OAMaskFlow: Occlusion-Aware Motion Mask for Scene Flow

Xiongfeng Peng, Zhihua Liu, Weiming Li, Yamin Mao, Qiang Wang

Samsung R&D Institute China-Beijing, China
{xf.peng, zhihua.liu, weiming.li, yamin18.mao, qiang.w}@samsung.com

Abstract

The scene flow estimation methods make significant progress by estimating pixel-wise 3D motion on implicitly learning a motion embedding using an end-to-end differentiable optimization framework. However, the motion embedding learned implicitly is insufficient for grouping pixels into rigid object in challenging regions, such as occlusion and inconsistent multi-view geometric properties. To address this issue, we propose a novel method for estimating scene flow called OAMaskFlow, which has three novelties. Firstly, we propose the concept of occlusion-aware motion (OAM) mask and generate the ground truth annotation through the photometric and geometry consistency. Secondly, we propose to supervise the motion embedding with the OAM mask to learn informative and reliable motion representation of the scene. Finally, a 3D motion propagation module is proposed to propagate high-quality 3D motion from reliable pixels to the challenging occluded regions. Experiments show that our proposed OAMaskFlow has reduced the $\mathrm{EPE}_{3\mathrm{D}}$ metric by 21.0% on the FlyingThings3D dataset and decreased SF-all metric by 24.3% on the KITTI scene flow benchmark than the baseline method RAFT-3D. Furthermore, we apply our proposed OAM mask in simultaneous localization and mapping (SLAM) to improve a state-of-the-art method DROID-SLAM. In comparison, the ATE metric has decreased by 65.7% and 58.3% on the TartanAir monocular and stereo datasets respectively.

Introduction

Scene flow task involves estimating both 3D structure and 3D motion of a complex and dynamic scene between a pair of video frames. This task has attracted increasing attention in recent years due to its importance in various applications such as robotics, augmented reality and autonomous driving.

Some previous scene flow methods (Behl et al. 2017; Ma et al. 2019; Yang and Ramanan 2021) assume that the scene can be well approximated as a collection of rigidly moving objects, such that the scene flow can be approximated by estimating the rigid motions of individual components. These methods construct a modular network to estimate scene flow based on multiple sub-tasks. For example, DRISF (Ma et al. 2019) jointly trains optical flow and instance segmentation networks, and then infers the 3D motions by differentiating through Gauss-Newton updates. RigidMask (Yang and Ramanan 2021) predicts the optical flow and segmentation masks of the background and multiple rigidly moving objects, and then parameterize them by their 3D rigid transformations to update 3D scene flow. One limitation of these approaches is that the network requires additional training of an instance segmentation branch for scene flow estimation. Another limitation is that each sub-task is independent, making it difficult to utilize the complementary properties between the sub-tasks in a unified network for overall optimization.

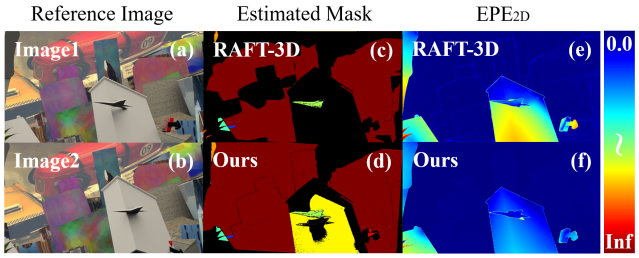

Figure 1: Visual comparison of our OAMaskFlow with RAFT-3D (Teed and Deng 2021b). (a) and (b) are input RGB image pair. We convert the motion embedding of RAFT-3D to mask and show it in (c). Our predicted OAM mask is shown in (d). Our OAM mask distinguishes between occluded pixels (black), static pixels (dark red), and dynamic objects (all other pixels). In contrast, the black pixels in the RAFT-3D method may be caused by either occlusion or large variance of motion embeddings, such as the black part in the middle of (c). (e) and (f) show the 2D end-point error ($\mathrm{EPE}_{2\mathrm{D}}$).(Best viewed in color.)

Recently, RAFT-3D (Teed and Deng 2021b) method proposes to learn dense rigid-motion embeddings and iteratively updates the dense SE3 motion field in an end-to-end differentiable architecture. Based on RAFT-3D, M-FUSE (Mehl et al. 2023) proposes to estimate the scene flow with multiple frame fusion of forward and backward flow. EMR-MSF (Jiang and Okutomi 2023) proposes a self-supervised monocular scene flow estimation by exploiting ego-motion rigidity and an ego-motion aggregation module. However, these methods implicitly learn motion embedding to softly group a pixel with its neighborhood. The learned motion embedding is not supervised which easily breaks the

3D motion consistency of the same rigid object, particularly for the pixels in challenging regions that are occluded or have poor matching quality to learn informative motion representation.

In this paper, we propose a novel scene flow estimation method called OAMaskFlow, which is an end-to-end differentiable optimization architecture with occlusion-aware motion (OAM) mask. Our proposed the concept of OAM mask differs from the instance segmentation mask in that it discriminates the pixels in the scene in terms of motion and occlusion instead of instance. For example, two different static instances will have the same label in the OAM mask, and the same instance will have different labels due to occlusion between two frames. We generate the ground truth annotation for the OAM mask through the constraints of photo-metric consistency and geometry consistency. With the OAM mask, we explicitly supervise the dense motion embedding to learn informative motion representation for grouping the neighbouring pixels of the same motion. To improve the accuracy of 3D motion in challenging regions, we propose a motion propagation module to spread the reliable 3D motion to these regions, such as occluded dynamic object regions and occluded static regions with an attention network. Finally, our proposed OAM mask can be applied in various other vision geometry estimation tasks, such as optical flow estimation, simultaneous localization and mapping (SLAM), etc., to help improving the performance.

