photometric loss $\|I_t(p_t) - \hat{I}_t(p_t)\|$ and use spatial transformer network [56] for supervising the training of the networks [2].

Theoretically, given a dense matched optical flow from all available p_t to p_s, when there is no non-rigid motion $\mathbf{M}$, Eq. (1) is convex with respect to $\mathbf{T}$ and $\mathbf{D}$, and could be solved through SVD [57] as commonly used in SfM methods [6]. This supports effective training of networks in previous works without motion modeling. In our case, $\mathbf{M}$ and $\mathbf{D}$ are two conjugate pieces of information, where there always exists a motion that can exactly compensate the error caused by depth. Considering matching p_t and p_s based on RGB could also be very noisy, this yields an ill-posed problem with trivial solutions. Therefore, designing an effective matching strategies, and adopting strong regularizations are necessary to provide effective supervision for the networks, which we will elaborate later.

Unsupervised learning of robust matching network. As discussed in Sec. 2, current unsupervised depth estimation methods [2,37,38,39,4] are mostly based solely on photometric error, *i.e.*$\|I_t(p_t) - \hat{I}_t(p_t)\|$, under Lambertian reflectance assumption and are not robust in natural scenes with lighting variations. More recently, supervision based on local structural errors, such as local image gradient [3], and structural similarity (SSIM) [58,1,40] yields more robust matching and shows additional improvement on depth estimation.

Structural matching has long been a center area for computer vision or optical flow based on SIFT [59] or HOG [60] descriptors. Most recently, unsupervised learning of dense matching [8] using deep CNN which integrates local and global context achieves impressive results according to the KITTI benchmark [1]. In our work, we adopt the unsupervised learning pipeline of occlusion-aware optical flow [8] and a light-weighted network architecture, *i.e.*PWC-Net [61], to learn a robust matching using our training dataset. We found that although PWC-Net is almost $10\times$ smaller than the network of FlowNet [55] which was adopted by [8], it produce higher matching accuracy in our unsupervised setting.

Holistic 3D motion parser (HMP). As described in Sec. 1, in order to apply the supervision, we need to distinguish between the motion from rigid background and dynamic moving objects. As illustrated in Fig. 2, we handle this through a HMP that takes multiple informations from the networks, and outputs the desired two motions.

Formally, four information are input to HMP: depth of both images $\mathbf{D}_s$ and $\mathbf{D}_t$, the learned optical flow $\mathbf{F}_{t\to s}$, the relative camera pose $\mathbf{T}_{t\to s}$ and a moving object segment mask $\mathbf{S}_t$ inside I_t, where the motion of rigid background $\mathbf{M}_b$ and dynamic moving objects $\mathbf{M}_d$ are computed as,

$$\begin{aligned}\mathbf{M}_b(p_t) &= \mathbf{V}(p_t)(1-\mathbf{S}_t(p_t))[\mathbf{T}_{t\to s}\phi(p_t|\mathbf{D}_t) - \phi(p_t|\mathbf{D}_t)]\\ \mathbf{M}_d(p_t) &= \mathbf{V}(p_t)\mathbf{S}_t(p_t)[\phi(p_t+\mathbf{F}_{t\to s}(p_t)|\mathbf{D}_s) - \phi(p_t|\mathbf{D}_t)]\end{aligned} \tag{2}$$

where $\phi(p_t|\mathbf{D}_t) = \mathbf{D}_t(p_t)\mathbf{K}^{-1}h(p_t)$ is a back projection function from 2D to 3D space. $\mathbf{V}$ is the visibility mask as mentioned in Eq. (1), which could be computed by estimating an optical flow $\mathbf{F}_{s\to t}$ as presented in [8]. We refer the reader to their original paper for further details due to the space limitation.

After HMP, the rigid and dynamic 3D motions are disentangled from the whole 3D motion, where we could apply various supervision accordingly based on our structural error and regularizations, which drives the learning of depth and motion networks.

[1] http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow

3.2 Training the networks.

In this section, we describe our loss design based on computed rigid and dynamic 3D motion from HMP. Specifically, as illustrated in Fig. 2, we adopt the network architecture from Yang *et al.* [4], which includes a shared encoder and two sibling decoders, estimating depth $\mathbf{D}$ and geometrical edge map $\mathbf{E}$ respectively, and a MotionNet estimating the relative camera poses. In this work, we also append a decoder with mirror connections in the same way with DepthNet to the MotionNet to output a binary segment mask $\mathbf{S}$ of the moving objects.

