EMR-MSF: Self-Supervised Recurrent Monocular Scene Flow Exploiting Ego-Motion Rigidity

Zijie Jiang Masatoshi Okutomi
Tokyo Institute of Technology
zjiang@ok.sc.e.titech.ac.jp, mxo@ctrl.titech.ac.jp

Abstract

*Self-supervised monocular scene flow estimation, aiming to understand both 3D structures and 3D motions from two temporally consecutive monocular images, has received increasing attention for its simple and economical sensor setup. However, the accuracy of current methods suffers from the bottleneck of less-efficient network architecture and lack of motion rigidity for regularization. In this paper, we propose a superior model named **EMR-MSF** by borrowing the advantages of network architecture design under the scope of supervised learning. We further impose explicit and robust geometric constraints with an elaborately constructed ego-motion aggregation module where a rigidity soft mask is proposed to filter out dynamic regions for stable ego-motion estimation using static regions. Moreover, we propose a motion consistency loss along with a mask regularization loss to fully exploit static regions. Several efficient training strategies are integrated including a gradient detachment technique and an enhanced view synthesis process for better performance. Our proposed method outperforms the previous self-supervised works by a large margin and catches up to the performance of supervised methods. On the KITTI scene flow benchmark, our approach improves the SF-all metric of the state-of-the-art self-supervised monocular method by 44% and demonstrates superior performance across sub-tasks including depth and visual odometry, amongst other self-supervised single-task or multi-task methods.*

1. Introduction

Scene flow estimation, which involves estimating both 3D structure and 3D motion of a dynamic scene from its two consecutive observations, has been receiving increasing attention due to its significance in areas such as robotics [10], augmented reality [22], and autonomous vehicles [35]. Recently, deep learning has demonstrated remarkable progress in the domain of scene flow estimation based on various input modalities, including stereo images [3, 24, 32, 41, 51, 40], RGB-D pairs [31, 39, 45, 33], or Lidar points [28, 18, 54, 56, 38, 55, 12, 7, 11, 52]. These methods, however, either require strict sensor calibrations (*e.g.*, stereo-based), or expensive devices (*e.g.*, RGB-D or Lidar-based) for achieving satisfactory performance, which restricts their widespread applications.

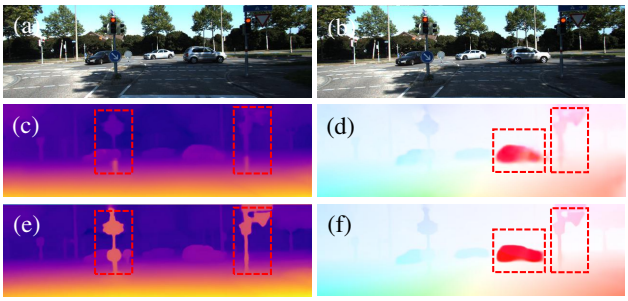


Figure 1: **Comparison between our method and [20].** (a) input first frame, (b) input second frame, (c) depth of first frame from [20], (d) synthesized optical flow from [20], (e) depth of first frame from our method, (f) synthesized optical flow from our method. Our method generates more regularized and detailed predictions as shown in red boxes.

On the other hand, monocular scene flow estimation methods [5, 57, 58, 62, 30, 26, 20, 21, 2] which only require a monocular camera for obtaining both 3D structure and 3D motion, have been presented as an economical yet effective solution for dynamic 3D perception. The methods [5, 57] combined with supervised learning have yielded promising results, yet the primary challenge facing them has been the limited availability of ground-truth training data. To address this limitation, several multi-task methods [58, 62, 30, 53, 26] have been proposed to jointly learn the depth, 2D optical flow and camera ego-motion networks from monocular sequences in a self-supervised manner, and the scene flow can be calculated from the outputs. Recently, [20, 21, 2] have shown it feasible to train a single network to directly estimate both depth and 3D scene flow from two

monocular images and outperform the previous multi-task methods. These methods typically build upon a standard optical flow pipeline (*e.g.*, PWC-Net [43] or RAFT [44]) as basis and adapt it for monocular scene flow. Despite the notable progress achieved by these methods, their accuracy still lags behind the supervised monocular methods by a large margin.

In this paper, we propose a novel approach for self-supervised monocular scene flow estimation, which outperforms the previous methods significantly as shown in Fig.1. To introduce explicit 3D geometry-oriented property, we follow the network architecture proposed in the supervised RGB-D method RAFT-3D [45] that iteratively refines a dense SE3 motion field for scene flow estimation. This improvement of architecture compared to previous methods directly improves the performance to a new level, but we argue that it still lacks the usage of *Ego-Motion Rigidity (EMR)*, an important prior that pixels in static regions should have the same SE3 motion as the ego-motion. A novel module named ego-motion aggregation (EMA) is thus proposed to jointly estimate ego-motion as well as a rigidity soft mask from the dense SE3 motion field. A new motion consistency loss is elaborately designed for constraining motion estimations in static areas represented by the rigidity soft mask. However, we notice that the network is inclined to select only a small subset of static regions which leads to a rigidity soft mask of low quality. To mitigate this problem, we adopt an efficient mask regularization loss to encourage the network to locate as many static regions as possible. Further performance improvement is attributed to our proposed training strategies including a gradient detachment technique and an improved view synthesis process.

Our main contributions are summarized as follows:

- We propose a novel self-supervised monocular scene flow estimation by incorporating 3D geometry-oriented network architecture property and exploiting ego-motion rigidity (EMR-MSF). To the best of our knowledge, we are the first method capable of jointly estimating depth, dense SE3 motion field and ego-motion from monocular images, as well as full scene flow derived from them.

- We introduce a novel ego-motion aggregation (EMA) module accompanied by a rigidity soft mask to precisely locate static regions for robust and accurate ego-motion estimation.

- We propose two new training losses to constrain the motion estimations in static regions, along with two effective training strategies to enhance accuracy as explained in Sec. 3.3.

