Self-supervised Learning of Decomposed Object-wise 3D Motion and Depth from Monocular Videos

Xiuzhe Wu*, Qihao Huang*, and Xiaojuan Qi

***Abstract*— In this paper, we propose a self-supervised method to learn 3D motion and depth from monocular videos. Our system contains a depth estimation module to predict depth and a new decomposed object-wise 3D motion (DO3D) estimation module to predict ego-motion and 3D object motion. Depth and motion are further combined to synthesize a novel video frame for self-supervised training. Our core component – DO3D – is a new motion disentanglement module that learns to predict camera ego-motion and instance-aware 3D object motion separately. More importantly, DO3D bypasses the difficulties in modeling the complicated non-rigid 3D object motion through learning an object-wise 6-DoF global transformation and a pixel-wise local 3D motion deformation field. Qualitative and quantitative experimental results on KITTI and DrivingStereo demonstrate the effectiveness and generalization ability of our model in depth estimation, optical flow estimation, and 3D motion estimation, especially in dynamic regions.**

I. INTRODUCTION

Estimating object-wise 3D motion and depth from monocular videos is a crucial and yet challenging problem in outdoor scene understanding with many applications in autonomous driving vehicles and robots. Recently, supervised data-driven approaches with deep learning have shown promising results [1], [2], [3] for 3D motion and depth estimation. However, it's very difficult to collect ground truth motion and depth data in a large quantity, and models trained on limited training data also suffer from generalization issues [4] in diversified application scenarios. To this end, we study self-supervised object-wise 3D motion and depth learning from a large amount of unlabeled monocular videos.

Although monocular videos – the projection of 3D scene with dynamic objects to the 2D camera planes – can provide important information about the 3D object motion and scene geometry, learning 3D object-wise motion and depth from monocular videos in a self-supervised manner is an ill-posed problem with several inherent challenges: 1) important scene structure information (*e.g.*. scale information) is missing due to 3D-to-2D projection; 2) 3D scene geometry (*e.g.* depth), camera ego-motion, and object-wise motion are entangled together in monocular videos, making it hard to infer 3D motion and depth directly; 3) 3D non-rigid motion patterns, *e.g.* pedestrian motion, are diversified and complicated, hence it is difficult to learn in a self-supervised manner.

Recent approaches [5], [6], [7], [8], [9] attempting to address the above challenges in self-supervised learning can be coarsely categorized into the following streams. One line of research [10], [5] assumes that the scene is static and ignores dynamic objects. Pixel filtering approaches such as auto-masking [5] are proposed to mitigate the effects of dynamic objects. Albeit advancing depth estimation, the above methods can not predict 3D object motion.

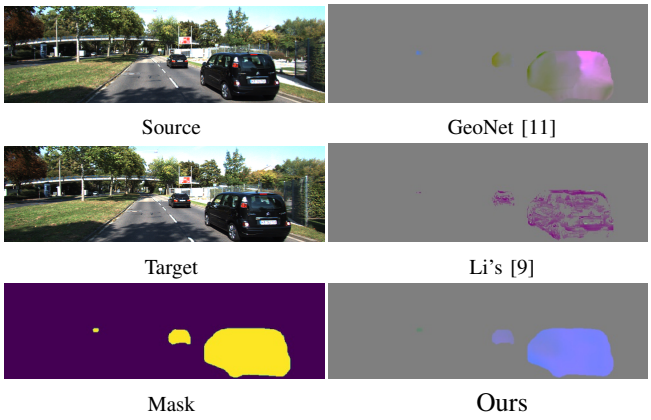

Fig. 1. 3D motion visualization of GeoNet [11], Li's [9], and ours. Our decomposed motion model predicts a consistent 3D motion filed than others by optical flow or direct methods. R,G,B color maps correspond to motion in x,y, z direction respectively.

Another stream tries to learn a residual 2D optical flow map [11] to model object motion. However, 2D optical flow cannot fully utilize the 3D constraints, *e.g.* the 3D motion of cars are mostly rigid, and also may not necessarily deliver high-quality 3D motion.

Most recently, several approaches [7], [9] have been proposed to jointly learn 3D motion and depth. Li *et al.* [9] propose to predict an extra 3D motion map for the entire image. Unfortunately, despite smoothness constraints, the approach cannot incorporate object-wise constraints, and the complicated 3D object non-rigid motion makes this approach struggle to achieve high-quality 3D motion estimation shown in Fig. 1. Dai *et al.* [7] leverage stereo frames, aligned foreground moving objects, and pre-computed ego-motion to learn object-wise rigid motion. However, the effectiveness of the approach strongly relies on object tracking and stereo image pairs, which is inflexible and complicated. Moreover, the approach only considers rigid object-wise motion and fails to produce high-fidelity 3D motion for non-rigid objects such as pedestrians.

In this paper, we propose a self-supervised joint 3D motion and depth estimation system with 3D object-wise motion disentanglement, namely DO3D, to resolve these challenges. Our system contains two major components which separately predict scene depth and 3D motion. Given depth and esti-

*Authors contribute equally to this work.
[1]Xiuzhe Wu, Qihao Huang and Xiaojuan Qi are with The University of Hong Kong. {xzwu, xjqi}@eee.hku.hk, and qihao.huang@connect.hku.hk

mated 3D motion, the 3D scene is projected to a new camera view to synthesize a video frame for self-supervised training.

The core component of our approach is a new 3D motion disentanglement module. The module first predicts camera ego-motion which is shared by all pixels in the scene. Further, to model the complicated 3D (non)-rigid object motion, we formulate the 3D object-wise motion as the composition of a global object-wise 6-DoF rigid transformation and a 3D pixel-wise motion deformation. The pixel-wise deformation refines the results and produces high-quality motion for non-rigid objects such as cyclists and pedestrians. Our formulation is inspired by the observation that the motion of many objects in outdoor scenes is globally rigid, *e.g*. cars, with small local adjustments, *e.g*. pedestrian movements as shown in Fig. 5. We use [12] to produce object instances, the mask map is shown in Fig. 1. Thanks to our formulation, the proposed approach can produce high-quality 3D object-wise motion and depth shown in Fig. 1.