Training losses. Given background motion $\mathbf{M}_b(p_t)$ in Eq. (2), we can directly apply the structural matching loss by comparing it with our trained optical flow $\mathbf{F}_{t\to s}$ and the two estimated depth maps $\mathbf{D}_t, \mathbf{D}_s$ ($\mathcal{L}_{st}$ in Eq. (3)). For moving objects $\mathbf{M}_d(p_t)$, we apply an edge-aware spatial smoothness loss for the motion map similar to that in [4]. This is based on the intuition that motions belong to a single object should be smooth in real world ($\mathcal{L}_{ms}$ in Eq. (3)). Last, for $\mathbf{S}_t$ which segments the moving object, similar to the explainability mask in [2], we avoid trivial solutions of treating every pixel as part of moving objects by encouraging zeros predictions inside the mask (($\mathcal{L}_{vis}$ in Eq. (3)).

In summary, the loss functions proposed in our work include,

$$
\begin{aligned}
\mathcal{L}_{st} &= \sum\nolimits_{p_t} |\mathbf{M}_b(p_t) - \hat{\mathbf{M}}_b(p_t)|, \\
\text{where,}\quad \hat{\mathbf{M}}_b(p_t) &= \mathbf{V}(p_t)(1-\mathbf{S}_t(p_t))(\phi(p_t + \mathbf{F}_{t\to s}(p_t)|\mathbf{D}_s) - \phi(p_t|\mathbf{D}_t)), \\
\mathcal{L}_{ms} &= \sum\nolimits_{p_t} (||\mathbf{M}_d(p_t)||^2 + \sum\nolimits_{p_n \in \mathcal{N}_{p_t}} |\mathbf{M}(p_t) - \mathbf{M}(p_n)|\kappa(p_t, p_n|\mathbf{E}_t), \\
\mathcal{L}_{vis} &= -\sum\nolimits_{p_t} \log(1-\mathbf{S}_t(p_t))
\end{aligned} \tag{3}
$$

where $\kappa(p_t, p_n|\mathbf{E}_t) = \exp\{-\alpha \max_{p\in\{p_t,p_n\}}(\mathbf{E}_t(p))\}$ is the affinity between two neighboring pixels, and $\mathcal{N}_{p_t}$ is a four neighbor set of pixel p_t, as defined in [4], which also helps to learn the EdgeNet.

In addition, in order to better regularize the predicted depths, we also add the depth normal consistency proposed in [3] for better regularization of depth prediction with normal information, and the losses corresponding to edge-aware depth and normal smoothness in the same way as [4], *i.e.*$\mathcal{L}_D, \mathcal{L}_N$ and $\mathcal{L}_e$ respectively. We use $\mathcal{L}_{dne}$ to sum them up, and please refer to the original papers for further details. Here, different from [4], we apply such losses for both $\mathbf{D}_s$ and $\mathbf{D}_t$.

Strong supervisions with bi-directional consistency. Although we are able to supervise all the networks through the proposed losses in Eq. (3), we find that the training converges slower and harder when train from scratch compared to the original algorithm [4]. The common solution to solve this is adding a strong supervision at the intermediate stages [62,63]. Therefore, we add a photometric loss without motion modeling for depth and camera motion prediction, and we apply the loss bi-directionally for both target image I_t and source image I_s. Formally, our bi-directional view synthesis cost is written as,

$$
\begin{aligned}
&\mathcal{L}_{bi-vs} = \sum\nolimits_{p_t} s(I_t(p_t), \hat{I}_t(p_t)|\mathbf{D}_t, \mathbf{T}_{t\to s}, I_s) + \sum\nolimits_{p_s} s(I_s(p_s), \hat{I}_t(p_s)|\mathbf{D}_s, \mathbf{T}_{s\to t}, I_t), \\
&\text{where,}\quad s(I(p), \hat{I}(p)|\mathbf{D}, \mathbf{T}, I_s) = |I(p) - \hat{I}(p)| + \beta * \text{SSIM}(I(p), \hat{I}(p))
\end{aligned} \tag{4}
$$

where the $\hat{I}_t(p)$ is the synthesized target image given $\mathbf{D}, \mathbf{T}, I_s$ in the same way with [2]. $s(*,*)$ is a similarity function which includes photometric distance and SSIM [58], and β is a balancing parameter.

Finally, our loss functional for depth and motion supervision from a monocular video can be summarized as,

$$\mathcal{L}_{mono} = \lambda_{st}\mathcal{L}_{st} + \lambda_{ms}\mathcal{L}_{ms} + \lambda_{vis}\mathcal{L}_{vis} + \sum\nolimits_l \{\lambda_{dne}\mathcal{L}^l_{dne} + \lambda_{vs}\mathcal{L}^l_{bi-vs}\} \quad (5)$$

where l indicates the level of image resolution, and four scales are used in the same way with [2].