- We conduct extensive experiments to verify the effectiveness of our proposed method, resulting in a 44% accuracy boost in the SF-all metric compared to the previous state-of-the-art method on the task of monocular scene flow estimation, as well as superior results in monocular depth and visual odometry.

2. Related Work

Scene flow. As first introduced in [48], scene flow estimation is defined as the task of jointly estimating 3D structures and 3D motions for each scene point. The early studies [1, 19, 50, 49, 51] are based on stereo inputs and approach the scene flow estimation as an energy minimization problem. Recently, deep learning has demonstrated powerful capabilities in end-to-end learning of scene flow estimation from stereo inputs [24, 32, 41]. Additionally, approaches that leverage pre-existing 3D structure through inputs of RGB-D sequences [31, 39, 45, 33] or Lidar points [28, 56, 38, 55, 12, 11, 52] have also been proposed for various scenarios.

Monocular scene flow. The advancement of deep learning techniques has facilitated the acquisition of scene flow solely from monocular images, with early methods relying on supervised learning [5, 57]. To exploit vast amounts of unlabeled data, a multitude of self-supervised multi-task approaches [58, 53, 62, 30, 26, 27] have been introduced that jointly predict depth, 2D optical flow, and camera motion from monocular sequences. While the recovery of scene flow is possible using the aforementioned outputs, the accuracy of such estimations is notably inadequate in temporally occluded areas. Hur et al. [20] first present a novel self-supervised model capable of inferring depth and 3D motion field from monocular sequences and surpass the performance of previous multi-task methods. Subsequent studies extend their method into a multi-frame model [21], or employ a recurrent network architecture [2] for better accuracy.

Rigidity in Scene Flow. Scene flow estimation can benefit from prior knowledge about rigidity, which assumes that pixels belonging to the same rigid object should undergo the same rigid transformation. To leverage the rigidity information in the scene, object detection or segmentation networks are commonly used to identify rigid instances and incorporated in scene flow estimation methods [32, 6, 40, 3] for better performance. Teed et al. [45] first propose the rigid-motion embeddings which softly and differentiably group pixels into rigid objects to exploit object-level rigidity. On the other hand, ego-motion rigidity, where the motion of pixels in static regions is constrained by the camera ego-motion, is widely used in self-supervised multi-task methods [58, 53, 62, 30, 26, 27] but often in a hard and non-differentiable way. In contrast, our proposed method jointly reasons ego-motion and rigidity soft mask in a fully differentiable manner, providing more robust and accurate scene flow estimation.

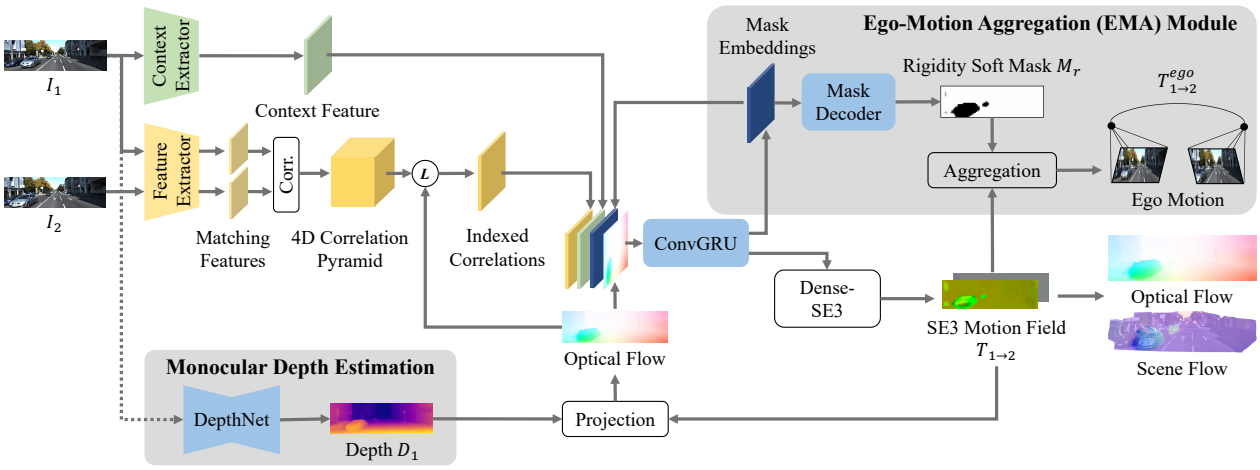


Figure 2: **Proposed network architecture**. We highlight the different parts from RAFT-3D [45] with the shaded boxes, including 1) end-to-end trainable monocular depth estimation that substitutes the estimated depths for fixed input depths in the original structure, 2) an ego-motion aggregation (EMA) module for inferring ego-motion along with a learnable rigidity soft mask for locating static regions.

3. Proposed Method

Given two temporally consecutive monocular images $\{I_1, I_2\} \in \mathbb{R}^{H\times W\times 3}$, our method aims to recover 1) the corresponding depth maps $D_1, D_2 \in \mathbb{R}^{H\times W\times 1}$, 2) the dense SE3 motion field $T_{1\to2} \in SE(3)^{H\times W}$ that assigns a rigid transformation to each pixel of I_1 to I_2, and 3) the ego-motion $T^{ego}_{1\to2} \in SE(3)$ from I_1 to I_2. The optical flow $F_{1\to2} \in \mathbb{R}^{H\times W\times 2}$ and scene flow $S_{1\to2} \in \mathbb{R}^{H\times W\times 3}$ from I_1 to I_2 can be further recovered from the estimated D_1 and $T_{1\to2}$. In the following sections, we will begin by providing an overview of the proposed network architecture which incorporates effective designs for 3D estimations from a supervised method (Sec. 3.1). Afterwards, we provide a detailed description of the proposed ego-motion aggregation (EMA) module that we utilize for estimating ego-motion, as well as a learnable rigidity soft mask for effectively locate static regions (Sec. 3.2). Finally, we elaborate on our self-supervised training in Sec. 3.3, which includes novel loss functions designed to fully exploit ego-motion rigidity, as well as improved training strategies.