Fig. 1 illustrates the estimated mask and error map of our OAMaskFlow and RAFT-3D (Teed and Deng 2021b). We convert motion embedding of the RAFT-3D algorithm to mask and visualize in (c). Our predicted OAM mask shows in (d). Our OAM mask distinguishes between occluded pixels (black), static pixels (dark red), and dynamic objects (all other pixels). In contrast, the black pixels in the RAFT-3D method may be caused by either occlusion or large variance of motion embeddings. It can be seen that our OAM mask is more complete than RAFT-3D. The 2D end-point error maps are shown in (e), (f) of the last column. It can be seen that RAFT-3D results have large 2D errors while ours are close to zero in most regions.

Our main contributions are summarized as follows:

- We propose a novel OAMaskFlow method to learn informative and reliable motion embedding by explicitly supervising the motion embedding with OAM mask in an end-to-end scene flow estimation network.
- The generated OAM mask is widely applicable to other vision geometry estimation tasks such as optical flow estimation and SLAM.
- We propose an attention module to propagate accurate and reliable scene flow estimation to the challenging occluded regions.
- Our method achieves state-of-the-art performance. Compared to the baseline RAFT-3D method (Teed and Deng 2021b), our OAMaskFlow reduces $\mathrm{EPE}_{3\mathrm{D}}$ metric by 21.0% on the FlyingThings3D dataset and SF-all metric by 24.3% on the KITTI scene flow benchmark. When used for SLAM, the ATE metric is reduced by 65.7% and 58.3% respectively on the TartanAir monocular and stereo datasets in comparison to the baseline DROID-SLAM method (Teed and Deng 2021a).

Related Work

Optical Flow Optical flow is the problem of estimating dense 2D pixel-level motion between a pair of frames. Recently, a large variety of network architectures have been proposed for the task (Hur and Roth 2019; Ilg et al. 2017; Jiang et al. 2021; Sun et al. 2018; Teed and Deng 2020; Xu et al. 2021; Zhang et al. 2021; Xu et al. 2022). Among these methods, the two most representative methods are the coarse-to-fine method PWC-Net (Sun et al. 2018) and the iterative refinement method RAFT (Teed and Deng 2020). They perform some sort of multi-stage refinements, either at multiple scales (Sun et al. 2018) or a single resolution (Teed and Deng 2020). For flow prediction at each stage, their pipelines are conceptually similar, which regress optical flow from a local cost volume with convolutions.

Scene Flow Scene flow is similar to optical flow, except that scene flow estimates a dense motion field in 3D space while the optical flow task is defined in 2D space. According to different inputs, some methods estimate sparse scene flow from point clouds, while others focus on estimating dense pixel-wise scene flow from RGB-D inputs.

Some methods (Liu, Qi, and Guibas 2019; Gu et al. 2019; Lang et al. 2023; Wang et al. 2020b; Wu et al. 2019; Shen et al. 2023; Wang et al. 2022) directly process point clouds and predict 3D scene flow. CamLiFlow (Liu et al. 2022) proposes a multi-stage and bidirectional fusion pipeline for based on camera-LiDAR fusion. CamLiFlow projects a small number of sparse points onto the image plane to fuse features from these two modalities. Instead, DELFlow (Peng et al. 2023) takes all points as input using the dense format of projected point clouds, which eliminates the density gap between sparse points and dense pixels, enabling effective feature fusion. CamLiRAFT (Liu et al. 2023) combines CamLiFlow and RAFT (Teed and Deng 2020) methods using the recurrent all-pairs field transforms.

There are also some methods to estimate scene flow from a pair of stereo or RGB-D frames. These methods (Behl et al. 2017; Ma et al. 2019; Yang and Ramanan 2020, 2021) divide scene flow estimation into multiple subtasks and build a modular network. One limitation of these approaches is that the network requires additional training of an instance segmentation branch for scene flow estimation. Another limitation is that each sub-task is independent, making it difficult to utilize their complementary properties in a unified network for overall optimization. RAFT-3D method (Teed and Deng 2021b) is an end-to-end differentiable architecture to iteratively update a dense field of pixel-wise SE3 motion from a pair of RGB-D frames. Based on RAFT-3D, M-FUSE (Mehl et al. 2023) proposes to estimate the scene flow with multiple frames fusion and EMR-MSF (Jiang and Okutomi 2023) proposes a self-supervised monocular scene flow estimation. However, these methods implicitly learn motion embedding to softly group the pixel with its neighborhood. The learned motion embedding is not supervised which easily break the 3D motion consistency of the same rigid object.

