Attentive and Contrastive Learning for Joint Depth and Motion Field Estimation

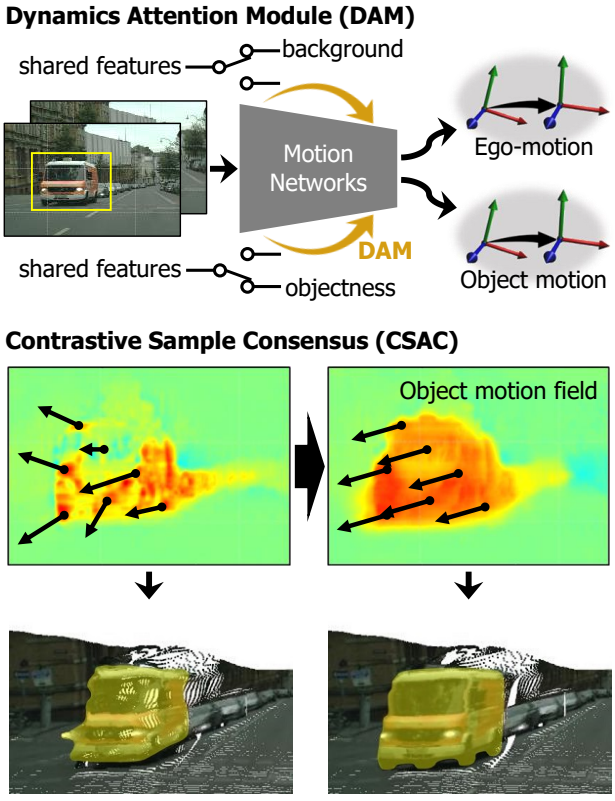
Seokju Lee Francois Rameau Fei Pan In So Kweon
Korea Advanced Institute of Science and Technology (KAIST)
{seokju91,rameau.fr,feipan664}@gmail.com, iskweon77@kaist.ac.kr

Abstract

Estimating the motion of the camera together with the 3D structure of the scene from a monocular vision system is a complex task that often relies on the so-called scene rigidity assumption. When observing a dynamic environment, this assumption is violated which leads to an ambiguity between the ego-motion of the camera and the motion of the objects. To solve this problem, we present a self-supervised learning framework for 3D object motion field estimation from monocular videos. Our contributions are two-fold. First, we propose a two-stage projection pipeline to explicitly disentangle the camera ego-motion and the object motions with dynamics attention module, called DAM. Specifically, we design an integrated motion model that estimates the motion of the camera and object in the first and second warping stages, respectively, controlled by the attention module through a shared motion encoder. Second, we propose an object motion field estimation through contrastive sample consensus, called CSAC, taking advantage of weak semantic prior (bounding box from an object detector) and geometric constraints (each object respects the rigid body motion model). Experiments on KITTI, Cityscapes, and Waymo Open Dataset demonstrate the relevance of our approach and show that our method outperforms state-of-the-art algorithms for the tasks of self-supervised monocular depth estimation, object motion segmentation, monocular scene flow estimation, and visual odometry.

1. Introduction

The simultaneous estimation of the camera motion and scene geometry is a fundamental research topic in 3D computer vision. Traditionally, this problem is tackled by feature-based methods [33], or direct approaches [14] that minimizes the photometric inconsistency among warped adjacent frames. A pioneering work based on deep neural network (DNN) [44] uses the photometric error map as a self-supervisory signal to jointly train a depth and a motion network. Inspired by this baseline structure, self-supervised depth, and motion learning framework has been widely studied [44, 30, 40, 32, 41] with an additional self-supervisory signal such as geometric consistency [32], optical flow [36], segmentation map [43], edge and normal map [41]. These photo-consistency-based optimization methods assume a static scene or require to mask out moving objects to disregard non-rigid motions. Such works aim at predicting the depth and ego-motion from a camera but are not suitable for



(a) Baseline (b) Ours

Figure 1. We introduce a unified motion modeling with our dynamics attention module (DAM) and a novel motion learning technique via contrastive sample consensus (CSAC). The last row shows synthesized views from the predicted depth and motion field. Compared to the baseline, our model learns the object motion fields in a more semantically plausible way, which enhances the distinction between the object and the background area.

dynamic scenes.

Recently, learning the objects' motion together with the camera's ego-motion and the depth has gain interest for dynamic scene understanding [6, 26, 19, 7, 8, 25, 12, 4, 23, 28]. We can distinguish mostly two types of approaches, namely the stereo-based and the monocular-based techniques. The stereo-based techniques [6, 26] take advantage of this sensor to disentangle the motion of static background and that of moving objects in the scene. For monocular systems, the ambiguity between the depth, ego-motion, and objects' motion becomes more intricate due to the unavailability of metric depth for each frame. Therefore, Monocular-based systems [7, 25, 28] rely on instance segmentation labels to reduce this ambiguity. Despite compelling results, the need for highly expensive human-labeled data constitute an important limitation for their deployment and reduce the interest of the self-supervised depth and motion prediction framework.

To reduce the data dependency problem and to offer more versatility, we propose a novel self-supervised learning framework for depth, camera motion, and object motion field estimation using weak semantic prior (*i.e.*, 2D object bounding boxes) as illustrated in Fig. 1. A major benefit of the proposed pipeline is that it helps to reduce the ambiguity between the camera's ego-motion and the objects' motion with cheaper data labels. The distinctive points of our approach are summarized as follows:

- ◇ We design a dynamics attention module that enables to train motion features dynamically when estimating the motion of a camera and objects through a two-stage projection. We highlight that motion features can be efficiently extracted by disentangling dynamic objects and static backgrounds through the simple mechanism of attention modules within the shared motion encoder.
- ◇ We propose a contrastive sample consensus for semantically plausible object motion field learning. Considering the rigid body characteristics of dynamic objects, we design a learning technique that effectively improves the capability to distinguish object's motion boundary.
- ◇ We show that the proposed scheme achieves favorable results in motion segmentation, monocular depth and scene flow estimation, and visual odometry on the KITTI, Cityscapes, Waymo Open Dataset.