3.1. Network Overview

Fig. 2 demonstrates the overview of our network. We highlight the different parts of our network compared to RAFT-3D [45], which is the basis of our network architecture, inside the shaded boxes. Our network consists of five stages: 1) monocular depth estimation, 2) feature extraction 3) correlation computing, 4) iterative refinement and 5) ego-motion aggregation. We first employ a monocular depth network to estimate the depth maps of input images instead of the fixed depths used in the original RAFT-3D structure. We adopt SDFA-Net [60] for depth estimation for its superior performance, which infers disparity from a monocular image under the assumption of a fixed baseline, and further converts the disparity into depth using pre-known focal length and baseline values. For feature extraction, correlation computing and iterative refinement, we utilize the designs of RAFT-3D, which include the construction of a 4D all-pairs correlation pyramid from extracted features of input images and the use of a ConvGRU unit followed by a Dense-SE3 layer for iterative residual refinement of the SE3 field estimate. The ego-motion aggregation (EMA) module is employed to further infer ego-motion from the estimated SE3 motion field, which is elaborated on in the next section. The 3D scene flow and 2D optical flow can be synthesized from estimated depth and SE3 motion field for various applications.

3.2. Ego-Motion Aggregation

As demonstrated in our ablation study 4.3, the joint learning of the depth and dense SE3 motion field in the self-supervised scenario can lead to significant ambiguities between the estimations of structure and motion, where the estimated SE3 motions of pixels belonging to the same rigid object, *e.g.*, the static regions, may be inconsistent. To mitigate such ambiguities, we incorporate the ego-motion estimation into the joint learning to provide additional constraints in static regions. We propose to aggregate the ego-motion from the estimated SE3 motion field in contrast to previous multi-task methods [53, 58, 62], which utilize a separate network to regress ego-motion from input images. Furthermore, to handle the dynamic regions which are non-relevant to ego-motion, we introduce a learnable rigidity

soft mask to predict per-pixel rigidity, thus locating static regions for stable ego-motion estimation.

Our ego-motion aggregation module proceeds in three steps, as shown in the upper-right corner of Fig. 2. We first incorporate the mask embeddings, a 16-channel feature map initialized to zero values, as new inputs and outputs to the convGRU unit, which is iteratively updated alongside the SE3 motion field. Next, we decode the mask embeddings using a mask decoder consisting of two convolutional layers and a sigmoid activation layer to obtain the rigidity soft mask M_r. The rigidity soft mask assigns a probability to each pixel, indicating the probability of it belonging to the static region. In the final step, we derive the ego-motion as an aggregation of estimated SE3 motion field based on the learned rigidity soft mask, which is formulated as:

$$T_{1\to2}^{ego} = \mathrm{Exp}(\frac{\sum M_r \mathrm{Log}(T_{1\to2})}{\sum M_r}), \tag{1}$$

where $\mathrm{Log}(\cdot)$ maps SE(3) components to the Lie algebra, and $\mathrm{Exp}(\cdot)$ performs the inverse operation.

As the ego-motion is differentiably computed from the SE3 motion field, the learning of ego-motion will implicitly impose constraints on the estimation of SE3 motion field. In the next section, we further combine the self-supervised losses with two new losses utilizing the ego-motion estimation and learned rigidity soft mask to explicitly regularize the motion estimations in static regions.

3.3. Self-supervised Training

3.3.1 Self-supervised Loss

To enable self-supervised training, the estimated depth D_1 of the first image and the SE3 motion field $T_{1\to2}$ are first converted into the scene flow representation $(u, v, \Delta D)$ with known camera intrinsics [33], where (u, v) denotes the standard optical flow $F_{1\to2}$, and ΔD denotes the depth change registered to the first frame I_1. We denote $\overline{D}_1 = D_1 + \Delta D$, which represents the transformed depth map registered to the first frame. We obtain the 2D rigid flow $F_{1\to2}^{ego}$ in the same manner by replacing $T_{1\to2}$ with $T_{1\to2}^{ego}$. The losses for our joint self-supervised learning are introduced as follows:

Temporal Photometric loss. We minimize the photometric differences between the original image and the synthesized images from flow field $F_{1\to2}$ and $F_{1\to2}^{ego}$, formulated by

$$L_p = \frac{1}{HW}\sum M_{noc} \odot pe(I_1, w(I_2, F_{1\to2})), \tag{2}$$

$$L_p^{ego} = \frac{1}{HW}\sum M_{ol} \odot M_{noc} \odot pe(I_1, w(I_2, F_{1\to2}^{ego})), \tag{3}$$

$$pe(I_a, I_b) = \frac{\alpha}{2}(1 - \mathrm{SSIM}(I_a, I_b)) + (1-\alpha)\left|I_a - I_b\right|, \tag{4}$$

where $\frac{1}{HW}\sum$ is used for the notation of the mean over all pixels and $\odot$ means element-wise multiplication. $w(\cdot,\cdot)$ is the view synthesis function with the flow field and $pe(\cdot,\cdot)$ measures the photometric difference between two images. The occlusion mask M_{noc} is derived from the forward-backward consistency check [34] using $F_{1\to2}$ and $F_{2\to1}$. We additionally use an outlier mask M_{ol} [23] for calculating L_p^{ego}, which masks out pixels with either large photometric errors mainly resulting from possible occluded or moving regions, or very small photometric errors mainly resulting from textureless regions. Note that M_r is not leveraged here for two reasons: 1) M_{ol} performs more stable than the learned mask M_r in the beginning of training. 2) M_{ol} can better locate pixels which are informative for learning ego-motion estimation.

Spatial Photometric Loss. To address scale ambiguity in monocular scene flow learning, we utilize stereo samples during training as proposed in previous works [20, 21, 2]. We use the stereoscopic image synthesis loss utilized in [60] to regularize depth estimation on an absolute scale and denote it as L_d in our method.