To evaluate the effectiveness of our approach, especially in dynamic scenes, we conduct experiments on two outdoor autonomous driving datasets – KITTI and DrivingStereo [13]. Compared with KITTI, DrivingStereo contains more dynamic scenes. Our method outperforms prior approaches in optical flow estimation and is on par with them on depth estimation, in both KITTI split and the new DrivingStereo subset. Moreover, we achieve much better 3D motion estimation results.

Our major contributions are summarized as below:

1) We propose a unified self-supervised framework to learn object-wise 3D motion and dense scene depth from monocular videos.
2) We present a new 3D motion estimation method with disentanglement to predict camera ego-motion, object-wise rigid motion and non-rigid deformation, exploiting real-world motion constraints.
3) Quantitative and qualitative results on two driving datasets show the superiority of our approach, especially in highly dynamic scenarios. We also put forward a new metric 3D endpoint error (3D EPE) for 3D motion/scene flow evaluation.

II. Related Work

a) Classical Geometry: Pioneer works of well-established multi-view geometry [14] date back to the concept of structure from motion (SfM) and stereo matching based on visual correspondence. Specifically, structured light [15], ToF [16], LiDAR [17], and stereo [15], [16] cameras are practically deployed to obtain relative scene geometry characteristics such as disparity and depth. Classical SfM relies on the assumption of the static scene and rigid motion. Thus, it perform poorly on highly dynamic or non-rigid scenarios. Besides, the high cost of LiDAR devices, the sparsity of depth, and the cumbersome calibration of the stereo cameras make them hard to be widely adopted. Recently, Luo *et al.* [18] utilize learning-based prior and leverage conventional SfM to obtain consistent video depth estimation. In contrast to classical geometry methods, our approach is trained in an end-to-end self-supervised manner, able to capture the 3D motion of dynamic objects.

b) Self-supervised Monocular Depth: The framework of self-supervised depth estimation is firstly proposed by Garg *et al.* [19] from stereoscopic videos and further extended into monocular fashion by Zhou *et al.* [10]. These learning-based approaches utilize large scale unlabeled streaming frames to jointly estimate scene depth, camera ego-motion based on novel view synthesis. However, one of the vital drawbacks is that they are incapable of predicting the accurate depth of moving objects robustly. Recent attempts such as GeoNet [11], Monodepth2 [5] try to minimize the gap by incorporating residual non-rigid optical flow and auto-masking for moving objects separately. In the meantime, other promising strategies excavate high-level semantic information followed by PackNet-SG [20] and feature-metric consistency followed by FeatDepth [21]. Our approach, based on Monodepth2 [5], aims to acquire the moving objects' depth by modeling their 3D motion physically. Our approach can serve as an extra module in current SOTA solutions for depth estimation to further boost performance.

c) Dynamic Objects Motion: To effectively model the 3D motion of dynamic objects, Ranftl *et al.* [22] and Kumar *et al.* [23] introduce piece-wise rigid approaches based on super-pixel, each super-pixel shares same rigid motion parameters, which may suffer from conflict motion prediction at the super-pixel boundaries. A few recent methods [6], [7], [8] attempt to estimation motion in self-supervised learning by disentangling the camera and object motion. GeoNet [11] fits the residual object motion from the viewpoint of 2D by flow consistency. It's still difficult to infer the explicit 3D motion pattern from the residual flow. Others propose to estimate each rigid motion then synthesize foreground moving objects apart from background for photo-metric measurements. Moreover, [6] only works on the stereo set, and [8] still exploits constraints from left-right view and offline alignment of objects' masks across frames. Dai *et al.* [7] pre-compute the ego-motion using off-line visual odometry, and utilize left-right photometric consistency. Also, [7] lose background context after segmenting foreground objects only. Most recently, Li *et al.* [9] alleviate this problem by predicting an implicit object motion map. In contrast to all the above approaches, our method is purely based on monocular frames and doesn't require the alignment of objects' masks. Besides, none of them consider modeling the deformation of non-rigid objects, while our decomposed non-rigid module is designed to predict pixel-wise deformations as a motion compensation.

III. Methods

In this section, we will first introduce preliminaries about scene geometry. Then, we will present an overview of our framework. Next, we will introduce different components and elaborate on details about object-wise motion modeling with rigid and non-rigid motion composition. Our loss functions for self-supervised learning are summarized in the final section.