Stereo to solve motion confusion. As discussed in our introduction (Sec. 1), reconstruction of moving objects in monocular video has projective confusion, which is illustrated in Fig. 3. The depth map (b) is predicted with Yang *et al.* [4], where the car in the front is running at the same speed and the region is estimated to be very far. This is because when the depth is estimated large, the car will stay at the same place in the warped image, yielding small photometric error during training in the model. Obviously, adding motion or smoothness as before does not solve this issue. Therefore, we have added stereo images (which are captured at the same time) into learning the depth network to avoid such confusion. As shown in Fig. 3 (c), the framework trained with stereo pairs correctly figures out the depth of the moving object regions.

(a) (b) (c)

Fig. 3: Moving object in the scene (a) causes large depth value confusion for framework trained with monocular videos, as shown in (b). This issue can be resolved by incorporating stereo training samples into the framework (c).

Formally, when corresponding stereo image I_c is additionally available for the target image I_t, we treat I_c as another source image, similar to I_s, but with known camera pose $\mathbf{T}_{t\to c}$. In this case, since there is no motion factor, we adopt the same loss of $\mathcal{L}_{dne}$ and $\mathcal{L}_{bi-vs}$ taken I_c, I_t as inputs for supervising the DepthNet. Formally, the total loss when having stereo images is,

$$\mathcal{L}_{mono-stereo} = \mathcal{L}_{mono} + \sum\nolimits_l \{\lambda_{dne}\mathcal{L}^l_{dne}(I_c) + \lambda_{vs}\mathcal{L}^l_{bi-vs}(I_c)\}. \quad (6)$$

where $\mathcal{L}_{dne}(I_c)$ and $\mathcal{L}_{bi-vs}(I_c)$ indicate the corresponding losses which are computed using stereo image I_c.

4 Experiments

In this section, we describe the datasets and evaluation metrics used in our experiments. And then present comprehensive evaluation of our framework on different tasks.

4.1 Implementation details

Our framework consists of three networks: DepthNet, FlowNet and MotionNet. The DepthNet + MotionNet and FlowNet are first trained on KITTI 2015 dataset separately. Then DepthNet and MotionNet are further finetuned with additional losses from HMP as in Sec. 3.

DepthNet architecture. A DispNet [46] like achitecture is adopted for DepthNet. Regular DispNet is based on an encoder-decoder design with skip connections and multi-scale side outputs. To train with stereo images, the output's channel for each scale is changed to 2, as in [1]. As in [4], the DepthNet has two sibling decoders which separately output depths and object edges. To avoid artifact grid output from decoder, the kernel size of decoder layers is set to be 4 and the input image is resized to be non-integer times of 64. All *conv* layers are followed by ReLU activation except for the top output layer, where we apply a sigmoid function to constrain the depth prediction within a reasonable range. Batch normalization [64] is performed on all *conv* layers. To increase the receptive field size while maintaining the number of parameters, dilated convolution with a dilation of 2 is implemented. During training, Adam optimizer [65] is applied with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate of 2×10^{-3} and batch size of 4. Other hyperparameters are set as in [4].

FlowNet architecture. A PWC-Net [61] is adopted as FlowNet. PWC-Net is based on an encoder-decoder design with intermediate layers warping CNN features for reconstruction. The network is optimized with Adam optimizer [65] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate of 1×10^{-4} for 100,000 iterations and then 1×10^{-4} for 100,000 iterations. The batch size is set as 8 and other hyperparameters are set as in [8].

MotionNet architecture. The MotionNet implements the same U-net [66] architecture as the Pose CNN in [2]. The 6-dimensional camera motion is generated after 7 *conv* layers and the motion mask is generated after symmetrical *deconv* layers.

For end-to-end finetuning of DepthNet and MotionNet with HMP, the hyperparameters are set as: $\lambda_{st} = 0.5$, $\lambda_{ms} = 0.25$, $\lambda_{vis} = 0.8$, $\lambda_{dne} = 0.2$, $\lambda_{vs} = 1.0$. The trade-off weight between photometric loss and SSIM loss is set as $\beta = 0.5$. All parameters are tuned on the validation set.

4.2 Datasets and metrics

Extensive experiments have been conducted on three different tasks: depth estimation, scene flow estimation and moving object segmentation. The results are evaluated on the KITTI 2015 dataset, using corresponding metrics.