2. Related Works

2.1. Joint Training of Depth and Motion from Monocular Videos

Zhou *et al*. [44] first propose a self-supervised depth and motion framework minimizing the photometric consistency across a monocular video. Following this publication, many improvements have been proposed [40, 32, 17, 41, 1, 10, 36, 20, 39]. Wang *et al*. [40] incorporate a second-order gradient descent-based pose refinement module, into end-to-end training. Yang *et al*. [41] introduce joint optimization of depth and motion with normal and edge information to force additional geometric loss while preserving the edges. Mahjoiurian *et al*. [32] enforce the geometric consistency across reconstructed 3D points as well as the photometric consistency. Bian *et al*. [1] and Chen *et al*. [10] also impose the depth consistency loss by comparing multiple predicted depth, but they further estimate dynamic objects' mask [1] or camera intrinsics [10]. Godard *et al*. [18] propose a minimum reprojection loss to handle occlusion robustly and a multi-scale sampling method to reduce artifacts. Ranjan *et al*. [36] introduce coordinated training frameworks composed of multiple neural networks for depth, camera motion, optical flow, and motion segmentation. Guizilini *et al*. [20] introduce a detail-preserving representation by learning representations that maximally propagate dense appearance and geometric information through 3D convolutions. Vasiljevic *et al*. [39] represent the depth with differentiable pixel-wise projection rays for learning with uncalibrated single viewpoint cameras. Recently, researches have been actively conducted to improve the performance of depth estimation in association with the semantic segmentation task. For instance, Klingner *et al*. [25] leverage semantic segmentation guidance to adaptively mask out the photometric inconsistency in dynamic scenes. Alternatively, Chen *et al*. [9] and Guizilini *et al*. [21] improve the performance of monocular depth estimation while enhancing semantic understanding by extracting features that are commonly related to semantics and geometry. They show that implicit feature learning through semantic prior knowledge can play an important role in 3D geometric perception.

2.2. Disentangling Camera and Object Motion

Disentangling local object motion from the global camera ego-motion is a key to improve the robustness of both depth and motion estimation in dynamic situations. Due to the motion ambiguity inherent to monocular-based techniques, most existing studies rely on stereo camera setup [6, 26]. This kind of system is advantageous in this context since metric scale depth (for each individual frame) coupled with semantic information (*e.g.*, 2D objects' bounding boxes [6] or segmentation labels [26]) offers privileged information to disentangle the ambiguity between the static background and the moving objects. While stereo-vision systems simplify the problem, solving this disentanglement using monocular cameras appears to be significantly more complex [19, 7, 25, 12, 28, 4]. Some of the works [7, 28] leverage instance segmentation map to estimate the motion field of individual objects. Casser *et al*. [7, 8] especially focus on designing a geometric structure in the learning process by modeling the scene and objects. Lee *et al*. [28, 27] focus on a geometrically correct

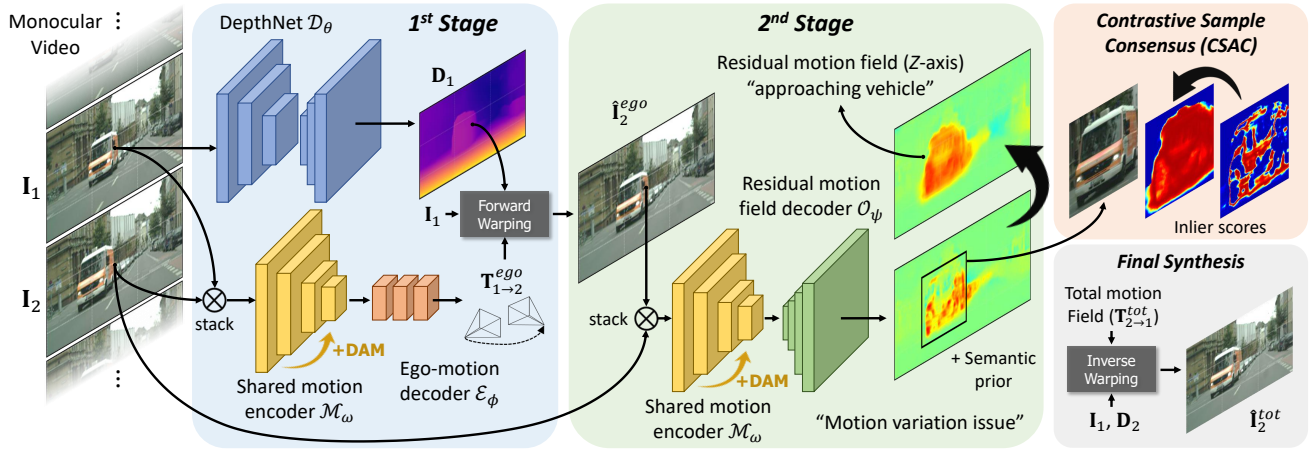


Figure 2. **Schematic overview of our self-supervised two-stage motion disentanglement.** First, we forward-warp $\mathbf{I}_1$ with the estimated camera motion and the depth map to synthesize $\hat{\mathbf{I}}^{ego}_2$. This ego-warped image is stacked with next frame, $\mathbf{I}_2$, and fed to the second stage projection to estimate the residual motion field. Finally, we generate a total composite motion field with the predicted ego-motion and residual motion field, and inverse-warp $\mathbf{I}_1$ to synthesize $\hat{\mathbf{I}}^{tot}_2$. The final synthesis is leveraged to optimize the networks as a self-supervisory signal. Our motion encoder is shared to estimate the ego-motion and the residual motion field. Each motion requires features extracted on the background and object area, respectively. This selective feature focusing is controlled by the proposed dynamics attention module (DAM). While learning the motion field, we propose contrastive sample consensus (CSAC) to solve the issue of local motion variation. Details of DAM and CSAC are elaborated in Sec. 3.2 and Sec. 3.3.

two-stage warping process that improves both photometric and geometric projection consistency in dynamic situations. Recent works attempt to disentangle the motion of objects with weaker semantic prior knowledge (*e.g.*, from pixel-level to box-level prior). Brazil *et al*. [4] learn a 3D object bounding box with their orientation and 3D confidence from a monocular video. Gordon *et al*. [19] introduce a motion field representation with bounding box information to train depth, ego-motion, and dynamic motion from uncalibrated cameras. Li *et al*. [29] extend the motion field representation with a motion sparsity loss without additional semantic prior knowledge. Gao *et al*. [15] propose attentional motion networks to adaptively focus on each object and background feature without semantic priors.