Geometric loss. To constrain the estimated motion field in 3D space, we exploit the geometric consistency between the transformed depth map $\overline{D}_1$ and estimated D_2:

$$L_g = \frac{1}{HW}\sum M_{noc} \odot ge(\overline{D}_1, w(D_2, F_{1\to2})), \tag{5}$$

$$ge(D_a, D_b) = \frac{|D_a - D_b|}{D_a + D_b}, \tag{6}$$

where $ge(\cdot,\cdot)$ measures the normalized difference [4] between two depth maps.

Smoothness loss. The k-th order edge-aware smoothness loss function is defined as:

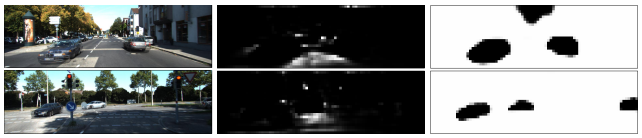
$$L_s(O) = \frac{1}{HW}\sum \left|\frac{\partial^k O}{\partial x^k}\right| e^{-\beta\left|\frac{\partial I_1}{\partial x}\right|} + \left|\frac{\partial^k O}{\partial y^k}\right| e^{-\beta\left|\frac{\partial I_1}{\partial y}\right|}, \tag{7}$$

where O is a dense prediction, which can be $\mathrm{Log}(T_{1\to2})$, D_1 and $F_{1\to2}$ in our case. We apply first-order edge-aware smoothness loss to $\mathrm{Log}(T_{1\to2})$ and D_1, denoted as $L_{s,t}$ and $L_{s,d}$ separately, and apply second-order edge-aware smoothness loss to $F_{1\to2}$ as $L_{s,f}$. The total smoothness loss is calculated as $L_s = \lambda_{st}L_{s,t} + \lambda_{sd}L_{s,d} + \lambda_{sf}L_{s,f}$.

Motion Consistency Loss. To further regularize the SE3 motion field in static regions, we propose to explicitly constrain the motion estimations in these regions to be consistent with the estimated ego-motion, formulated as:

$$L_c = \frac{1}{HW}\sum M_r \odot |\mathrm{Log}(T_{1\to2}) - \mathrm{Log}(T_{1\to2}^{ego})|, \tag{8}$$

Mask Regularization loss. We observe that the estimated rigidity soft mask tends to degenerate during training. This is intuitively reasonable since theoretically the ego-motion



Overlayed inputs W/o L_m W L_m

Figure 3: **Visualization of estimated rigidity soft masks**. The middle column shows the degeneration cases of estimated rigidity soft mask, which is solved by introducing L_m during training.

can be represented as the SE3 motion of any single pixel in static regions, thus the rigidity soft mask is inclined to select only a small subset of static regions due to L_c. To address this problem, we propose a mask regularization loss to encourage the rigidity soft mask to locate static regions as many as possible for fully exploiting ego-motion rigidity in static regions, which is formulated as:

$$L_m = \frac{1}{HW} \sum \frac{1 - M_r}{\gamma + M_r}, \tag{9}$$

where γ is a hyper-parameter. We provide a visual comparison of the estimated rigidity soft mask without and with L_m in Fig. 3.

Total Loss. We calculate losses for both the final and intermediate estimations from our recurrent structure. We use an upper-right index $(\cdot)^i$ to denote the losses related to the i-th iteration. The total loss of our method can be summarized as:

$$\begin{aligned} L_{total} = L_d + \sum_{i=1}^{N} \zeta^{N-i} \big(& L_p^i + L_p^{ego,i} \\ & \lambda_g L_g^i + \lambda_s L_s^i + \lambda_c L_c^i + \lambda_m L_m^i \big) , \end{aligned} \tag{10}$$

where N is the iteration number, ζ is the weight decay factor, and $\lambda = [\lambda_g, \lambda_s, \lambda_c, \lambda_m]$ is the set of hyper-parameters balancing different losses.

3.3.2 Improved Training Strategies

Gradient Detachment. Our loss functions except L_d are calculated for both the final and intermediate estimations of motion field and ego-motion for preventing divergence of training. However, joint learning of depth and coarse motion estimations from early iterations can hinder the learning of the depth network. To address this issue, we propose to detach the gradients of depth estimations when calculating losses using intermediate motion estimations, which ensures that joint learning only occurs when the finest motion estimations are utilized.

Improved view synthesis process. We leverage the full-image warping technique proposed in [42] to provide better supervisory signals at image boundaries during the calculation of photometric loss, which uses cropped images as inputs to the network, but refers to the uncropped images when performing view synthesis. We further leverage this idea during the calculation of geometric loss in Eqn. 5, where we refer to the estimated depths of uncropped images for depth synthesis.

4. Experimental Results

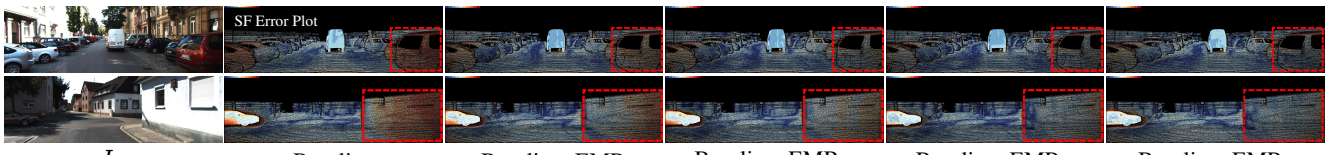
Our proposed method is evaluated on various tasks including scene flow, monocular depth, and visual odometry.