KITTI 2015. KITTI 2015 dataset provides videos in 200 street scenes captured by stereo RGB cameras, with sparse depth ground truths captured by Velodyne laser scanner. 2D flow and 3D scene flow ground truth is generated from the ICP registration of point cloud projection. The moving object mask is provided as a binary map to distinguish background and foreground in flow evaluation. During training, 156 stereo videos excluding test and validation scenes are used.The monocular training sequences are constructed with three consecutive frames in the left view, while stereo training pairs

are constructed with left and right frame pairs, resulting in a total of 22,000 training samples.

For depth evaluation, two test splits of KITTI 2015 are proposed: the official test set consisting of 200 images (KITTI split) and the test split proposed in [5] consisting of 697 images (Eigen split). The official KITTI test split provides ground truth of better quality compared to Eigen split, where less than 5% pixels in the input image has ground truth depth values. For better comparison with other methods, the depth evaluation is conducted on both splits. For scene flow and segmentation evaluation, as the flow ground truth is only provided for KITTI split, our evaluation is conducted on the 200 images in KITTI test split.

Cityscapes. Cityscapes is a city-scene dataset captured by stereo cameras in 27 different cities. As depth ground truth is not available, Cityscapes is only used for training and the training samples are generated from 18 stereo videos in the training set, resulting in 34,652 samples.

Metrics. The existing metrics of depth, scene flow and segmentation have been used for evaluation, as in [5], [42] and [67]. For depth and scene flow evaluation, we have used the code by [1] and [42] respectively. For foreground segmentation evaluation, we implemented the evaluation metrics in [67]. The definition of each metric used in our evaluation is specified in Tab. 1. In which, x^* and x' are ground truth and estimated results ($x \in \{d, sf\}$). n_{ij} is the number of pixels of class i segmented into class j. t_i is the total number of pixels in class i. n_{cl} is the total number of classes, which is equal to 2 in our case.

Table 1: From top row to bottom row: depth, scene flow and segmentation evaluation metrics.

Abs Rel: $\frac{1}{\lvert D\rvert}\sum_{d'\in D}\lvert d^*\text{–}d'\rvert/d^*$	Sq Rel: $\frac{1}{\lvert D\rvert}\sum_{d'\in D}\lvert\lvert d^*\text{–}d'\rvert\rvert^2/d^*$
RMSE: $\sqrt{\frac{1}{\lvert D\rvert}\sum_{d'\in D}\lvert\lvert d^*\text{–}d'\rvert\rvert^2}$	RMSE log: $\sqrt{\frac{1}{\lvert D\rvert}\sum_{d'\in D}\lvert\lvert \log d^*\text{–}\log d'\rvert\rvert^2}$
D1, D2: $\frac{1}{\lvert D\rvert}\sum_{d'\in D}\lvert d^*\text{–}d'\rvert$	SF: $\frac{1}{\lvert SF\rvert}\sum_{sf'\in SF}\lvert sf^*\text{–}sf'\rvert$
pixel acc. $\frac{\sum_i n_{ii}}{\sum_i t_i}$	mean acc. $\frac{1}{n_{cl}}\sum_i \frac{n_{ii}}{t_i}$
mean IoU $\frac{1}{n_{cl}}\sum_i \frac{n_{ii}}{t_i+\sum_j n_{ji}+n_{ii}}$	f.w. IoU: $\frac{1}{\sum_i t_i}\sum_i \frac{n_{ii}}{t_i+\sum_j n_{ji}+n_{ii}}$

4.3 Depth evaluation

Experiment setup. The depth experiments are conducted on KITTI 2015 and Cityscapes. For KITTI test split, the given depth ground truth is used for evaluation. For Eigen test split, synchronized Velodyne points are provided and these sparse points are projected and serve as depth ground truth. Only pixels with ground truth depth values are evaluated. The following evaluations are performed to present the depth results: (1) ablation study of our approach; (2) depth estimation performance comparison with SOTA methods.

Ablation study. We explore the effectivness of each component in our framework. Several variant results are generated for evaluation, which include: (1) DepthNet trained with only monocular training sequences (Ours (mono)); (2) DepthNet trained with

monocular samples and then finetuned with HMP (Ours (mono+HMP)); (3) DepthNet without finetuning from 3D solver loss (Ours w/o HMP). For traning with only monocular sequences, the left and right sequences are considered independently, thus resulting in 44,000 training samples. The quantitative results of different variants are presented in Tab. 2. Although these three variants use the same amount of data, our approach trained with both stereo and sequential samples shows large performance boost over using only one type of training samples, proving the effectiveness of incorporating stereo into training. With the finetuning from HMP, the performance is further improved.