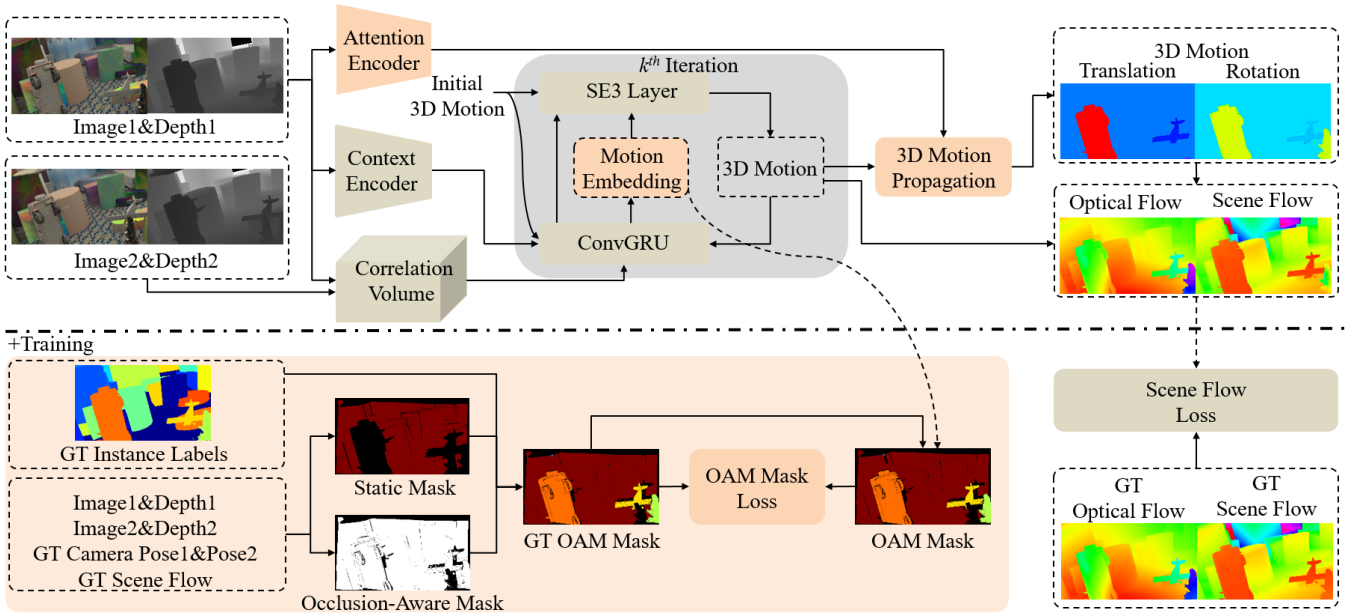

Figure 2: Overview of our OAMaskFlow pipeline with supervision. We highlight the different parts between our OAMaskFlow and RAFT-3D (Teed and Deng 2021b) in orange red, including 1) supervised motion embedding with our generated ground truth OAM mask, 2) 3D motion propagation module to propagate the reliable 3D motion to the occluded regions with attention.

Our Method

Fig. 2 illustrates our OAMaskFlow pipeline. Given a pair of synchronized RGB $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ and depth frames $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, the network outputs a dense transformation field $\boldsymbol{T} \in SE(3)^{H\times W}$ that represents the 3D motion between the pair of frames. Each 3D motion can be decomposed into a rotation component and a translation component, and it can be projected onto the image to recover dense optical flow and dense scene flow of the pair. At the beginning, the SE3 motion field $\boldsymbol{T}$ is initialized with an identity rotation and zero translation matrix. Different from RAFT-3D (Teed and Deng 2021b), there are two main differences in OAMaskFlow network. One is that we generate the ground truth OAM mask to supervise the motion embedding in the training stage. The other is the 3D motion propagation module to propagate the reliable 3D motion to the challenging regions with a learned attention feature. We introduce them as follows.

OAM Mask Generation

To enhance the pixel-wise motion representation, it's necessary to supervise the motion embeddings of each pixel to ensure 3D motion consistency with its neighborhood. However, it is non-trivial to supervise the motion embedding because there are only 3D scene flow labels instead of dense motion labels in the scene. It's also not easy to convert the scene flow labels into the motion labels.

To alleviate the problem, we compromise by generating a motion mask to distinguish the motion of the scene and then use it to supervise the motion embedding. From a 3D motion perspective, a scene can be divided into static regions and dynamic objects. Static regions include the background and some static objects that have the same motion as the camera ego-motion. Dynamic objects move differently from each other, and each pixel on the same rigid object has the same motion. The challenging occluded regions from two different perspectives are also considered to improve the representation of motion embedding. In the following sections, we introduce how to generate the static mask, occlusion-aware mask and OAM mask under the constraint of geometric and photo-metric consistency.

Static Mask As the camera ego-motion and static regions share the same motion, the matching pairs exhibit identical intensity and depth when one image is warped to the other using the ground truth camera poses. This consistency of depth and intensity is used to identify static regions in the scene.

Assuming that the ground truth camera poses $\boldsymbol{T}_1, \boldsymbol{T}_2$ of frame $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ are known, each pixel $\boldsymbol{x}$ on image $\boldsymbol{I}_1$ can find its correspondence $\boldsymbol{x'}$ on $\boldsymbol{I}_2$, where $\boldsymbol{x'} = \pi(\boldsymbol{T}_2\boldsymbol{T}_1^{-1}\pi^{-1}(\boldsymbol{x}, \boldsymbol{D_1}(\boldsymbol{x})))$. We evaluate the photo-metric consistency of matching pair $(\boldsymbol{x}, \boldsymbol{x'})$ via Eq. 1 and calculate the depth consistency by comparing the projected depth $\boldsymbol{d'}$ with the value $\boldsymbol{D}_2(\boldsymbol{x}')$ from depth image $\boldsymbol{D}_2$ in Eq. 2. If both errors are less than given thresholds, then the pixel satisfies the depth and intensity consistency and is considered as static. Or else, it is dynamic. Here, The projection function $\pi(\cdot)$ maps a 3D point $\boldsymbol{X}$ to its projected pixel coordinates $\boldsymbol{x}$. In this way, we get static mask $\boldsymbol{m}^{static}$ and express in Eq. 3. The hyperparameters Th_1, Th_2 are set $25, 0.005$ respectively.

$$\boldsymbol{E}^p(\boldsymbol{x}) = ||\boldsymbol{I}_1(\boldsymbol{x}) - \boldsymbol{I}_2(\boldsymbol{x'})||_2 \quad (1)$$

$$\boldsymbol{E}^d(\boldsymbol{x}) = ||\boldsymbol{D}_2(\boldsymbol{x'}) - \boldsymbol{d'}||_2 \quad (2)$$

$$\boldsymbol{m}^{static}(x) = \begin{cases} 1 & \boldsymbol{E}^p(x) < Th_1 \; and \; \boldsymbol{E}^d(x) < Th_2 \\ 0 & other \end{cases} \tag{3}$$

With the static mask $\boldsymbol{m}^{static}$, we can identify the static pixels from the scene. An example of a static mask is shown in Fig. 2. It is important to note that some static rigid objects may have the same labels as the background.

Occlusion-Aware Mask To maintain 3D motion consistency of a rigid object from two different perspectives, the occluded region should be excluded from the valid region because it does not match well between two different perspectives and does not provide an informative motion representation for rigid objects.