4.1. Implementation Details

We implement our network with Pytorch [37]. All components of our network are trained from scratch, except the encoder in the depth network and the context extractor, which use ImageNet [9] pretrained weights. We use the Adam optimizer [29] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to train our network. During training, the images are first resized into the resolution of 800×240, and cropped off the top, bottom, left and right 10% pixels to obtain the input images of 640×192 to leverage the improved view synthesis process. During test, the images are resized into 640×192 for processing and the results are bilinearly rescaled back to the original size for evaluation. We use a two-staged training process for better stability of our method. During the first stage, we separately train the depth network using spatial photometric loss L_d and depth smoothness loss $L_{s,d}$. Then, we train our full network using the total loss L_{total} for the rest epochs. The training is carried on for 50 epochs total, 20 epochs for the first stage, and 30 epochs for the second stage. The initial learning rate is set to 1e-4, and downgraded by half at epoch 20, 25, 30, and 40. The hyper-parameters of our method are set as: $[\alpha, \beta, \gamma, \zeta] = [0.15, 10, 1, 0.9], [\lambda_{s,t}, \lambda_{s,d}, \lambda_{s,f}] = [0.001, 1, 1], [\lambda_g, \lambda_s, \lambda_c, \lambda_m] = [0.1, 0.1, 0.1, 0.1]$, $N = 12$. For data augmentation, we employ random color augmentation, random horizontal flipping and random time order switching. We use the LieTorch [46] library to perform backpropagation of the SE3 motion field.

4.2. Datasets and Evaluation Metrics

Datasets. For the scene flow task, we use the same data setting as previous self-supervised monocular scene flow methods [20, 21, 2], which use KITTI Scene Flow Training and Testing as two test sets, and spilt the remaining data into 25801 samples for training and 1684 samples for validation. For comparison in the task of monocular depth estimation, we follow the data split used in [60], but remove the samples which are the last images of sequences, which gives us 22568 samples for training and 1774 for validation. The depth evaluation is conducted on the Eigen Test



I_t | Baseline | Baseline+EMR | Baseline+EMR+ MRL | Baseline+EMR+ MRL+GD | Baseline+EMR+ MRL+GD+IVS

Figure 4: **Qualitative ablation study of proposed components.** The erroneous predictions are gradually reduced by incorporating proposed components as shown in the red boxes.

EMR	MRL	GD	IVS	D1-all ↓	D2-all ↓	F1-all ↓	SF-all ↓	EPE-noc ↓	EPE-occ ↓	EPE-all ↓
-	-	-	-	13.20	21.90	14.16	28.15	2.78	12.57	4.83
√	-	-	-	9.99	18.25	13.55	24.21	2.68	10.85	4.44
√	√	-	-	9.83	17.25	13.65	23.07	2.69	10.17	4.23
√	√	√	-	9.39	16.90	13.51	22.86	2.65	10.04	4.21
-	-	√	√	11.73	18.76	12.54	24.80	2.74	7.63	3.81
√	√	√	√	**9.03**	**15.42**	**11.93**	**21.17**	**2.53**	**7.07**	**3.56**

Table 1: **Quantitative ablation study of key components.** EMR: Ego-Motion Rigidity, MRL: Mask Regularization Loss, GD: Gradient Detachment, IVS: Improved View Synthesis. All components effectively improve the performance, especially the EMR component.

Iter. Num.	D1-all ↓	D2-all ↓	F1-all ↓	SF-all ↓	Runtime
2	9.03	15.42	11.93	21.17	127 ms
4	8.65	13.93	**11.36**	19.05	151 ms
8	8.38	13.14	11.76	18.31	204 ms
12	**8.37**	**12.86**	11.58	**18.11**	250 ms

Table 2: **Ablation study of the iteration number.** More iterations give better performance up to about 12, but with a slower speed.

split [13], which contains 697 images with ground-truth labels. For the task of visual odometry, we use the official odometry data split, which uses Seq. 00-08 for training and Seq. 09-10 for testing, as done in [53, 61, 25].

Metrics. We follow the evaluation metric of KITTI Scene Flow benchmark [35] for scene flow estimation, which evaluates the outlier rate of the disparity for the reference frame (D1-all) and for the target image mapped into the reference frame (D2-all), as well as of the optical flow (F1-all). The outlier rate of the scene flow (SF-all) is obtained by checking if a pixel is an outlier on either of them. For monocular depth evaluation, we use the publicly used metrics, including: Abs Rel, Sq Rel, RMSE, logRMSE, A1= δ<1.25, A2= δ<1.25^2 and A3= δ<1.25^3. For visual odometry evaluation, we adopt the KITTI odometry criterion, which reports the average translational error T_{rel} and rotational error R_{rel} of possible sub-sequences of length (100, 200, 800) meters as the main criteria.

4.3. Ablation Studies

We first conduct ablation studies to verify the effectiveness of each proposed component of our method on the task of scene flow estimation, including 1) ego-motion rigidity (EMR), which includes the ego-motion aggregation module and losses for L_p^{ego} and L_c, 2) mask regularization loss L_r (MRL), 3) gradient detachment technique (GD), and 4) improved view synthesis (IVS). For efficiency, the ablation studies are conducted using iteration number equal to 2. We report both the scene flow metrics and end-point-error (EPE) of synthesized optical flow in Tab. 1. Each proposed component proves to be effective in improving the overall scene flow accuracy. The largest performance gain is obtained by exploiting the ego-motion rigidity, which is in line with our expectation that ego-motion rigidity is an important prior in the task of scene flow estimation. Fig. 4 gives a visualization of the achieved error reduction on SF-all error plots from each component. The erroneous estimations in static regions and image boundaries are largely reduced by incorporating our contributions. The ablation study on the iteration number is reported in Tab. 2. The performance is about to reach convergence when the iteration number is 12. We also report the runtime for efficiency comparison, which is tested on a single GTX 3090 device for each model. For the following experiments, we always set the iteration number to 12.