Comparison with state-of-the-art. Following the tradition of other methods [5,2,1], our framework is trained with two strategies: (1) trained with KITTI data only; (2) trained with Cityscapes data and then finetuned with KITTI data (CS+K). The maximum of depth estimation on KITTI split is capped at 80 meters and the same crop as in [5] is applied during evaluation on Eigen split.

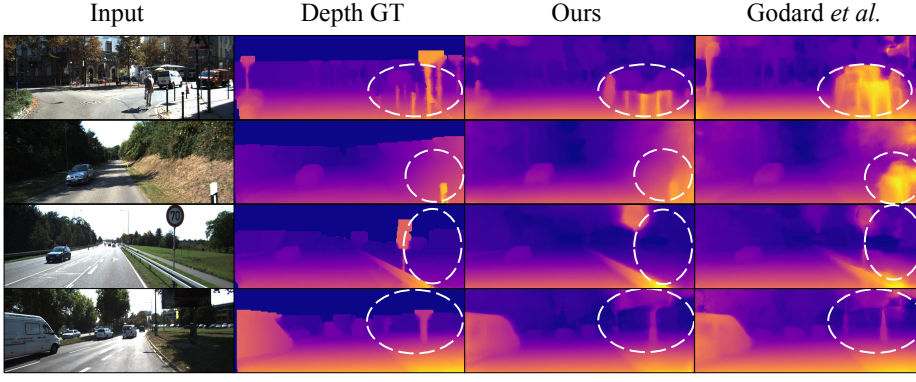

Fig. 4: Visual comparison between Godard *et al.*[1] and our results on KITTI test split. The depth ground truths are interpolated and all images are reshaped for better visualization. For depths, our results have preserved the details of objects noticeably better (as in white circles).

Tab. 2 shows the comparison of ours performance and recent SOTA methods. Our approach outperforms current SOTA unsupervised methods [2,68,4,1] on almost all metrics by a large margin when trained with KITTI data. When trained with more data (CS+K), our method still shows the SOTA performance on the "Abs Rel" metric. Some depth estimation visualization results are presented in Fig. 1, comparing with results from [1]. Our depth results have preserved the details of the scene noticeably better.

4.4 Scene flow evaluation

Experiment setup. The scene flow evaluation is performed on KITTI 2015 dataset. For 200 frames pairs in KITTI test split, the depth ground truth of the two consecutive frames (t and $t+1$) and the 2D optical flow ground truth from frame t to frame $t+1$ are provided. Following the KITTI benchmark evaluation toolkit, the scene flow evaluation is conducted on the two depth results and optical flow results. As the unsupervised

Table 2: Monocular depth evaluation results on KITTI split (upper part) and Eigen split(lower part). Results of [2] on KITTI test split are generated by training their released model on KITTI dataset. All results are generated by model trained on KITTI data only unless specially noted. "pp" denotes post processing implemented in [1].

Method	Split	Stereo	Lower the better				Higher the better		
			Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Train mean	KITTI		0.398	5.519	8.632	0.405	0.587	0.764	0.880
Zhou *et al.*[2]			0.216	2.255	7.422	0.299	0.686	0.873	0.951
LEGO[4]			0.154	1.272	6.012	0.230	0.795	0.932	0.975
Wang *et al.*[37]			0.151	1.257	5.583	0.228	0.810	0.936	0.974
Godard *et al.*[1]		✓	0.124	1.388	**6.125**	0.217	0.841	0.936	0.975
Ours (mono)			0.137	1.326	6.232	0.224	0.806	0.927	0.973
Ours (mono+HMP)			0.131	1.254	6.117	0.220	0.826	0.931	0.973
Ours (w/o HMP)		✓	0.117	1.163	6.254	0.212	0.849	0.932	0.975
Ours		✓	**0.109**	**1.004**	6.232	**0.203**	**0.853**	**0.937**	**0.975**
Godard *et al.*[1] (CS+K+pp)		✓	0.100	**0.934**	**5.141**	**0.178**	**0.878**	**0.961**	0.986
Ours (CS+K)		✓	**0.099**	0.986	6.122	0.194	0.860	0.957	**0.986**
Train mean	Eigen		0.403	5.530	8.709	0.403	0.593	0.776	0.878
Zhou *et al.*[2]			0.208	1.768	6.856	0.283	0.678	0.885	0.957
UnDeepVO[38]		✓	0.183	1.730	6.570	0.268	-	-	-
LEGO[4]			0.162	1.352	6.276	0.252	0.783	0.921	0.969
Mahjourian *et al.*[39]			0.163	1.240	6.220	0.250	0.762	0.916	0.968
Godard *et al.*[1]		✓	0.148	1.344	**5.927**	0.247	0.803	0.922	0.964
Ours		✓	**0.127**	**1.239**	6.247	**0.214**	**0.847**	**0.926**	**0.969**
Godard *et al.*[1] (CS+K+pp)		✓	0.118	**0.923**	**5.015**	0.210	**0.854**	**0.947**	0.976
Ours (CS+K)		✓	**0.114**	1.074	5.836	**0.208**	0.856	0.939	**0.976**

method generates depth/disparity up to a scale, we rescale the depth estimation by a factor to make the estimated depth median equal to ground truth depth median.