Given the ground truth of scene flow $\boldsymbol{f}^{gt}$, for each pixel $\boldsymbol{x} = (u, v)$ in image $\boldsymbol{I}_1$, we get a matched pair $(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{f}^{gt})$ between $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. If the pair is visible in both views, they have consistent intensity, or else, it is occluded by the foreground objects or out of image boundary. Therefore, we identify the occluded pixels by evaluating the photo-metric error E^p of the matching pair which is described with Eq. 4. Eq. 5 represents occlusion-aware mask $\boldsymbol{m}^{occ}$.

$$\boldsymbol{E}^p(\boldsymbol{x}) = ||\boldsymbol{I}_1(\boldsymbol{x}) - \boldsymbol{I}_2(\boldsymbol{x} + \boldsymbol{f}^{gt})||_2 \tag{4}$$

$$\boldsymbol{m}^{occ}(x) = \begin{cases} 1 & \boldsymbol{E}^p(x) < Th_1 \\ 0 & other \end{cases} \tag{5}$$

Fig. 2 shows an example of occlusion mask and the occluded regions are marked in black. The majority of occluded regions occur at the intersection of the foreground and background regions.

OAM Mask With the static mask $\boldsymbol{m}^{static}$ and the occlusion-aware mask $\boldsymbol{m}^{occ}$, we can roughly discriminate each pixel in the scene based on occlusion and motion. To differentiate between various dynamic objects, which may have different motions, the ground truth instance labels are effective. Assuming the ground truth instance labels $\boldsymbol{O}^{gt} \in \{1, ..., k, ..., K\}$, we set the motion label of each dynamic object as the object class label and the OAM mask $\boldsymbol{m}^{OAM}$ is described as in Eq. 6. Here we set the motion label of the static class to a fixed value of $K + 1$.

The scene is classified into three categories using the OAM mask $\boldsymbol{m}^{OAM}$. The first category includes non-occluded dynamic objects, each with potentially different motion from each other and the camera ego-motion. The second category comprises non-occluded static regions with the same motion as the camera ego-motion. This includes both the non-occluded background and non-occluded static objects. The final category pertains to challenging occluded regions, where pixels cannot be simultaneously viewed from two different perspectives, and the motion label is set to 0.

$$\boldsymbol{m}^{OAM}(x) = \begin{cases} \boldsymbol{O}^{gt}(x) & \boldsymbol{m}^{static}(x) = 0 \; and \; \boldsymbol{m}^{occ}(x) = 1 \\ K + 1 & \boldsymbol{m}^{static}(x) = 1 \; and \; \boldsymbol{m}^{occ}(x) = 1 \\ 0 & other \end{cases} \tag{6}$$

Fig. 3 illustrates the generated ground truth OAM mask examples of two datasets FlyingThings3D (Mayer et al. 2016) and TartanAir (Wang et al. 2020a) respectively. It is noticeable that some instance objects share the same label as the background because of their static attributes. The OAM embedding can be supervised explicitly using the mask $\boldsymbol{m}^{OAM}$. Next section introduces how to generate the predicted OAM mask with the OAM embedding.

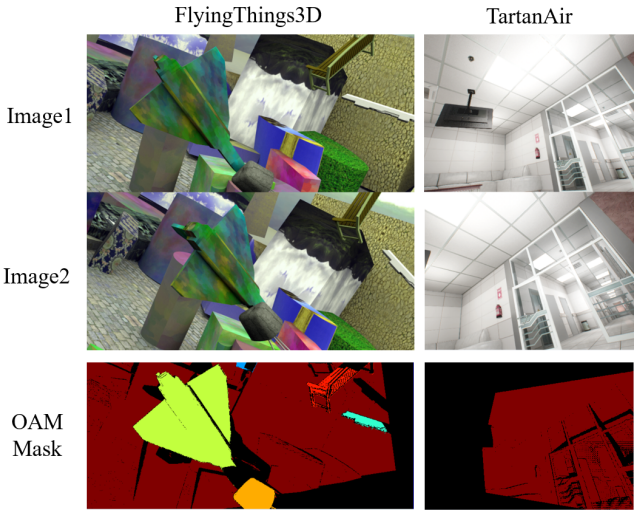

Figure 3: Ground truth OAM mask examples of FlyingThings3D and TartanAir datasets.

Given two OAM embedding vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, similar as the baseline RAFT-3D method (Teed and Deng 2021b), we compute an affinity and weighted it on the re-projection error for further optimization in Dense-SE3 layer. Please refer to the RAFT-3D paper for details.

Supervision with OAM Mask

With the learned OAM embedding $\boldsymbol{v}^{est} \in R^{D \times H \times W}$, in this section, we introduce the OAM mask loss to supervise the embedding with the ground truth OAM mask and the total loss of OAMaskFlow.

OAM Mask Loss We firstly compute the predicted mean embedding of each category. For easy computing, the ground truth OAM mask m^{gt} is decomposed into K binary mask $\boldsymbol{m}_k^{gt}$. For each binary mask, pixels with mask value 1 belong to the same motion category, otherwise they are equal to 0. We add up the embeddings of the same motion category and then normalize them to get the mean embedding $\boldsymbol{t}_k^{est}$ in Eq. 7. Next, we compute the similarity between the learned OAM embedding and the mean embedding to get the predicted binary OAM mask m_k^{est} in Eq. 8. The higher the similarity, the greater the probability of belonging to the same motion. Finally, to bring pixels with the same category closer together, pixels with different categories are separated by a large margin. The OAM mask loss is defined as the cross entropy loss between the predicted OAM mask m^{est} and the ground truth OAM mask m^{gt} in Eq. 9.

$$\boldsymbol{t}_k^{est} = \frac{1}{||\boldsymbol{m}_k^{gt}||_1} \sum_{H,W} \boldsymbol{m}_k^{gt} \boldsymbol{v}^{est} \tag{7}$$

$$\boldsymbol{m}_k^{est} = 2 * \sigma(-||\boldsymbol{v}^{est} - \boldsymbol{t}_k^{est}||^2) \tag{8}$$

$$L_{mask} = -||(\boldsymbol{m}^{gt}\log(\boldsymbol{m}^{est}) + (1-\boldsymbol{m}^{gt})\log(1-\boldsymbol{m}^{est}))||_1 \tag{9}$$