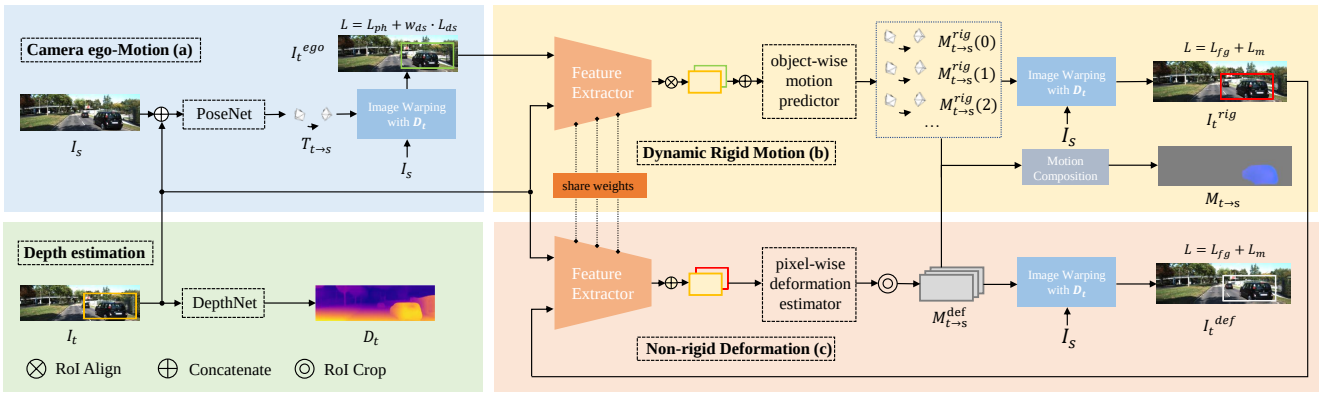

Fig. 2. Framework overview. Our system requires two consecutive video frames for camera ego-motion prediction (a). The reconstructed image I_t^{ego} and original I_t are incorporated into dynamic rigid motion (b) to learn object-wise rigid motion $M_{t\to s}^{rig}$. Further, the residual non-rigid deformation (c) exploits I_t^{rig} and I_t to recover non-rigid deformation $M_{t\to s}^{def}$. Each piece-wise training learning objective $\mathcal{L}$ is attached around the synthesized image.

A. Preliminaries about Scene Geometry

As a video is essentially the projection of a 3D scene onto different image planes, two video frames are linked together by the 3D scene geometry, camera ego-motion, and dynamic object motion. We aim to utilize the geometric relationships to recover the underlying 3D scene geometry and motion from monocular videos. The scene geometry model will be briefly introduced below.

Here, we represent two video frames as the source video frame I_s, $s \in \{t-1, t+1\}$ and the target video frame I_t. Let p_t denote the 2D homogeneous pixel grid coordinates of pixels in I_t, K represent the camera intrinsic matrix, and D_t denote the depth map. Then, the corresponding 3D location of the pixel p_t in the target camera coordinate system is

$$P_t = D_t(p_t)K^{-1}p_t. \tag{1}$$

Further, given the 3D point P_t, its location $P_{t\to s}$ at the source frame timestamp (w.r.t the source camera coordinate system) is determined by two factors: the relative camera motion (*i.e.,* rotation and translation) from the target to the source $T_{t\to s}$ and the dynamic 3D object motion $M_{t\to s}$ (w.r.t the target camera coordinate system). $M_{t\to s}$ is zero for static objects or backgrounds. The point location $P_{t\to s}$ can be obtained by combining the effects of camera motion $T_{t\to s}$ and object motion $M_{t\to s}$ as

$$P_{t\to s} = T_{t\to s}(P_t + M_{t\to s}). \tag{2}$$

Finally, the 3D point $P_{t\to s}$ will be projected to $p_{t\to s}$ at the source image plane via

$$p_{t\to s} \sim KP_{t\to s}, \tag{3}$$

where $p_{t\to s}$ is the corresponding 2D homogeneous coordinate location of target pixel p_t at the source image. Pixels in I_t and I_s are connected via $p_{t\to s}$. Given the correspondence between pixels at I_t and I_s from $p_{t\to s}$, pixel $I_t(p_t)$ can be reconstructed by $I_s(p_{t\to s})$ at the source image. As $p_{t\to s}$ is continuous, we use bi-linear sampling method to interpolate the pixel values of its four nearest-neighbors following [10], [24] to obtain $I_s(p_{t\to s})$. This is the base for constructing the self-supervised loss.

Given the original pixel coordinates p_t and the transformed one $p_{t\to s}$, the optical flow is calculated by

$$f_{t\to s} \sim p_{t\to s} - p_t. \tag{4}$$

B. Our Approach

In this section, we will elaborate on how we use the scene geometry model to design a neural system for jointly estimating depth D_t, 3D motion (*i.e.,* camera ego-motion $T_{t\to s}$ and 3D motion $M_{t\to s}$) and how the system is trained in a self-supervised manner.

1) Our Network: An overview of the system is shown in Fig. 2. The inputs to the system are two consecutive video frames I_t and I_s. The system contains two major modules for depth estimation (Fig. 2: Depth Estimation) and motion estimation (Fig. 2: (a)-(c)). .

The depth estimation module is a fully convolutional neural network – DepthNet – which processes each frame separately and produces depth maps D_t for each input frame. Our DepthNet adopts U-Net structure with skip-connections based on a ResNet18 backbone following [5].

The motion estimation module contains a pose estimation network – PoseNet – to estimate camera ego-motion $T_{t\to s}$, and an object motion prediction network – MotionNet – to produce object-wise motion $M_{t\to s}$. The PoseNet (see Fig. 2 (a)) is a convolutional neural network with fully connected layers to output the 6-DoF parameters including Pitch, Roll, Yaw and translations along three coordinates. Then, the 6-DoF parameters are converted to a transformation matrix representing the relative camera motion $T_{t\to s}$.

To relieve the difficulties of estimating complicated motion patterns for various objects, MotionNet is designed to have two components: an object-wise rigid motion predictor (see Fig. 2 (b)) and a pixel-wise motion deformation estimator (see Fig. 2 (c)). The object-wise motion predictor outputs 6-DoF global rigid transformation $M_{t\to s}^{rig}(i)$ for each object instances i (see Fig. 2 (b)) to effectively model rigid motion and produce a reasonably good global initialization for modeling non-rigid motion. For notation simplicity and consistency, we assign the estimated object-wise rigid motion

to the corresponding pixels and thus obtain a rigid motion map denoted as $M^{rig}_{t\to s}$. Further, the pixel-wise motion deformation estimator is designed to learn a pixel-wise motion deformation map $M^{def}_{t\to s}$ (see Fig. 2 (c)) by refining the rigid object-wise motion. The object motion is obtained by combining the rigid motion and deformation as

$$M_{t\to s} = M^{rig}_{t\to s}(P_t + M^{def}_{t\to s}) - P_t. \tag{5}$$

The object information is obtained by applying a pre-trained Mask R-CNN [12] model to produce the instance-wise object masks. We will show the design of MotionNet in the following section.