3. Methodology

We introduce a two-stage pipeline for joint depth and motion learning. Our main objective is to disentangle the camera's and objects' motion in a self-supervised manner. In this section, we present the two projection stages composing our system, and the networks: DepthNet, and MotionNet with a shared encoder and two branch decoders. Further, we detail our dynamics attention module and contrastive sample consensus for modeling camera and object motions with the semantic guidance.

3.1. Two-Stage Motion Disentanglement

The overall schematic framework of the proposed method is illustrated in Fig. 2. The self-supervision of our architecture is achieved by warping the source frame $\mathbf{I}_1$ to its adjacent target frame $\hat{\mathbf{I}}_2$, where $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ is an RGB image sampled from a monocular video. The residual error resulting from this warping is used as a training signal.

Stage 1 – depth and ego-motion: We first predict each source and target view's depth map ($\mathbf{D}_1$, $\mathbf{D}_2$) via our DepthNet $\mathcal{D}_\theta : \mathbb{R}^{3\times H\times W} \to \mathbb{R}^{1\times H\times W}$ with trainable parameters θ. By concatenating two sequential images and depth maps ($\mathbf{I}_1$, $\mathbf{D}_1$, $\mathbf{I}_2$, $\mathbf{D}_2$) as an input, our proposed motion encoder $\mathcal{M}_\omega : \mathbb{R}^{8\times H\times W} \to \mathbb{R}^{c_k\times h_k\times w_k}$ with trainable parameters ω extracts bottleneck motion features. Using the last layer's bottleneck feature as an input for the ego-motion decoder $\mathcal{E}_\phi : \mathbb{R}^{c_k\times h_k\times w_k} \to \mathbb{R}^6$ with trainable parameters ϕ, we estimate the six-dimensional (three translations and Euler angles) relative transformation vector $\mathbf{T}^{ego}_{1\to2}$ as a forward camera ego-motion. We then synthesize the ego-warped image $\hat{\mathbf{I}}^{ego}_2$ and its depth map $\hat{\mathbf{D}}^{ego}_2$ as outputs of the first stage as

$$\{\hat{\mathbf{I}}^{ego}_2, \hat{\mathbf{D}}^{ego}_2\} = \mathcal{F}_{fwd}(\mathbf{I}_1, \mathbf{D}_1, \mathbf{T}^{ego}_{1\to2}, \mathbf{K}), \tag{1}$$

where $\mathcal{F}_{fwd}$ is a forward projection function proposed in [28], and $\mathbf{K} \in \mathbb{R}^{3\times3}$ is a given camera intrinsic matrix. We postulate that this ego-warped image and its projected depth are structurally aligned to the target view except for occluded and disoccluded regions if there are no moving objects.

Stage 2 – residual motion field: In the second projection stage, we residually predict a motion translation field. Since we synthesize the target view with the predicted camera motion, we conjecture that the misaligned regions are caused from local object motions. Using this clue, we model the object motion as a residual motion field $\mathbf{T}^{res}$ using

our motion encoder and a residual motion field decoder $\mathcal{O}_\psi : \mathbb{R}^{c_k \times h_k \times w_k} \rightarrow \mathbb{R}^{3 \times H \times W}$ with trainable parameters ψ. Specifically, we concatenate the outputs from the first projection stage and target frame's image and depth ($\hat{\mathbf{I}}_2^{ego}$, $\hat{\mathbf{D}}_2^{ego}$, $\mathbf{I}_2$, $\mathbf{D}_2$) as an input for the motion encoder. We again feed this to our motion encoder, and its output features are fed to the residual motion field decoder. This local object motion is represented only with a 3D translation field to reduce the rotation ambiguity from the camera. Finally, we compose the total motion field $\mathbf{T}^{tot} \in \mathbb{R}^{6 \times H \times W}$ from the predicted ego-motion[1] and residual motion field through pixel-wise matrix multiplication. Given this total motion field, source image, and target depth map, we synthesize the final target image and its depth map as

$$\{\hat{\mathbf{I}}_2^{tot}, \hat{\mathbf{D}}_2^{tot}\} = \mathcal{F}_{inv}(\mathbf{I}_1, \mathbf{D}_2, \mathbf{T}_{2\rightarrow1}^{tot}, \mathbf{K}), \tag{2}$$

where $\mathcal{F}_{inv}$ is our pixel-wise inverse projection function. We optimize the whole frameworks by minimizing the photometric and geometric errors between $\{\mathbf{I}_2, \mathbf{D}_2\}$ and $\{\hat{\mathbf{I}}_2^{tot}, \hat{\mathbf{D}}_2^{tot}\}$. The loss functions will be discussed in Sec. 3.4.

3.2. Dynamics Attention Module

Since the camera motion and the residual motion field estimation are two complementary tasks, we postulate that using a shared encoder for these two tasks would improve their efficiency and the motion feature representation. To maximize this effect, we propose *dynamics attention module (DAM)* as described in Fig. 3. The encoding part of our unified motion networks is based on the ResNet-18 [22] structure. As an input for the networks, we concatenate two consecutive images and depth maps, which has eight channels in total. Motion features are learned while passing through each residual layer of the encoder. In this process, we attach DAM after the residual layers (ResLayer-2, -3, and -4) to selectively extract the ego-motion and residual motion features. We design DAM by referring to the generic self-attention structure that is transformed after context modeling introduced in GCNet [5]. To be specific, we first squeeze the channel dimension with two 1×1 *conv* layers and generate a spatial attention map via *softmax* along the spatial dimension. This spatial attention is multiplied to the input feature, which represents a global attention pooling for context embedding. Then, with the reduction ratio set to $r = 4$, the pooled feature is transformed with a bottleneck of two 1×1 *conv* layers. Finally, the transformed feature is added to the input feature in element-wise for feature fusion. This self-attention module is applied to each ego-motion and residual motion feature. If the motion feature is extracted to predict the camera motion, the ego-motion attention module is activated, and if estimating the residual motion field, we operate the residual motion

[1]Since the final warping is inverse direction, the direction of ego-motion also should be inverted, which is estimated by feeding a reverse-ordered input ($\mathbf{I}_2$, $\mathbf{D}_2$, $\mathbf{I}_1$, $\mathbf{D}_1$) to the ego-motion networks.