Total Loss In OAMaskFlow network, we calculate losses for both the final and intermediate estimations from the recurrent structure. The total loss L_{total} is a weighted sum of scene flow loss and mask loss. The hyperparameter w is set 0.1.

$$L_{total} = \sum_{i=1}^{N+1} \zeta^{N+1-i} L_{flow}^i + w * \sum_{i=1}^{N} \zeta^{N-i} L_{mask}^i \tag{10}$$

where N is the iteration number, ζ is the weight decay factor, scene flow loss is defined as the L1 loss of $L_{flow} = ||\boldsymbol{f}^{gt} - \boldsymbol{f}^{est}||_1$. Each 3D motion $\boldsymbol{T}_k$ can be projected onto the image to recover the flow by $\boldsymbol{f}_k^{est} = \pi(\boldsymbol{T}_k \pi^{-1}(\boldsymbol{x})) - \boldsymbol{x}$.

3D Motion Propagation

To get reliable 3D motions for the challenging occluded regions, we propose a 3D motion propagation module that propagates the accurately estimated 3D motion from the non-occluded regions to the occluded regions. As pixels in occluded regions may have a similar appearances or structures to their non-occluded neighborhood, we extract attention feature $\boldsymbol{F}$ from the reference image to measure the feature self-similarity globally. The feature similarity for each pixel in $\boldsymbol{F}$ with respect to all pixels in the image is computed using a simple matrix multiplication to calculate their correlations. The final SE3 field $\widetilde{\boldsymbol{T}}$ is computed with Eq. 11 by weighting the normalized similarity on the SE3 field $\boldsymbol{T}$. Here $\sqrt{D}$ is a normalization factor to avoid large values after the dot-product operation.

$$\widetilde{\boldsymbol{T}} = softmax(\frac{\boldsymbol{F}\boldsymbol{F}^T}{\sqrt{D}})\boldsymbol{T} \tag{11}$$

We refer to GMFlow (Xu et al. 2022) for 3D motion propagation module, but there are two differences. One is propagation on different tasks. GMFlow is used to estimate 2D flow while we estimate 3D motion(translation and rotation). The other is propagation on different regions. Our OAMaskFlow propagates reliable motion to several fine-grained categories, such as occluded, non-occluded background and non-occluded dynamic instance regions. While in GMFlow, there is no detailed division of the non-occluded regions.

Experiments

To validate our OAMaskFlow method, we perform experiments on the synthetic FlyingThings3D (Mayer et al. 2016) and the real-world KITTI (Menze and Geiger 2015) datasets. FlyingThings3D consists of stereo and RGB-D images rendered with multiple moving objects along randomized 3D trajectories from ShapeNet (Chang et al. 2015). It is the most diverse and challenging, and contains dense, accurate, and multi-task ground truth. KITTI consists of autonomous stereo images with sparse and multi-task ground truth.

Furthermore, we extensively apply OAM mask into DROID-SLAM framework (Teed and Deng 2021a) and make experiments on the synthetic TartanAir SLAM challenge (Wang et al. 2020a) dataset to validate our proposed OAM mask effectiveness. The SLAM challenge corresponding to the synthetic TartanAir dataset is one of the official challenges in the CVPR 2020 SLAM workshop. The synthetic TartanAir dataset contains dense, accurate, and multi-task ground truth.

FlyingThings3D

Data Preprocessing Following previous work (Teed and Deng 2021b), we use the FlyingThings3D dataset to train the model. The training set is augmented by randomly cropping and resizing, and the intrinsic values are adjusted according to the training data. Following FlowNet3D (Liu, Qi, and Guibas 2019), approximately 2000 test examples are sampled from the FlyingThings3D test set for evaluation.

Training The FlyingThings3D dataset provides not only the ground truth of the scene flow, but also the ground truth of the instance labels and camera poses. This is convenient for computing the ground truth of the OAM mask. We train our network for 200k iterations with a batch size of 4 and a crop size of [320, 720]. The initial learning rate is 2×10^{-4} and decays linearly during training.

Evaluation Metrics Following RAFT-3D (Teed and Deng 2021b), we evaluate our network with 2D end-point error ($\mathrm{EPE_{2D}}$) and 3D end-point error ($\mathrm{EPE_{3D}}$), as well as threshold metrics ($\mathrm{ACC_{1px}}$, $\mathrm{ACC_{.05m}}$ and $\mathrm{ACC_{.1m}}$, which measure the proportion of error within a threshold.

Quantitative Comparison The quantitative results on the FlyingThings3D dataset are listed in Table 1. We notice that $\mathrm{EPE_{2D}}$ and $\mathrm{EPE_{3D}}$ of our OAMaskFlow have decreased by 20.4% and 21.0% respectively in comparison with the baseline RAFT-3D method which verifies the proposed OAM mask and motion propagation effectiveness. In our OAMaskFlow network, we modify the context encoder branch with the same structure as the correlation encoder branch except that the output dimension is 512. With this design, the parameters of the entire network have decreased from 45.8M to 7.7M, reducing parameters by 83.2%. In addition, our OAMaskFlow also exceeds the best CamLiRAFT method in 2D and 3D threshold metrics with less parameters.