2) MotionNet: Given the estimated camera-ego motion $T_{t\to s}$ from PoseNet, we first obtain I^{ego}_t (see Fig. 2) by sampling the source image I_s according to $p_{t\to s}$. The process is detailed in Sec. III-A while $p_{t\to s}$ is computed with only $T_{t\to s}$ (*i.e.* $M_{t\to s}$ is zero). This process transfers the source image I_s into the camera coordinate system I^{ego}_t to eliminate the motion caused by the camera movement, facilitating the follow-up dynamic object motion estimation.

Further, as shown in Fig. 2, our MotionNet takes I_t and I^{ego}_t as inputs, and then predicts a dynamic object motion map $M_{t\to s}$ (w.r.t the target coordinate system) through predicting and combining the object-wise rigid motion $M^{rig}_{t\to s}$ (see Fig. 2) and pixel-wise motion deformation $M^{def}_{t\to s}$ (see Fig. 2) as Eq. (5), which will be detailed as below.

a) Dynamic Rigid Motion: The dynamic rigid motion component takes I^{ego}_t and I_t as inputs, and the goal is to estimate an object-wise 6-DoF rigid transformation matrix $M^{rig}_{t\to s}(i)$ for each moving object i. The object information is from an off-the-shelf instance segmentation network – Mask R-CNN [12]. The dynamic rigid motion network employs several convolutional layers to extract the feature representation separately given I^{ego}_t and I_t (see Fig. 2: Feature Extractor and Fig. 3). We use RoI Align [12] to extract object-wise features from each encoded feature map. The original object bounding box is from Mask R-CNN [12] which is enlarged by 20 pixels to incorporate more background context information. In this stage, we remove objects that has already been well reconstructed in I^{rig}_t with camera ego-motion, indicating static objects. The extracted RoI features from I_t and I^{ego}_t are concatenated (see Fig. 3) and fed into a network with several convolutional layers followed by a fully connected layer to regress 6-DoF rigid motion $M^{rig}_{t\to s}(i)$ for each object instance i .

b) Non-rigid Deformation: Given the estimated object-wise 6-DoF rigid transformation $M^{rig}_{t\to s}(i)$, the I^{rig}_t is obtained by sampling from I_s through transforming each instance. This process is achieved by computing $p_{t\to s}$ considering D_t, $T_{t\to s}$, and $M_{t\to s}$ with $M^{def}_{t\to s}$ being zero following Eq. (1) – (3). The formation of I^{rig}_t considers camera-ego motion and object-wise rigid motion, eliminating their effects and facilitating the estimation of pixel-wise non-rigid deformation. In this stage, we remove objects that have already been well reconstructed in I^{rig}_t with camera ego-motion and object-wise rigid motion, indicating static objects or rigid objects. The feature extractor in dynamic

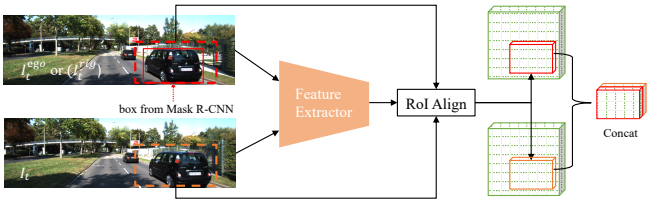

Fig. 3. RoI Align to learn object motion from context.

rigid motion estimation is reused to encode features of I_t and I^{rig}_t which are concatenated together and fed into the pixel-wise motion deformation estimator. This estimator contains several convolution and up-sample layers, and outputs a pixel-wise deformation maps with three channels representing motion in $x, y, z-$ directions.

3) Self-supervised Learning: In the following, we show how we can train the network in a self-supervised manner.

Based on the estimated depth map D_t, camera ego-motion $T_{t\to s}$, and the object motion $M_{t\to s}$, we can calculate the pixel correspondence $p_{t\to s}$ between target frame and source frame using Eq. (1) – (3). Based on this correspondence, the target frame is reconstructed by sampling the corresponding pixels from the source frame I_s. As before, the reconstructed target frame $\hat{I}_t$ can be obtained by utilizing bi-linear interpolation following [10], [24]. $\hat{I}_t$ can be I^{ego}_t, I^{rig}_t and I^{def}_t which are reconstructed frames with $T_{t\to s}$, $\{T_{t\to s}, M^{rig}_{t\to s}\}$, and $\{T_{t\to s}, M^{rig}_{t\to s}, M^{def}_{t\to s}\}$ respectively. Our self-supervised objective is built by comparing the reconstructed frame $\hat{I}_t$ with the observed frame I_t as shown in Fig. 2.

We use the photo-metric reconstruction error loss as Eq. (6) to measure the discrepancy between $\hat{I}_t$ and I_t. It encourages the network to learn to estimate D_t, $T_{t\to s}$ and $M_{t\to s}$ ($M^{rig}_{t\to s}(i)$ for all instances i and $M^{def}_{t\to s}$) that can produce I_t given I_s.

$$\mathcal{L}_{ph} = \frac{\alpha}{2}(1 - \text{SSIM}(\hat{I}_t, I_t)) + (1-\alpha)\|\hat{I}_t - I_t\|_1, \tag{6}$$

where α is a hyper-parameter balancing the SSIM [25] term and l_1 pixel-wise difference.

Further, we also incorporate a smoothness loss $\mathcal{L}_{ds}$ on the produced depth map to encourage local smoothness:

$$\mathcal{L}_{ds} = |\partial_x D^*_t| e^{-|\partial_x I_t|} + |\partial_y D^*_t| e^{-|\partial_y I_t|}, \tag{7}$$

where $D^*_t = D_t/\overline{D_t}$ is the mean-normalized inverse depth map following [5].