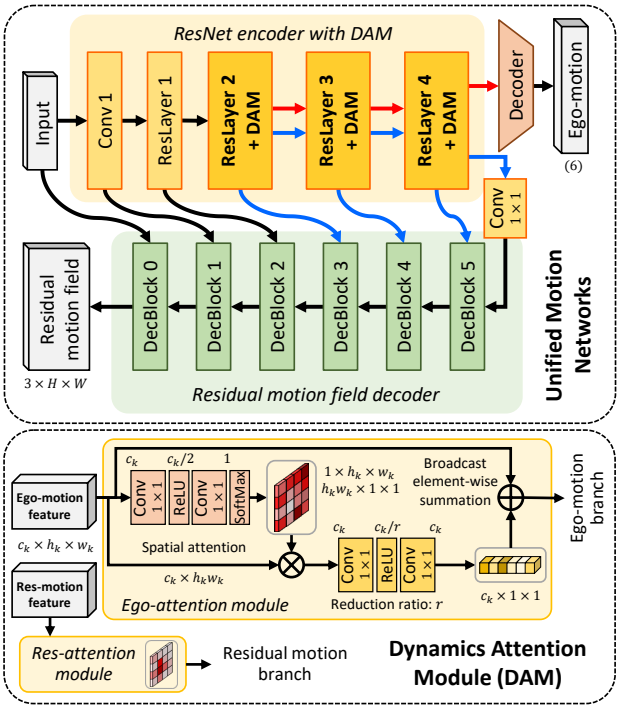


Figure 3. **Overall structure of unified motion networks with dynamics attention module (DAM).** Our motion networks consist of a shared motion encoder with an attention module for each residual layer (ResLayer), and two motion prediction branches: ego-motion decoder and residual motion (res-motion) field decoder. DAM has two self-attention modules to adaptively focus on background and local dynamic objects.

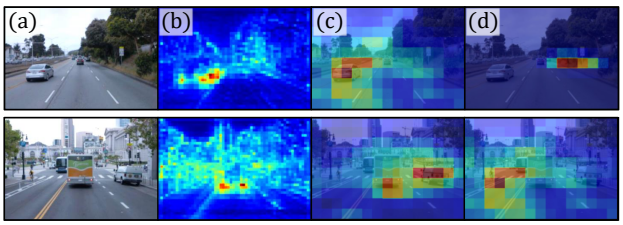


Figure 4. **Qualitative results of attention maps.** (a) Input target images from Waymo Open Dataset. (b) Aggregated residual motion attentions on ResLayer-2 and -3 (mid-level). (c) and (d) residual motion and ego-motion attentions on ResLayer-4 (high-level, visually overlaid on the input image).

attention module. Since the ego-motion and residual motion are in complementary relation, DAM enables selective motion focusing for each motion decoding as demonstrated in Fig. 4. The ego-motion decoder is designed with four *conv* layers to process the output feature of the last encoding layer. The residual motion field decoder is composed of six decoding blocks (DecBlock). Each decoding block aggregates output features of the bottom block and the encoding layer.

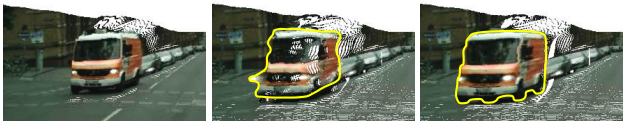


Figure 5. **Left**: 3D point cloud visualization of $\mathbf{I}_1$ and $\mathbf{D}_1$. **Middle**: a motion variation issue occurs on view synthesis ($\mathbf{I}_1 \rightarrow \mathbf{I}_2$) with the baseline training. The networks try to minimize the photometric errors on headlight of the bus, while motions on homogeneous regions are not activated well. **Right**: Training the motion field with contrastive sample consensus (CSAC) to regularize the motion vectors for every pixel on the moving object. Yellow line indicates boundary of the object.

3.3. Contrastive Sample Consensus

Motion variation issue While learning the residual motion field, our motion networks are trained to minimize the local errors triggered by an individual object motion. However, due to the limitation of self-supervisory optimization by photometric consistency, motion fluctuation[2] occurs during training. As shown in Fig. 5, the discriminative regions, *e.g.*, headlights, tend to be inferred with high motion response, while the homogeneous regions, *e.g.*, windows, have relatively small motion. To mitigate this issue, we propose *contrastive sample consensus (CSAC)* to boost the motion consistency.

Motion regularization via CSAC Given a semantic prior as a 2D object box and its geometric prior (depth), we design a differentiable regularization module combining traditional random sampling and recent deep learning techniques. This regularization relies on two assumptions:

Assumption 1 (geometric prior) *Each 2D detection box contains a potentially movable object, and it belongs to the foreground region.*

Assumption 2 (semantic prior) *The motion vectors in each box are distributed into two groups (background: small, object: large), and those belonging to the object group should converge to a single motion vector considering its rigidity under a short time period.*

From these assumptions, we train the motions from the foreground and background by *motion-repulsive* embedding as introduced in Algorithm 1. In this algorithm, first we estimate the initial foreground mask from our predicted depth map on the detection box using [34] on the depth values (line 3). From this initial binary segmentation mask, we iteratively

[2]Previous works [19, 29] have alleviated this issue by applying motion smoothness term. This is fair, but only nearby motion vectors are regularized. On the other hands, our regularization method plays with the distribution of motion vectors. Considering the rigidity of the moving objects, *e.g.*, mostly vehicles on traffic roads, we postulate that boosting consistency over a set of whole motion vectors for each object is more helpful to learn semantically plausible object motion field.