Ablation study. We explore the effectiveness of HMP and other loss terms by several ablation experiments: (1)excluding the HMP module from our framework (Ours w/o HMP); (2) DepthNet trained with monocular samples (Ours (mono)). The scene flow evaluation results of different variants are presented in Tab. 3. As the same trend in depth evaluation, both incorporating stereo examples into training and finetuning with HMP help improve the scene flow performance.

Comparison with other methods. The comparison with current SOTA scene flow methods are presented in Tab. 3. Note that all supervised methods use the stereo image pairs to generate the disparity estimation during testing. The performance of "Ours w/o HMP" is further improved with scene flow solver, proving the capability of facilitating depth learning through optical flow in the proposed HMP. The depth, flow and scene flow errors are visualized in Fig. 5.

4.5 Moving object segmentation

We evaluate the moving object segmentation performance to test the capability of capturing foreground motion in our framework.

Experiment setup. The moving object segmentation is evaluated on KITTI 2015 dataset. "Object map" ground truth is provided in this dataset to dinstinguish foreground and

Table 3: Scene flow performances of different methods on KITTI 2015 dataset. Upper part includes results of supervised methods and the bottom part includes unsupervised methods.

	Supervision	D1			D2			FL		
		bg	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*
OSF[42]	Yes	4.00	8.86	4.74	5.16	17.11	6.99	6.38	20.56	8.55
ISF[43]	Yes	3.55	3.94	3.61	4.86	4.72	4.84	6.36	7.31	6.50
Ours w/o HMP	No	24.22	27.74	26.38	68.84	71.36	69.68	25.34	28.00	25.74
Ours(mono)	No	26.12	30.27	30.54	23.94	73.85	68.47	25.34	28.00	25.74
Ours	No	23.62	27.38	26.81	18.75	70.89	60.97	25.34	28.00	25.74

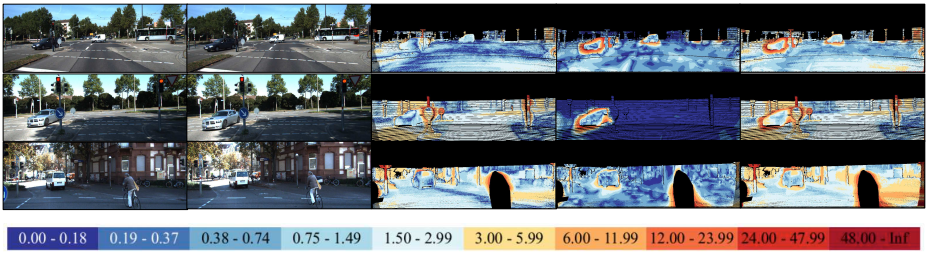

Fig. 5: Errors in scene flow evaluation. The left two columns show the two consecutive frames as input. The other three columns show the error in depth, flow and scene flow evaluation. The color code of error is following the tradition of [42].

background in flow evaluation. Such dense motion mask serve as ground truth in our segmentation evaluation. Fig. 6 (second column) shows some visualization of segmentation ground truths.

For better quantitative comparison, we propose several baseline methods to do moving object segmentation, including: (1) Using segment mask from MotionNet in the same way as explainability mask of [2] with our learning pipeline by removing HMP; (2) Compute a residual flow map by substracting 3D flow induced by camera motion (using $\mathbf{T}_{t\to s}, \mathbf{D}_t, \mathbf{V}_t$) from the full 3D scene flow (using $\mathbf{F}_{t\to s}, \mathbf{D}_t, \mathbf{D}_s, \mathbf{V}_t$). Then, we apply a two-class Gaussian Mixture Model (GMM) to fit the flow magnitude, on which do graph cut to generate the segmentation results (Graphcut on residual flow). We leave the segmentation details in supplimentary material due to space limit.