MotionNet focuses on object-wise motion estimation, hence we define a foreground loss as:

$$\mathcal{L}_{fg} = m_t \cdot (\mathcal{L}_{ph} + \beta \cdot \mathcal{L}_{ds}), \tag{8}$$

where m_t is the foreground mask in the target frame. β is set to 0.001 in our paper, similar to [5].

We also introduce a mask loss to regularize the reconstruction in the semantic space. The mask loss $\mathcal{L}_m$ is defined as:

$$\mathcal{L}_m = 1 - IoU(\hat{m}, m_t), \tag{9}$$

where $\hat{m}_t$ represents the reconstructed semantic mask accroding to the estimated geometric model, and $IoU(\cdot,\cdot)$

computes Intersection over Union (IoU) between $\hat{m}_t$ and the mask m_t of frame t. Finally, $\mathcal{L}_{ph}$, $\mathcal{L}_{ds}$, $\mathcal{L}_{fg}$ and $\mathcal{L}_m$ are combined to be the overall loss:

$$\mathcal{L} = \omega_{ph} \cdot \mathcal{L}_{ph} + \omega_{ds} \cdot \mathcal{L}_{ds} + \omega_{fg} \cdot \mathcal{L}_{fg} + \omega_m \cdot \mathcal{L}_m, \quad (10)$$

where hyper-parameters are $\omega^* = [\omega_{ph}, \omega_{ds}, \omega_{fg}, \omega_m]$ including the loss weight of the photo-metric loss, foreground loss and mask loss respectively.

The whole framework is trained in a piece-wise manner: 1) Train the DepthNet and PoseNet using I_t^{ego} and I^t, where $\omega^* = [1, 0.001, 0, 0]$; 2) Fix DepthNet and PoseNet from stage 1), and train the object-wise rigid motion estimator together with feature extractor using I_t^{rig} and I_t where $\omega^* = [0, 0, 1, 1]$; 3) Fix DepthNet, PoseNet, feature extractor, and object-wise rigid motion estimator, and train pixel-wise motion deformation estimator using I_t^{rig} and I_t where $\omega^* = [0, 0, 1, 1]$. In training the MotionNet, we aim to estimate motion for dynamic objects at this stage and we hence only considers object regions. As all the components are differentiable, the network can be optimized by minimizing the self-supervised loss.

IV. Experiments

A. Implementation Details

The entire network is trained on a single RTX 2080 Ti GPU with Adam [26] optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$), and all RGB inputs are resized to 640×192. We follow Monodepth2 [5] to construct DepthNet and PoseNet which are trained with batch size 12 and learning rate 10^{-4}. In training the MotionNet, the learning rate is 10^{-5} with batch-size 12.

B. Datasets

We conduct experiments on the KITTI 2015 dataset [17] and DrivingStereo dataset [13]. The KITTI 2015 dataset consists of image sequences for 200 driving scenes at 10 frames per second. The image resolution is around 375×1242. Nevertheless, motion patterns and dynamic objects are still limited in the KITTI dataset. Thus, we introduce the DrivingStereo dataset that covers highly dense and dynamic scenarios. The frame resolution in the DrivingStereo dataset is around 400×881.

1) KITTI Eigen Depth Split: The network training and the depth estimation (DepthNet) evaluation is conducted on the Eigen depth split [27] on KITTI 2015. We follow [5] and remove the invalid static frames are removed in pre-processing step. Finally, the training, validation and test set contains $39,810$, $4,424$, and 673 images respectively. As the Eigen split does not provide optical flow ground truth for motion evaluation, we apply the state-of-the-art optical flow estimation approach RAFT [1] to produce the pseudo ground truth optical flow for evaluating motion.

2) KITTI Optical Flow Split: To better evaluate the motion with optical flow ground truth, we also adopt the KITTI 2015 flow split and employ its training split (200 images) to validate our model in optical flow and 3D motion estimation.

3) DrivingStereo Train/Test Split: As existing methods are not evaluated on the DrivingStereo dataset, we randomly split the dataset into two subsets – train split and test split. The train and test split consists of $39,805$ and 150 images respectively. Due to the lack of optical flow ground truth, we also introduce RAFT [1] to predict dense pseudo ground truth for motion evaluation.

C. Evaluation metrics

1) Depth: We follow the depth evaluation metrics in [28]: absolute relative error (Abs Rel), square relative error (Sq Rel), root mean square error (RMSE) and root mean square logarithmic error (RMSE log). To eliminate the influence of scale which is missing in monocular scenario, Godard *et al.* [28] scale the predicted depth frame by the median of the ground truth in evaluation. Besides, $\delta < \sigma$ represents the percentage of depth predictions whose ratio with respect to the ground truth and inverse ratio with respect to the ground truth are lower than σ. In addition, the predicted monocular depth is capped to 80m during evaluation following [5].

2) Optical Flow: For the optical flow estimation task, we use the average endpoint error (EPE) metric [29]. To better assess the motion estimation of objects (our MotionNet), we divide an image into background (bg) and foreground (fg) regions, and evaluate our model in bg and fg regions separately. Foreground and background are splitted by instance masks from Mask R-CNN [12].

3) 3D Motion/Scene Flow: To better evaluate the performance of 3D motion estimation, we propose a new scene flow metric: **3D EPE**, which directly measures the distance of each point's 3D movements in two consecutive frames. This is more suitable in comparison with the provided – bad pixel percentage (BPP) – for 3D scene flow evaluation, which calculates the percentage of pixels with both good depth estimation and optical flow. In contrast, the 3D EPE directly measures the quality of estimated 3D motion.