4.4. Comparison with State of the Art Methods

Scene Flow. We compare our method with other state-of-the-art monocular scene flow methods on both KTTII Scene Flow Training set and Testing Set as shown in Tab. 3. Our method achieves the best performance among all methods based on self-supervised learning, and even outperforms Mono-SF [5], which is a hybrid method based on the combination of supervised monocular depth estimation and energy minimization. In Fig. 5, we visualize the estimations and error maps of our method and other methods on sam-

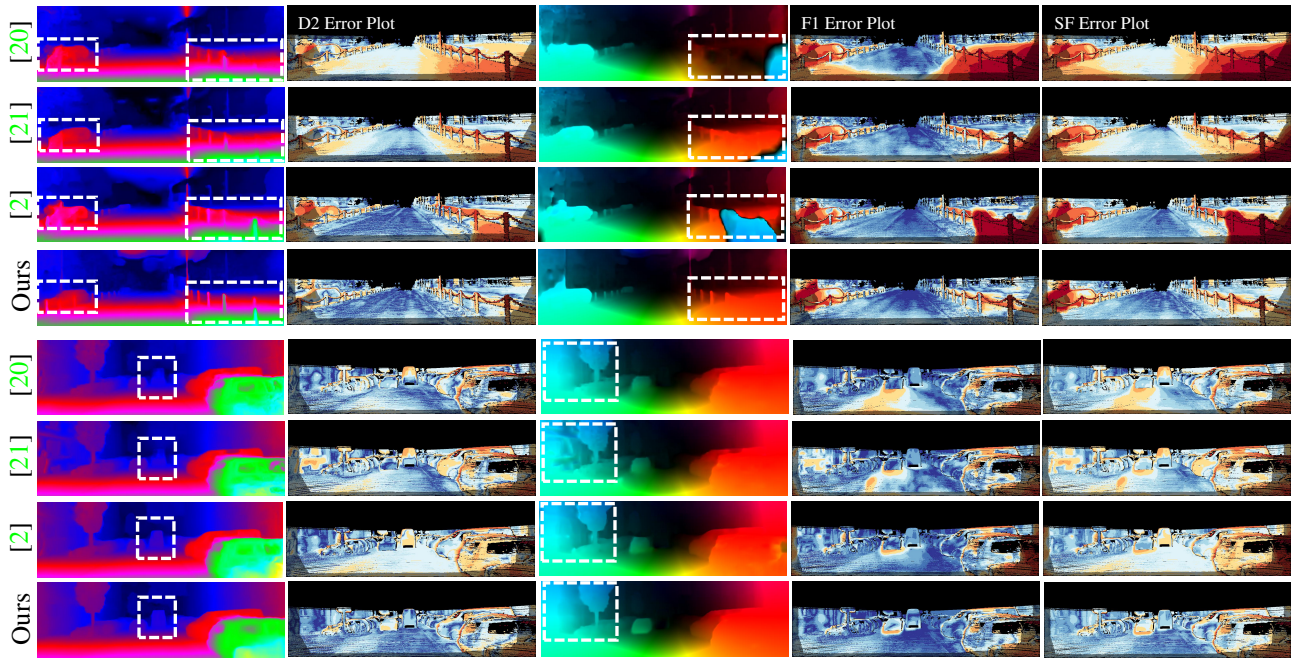


Figure 5: **Qualitative evaluation on KITTI Scene Flow Testing set.** We compare our method with Self-Mono-SF [20], Multi-Mono-SF [21] and RAFT-MSF [2] for two scenes using the visualizations provided by the KITTI benchmark [35]. From left to right: disparity visualization of I_t, $D2$ error plot, optical flow visualization, corresponding $F1$ error plot and combined SF error plot.

Method	KITTI Scene Flow Training Set				KITTI Scene Flow Testing Set			
	D1-all ↓	D2-all ↓	F1-all ↓	SF-all ↓	D1-all ↓	D2-all ↓	F1-all ↓	SF-all ↓
Mono-SF [5]	16.72	18.97	11.85	21.60	16.32	19.59	12.77	23.08
GeoNet [58]	49.54	58.17	37.83	71.32	-	-	-	-
DF-Net [62]	46.50	61.54	27.47	73.30	-	-	-	-
EPC++ [30]	23.84	60.32	19.64	-	-	-	-	-
Self-Mono-SF [20]	31.25	34.86	23.49	47.05	34.02	36.34	23.54	49.54
Multi-Mono-SF [21]	27.33	30.44	18.92	39.82	30.78	34.41	19.54	44.04
RAFT-MSF [2]	18.34	23.65	17.51	30.97	21.21	27.51	18.37	34.98
EMR-MSF (Ours)	**8.37**	**12.86**	**11.58**	**18.11**	**9.70**	**14.51**	**11.93**	**19.74**

Table 3: **Quantitative evaluation of the scene flow on the KITTI Scene Flow Training set and Testing set.** The best results are in **bold**.

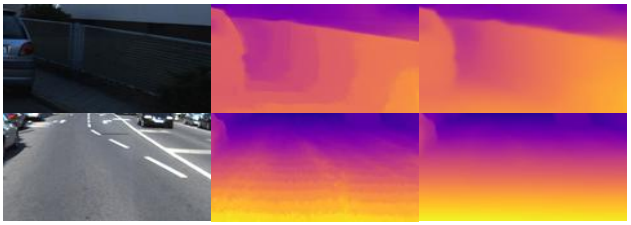
ples from KITTI Scene Flow Testing set. In the highlighted regions, our method shows better regularized and detailed estimations compare to other methods which give no consideration to exploit ego-motion rigidity. The error maps of various metrics are provided for better visualization.

Monocular Depth. We compare our method trained on the KITTI Eigen split with other state-of-the-art monocular depth methods as shown in Tab. 4. Our method achieves the best performance in 4 metrics among all compared methods and second best in the left 3 metrics. A visual comparison between our results and [60] is given in Fig. 6. Our method produces smoother depth estimations than [60], which we attributes to the jointly learning of depth and motion.