Evaluation results. We compare our segmentation results from the motion mask and those from the two baseline methods. As the Tab. 4 shows, our segmentation results from the motion mask shows superior performance compared to the masks applied in depth reconstruction or masks calculated from the scene flow residual. Visualization examples of segmentation are presented in Fig. 6. Our segmentation results are focused on moving object compared to the explainability masks similar to [2], which is optimized to filter out any reconstruction error.

5 Conclusion

In this paper, we proposed a self-supervised framework for joint 3D geometry and dense object motion learning. A novel depth estimation framework is proposed to model better depth estimation and also the ego-motion. A holistic 3D motion parser (HMP) is

Table 4: Foreground moving object segmentation performance on KITTI 2015 dataset.

	pixel acc.	mean acc.	mean IoU	f.w. IoU
Explainability mask	70.32	58.24	41.95	67.56
Graphcut on residual flow	75.05	67.26	50.83	71.32
Ours	**88.71**	**74.59**	**52.25**	**86.53**

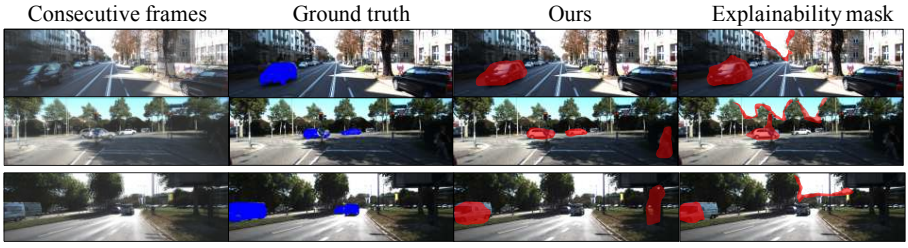

Fig. 6: Moving object segmentation results.

proposed to model the consistency between depth and 2D optical flow estimation. Such consistency is proved to be helpful for supervising depth learning. We conducted comprehensive experiments to present the performance. On KITTI dataset, our approach achieves SOTA performance on all depth, scene flow and moving object segmentation evaluations. In the future, we would like to extend our framework to other motion video data sets containing deformable and articulated non-rigid objects such as MoSeg [53] *etc*., in order to make the learning as general as possible.

References

1. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. (2017) 1, 4, 6, 9, 10, 11, 12
2. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR. (2017) 1, 2, 4, 5, 6, 7, 8, 9, 11, 12, 13
3. Yang, Z., Wang, P., Xu, W., Zhao, L., Ram, N.: Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In: AAAI. (2018) 1, 4, 6, 7
4. Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R.: Lego: Learning edge with geometry all at once by watching videos. In: CVPR. (2018) 1, 2, 4, 5, 6, 7, 8, 9, 11, 12
5. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NIPS. (2014) 1, 10, 11
6. Wu, C., et al.: Visualsfm: A visual structure from motion system. (2011) 2, 3, 6
7. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: Sfm-net: Learning of structure and motion from video. CoRR **abs/1704.07804** (2017) 2, 4
8. Wang, Y., Yang, Y., Yang, Z., Wang, P., Zhao, L., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: CVPR. (2018) 2, 6, 9
9. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE transactions on pattern analysis and machine intelligence **30**(5) (2008) 878–892 2
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR. (2012) 3
11. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: Bmvc. Volume 11. (2011) 1–11 3

12. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics **31**(5) (2015) 1147–1163 3
13. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: ECCV. (2014) 3
14. Newcombe, R.A., Lovegrove, S., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: ICCV. (2011) 3
15. Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. International Journal of Computer Vision **107**(2) (2014) 101–122 3
16. Taylor, J., Jepson, A.D., Kutulakos, K.N.: Non-rigid structure from locally-rigid motion. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 2761–2768 3
17. Kumar, S., Dai, Y., Li, H.: Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. ICCV (2017) 3, 4
18. Kumar, S., Dai, Y., Li, H.: Multi-body non-rigid structure-from-motion. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 148–156 3
19. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. In: ICCV. (2007) 3
20. Prados, E., Faugeras, O.: Shape from shading. Handbook of mathematical models in computer vision (2006) 375–388 3
21. Kong, N., Black, M.J.: Intrinsic depth: Improving depth transfer with intrinsic images. In: ICCV. (2015) 3
22. Schwing, A.G., Fidler, S., Pollefeys, M., Urtasun, R.: Box in the box: Joint 3d layout and object reasoning from single images. In: ICCV. (2013) 3
23. Srajer, F., Schwing, A.G., Pollefeys, M., Pajdla, T.: Match box: Indoor image matching via box-like scene estimation. In: 3DV. (2014) 3
24. Wang, X., Fouhey, D., Gupta, A.: Designing deep networks for surface normal estimation. In: CVPR. (2015) 4
25. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV. (2015) 4
26. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 239–248 4
27. Li, J., Klein, R., Yao, A.: A two-streamed network for estimating fine-scaled depth maps from single rgb images. In: ICCV. (2017) 4
28. Karsch, K., Liu, C., Kang, S.B.: Depth transfer: Depth extraction from video using non-parametric sampling. IEEE transactions on pattern analysis and machine intelligence **36**(11) (2014) 2144–2158 4
29. Ladicky, L., Shi, J., Pollefeys, M.: Pulling things out of perspective. In: CVPR. (2014) 4
30. L. Ladicky, Zeisl, B., Pollefeys, M., et al.: Discriminatively trained dense surface normal estimation. In: ECCV. (2014) 4
31. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B.L., Yuille, A.L.: Towards unified depth and semantic prediction from a single image. In: CVPR. (2015) 4
32. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: CVPR. (June 2015) 4
33. Li, B., Shen, C., Dai, Y., van den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: CVPR. (2015) 4
34. Wang, P., Shen, X., Russell, B., Cohen, S., Price, B.L., Yuille, A.L.: SURGE: surface regularized geometry estimation from a single image. In: NIPS. (2016) 4
35. Xie, J., Girshick, R., Farhadi, A.: Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In: ECCV. (2016) 4