Algorithm 1 Regularization scheme of residual motion field

Input: Set of motion vector $\mathbf{V} = \{v_1, ..., v_n\}$, Set of predicted depth $\mathbf{D} = \{d_1, ..., d_n\}$ for every n-pixel in a detected box

Output: CSAC loss $\mathcal{L}_{csac}$ for a detected box

1: **function** REGULARIZER($\mathbf{V}, \mathbf{D}$)
2: $\quad\mathcal{L}_{csac} \leftarrow 0$ ▷ initialize CSAC loss for this detected box
3: $\quad\mathbf{M}_f \leftarrow$ FGMASK($\mathbf{D}$) ▷ $m_k \in \mathbf{M}_f$ is 1 if foreground (*fg*)
4: $\quad\mathbf{V}_f \leftarrow \{v_k | v_k \in \mathbf{V} \wedge m_k = 1\}$ ▷ *fg* motion set
5: $\quad\mathbf{V}_b \leftarrow \{v_k | v_k \in \mathbf{V} \wedge m_k = 0\}$ ▷ *bg* motion set
6: $\quad$**for** $\mathcal{V} \leftarrow \{\mathbf{V}_f, \mathbf{V}_b\}$ **do** ▷ for both *fg* and *bg* iterations
7: $\quad\quad S_{max} \leftarrow 0$ ▷ initialize inlier score
8: $\quad\quad$**for** $i \leftarrow 1$ to N **do** ▷ CPU
9: $\quad\quad\quad v_h \leftarrow \langle$random hypothesis from $\mathcal{V}\rangle$
10: $\quad\quad\quad\mathbf{S} \leftarrow$ CALCSCORE($v_h, \mathcal{V}$) ▷ for every $v_k \in \mathcal{V}$
11: $\quad\quad\quad S_i \leftarrow \sum \mathbf{S}$
12: $\quad\quad\quad$**if** $S_{max} < S_i$ **then**
13: $\quad\quad\quad\quad S_{max} \leftarrow S_i$
14: $\quad\quad\quad\quad\bar{v} \leftarrow$ REFINEVEC($\mathbf{S}, \mathcal{V}$) ▷ motion refinement
15: $\quad\quad\quad$**end if**
16: $\quad\quad$**end for**
17: $\quad\quad\mathcal{L}_{csac} \leftarrow \mathcal{L}_{csac} +$ CALCPENALTY($\bar{v}, \mathcal{V}$) ▷ GPU
18: $\quad$**end for**
19: $\quad$**return** $\mathcal{L}_{csac}$
20: **end function**

estimate the representative motion for the foreground and background through a random sampling technique. During the iteration, we measure the *L1-norm* between the hypothesis v_h and query vectors v_q for each translation axis, and calculate the inlier scores (line 10) as

$$\mathbf{S} = \sum_{v_q \in \mathcal{V}} \mathcal{F}_{inlier}\left(\left|\frac{v_h - v_q}{v_h}\right|_1\right), \tag{3}$$

where $\mathcal{F}_{inlier}$ is designed as

$$\mathcal{F}_{inlier}(\mathbf{x}) = 1 - \sigma(\alpha \cdot (\mathbf{x} - \beta)), \tag{4}$$

which is a soft inlier counting with a *sigmoid* function σ, proposed by [2, 3]. In our case, α and β are set to 30.0 and 0.2 respectively based on cross-validation. Then, we measure the iteration score S_i by simply aggregating the inlier scores to find the best hypothesis (line 11). The motion refinement is operated by multiplying the query vectors and inlier scores as weights (line 14). Note that these iterations are processed by the CPU since we do not require gradients for motion estimation. Once we get the refined motion vector $\bar{v}$, we calculate the contrastive penalty loss (line 17), imposed for each foreground and background as

$$\begin{aligned}\mathcal{L}^f_{csac} &= \sum_{v_q \in \mathbf{V}_f} \{1 - \mathcal{F}_{inlier}(max(0, (|\bar{v}_f| - |v_q|)/|\bar{v}_f|))\}, \\ \mathcal{L}^b_{csac} &= \sum_{v_q \in \mathbf{V}_b} \{1 - \mathcal{F}_{inlier}(max(0, (|v_q| - |\bar{v}_b|)/|\bar{v}_b|))\},\end{aligned} \tag{5}$$

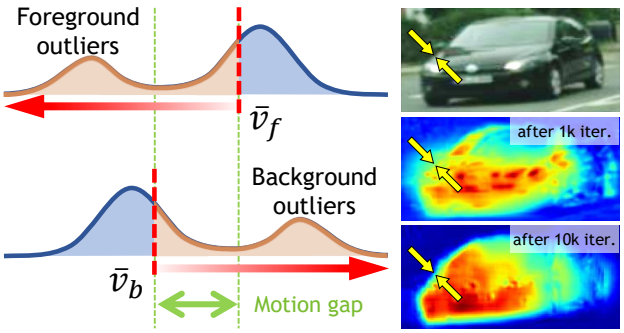


Figure 6. **Left**: Schematic of the distributions of foreground and background motions. We penalize small foreground motions and large background motions via CSAC, which eventually increases motion gap between the foreground and background. **Right**: Visualization of object motion inliers. As the learning progresses ($1k \rightarrow 10k$ iterations), the boundary of motion becomes clearer.

where $|\bar{v}_f|$ and $|\bar{v}_b|$ are the magnitudes of refined foreground and background motions respectively. This operation is processed by the GPU to perform the gradient propagation. In this equation, we penalize the foreground motions smaller than $|\bar{v}_f|$, and background motions larger than $|\bar{v}_b|$, in order to enhance motion contrast between the foreground and background, and this also meets our **Assumption 2**. Fig. 6 illustrates the importance of our motion contrast enhancement process to learn more accurate motion boundaries. Our final residual motion field regularization loss $\mathcal{L}_{mr}$ is the summation of $\mathcal{L}^f_{csac}$ and $\mathcal{L}^b_{csac}$ (per-box losses), and normalized to perform per-pixel loss for each mini-batch.

3.4. Training Scheme

Multi-phase joint training The proposed learning system is composed of complicated submodules. Although it is possible to gradually converge through end-to-end training, we propose a multi-phase learning technique for efficient and fast convergence. We summarize the training scheme in Table 1. Assigned tasks for each phase are jointly trained. In *phase-1*, since the residual motion branch is inactivated, we set the residual motion to be zero.