Visual Odometry. Finally, we compare the performance of our method trained on the KITTI Odometry split with other monocular methods in the task of visual odometry, including ORB-SLAM2 [36], a traditional method, as well as other self-supervised learning-based methods. We provide both results of ORB-SLAM2 with and without loop closure. For evaluating monocular methods, we perform the scale alignment to align the predicted up-to-scale trajectories to the ground-truth associated poses using [47]. Since our method leverages stereo samples during training, it is possible for our method to predict trajectories on a

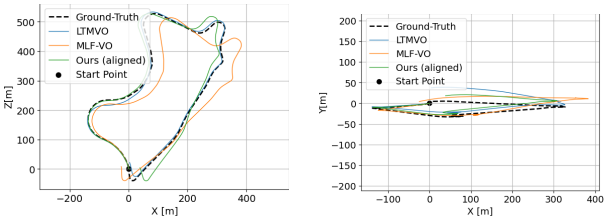
Method	Sup.	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	A1 ↑	A2 ↑	A3 ↑
Monodepth2 [15]	S	0.109	0.873	4.960	0.209	0.864	0.948	0.975
FAL-Net [17]	S	0.097	0.590	3.991	0.177	0.893	0.966	0.984
PLADE-Net [16]	S	0.092	0.626	4.046	0.175	0.896	0.965	0.984
SDFA-Net [60]	S	**0.090**	**0.538**	**3.896**	**0.169**	**0.906**	**0.969**	**0.985**
EPC++ [30]	MS	0.127	0.936	5.008	0.209	0.841	0.946	0.979
Self-Mono-SF [20]	MS	0.125	0.978	4.877	0.208	0.851	0.950	0.978
Monodepth2 [15]	MS	0.106	0.818	4.750	0.196	0.874	0.957	0.979
DIFFNet [59]	MS	0.101	0.749	4.445	0.179	0.898	0.965	0.983
RAFT-MSF [2]	MS	0.093	0.781	4.321	0.186	0.901	0.960	0.981
EMR-MSF (Ours)	MS	**0.088**	**0.552**	**3.946**	**0.169**	**0.905**	**0.970**	**0.986**

Table 4: **Quantitative evaluation of the monocular depth on the KITTI Eigen split.** S: trained on stereo pairs. MS: trained on stereo videos. The best results are in **bold**.



RGB SDFA-Net [60] EMR-MSF (Ours)

Figure 6: **Visualization of estimated depth.** We compare our results with SDFA-Net [60].



(a) Top view (b) Front view

Figure 7: **Trajectories on Sequence 09 of KITTI Odometry benchmark**. Both the top view and front view are provided for better visualization.

Method	Seq.09		Seq.10	
	t_{err} (%) ↓	r_{err} (°/100) ↓	t_{err} (%) ↓	r_{err} (°/100) ↓
ORB-SLAM2 (w/o LC) [36]	10.03	0.29	3.64	0.32
ORB-SLAM2 (w LC) [36]	3.48	0.39	3.46	0.38
GeoNet [58]	39.43	14.30	28.99	8.85
Monodepth2 [15]	17.22	3.86	11.72	5.35
EPC++ [30]	8.84	3.34	8.86	3.18
LTMVO [61]	3.49	1.00	5.81	1.80
MLF-VO [25]	3.90	1.41	4.88	1.38
EMR-MSF (Ours)	3.49	0.78	3.11	1.04
EMR-MSF (Ours, aligned)	**3.30**	**0.78**	**2.35**	**1.04**

Table 5: **Quantitative evaluation of the visual odometry.** The best results are highlighted by **bold** style.

real scale. For a fair comparison, we provide both aligned and not aligned trajectories of our method in the table. As shown in Table 5, our method outperforms the previous self-supervised learning-based methods in all metrics, and even achieves better accuracy than traditional methods with loop closure in terms of the t_{err} metric. This demonstrates the effectiveness of our ego-motion aggregation module in improving the accuracy of visual odometry. We also provide a qualitative comparison of the estimated trajectories from our method, LTMVO [61], and MLF-VO [25] in Fig. 7. Our method yields trajectories with overall smaller drifts than the other methods.

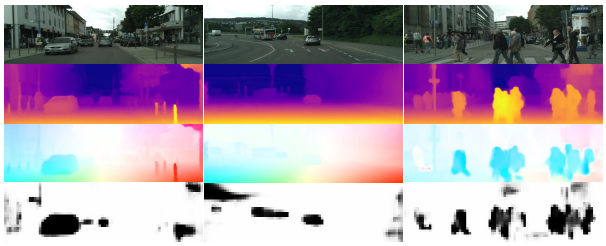


Figure 8: **Generalization test on Cityscapes [8].** From top to bottom: input first frame, estimated depth of first frame, synthesized optical flow, estimated rigidity soft mask.

4.5. Generalization Ability

We use the Cityscapes dataset [8] to test the generalization ability of our model trained on the KITTI dataset [14]. Several visual samples are provided in Fig. 8. Our method remarkably generalizes to unseen data, including some significantly dynamic scenes which are rarely present in the training data, such as the presence of numerous pedestrians crossing before the vehicle. More generalization examples

can be found in our supplementary material.

5. Conclusions

In this paper, we have proposed a novel self-supervised monocular method named EMR-MSF for scene flow estimation. Our methods incorporates a 3D geometry-oriented network architecture with novel designs to exploit ego-motion rigidity, which results in well-regularized scene flow estimations from solely monocular images. Our proposed approach demonstrates promising potential for monocular dynamic 3D perception and is capable of various computer tasks including scene flow, optical flow, depth, and ego-motion estimation.

References

[1] Tali Basha, Yael Moses, and Nahum Kiryati. Multi-view scene flow estimation: A view centered variational approach. *IJCV*, 101:6–21, 2013. 2

[2] Bayram Bayramli, Junhwa Hur, and Hongtao Lu. Raft-msf: Self-supervised monocular scene flow using recurrent optimizer. *arXiv preprint arXiv:2205.01568*, 2022. 1, 2, 4, 5, 7, 8

[3] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *Proc. CVPR*, 2017. 1, 2

[4] 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. *NIPS*, 2019. 4

[5] Fabian Brickwedde, Steffen Abraham, and Rudolf Mester. Mono-sf: Multi-view geometry meets single-view depth for monocular scene flow estimation of dynamic traffic scenes. In *Proc. ICCV*, 2019. 1, 2, 6, 7

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

[7] Wencan Cheng and Jong Hwan Ko. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *Proc. ECCV*, 2022. 1

[8] 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 *Proc. CVPR*, 2016. 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 5

[10] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Rigid scene flow for 3d lidar scans. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2016. 1