KITTI

Training and Testing With limited data on the KITTI scene flow benchmark, we finetune our pretrained model on KITTI benchmark for an additional 50k iterations with a learning rate of 1×10^{-4}. We crop the image to [288, 960] and perform spatial and photo-metric augmentation. Since there is no depth on the KITTI scene flow benchmark, we use GA-Net (Zhang et al. 2019) to generate the depth map as our input. Additionally, the KITTI benchmark lacks the ground truth of camera pose, and does not have accurate depth estimation and instance segmentation results. Therefore, we freeze the parameters of motion embedding branch when finetuning the model. It also means that our method,

Method	Input	2D Metrics		3D Metrics		Parameters
		$EPE_{2D}\downarrow$	$ACC_{1px}\uparrow$	$EPE_{3D}\downarrow$	$ACC_{.05m}\uparrow$	
FlowNet3D(Liu, Qi, and Guibas 2019)	Point Clouds	-	-	0.169	25.4%	1.2M
PointPWC-Net(Wu et al. 2019)	Point Clouds	-	-	0.132	44.3%	7.7M
FLOT(Puy, Boulch, and Marlet 2020)	Point Clouds	-	-	0.156	34.3%	-
CamLiFlow(Liu et al. 2022)	RGB+Point Clouds	2.18	84.3%	0.061	85.6%	7.7M
DELFlow(Peng et al. 2023)	RGB+Point Clouds	2.02	85.9%	0.058	86.7%	-
CamLiRAFT(Liu et al. 2023)	RGB+Point Clouds	**1.73**	87.5%	**0.049**	88.4%	8.4M
FlowNet2(Ilg et al. 2017)	RGB	5.05	72.8%	-	-	162.5M
PWC-Net(Sun et al. 2018)	RGB	6.55	64.3%	-	-	9.4M
RAFT(Teed and Deng 2020)	RGB	3.12	81.1%	-	-	5.3M
RAFT-3D(Teed and Deng 2021b)	RGB+Depth	2.45	86.3%	0.062	87.8%	45.8M
Our OAMaskFlow	RGB+Depth	1.95	**89.0%**	**0.049**	**89.7%**	7.7M

Table 1: Quantitative comparison results with the state-of-the-art methods on the FlyingThings3D dataset.

Method	Input	Disparity 1			Disparity 2			Optical Flow			Scene Flow		
		bg↓	fg↓	all↓	bg↓	fg↓	all↓	bg↓	fg↓	all↓	bg↓	fg↓	all↓
OpticalExp(Yang and Ramanan 2020)	RGB+Point Clouds	1.48	3.46	1.81	3.39	8.54	4.25	5.83	8.66	6.30	7.06	13.44	8.12
ISF(Behl et al. 2017)	RGB+Point Clouds	4.12	6.17	4.46	4.88	11.34	5.95	5.40	10.29	6.22	6.58	15.63	8.08
CamLiFlow(Liu et al. 2022)	RGB+Point Clouds	-	-	1.81	-	-	3.19	-	-	4.05	-	-	5.62
CamLiFlow(Background)(Liu et al. 2022)	RGB+Point Clouds	1.48	3.46	1.81	1.92	8.14	2.95	2.31	**7.04**	3.10	2.87	12.23	4.43
DELFlow(Background)(Peng et al. 2023)	RGB+Point Clouds	**1.40**	**2.91**	**1.65**	**1.90**	**7.50**	**2.84**	2.27	7.10	3.07	2.87	**11.69**	4.34
CamLiRAFT(Liu et al. 2023)	RGB+Point Clouds	-	-	1.81	-	-	3.02	-	-	3.43	-	-	4.97
CamLiRAFT(Background)(Liu et al. 2023)	RGB+Point Clouds	1.48	3.46	1.81	1.91	8.11	2.94	**2.08**	7.37	**2.96**	2.68	12.16	4.26
SSF(Ren et al. 2017)	RGB	3.55	8.75	4.42	4.94	17.48	7.02	5.63	14.71	7.14	7.18	24.58	10.07
Sense(Jiang et al. 2019)	RGB	2.07	3.01	2.22	4.90	10.83	5.89	7.30	9.33	7.64	8.36	15.49	9.55
PRSM(Vogel, Schindler, and Roth 2015)	RGB	3.02	10.52	4.27	5.13	15.11	6.79	5.33	13.40	6.68	6.61	20.79	8.97
ACOSF(Li, Ma, and Liao 2021)	RGB	2.79	7.56	3.58	3.82	12.74	5.31	4.56	12.00	5.79	5.61	19.38	7.90
DRISF(Ma et al. 2019)	RGB	2.16	4.49	2.55	2.90	9.73	4.04	3.59	10.40	4.73	4.39	15.94	6.31
RigidMask(Yang and Ramanan 2021)	RGB+Depth	1.53	3.65	1.89	2.09	8.92	3.23	2.63	7.85	3.50	3.25	13.08	4.89
RAFT-3D(Teed and Deng 2021b)	RGB+Depth	1.48	3.46	1.81	2.51	9.46	3.67	3.39	8.79	4.29	4.27	13.27	5.77
M-FUSE(Mehl et al. 2023)	RGB+Depth	**1.40**	**2.91**	**1.65**	2.14	8.10	3.13	2.66	7.47	3.46	3.43	11.84	4.83
Our OAMaskFlow	RGB+Depth	1.48	3.46	1.81	**1.90**	7.72	2.87	2.32	7.68	3.21	2.86	11.93	4.37
Our OAMaskFlow(Background)	RGB+Depth	1.48	3.46	1.81	**1.90**	7.80	2.89	2.10	7.80	3.05	**2.64**	12.05	**4.21**

Table 2: Quantitative comparison results on KITTI Scene Flow benchmark. 'bg', 'fg' and 'all' denote background regions, foreground regions and all pixels respectively. The value is the percentage of erroneous pixels with lower being better. 'Background' refers to the results obtained after background refinement.

like the RAFT-3D, does not use other ground truth labels on KITTI except scene flow labels.

Quantitative Comparison We quantitatively compares our OAMaskFlow with the state-of-the-art methods on KITTI2015 scene flow benchmark on Table 2. It can be seen that, with the OAM and motion propagation modules, the percentage of erroneous pixels in 'all' regions has reduced from 5.77% to 4.37% in comparison with the baseline, which verifies importance of OAM and motion propagation modules. Following the CamLiFlow(Liu et al. 2022) and CamLiRAFT(Liu et al. 2023) methods, we perform additional background refinement on our results. Overall, we can see that our OAMaskFlow achieves the lowest scene flow error on both all pixels and background regions with the background refinement strategy on the KITTI2015 scene flow benchmark leaderboard. This result also shows that motion embedding branch has a good generalization capability from synthetic to real-world.