Self-supervised objective Our complete objective function is composed of *phase-1* loss $\mathcal{L}^{p_1}$ and *phase-2* loss $\mathcal{L}^{p_2}$ defined as

$$\begin{aligned} \mathcal{L}^{p_1} &= \lambda_p \mathcal{L}_p + \lambda_g \mathcal{L}_g + \lambda_s \mathcal{L}_s + \lambda_h \mathcal{L}_h, \\ \mathcal{L}^{p_2} &= \lambda_{mr} \mathcal{L}_{mr} + \lambda_{ms} \mathcal{L}_{ms} + \lambda_{mp} \mathcal{L}_{mp} + \lambda_{mc} \mathcal{L}_{mc}, \end{aligned} \tag{6}$$

where loss weights are grouped as $\Lambda^{p_1} = \{\lambda_p, \lambda_g, \lambda_s, \lambda_h\}$ and $\Lambda^{p_2} = \{\lambda_{mr}, \lambda_{ms}, \lambda_{mp}, \lambda_{mc}\}$, and each sub-loss is summarized as follows:

$\mathcal{L}_p$ and $\mathcal{L}_g$: Photometric and geometric consistency losses defined in [1, 15]. Occluded and disoccluded regions are masked out by geometric inconsistency map [28]. We replace their global and object-wise motion transformation to our pixel-wise motion representation.

$\mathcal{L}_s$: Generic edge-aware depth smoothness term, which is standardized in CC [36].

$\mathcal{L}_h$: Object scale constraint loss with height prior, introduced in Struct2Depth [7]. We use box height as the prior.

$\mathcal{L}_{mr}$: Proposed motion field regularization loss via CSAC.

$\mathcal{L}_{ms}$: We newly propose a reparametrized edge-aware motion smoothness loss. Compared to the *motion-repulsive* embedding by our motion contrastive learning, we add *motion-attractive* embedding to merge the near motion vectors locally. To prevent blurry inference near object boundary, we reparametrize the gradient of the edges with τ as

$$\mathcal{L}_{ms} = \sum (\nabla \mathbf{T}^{res} \cdot e^{-\nabla \mathbf{D}/\tau})^2, \tag{7}$$

where we set $\tau = 0.1$ in our training.

$\mathcal{L}_{mp}$ and $\mathcal{L}_{mc}$: Motion sparsity and consistency losses, which are introduced in [19, 29].

Phase	Joint tasks			Training parameters	Losses
	Depth	Ego-	Res-		
1^{st}	✓	✓	–	$\{\theta, \omega, \phi\}$	$\mathcal{L}^{p_1}$
2^{nd}	–	✓	✓	$\{\omega, \phi, \psi\}$	$\mathcal{L}^{p_1} + \mathcal{L}^{p_2}$
3^{rd}	✓	✓	✓	$\{\theta, \omega, \phi, \psi\}$	$\mathcal{L}^{p_1} + \mathcal{L}^{p_2}$

Table 1. Multi-phase joint training scheme between the three tasks: depth, ego-motion (ego-), and residual motion (res-) estimation.

4. Experiments

In this section, we validate our proposed methods: DAM and CSAC. We report and analyze experimental results for the tasks of monocular depth and scene flow estimation, motion segmentation, and visual odometry. For the sake of fairness, all competing techniques are purely monocular, moreover, we rule out the impact of different network architectures, *e.g.*, PackNet [20] and DispNet [44].

4.1. Implementation Details

Networks We design DepthNet with ResNet18-based ImageNet [37] pretrained encoder and decoder structure. The decoder has the same structure as Monodepth2 [18], and its output is a single-scale inverse depth map with a *sigmoid* activation. For MotionNet, we use ImageNet pretrained ResNet18 encoder with DAM followed by two motion branches: ego-motion decoder with three convolutional layers, and residual motion field decoder proposed by Gordon *et al.* [19]. Each block of the motion field decoder refines the motion feature from the previous block concatenated with its symmetrically corresponding output feature from the encoding block.

Training Our system is implemented in PyTorch [35] and trained using the ADAM optimizer [24] with the initial learning rate of 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ on 2×Nvidia

Model	# Params.	AbsRel	SqRel	$\delta_{1.25}$
Separated encoders	33.45 M	0.119	0.985	86.2
Shared encoder with DAM	**22.77 M**	**0.116**	**0.894**	**86.9**

Table 2. **Ablation study on shared motion encoder with DAM.** Numbers are reported after *phase-3*. Using a single motion encoder yields better performance on monocular depth estimation with fewer parameters.

Models	DAM	CSAC	*phase-1*		*phase-3*	
			all	*obj*	*all*	*obj*
A1	–	–	0.126	0.202	0.120	0.199
A2	–	✓			0.113	0.190
A3	✓	–	**0.121**	**0.196**	0.116	0.191
A4	✓	✓			**0.109**	**0.182**

Table 3. **Ablation study on DAM and CSAC.** We measure AbsRel errors after *phase-1* and *phase-3* on both *all* and *obj* areas.

RTX 2080 GPUs. We set the mini-batch size to 4 and each epoch is trained with 1,000 randomly sampled sequences following the augmentation policy of SC-SfM [1]. We train *phase-1* and *phase-2* for 10 epochs respectively. The loss weights, Λ^{p_1} and Λ^{p_2}, are tuned differently depending on the dataset and training phase. We describe this in detail in the `supplement`. To be brief, we empirically found that every objective shows stable convergence when the magnitude of each weighted per-pixel loss ($\lambda\mathcal{L}$) is 0.05 times the weighted photometric loss ($\lambda_p\mathcal{L}_p$).

Dataset Our system is trained and validated in KITTI [16], Cityscapes [11], and Waymo Open Dataset [38]. For KITTI and Cityscapes, we utilize the VIS annotations [28] for testing motion segmentation, and detection prior in CSAC training with a random margin up to 10% for generating detection box. The input resolution is set to 832×256 for KITTI and Cityscapes, and 480×320 for Waymo Open Dataset.