We generate 3D motion ground truth using the optical flow and depth ground truth provided by the KITTI dataset. Pixel-wise correspondence is obtained via optical flow, and the 3D point location can be derived with the provided depth ground truth. The camera-ego motion can be robustly estimated via [4]. The source pixel coordinates are converted into the target coordinate system based on the global camera ego-motion and the depth ground truth. Then, we can obtain the 3D EPE pseudo ground truth through subtracting the point locations at the source time-step from the corresponding point location at the target time-step.

D. Main results

In the following, results are compared in terms of motion and monocular depth estimation. To evaluate motion, we adopt the optical flow estimation task (see Sec. IV-D.1), measuring the 2D projection of the estimated 3D motion, and the 3D motion estimation, assessing the 3D EPE of the estimated 3D motion ((see Sec. IV-D.1). The depth evaluation performance is presented in Sec. IV-D.2. We perform model adaptivity and generalization analysis in Sec. IV-D.2

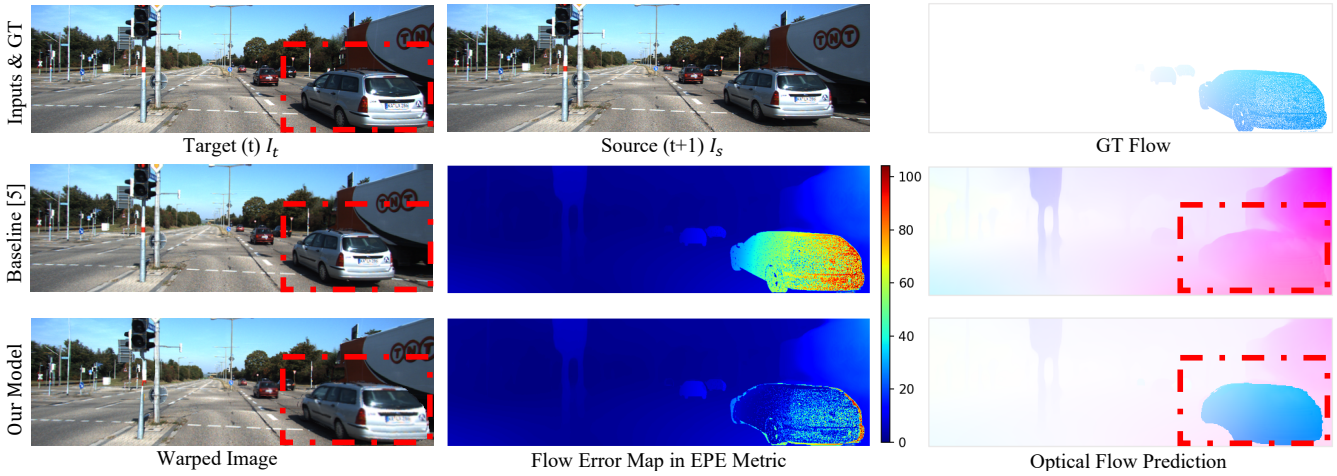

Fig. 4. Detailed qualitative comparisons with our baseline Monodepth2. Our method revises the dynamic foreground optical flow through motion decomposition modules. Red dash boxes illustrate the major differences between our novel view synthesis I^{def} and baseline I^{Mono2} through the predicted optical flow separately. GT flow provided by KITTI split is visualized on the upper right side. Best viewed in color.

1) Optical Flow Evaluation: We conduct both quantitative and qualitative comparisons on the KITTI and DrivingStereo datasets to validate the performance on 3D motion estimation. We don't directly compare with methods based on pixel matching [30], [31] in that the optical flow based on matching can not reflect the 3D motion modeling capability which is our focus. Therefore, we compare with representative works in geometry based optical flow estimation approach – GeoNet [11] and Monodepth2 [5]. The results on the KITTI dataset and the DrivingStereo dataset are shown in Tab. I, Tab. II and Tab. V.

"fg" and "bg" represent the evaluation on foreground and background regions, "Noc" stands for non-occluded regions, and "Occ" indicates occluded regions. GeoNet [11] rectifies rigid flow with a ResFlowNet. Thus, we compare two versions of GeoNet, which are GeoNet with or without ResFlowNet, labelled by $\text{GeoNet}^{\text{res}}$ and $\text{GeoNet}^{\text{rig}}$ respectively. We use "Monodepth2" as our baseline. DO3D^{rig} represents our model with object rigid motion revised and DO3D^{def} is our full model.

a) Evaluation on KITTI Dataset: On the optical flow split: 1) benefited from our MotionNet which estimates 3D object motion, our overall model DO3D^{def} significantly outperforms all other methods in foreground object regions; 2) our model also achieves the best overall performance in both occluded and non-occluded regions, demonstrating the effectiveness of our approach; 3) compared with our baseline – Monodepth2, our full model reduces the EPE in foreground regions almost by 50%, *e.g.* 14.86 vs 30.33 in Noc regions and 15.87 vs 31.05 in Occ regions; and different components of our model consistently reduce the error for foreground objects, *e.g.* the original "fg" EPE in non-occluded regions is reduced from 30.33 to 16.33 with the rigid-motion estimation component, which is further reduced to 14.86 with pixel-wise motion deformation component, and the trend is consistent for occluded regions.

Quantitative results on the Eigen-split are shown in Tab. II.

Methods	Noc ↓			Occ ↓		
	bg	**fg**	all	bg	**fg**	all
$\text{GeoNet}^{\text{res}}$ [11]	3.77	19.07	8.17	6.76	20.23	10.86
$\text{GeoNet}^{\text{rig}}$ [11]	5.98	28.06	11.47	8.24	29.00	13.40
Monodepth2 [5]	4.11	30.33	10.95	5.48	31.05	11.85
DO3D^{rig}	4.10	16.33	7.29	5.44	17.18	8.53
DO3D^{def}	4.15	**14.86**	6.86	5.49	**15.87**	8.18

TABLE I
OPTICAL FLOW ABLATION STUDY OF KITTI 2015 OPTICAL FLOW TRAINING SPLIT.