36. Garg, R., G, V.K.B., Reid, I.D.: Unsupervised CNN for single view depth estimation: Geometry to the rescue. ECCV (2016) 4
37. Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: CVPR. (2018) 4, 6, 12
38. Li, R., Wang, S., Long, Z., Gu, D.: Undeepvo: Monocular visual odometry through unsupervised deep learning. ICRA (2018) 4, 6, 12
39. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. arXiv preprint arXiv:1802.05522 (2018) 4, 6, 12
40. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. arXiv preprint arXiv:1803.02276 (2018) 4, 6
41. Vedula, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. IEEE transactions on pattern analysis and machine intelligence **27**(3) (2005) 475–480 4
42. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: CVPR. (2015) 4, 10, 13
43. Behl, A., Jafari, O.H., Mustikovela, S.K., Alhaija, H.A., Rother, C., Geiger, A.: Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In: CVPR. (2017) 2574–2583 4, 13
44. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 1377–1384 4
45. Lv, Z., Beall, C., Alcantarilla, P.F., Li, F., Kira, Z., Dellaert, F.: A continuous optimization approach for efficient and accurate scene flow. In: ECCV, Springer (2016) 757–773 4
46. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: CVPR. (2016) 4, 9
47. Fragkiadaki, K., Arbelaez, P., Felsen, P., Malik, J.: Learning to segment moving objects in videos. In: CVPR. (2015) 4083–4090 4
48. Yoon, J.S., Rameau, F., Kim, J., Lee, S., Shin, S., Kweon, I.S.: Pixel-level matching for video object segmentation using convolutional neural networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE (2017) 2186–2195 4
49. Tokmakov, P., Schmid, C., Alahari, K.: Learning to segment moving objects. arXiv preprint arXiv:1712.01127 (2017) 4
50. Wang, W., Shen, J., Yang, R., Porikli, F.: Saliency-aware video object segmentation. IEEE transactions on pattern analysis and machine intelligence **40**(1) (2018) 20–33 4
51. Faktor, A., Irani, M.: Video segmentation by non-local consensus voting. In: BMVC. Volume 2. (2014) 8 4
52. Yang, Z., Gao, J., Nevatia, R.: Spatio-temporal action detection with cascade proposal and location anticipation. arXiv preprint arXiv:1708.00042 (2017) 4
53. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: European conference on computer vision, Springer (2010) 282–295 4, 14
54. Kim, K., Yang, Z., Masi, I., Nevatia, R., Medioni, G.: Face and body association for video-based face recognition. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE (2018) 39–48 4
55. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR. (2017) 5, 6
56. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in Neural Information Processing Systems. (2015) 2017–2025 5
57. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9**(2) (1992) 137–154 6

58. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4) (2004) 600–612 6, 8
59. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2) (2004) 91–110 6
60. Lowe, D.G.: Object recognition from local scale-invariant features. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Volume 2., Ieee (1999) 1150–1157 6
61. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. arXiv preprint arXiv:1709.02371 (2017) 6, 9
62. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014) 7
63. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Artificial Intelligence and Statistics. (2015) 562–570 7
64. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. (2015) 9
65. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
66. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241 9
67. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015) 10
68. Kuznietsov, Y., Stuckler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. (2017) 11