4.2. Ablation Study

To quantify the impact of our motion encoding using DAM and the motion regularization via CSAC, we propose various ablation studies. In this experiment, the training (90%) and validation (10%) sets are randomly split from KITTI raw monocular videos. We repeat the training 5 times and average the performance of monocular depth estimation. First, we perform an ablation to verify our motivation on sharing the motion encoder for MotionNet, while estimating the camera and object motion at the same time. As shown in Table 2, we achieve better performance with fewer number of trainable parameters, compared to the model with separated encoders. We, thus, conclude that our invertible attention mechanism enhances the capability of motion disentangling, which produces a better motion feature representation. Second, we proceed ablation integrated with both DAM and CSAC. In this case, we measure AbsRel error on both entire (*all*) and object (*obj*) regions. As demonstrated in Table 3, we conduct four ablations (*A1*∼*A4*) according to our proposed models. We observe that in *phase-1*, feature extraction with

Method	Semantic prior	D1			D2		
		bg	*fg*	*all*	*bg*	*fg*	*all*
DF-Net [45]	–	–	–	46.5	–	–	61.5
GeoNet [42]	–	–	–	49.5	–	–	58.2
CC [36]	–	35.0	42.7	36.2	–	–	–
SC-SfM [1]	–	36.0	46.5	37.5	–	–	–
EPC++ (mono) [31]	–	30.7	34.4	32.7	**18.4**	84.6	65.6
Insta-DM [28]	instance	**26.8**	**30.4**	**27.4**	28.9	**32.3**	**29.4**
Ours (DAM+CSAC)	box	28.6	32.5	29.8	30.5	35.7	32.6

Table 4. **Evaluation on the KITTI Scene Flow 2015 training set.** We validate the disparity compared to recent monocular-based training methods. **Bold**: Best, Underbar: Second best.

	Before reg.	After reg.	Li *et al.* [29]	CC [36]
KITTI-VIS	0.483	**0.813**	0.689	0.571
Cityscapes-VIS	0.416	**0.785**	0.620	–

Table 5. **Results (mean IoU) of object (vehicle class) segmentation with a given box prior.** The results show that CSAC regularization improve the performance on semantic perception task.

DAM (*A3* and *A4*) has a marginal improvement on depth estimation. After training MotionNet with residual motion field (*phase-2*), we notice a significant improvement while refining the depth in *phase-3*. In addition, regularization through CSAC further improves the depth estimation on the object area by providing a rigid body constraint. We conclude that our modules play an important role in enhancing the performance of joint depth and motion estimation.

4.3. Monocular Scene Flow Estimation

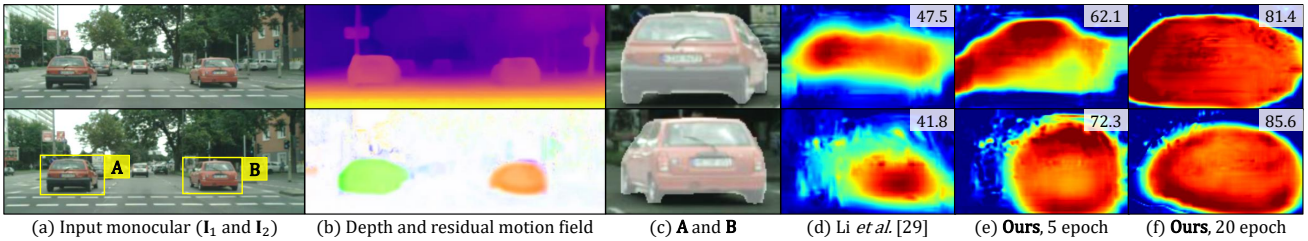
To validate our pixel-wise motion and depth estimation simultaneously, we assessed our monocular scene flow estimation on the KITTI Scene Flow 2015 training set, as shown in Table 4. We compare our method with existing monocular-based training methods. Compared to the methods not using semantic priors, we achieve more than 57.8% accuracy gain for estimating the disparity of objects on the target image (D2-*fg*). Despite using weaker priors, we obtain competitive results against techniques relying on strong semantic prior, such as, Insta-DM [28].

4.4. Motion Segmentation

We demonstrate object motion segmentation on KITTI Scene Flow 2015 training set and Cityscapes with VIS annotation [28]. This is enabled by leveraging inlier scores (threshold of 0.5) of the best hypothesis, as designed in Eq. (4). Table 5 demonstrates that our regularization with contrastive learning based on the geometric prior significantly improves the performance of semantic perception task. We demonstrate qualitative results in Fig. 7.

4.5. Monocular Depth Estimation and Visual Odometry

Finally, we provide comparisons to state-of-the-arts [18, 29, 7, 19, 25] of self-supervised monocular depth and



(a) Input monocular ($\mathbf{I}_1$ and $\mathbf{I}_2$) (b) Depth and residual motion field (c) **A** and **B** (d) Li *et al.* [29] (e) **Ours**, 5 epoch (f) **Ours**, 20 epoch

Figure 7. **Qualitative results of our depth and residual motion field on Cityscapes test set.** Each scene shows (a) consecutive input images with a box prior, (b) our networks outputs, (c) object images with GT mask, (d) motion inliers of the previous method [29] and ours after training (e) 5 and (f) 20 epoch. The inlier maps are normalized in the same scale, and we indicate their mean IoU from the GT.

Method	Semantic prior	Training	Test	Error metric ↓				Accuracy metric ↑		
				AbsRel	SqRel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [18]	–	K	K	0.115	0.903	4.863	0.193	0.877	0.959	**0.981**
Li *et al.* [29]	–	K	K	0.130	0.950	5.138	0.209	0.843	0.948	0.978
Struct2Depth [7]	instance	K	K	0.141	1.026	5.290	0.215	0.816	0.945	0.979
Gordon *et al.* [19]	box	K	K	0.128	0.959	5.230	0.212	0.845	0.947	0.976
SGDepth [25]	segment	K	K	**0.113**	**0.835**	**4.693**	**0.191**	**0.879**	**0.961**	**0.981**
Ours (DAM+CSAC)	box	K	K	0.114	0.876	4.715	**0.191**	0.872	0.955	**0.981**
Gordon *et al.* [19]	box	C+K	K	0.124	0.930	5.120	0.206	0.851	0.950	0.978
Ours (DAM+CSAC)	box	C+K	K	**0.111**	**0.805**	**4.708**	**0.187**	**0.875**	**0.962**	**0.981**
Li *et al.* [29]	–	C	C	0.119	1.290	6.980	0.190	0.846	**0.952**	**0.982**
Struct2Depth [8]	instance	C	C	0.145	1.737	7.280	0.205	0.813	0.942	0.978
Gordon *et al.* [19]	box	C	C	0.127	1.330	6.960	0.195	0.830	0.947	0.981
Ours (DAM+CSAC)	box	C	C	**0.116**	**1.213**	**6.695**	**0.186**	**0.852**	0.951	**0.982**
Monodepth2 [18]	–	W	W	0.168	1.738	7.947	0.230	–	–	–
Li *et al.* [29]	–	W	W	0.162	1.711	7.833	0.223	–	–	–
Struct2Depth [7]	instance	W	W	0.180	1.782	8.583	0.244	–	–	–
Ours (DAM+CSAC)	box	W	W	**0.148**	**1.686**	**7.420**	**0.210**	–	–	–

Table 6. Monocular depth estimation results on the KITTI (K) Eigen test set, Cityscapes (C) test set, and Waymo Open Dataset (W). Models pretrained on Cityscapes and fine-tuned on KITTI are denoted by 'C+K'. Due to the page limit, we only indicate our final model (DAM+CSAC), and methods using strong semantic priors (*e.g.*, video instance segmentation) are ruled out. Full table is demonstrated in the `supplement`. For each partition, **Bold**: Best, Underbar: Second best.