Our method again achieves the best overall performance which demonstrates especially superior performance in the foreground regions.

Qualitative results are shown in Fig. 4. We can see that optical flow of the car in red dash box is fully revised with our MotionNet. The error map in the middle column also shows that our predicted optical flow from 3D motion is much more accurate than the baseline. Finally, we visualize the learning 3D motion in Fig. 5 ($M_{t\to s}$) which demonstrates the movement of the cyclist and the car. The 3D motion map is visualized as a color image where R,G,B corresponds to motion along x, y and z directions respectively. Our intermediate reconstructed images are also shown in Fig. 5. With the pixel-wise motion deformation estimator, the reconstructed quality for cyclist has been improved.

b) Evaluation on DrivingStereo Dataset: To further manifest that approach performs well in complex and dynamic scenes, we use DrivingStereo dataset to train both our proposed models and the baseline Monodepth2 from scratch for fair comparisons. We believe that it's necessary to test effectiveness of models in a dynamic dataset with diversified motion patterns. According to the experimental results in Tab. V, our method performs significantly better than the baseline (*e.g.* "fg" EPE is reduced to 24.03 and overall EPE

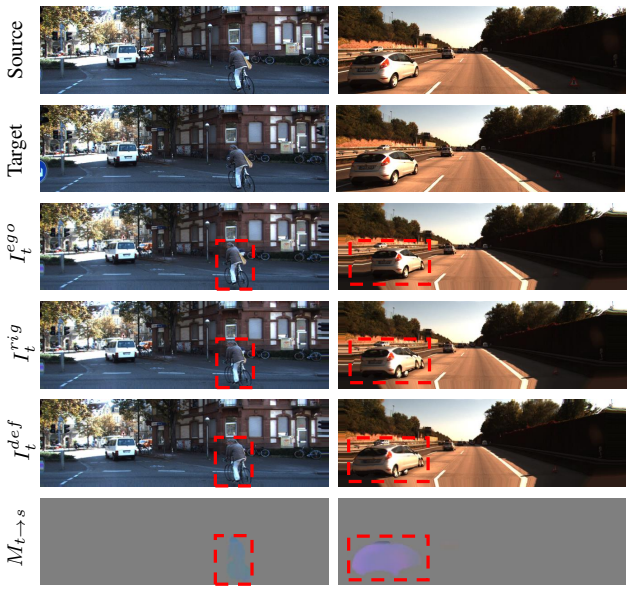

Fig. 5. Qualitative comparisons between DO3D^{rig} and DO3D^{def}.

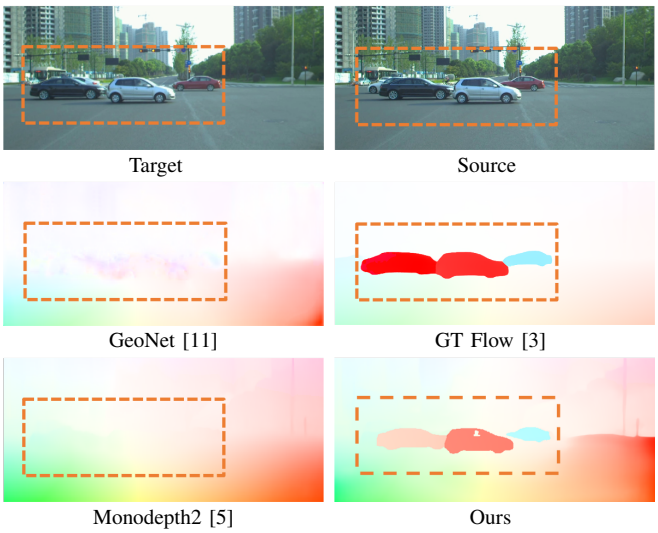

Fig. 6. Qualitative results of the DrivingStereo split to test the generalization ability in highly dynamic cases without finetuning. Our motion decomposition module revises the foreground optical flow of moving objects explicitly. Best viewed in color.

Methods	KITTI Depth		DrivingStereo	
	bg ↓	**fg ↓**	bg ↓	**fg ↓**
$\text{GeoNet}^{\text{res}}$ [11]	4.19	8.73	19.58	36.38
$\text{GeoNet}^{\text{rig}}$ [11]	6.82	12.55	24.78	43.42
Monodepth2 [5]	4.38	9.32	20.39	49.90
DO3D^{rig}	4.37	**8.26**	20.28	38.63
DO3D^{def}	4.40	8.41	20.34	**34.81**

TABLE II

OPTICAL FLOW ABLATION STUDY OF KITTI DEPTH SPLIT (673 IMAGES) AND DRIVINGSTEREO IN EPE METRIC. FOR THE DRIVINGSTEREO RESULT, MODELS ARE ALL TRAINED ON KITTI AND DIRECTLY EVALUATED ON THE DRIVINGSTEREO DATASET.

Methods	Monodepth2 [5]	DO3D^{rig}	DO3D^{def}
3D EPE ↓	3.062	2.296	**2.290**

TABLE III

3D MOTION ABLATION STUDY OF KITTI 2015 OPTICAL FLOW TRAINING SPLIT IN 3D EPE METRIC.

is reduced to 23.60). The encouraging results demonstrate that our model is able to learn 3D complicated motion in fast-moving and complicated driving scenarios. This further verifies the effectiveness of our proposed MotionNet.

2) Monocular Depth Evaluation: We evaluate the depth estimation performance of DepthNet on the KITTI dataset and the DrivingStereo dataset. The whole model is finetuned end-to-end. Results are shown in Tab. IV (KITTI) and Tab. V. Although the focus of our approach is to model 3D motion, the full model still improves the performance of the baseline Monodepth2 [5] benefited from a better model representing the underlying geometric rules. Qualitative results shown in Fig. 7 also demonstrate the superiority of our proposed method, especially at the object boundaries. Depth estimation can also benefit from well-learned 3D movements.