[11] Lihe Ding, Shaocong Dong, Tingfa Xu, Xinli Xu, Jie Wang, and Jianan Li. Fh-net: A fast hierarchical network for scene flow estimation on real-world point clouds. In *Proc. ECCV*, 2022. 1, 2

[12] Guanting Dong, Yueyi Zhang, Hanlin Li, Xiaoyan Sun, and Zhiwei Xiong. Exploiting rigidity constraints for lidar scene flow estimation. In *Proc. CVPR*, 2022. 1, 2

[13] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NIPS*, 2014. 6

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Intl. J. of Robotics Research*, 32(11):1231–1237, 2013. 8

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

[16] Juan Luis Gonzalez and Munchurl Kim. Plade-net: towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In *Proc. CVPR*, 2021. 8

[17] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *NIPS*, 2020. 8

[18] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proc. CVPR*, 2019. 1

[19] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *Proc. ICCV*, 2007. 2

[20] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Proc. CVPR*, 2020. 1, 2, 4, 5, 7, 8

[21] Junhwa Hur and Stefan Roth. Self-supervised multi-frame monocular scene flow. In *Proc. CVPR*, 2021. 1, 2, 4, 5, 7

[22] Mariano Jaimez, Mohamed Souiai, Jörg Stückler, Javier Gonzalez-Jimenez, and Daniel Cremers. Motion cooperation: Smooth piece-wise rigid scene flow from rgb-d images. In *Proc. 3DV*, 2015. 1

[23] Hualie Jiang, Laiyan Ding, Zhenglong Sun, and Rui Huang. Dipe: Deeper into photometric errors for unsupervised learning of depth and ego-motion from monocular videos. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2020. 4

[24] Huaizu Jiang, Deqing Sun, Varun Jampani, Zhaoyang Lv, Erik Learned-Miller, and Jan Kautz. Sense: A shared encoder network for scene-flow estimation. In *Proc. ICCV*, 2019. 1, 2

[25] Zijie Jiang, Hajime Taira, Naoyuki Miyashita, and Masatoshi Okutomi. Self-supervised ego-motion estimation based on multi-layer fusion of rgb and inferred depth. In *Proc. Intl. Conf. on Robotics and Automation*, 2022. 6, 8

[26] Yang Jiao, Trac D Tran, and Guangming Shi. Effiscene: Efficient per-pixel rigidity inference for unsupervised joint learning of optical flow, depth, camera pose and motion segmentation. In *Proc. CVPR*, 2021. 1, 2

[27] Liang Liu, Guangyao Zhai, Wenlong Ye, and Yong Liu. Unsupervised learning of scene flow estimation fusing with local rigidity. In *Proc. IJCAI*, 2019. 2

[28] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proc. CVPR*, 2019. 1, 2

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[30] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE PAMI*, 42(10):2624–2641, 2019. 1, 2, 7, 8

[31] Zhaoyang Lv, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James M Rehg, and Jan Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In *Proc. ECCV*, 2018. 1, 2

[32] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *Proc. CVPR*, 2019. 1, 2

[33] Lukas Mehl, Azin Jahedi, Jenny Schmalfuss, and Andrés Bruhn. M-fuse: Multi-frame fusion for scene flow estimation. In *Proc. WACV*, 2023. 1, 2, 4

[34] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018. 4

[35] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. CVPR*, 2015. 1, 6, 7

[36] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 2017. 7, 8

[37] 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. *NIPS*, 2019. 5

[38] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *Proc. ECCV*, 2020. 1, 2

[39] Yi-Ling Qiao, Lin Gao, Yu-Kun Lai, Fang-Lue Zhang, Ming-Ze Yuan, and Shihong Xia. Sf-net: Learning scene flow from rgb-d images with cnns. In *Proc. BMVC.*, 2018. 1, 2

[40] Zhile Ren, Deqing Sun, Jan Kautz, and Erik Sudderth. Cascaded scene flow prediction using semantic segmentation. In *Proc. 3DV*, 2017. 1, 2

[41] René Schuster, Oliver Wasenmuller, Georg Kuschk, Christian Bailer, and Didier Stricker. Sceneflowfields: Dense interpolation of sparse scene flow correspondences. In *Proc. WACV*, 2018. 1, 2

[42] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proc. CVPR*, 2021. 5

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proc. CVPR*, 2018. 2

[44] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. ECCV*, 2020. 2

[45] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proc. CVPR*, 2021. 1, 2, 3

[46] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *Proc. CVPR*, 2021. 5

[47] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE PAMI*, 13(04):376–380, 1991. 7

[48] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proc. ICCV*, 1999. 2

[49] Christoph Vogel, Stefan Roth, and Konrad Schindler. View-consistent 3d scene flow estimation over multiple frames. In *Proc. ECCV*, 2014. 2

[50] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proc. ICCV*, 2013. 2

[51] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a piecewise rigid scene model. *IJCV*, 115(1):1–28, 2015. 1, 2

[52] Guangming Wang, Yunzhe Hu, Zhe Liu, Yiyang Zhou, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. What matters for 3d scene flow network. In *Proc. ECCV*, 2022. 1, 2

[53] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proc. CVPR*, 2019. 1, 2, 3, 6

[54] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen. Flownet3d++: Geometric losses for deep scene flow estimation. In *Proc. WACV*, 2020. 1

[55] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. Pv-raft: point-voxel correlation fields for scene flow estimation of point clouds. In *Proc. CVPR*, 2021. 1, 2

[56] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *Proc. ECCV*, 2020. 1, 2

[57] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Proc. ECCV*, 2020. 1, 2

[58] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proc. CVPR*, 2018. 1, 2, 3, 7, 8

[59] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *Proc. BMVC.*, 2021. 8

[60] Zhengming Zhou and Qiulei Dong. Self-distilled feature aggregation for self-supervised monocular depth estimation. In *Proc. ECCV*, 2022. 3, 4, 5, 7, 8

[61] Yuliang Zou, Pan Ji, Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. Learning monocular visual odometry via self-supervised long-term modeling. In *Proc. ECCV*, 2020. 6, 8

[62] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proc. ECCV*, 2018. 1, 2, 3, 7