Method	Seq. 09	Seq. 10
SfM-Learner [44]	0.021 ± 0.017	0.020 ± 0.015
GeoNet [42]	0.012 ± 0.007	0.012 ± 0.009
CC [36]	0.012 ± 0.007	0.012 ± 0.008
Struct2Depth [7]	0.011 ± 0.006	0.011 ± 0.010
GLNet [10]	0.011 ± 0.006	$\mathbf{0.011 \pm 0.009}$
SGDepth [25]	0.017 ± 0.009	0.014 ± 0.010
Ours (w/o DAM)	0.012 ± 0.006	$\mathbf{0.011 \pm 0.009}$
Ours (w/ DAM)	$\mathbf{0.010 \pm 0.011}$	$\mathbf{0.011 \pm 0.009}$

Table 7. Absolute trajectory error (ATE) on KITTI-VO.

Method	Seq. 09		Seq. 10	
	t_{err}	r_{err}	t_{err}	r_{err}
GeoNet [42]	39.4	14.3	29.0	8.6
SC-SfM [1]	11.2	3.4	10.1	5.0
Ours (w/o DAM)	9.7	3.4	9.9	4.8
Ours (w/ DAM)	**8.9**	**3.3**	**9.5**	**4.7**

Table 8. Relative translation t_{err} (%) and rotation r_{err} ($^\circ/100m$) errors on KITTI-VO.

ego-motion estimation based on monocular training. We compare the depth estimation on KITTI Eigen split [13], Cityscapes [11], and Waymo Open Dataset [38], and all the compared methods are based on the ResNet18 backbone. As shown in Table 6, our final model with DAM and CSAC outperforms all published self-supervised methods with weak semantic priors (up to box prior). Qualitative results are demonstrated in the `supplement`.

We also demonstrate visual odometry on KITTI-VO in Table 7 and Table 8. In these experiments, our model with DAM outperforms state-of-the-arts using monocular self-supervised training. We conclude that our attention module favorably works in estimating the camera ego-motion.

5. Conclusion

We proposed a novel self-supervised learning framework to estimate the motion field of a dynamic scene from a monocular camera. First, our approach heavily relies on a novel attention module dedicated to the disentanglement of the camera ego-motion and the objects' motions, which has proven to be effective to improve the overall performance of our network. Second, we designed an object motion field estimation through contrastive sample consensus. With given geometric and semantic priors, we leverage a motion-repulsive embedding near object motion boundaries to estimate more accurate motion field. The effectiveness of our system has been demonstrated on various driving datasets. One remaining limitation is the rigid body assumption of the moving objects. While our approach is suitable for a traffic scene containing vehicles, it is not appropriate for deformable objects such as pedestrians. Therefore, we leave this problem as our future direction to improve the applicability of the technique to more diverse scenarios.

Acknowledgement

This work was supported under the framework of international cooperation program managed by the National Research Foundation of Korea (NRF-2020M3H8A1115028, FY2021). This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068.

References

[1] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, 2019. 2, 6, 7, 8

[2] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017. 5

[3] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *CVPR*, 2018. 5

[4] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, 2020. 2, 3

[5] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCV Workshop*, 2019. 4

[6] Zhe Cao, Abhishek Kar, Christian Hane, and Jitendra Malik. Learning independent object motion from unlabelled stereoscopic videos. In *CVPR*, 2019. 2

[7] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019. 2, 6, 7, 8

[8] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *CVPR Workshop*, 2019. 2, 8

[9] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *CVPR*, 2019. 2

[10] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *ICCV*, 2019. 2, 8

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 7, 8

[12] Qi Dai, Vaishakh Patil, Simon Hecker, Dengxin Dai, Luc Van Gool, and Konrad Schindler. Self-supervised object motion and depth estimation from video. In *CVPR Workshop*, 2020. 2

[13] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 8

[14] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014. 1

[15] Feng Gao, Jincheng Yu, Hao Shen, Yu Wang, and Huazhong Yang. Attentional separation-and-aggregation network for self-supervised depth-pose learning in dynamic scenes. In *CoRL*, 2020. 3, 6

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 7

[17] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 2

[18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 2, 6, 7, 8

[19] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *ICCV*, 2019. 2, 3, 5, 6, 7, 8

[20] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, 2020. 2, 6

[21] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *ICLR*, 2020. 2

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[23] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *CVPR*, 2020. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[25] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *ECCV*, 2020. 2, 7, 8

[26] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning residual flow as dynamic motion from stereo videos. In *IROS*, 2019. 2

[27] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Instance-wise depth and motion learning from monocular videos. In *NeurIPS Workshop*, 2020. 2

[28] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *AAAI*, 2021. 2, 3, 6, 7

[29] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *CoRL*, 2020. 3, 5, 6, 7, 8

[30] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *ICRA*, 2018. 1

[31] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ramkant Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 7

[32] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *CVPR*, 2018. 1, 2

[33] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 2015. 1

[34] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979. 5

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[36] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, 2019. 1, 2, 6, 7, 8

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015. 6

[38] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 7, 8

[39] Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Wolfram Burgard, Greg Shakhnarovich, and Adrien Gaidon. Neural ray surfaces for self-supervised learning of depth and ego-motion. In *3DV*, 2020. 2

[40] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018. 1, 2

[41] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *CVPR*, 2018. 1, 2

[42] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 7, 8

[43] Junming Zhang, Katherine A Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. Dispsegnet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery. *Robotics and Automation Letters*, 2019. 1

[44] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 1, 2, 6, 8

[45] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018. 7