3) 3D Motion Evaluation: Quantitative results of 3D EPE are shown in Tab. III. In comparison with the baseline Monodepth2, our proposed model significantly reduces the 3D EPE error (DO3D^{def}) by around 25%. Fig. 1 shows the qualitative results of the predicted 3D motion ($M_{t\to s}$): the car is moving along the z-axis with small motion along the x-axis. All results demonstrate that our full model DO3D^{def} performs well in estimating 3D motion.

4) More Analysis on Generalization:

a) Train on KITTI and Evaluate on DrivingStereo: We use the model trained on KITTI to evaluate the DrivingStereo sequences. The optical flow results are shown in Tab. II (DrivingStereo) and the depth estimation results are shown in Tab. IV (DrivingStereo). Our approach also achieves SOTA performance in this high dynamic cross-dataset evaluation setting. Qualitative results are shown in Fig. 6. GeoNet and Monodepth2 both fail in learning object motion (*e.g.* cars) in this new dataset. However, our model still succeeds in estimating accurate motion and is able to preserve the sharp details at the object boundaries. Our model outperforms the baseline in the depth estimation. The above demonstrates that our model also exhibits good generalization abilities.

V. CONCLUSION

We have presented a self-supervised framework to estimate the dynamic motion of moving objects and monocular dense scene depth jointly. The core component is MotionNet which combines object-wise rigid motion and pixel-wise motion deformation to represent the complicated 3D object motion. Our extensive experiments on optical flow and depth estimation demonstrate the superiority of our model. For the future direction, we would like to combine temporal information to derive a consistent geometry model for 3D scene reconstruction.

Datasets	Methods	Size	Arch.	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
KITTI Eigen Depth Split	GeoNet* [11]	418×126	ResNet50	0.149	1.060	5.567	0.226	0.796	0.935	0.975
	Li *et al.* [9]	416×128	ResNet50	0.130	0.950	5.138	0.209	0.843	0.948	0.978
	Lee *et al.* [8]	832×256	ResNet50	0.124	1.009	5.176	0.208	0.839	0.942	0.980
	Monodepth2 [5]	640×192	ResNet18	0.115	0.903	4.863	**0.193**	**0.877**	**0.959**	**0.981**
	$\text{DO3D}^{\text{def}\dagger}$	640×192	ResNet18	**0.114**	**0.890**	**4.841**	**0.193**	**0.877**	**0.959**	**0.981**
DrivingStereo Split	Monodepth2* [5]	640×192	ResNet18	0.157	1.976	7.814	0.217	0.801	0.942	0.979
	$\text{DO3D}^{\text{def}\dagger}$	640×192	ResNet18	**0.156**	**1.935**	**7.733**	**0.214**	**0.804**	**0.944**	**0.980**

TABLE IV

QUANTITATIVE RESULTS OF MONOCULAR DEPTH ESTIMATION ON KITTI 2015 [17] EIGEN SPLIT AND DRIVINGSTEREO [13] SPLIT.

RGB Monodepth2 Ours

Fig. 7. Depth visualization of Monodepth2 and our method. Best viewed in color.

Methods	Optical Flow Estimation		
	bg ↓	**fg** ↓	all ↓
Monodepth2	24.40	66.63	30.02
DO3D^{rig}	22.50	24.68	23.48
DO3D^{def}	22.65	**24.03**	**23.60**

TABLE V

ADDITIONAL COMPARISONS ON DRIVINGSTEREO OPTICAL FLOW EVALUATION. ALL THE MODELS ARE **RE-TRAINED FROM SCRATCH**.

REFERENCES

[1] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *ECCV*, 2020.

[2] Z. Teed and J. Deng, "Deepv2d: Video to depth with differentiable structure from motion," in *ICLR*, 2020.

[3] Z. Teed and J. Deng, "Raft-3d: Scene flow using rigid-motion embeddings," in *arXiv:2012.00726*, 2020.

[4] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, "Towards better generalization: Joint depth-pose learning without posenet," in *CVPR*, 2020, pp. 9151–9161.

[5] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, 2019.

[6] Z. Cao, A. Kar, C. Hane, and J. Malik, "Learning independent object motion from unlabelled stereoscopic videos," in *CVPR*, 2019.

[7] Q. Dai, V. Patil, S. Hecker, D. Dai, L. Van Gool, and K. Schindler, "Self-supervised object motion and depth estimation from video," in *CVPRW*, 2020.

[8] S. Lee, S. Im, S. Lin, and I. S. Kweon, "Instance-wise depth and motion learning from monocular videos," in *NeurIPSw*, 2020.

[9] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," in *CoRL*, 2020.

[10] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.

[11] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, 2018.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[13] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *CVPR*, 2019.

[14] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[15] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," in *CVPRW*, 2017.

[16] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (tof) cameras: A survey," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1917–1926, 2011.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[18] X. Luo, J. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," in *SIGGRAPH*. ACM, 2020.

[19] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *ECCV*, 2016.

[20] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically-guided representation learning for self-supervised monocular depth," in *ICLR*, 2020.

[21] C. Shu, K. Yu, Z. Duan, and K. Yang, "Feature-metric loss for self-supervised learning of depth and egomotion," in *ECCV*, 2020.

[22] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *CVPR*, 2016.

[23] S. Kumar, R. S. Ghorakavi, Y. Dai, and H. Li, "Dense depth estimation of a complex dynamic scene without explicit 3d motion estimation," in *CVPR*, 2019.

[24] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *NeurIPS*, 2015.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.

[27] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015.

[28] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017, pp. 270–279.

[29] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International journal of computer vision*, vol. 92, no. 1, pp. 1–31, 2011.

[30] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *CVPR*, 2019.

[31] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *PAMI*, vol. 42, no. 10, pp. 2624–2641, 2019.