TartanAir

OAM-DROID-SLAM Furthermore, we extensively apply OAM mask into DROID-SLAM (Teed and Deng 2021a) by weighting the dense flow and make experiments on TartanAir SLAM challenge (Wang et al. 2020a) dataset to validate our OAM mask effectiveness. Similar to DROID-SLAM, we use pose loss and flow loss to supervise the network, except that we also use OAM loss with OAM mask, and name it OAM-DROID-SLAM. In SLAM task, all dynamic objects are outliers and only static mask via Eq. 3 contributes to the performance. Therefore, we use the same ground truth labels as DROID-SLAM and do not need to use additional instance segmentation labels.

Training We train OAM-DROID-SLAM network on four Nvidia 3090 GPUs with two stages on TartanAir (Wang et al. 2020a) dataset. TartanAir dataset is a synthetic dataset that collects in photo-realistic simulation environments with various light conditions, weather and moving objects. It covers 30 scenarios of synthetic world and has the stereo RGB image, depth image, ground truth segmentation, ground truth optical flow, and ground truth camera poses. In the first stage, similar to DROID-SLAM, the network is trained on monocular video stream with 384×512 resolution for 250k steps. In the second stage, we freeze other parameters and then only learn the OAM mask branch for 50k steps.

Quantitative Comparison Table 3 shows the trajectory error comparison of OAM-DROID-SLAM with the state-of-the-art methods on all hard sequences of TartanAir monocular SLAM challenge. We evaluate absolute trajectory er-

	Mono(ATE(m))↓	MH000	MH001	MH002	MH003	MH004	MH005	MH006	MH007	Avg
VO	DeepV2D(Teed and Deng 2018)	6.15	2.12	4.54	3.89	2.71	11.55	5.53	3.76	5.03
	TartanVO(Wang, Hu, and Scherer 2021)	4.88	0.26	2.00	0.94	1.07	3.19	1.00	2.04	1.92
	DROID-VO(Teed and Deng 2021a)	0.49	**0.08**	**0.10**	**0.06**	1.79	**0.53**	0.14	0.57	0.47
	OAM-DROID-VO	**0.35**	**0.08**	0.11	**0.06**	**0.05**	0.62	**0.09**	**0.30**	**0.21**
SLAM	ORB-SLAM(Mur-Artal, Montiel, and Tardos 2015)	1.30	0.04	2.37	2.45	-	-	21.47	2.73	-
	DROID-SLAM(Teed and Deng 2021a)	**0.07**	0.05	0.03	**0.03**	1.81	**0.51**	0.15	0.15	0.35
	OAM-DROID-SLAM	**0.07**	**0.04**	**0.02**	0.04	**0.01**	0.61	**0.09**	**0.06**	**0.12**

Table 3: Quantitative comparison of our method with the state-of-the-art methods on TartanAir monocular SLAM challenge.

	Stereo(ATE(m))↓	SH000	SH001	SH002	SH003	SH004	SH005	SH006	SH007	Avg
VO	TartanVO(Wang, Hu, and Scherer 2021)	2.52	1.61	3.65	0.29	3.36	4.74	3.72	3.06	2.87
	DROID-VO(Teed and Deng 2021a)	0.08	**0.20**	**0.09**	**0.01**	0.39	0.21	0.23	**0.03**	0.15
	OAM-DROID-VO	**0.07**	**0.20**	0.11	**0.01**	**0.30**	**0.13**	**0.19**	0.04	**0.13**
SLAM	ORB-SLAM2(Mur-Artal and Tardós 2017)	0.05	6.67	-	-	-	-	**0.10**	-	-
	DROID-SLAM(Teed and Deng 2021a)	**0.02**	0.06	0.05	**0.01**	0.13	0.44	0.22	**0.01**	0.12
	OAM-DROID-SLAM	**0.02**	**0.05**	**0.04**	**0.01**	**0.05**	**0.10**	0.14	**0.01**	**0.05**

Table 4: Quantitative comparison of our method with the state-of-the-art methods on TartanAir stereo SLAM challenge.

ror (ATE) (Sturm et al. 2012) after optimizing the scale (Sim3). It can be seen that the ATE of visual odometry (VO) and SLAM with the OAM mask have reduced by 55.3% and 65.7% respectively than the baseline DROID-SLAM method (Teed and Deng 2021a).

Table 4 shows the trajectory error comparison of OAM-DROID-SLAM with the state-of-the-art methods on all hard sequences of TartanAir stereo SLAM challenge. It can be seen that the ATE of VO and SLAM with the OAM mask have reduced by 13.3% and 58.3% respectively than the baseline DROID-SLAM method (Teed and Deng 2021a).

Ablation Study

We conduct ablation studies to verify our proposed modules on the FlyingThings3D dataset. Table 5 validates the effectiveness of our proposed OAM mask and motion propagation module. For fast validation, RAFT-3D without Laplacian module is used as our baseline. We supervise motion embedding with the OAM mask and then add the motion propagation module. We notice that EPE_{2D} and EPE_{3D} are gradually reduced with our proposed components. Especially, our OAMaskFlow method decreases EPE_{2D} and EPE_{3D} by 19.7% and 16.1% in comparison to the baseline RAFT-3D method on the FlyingThings3D dataset. Fig. 4 shows the comparison between the estimated mask, the estimated OAM mask and the ground truth OAM mask on the FlyingThings3D dataset. It is observed that the estimated OAM mask is complete and consistent with the ground truth. Our OAM mask distinguishes between occluded pixels (black), static pixels (dark red), and dynamic objects (all other pixels). In contrast, the black pixels in the baseline RAFT-3D method may be caused by either occlusion or large variance of motion embeddings. We highlight the large difference regions with yellow rectangles.

Conclusion

In this paper, we propose a novel OAMaskFlow method to learn motion consistency representative with the supervision of generated OAM mask in an end-to-end scene flow estimation network. We further extensively applies the OAM mask in DROID-SLAM framework by weighting the dense flow. The experimental results validate the effectiveness of our proposed OAM mask in the scene flow and SLAM tasks. One limitation of our approach is that if the entire object is occluded, we cannot accurately estimate the scene flow of the object due to insufficient measurements. The other limitation of our approach is that the depth data is not fully explored. In our approach, we simply concatenate RGB and depth image as input, similar to the baseline RAFT3D, which limits ability to utilize most 3D structural information from depth. In the future, one promising research direction is the scene flow estimation from the image and point cloud with the OAM mask. The other is the scene flow estimation with 3D Gaussian Splatting.

Components	2D Metrics		3D Metrics		
	EPE_{2D}↓	ACC_{1px}↑	EPE_{3D}↓	$\mathrm{ACC}_{.05m}$↑	$\mathrm{ACC}_{.1m}$↑
Baseline(w/o Laplacian)	2.39	85.9%	0.062	85.1%	89.5%
+ OAM mask	2.01	87.4%	0.054	86.9%	90.8%
+ OAM mask + propagation	**1.92**	**87.5%**	**0.052**	**87.3%**	**91.1%**

Table 5: Ablation study on the FlyingThings3D dataset. Based on RAFT-3D (Teed and Deng 2021b), we supervise the motion embedding with OAM mask and then add the motion propagation (propagation) module. It can be seen that EPE_{2D} and EPE_{3D} are decreased gradually with our proposed components.

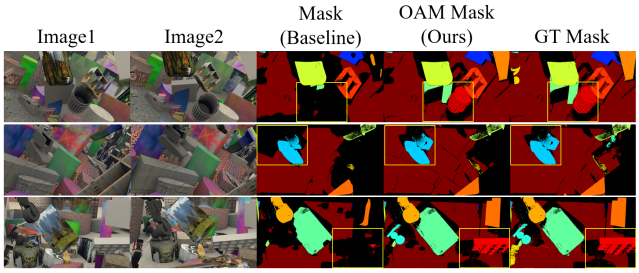

Figure 4: The comparison between the estimated mask, the estimated OAM mask and the OAM mask of ground truth on the FlyingThings3D dataset. We highlight the large difference regions with yellow rectangles.

References

Behl, A.; Hosseini Jafari, O.; Karthik Mustikovela, S.; Abu Alhaija, H.; Rother, C.; and Geiger, A. 2017. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *Proceedings of the IEEE International Conference on Computer Vision*, 2574–2583.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Gu, X.; Wang, Y.; Wu, C.; Lee, Y. J.; and Wang, P. 2019. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3254–3263.

Hur, J.; and Roth, S. 2019. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5754–5763.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2462–2470.

Jiang, H.; Sun, D.; Jampani, V.; Lv, Z.; Learned-Miller, E.; and Kautz, J. 2019. Sense: A shared encoder network for scene-flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3195–3204.

Jiang, S.; Campbell, D.; Lu, Y.; Li, H.; and Hartley, R. 2021. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9772–9781.

Jiang, Z.; and Okutomi, M. 2023. EMR-MSF: Self-Supervised Recurrent Monocular Scene Flow Exploiting Ego-Motion Rigidity. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 69–78.

Lang, I.; Aiger, D.; Cole, F.; Avidan, S.; and Rubinstein, M. 2023. SCOOP: Self-Supervised Correspondence and Optimization-Based Scene Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5281–5290.

Li, C.; Ma, H.; and Liao, Q. 2021. Two-stage adaptive object scene flow using hybrid cnn-crf model. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 3876–3883.

Liu, H.; Lu, T.; Xu, Y.; Liu, J.; Li, W.; and Chen, L. 2022. CamLiFlow: Bidirectional camera-LiDAR fusion for joint optical flow and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5791–5801.

Liu, H.; Lu, T.; Xu, Y.; Liu, J.; and Wang, L. 2023. Learning Optical Flow and Scene Flow with Bidirectional Camera-LiDAR Fusion. *arXiv preprint arXiv:2303.12017*.

Liu, X.; Qi, C. R.; and Guibas, L. J. 2019. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 529–537.

Ma, W.-C.; Wang, S.; Hu, R.; Xiong, Y.; and Urtasun, R. 2019. Deep rigid instance scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3614–3622.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4040–4048.

Mehl, L.; Jahedi, A.; Schmalfuss, J.; and Bruhn, A. 2023. M-FUSE: Multi-frame fusion for scene flow estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020–2029.

Menze, M.; and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3061–3070.

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 1147–1163.

Mur-Artal, R.; and Tardós, J. D. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 1255–1262.

Peng, C.; Wang, G.; Lo, X. W.; Wu, X.; Xu, C.; Tomizuka, M.; Zhan, W.; and Wang, H. 2023. DELFlow: Dense Efficient Learning of Scene Flow for Large-Scale Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16901–16910.

Puy, G.; Boulch, A.; and Marlet, R. 2020. Flot: Scene flow on point clouds guided by optimal transport. In *European conference on computer vision*, 527–544.

Ren, Z.; Sun, D.; Kautz, J.; and Sudderth, E. 2017. Cascaded scene flow prediction using semantic segmentation. In *2017 International Conference on 3D Vision (3DV)*, 225–233.

Shen, Y.; Hui, L.; Xie, J.; and Yang, J. 2023. Self-Supervised 3D Scene Flow Estimation Guided by Superpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5271–5280.

Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; and Cremers, D. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 573–580.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8934–8943.

Teed, Z.; and Deng, J. 2018. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 402–419.

Teed, Z.; and Deng, J. 2021a. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 16558–16569.

Teed, Z.; and Deng, J. 2021b. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8375–8384.

Vogel, C.; Schindler, K.; and Roth, S. 2015. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 1–28.

Wang, G.; Hu, Y.; Liu, Z.; Zhou, Y.; Tomizuka, M.; Zhan, W.; and Wang, H. 2022. What Matters for 3D Scene Flow Network. In *European Conference on Computer Vision*.

Wang, W.; Hu, Y.; and Scherer, S. 2021. Tartanvo: A generalizable learning-based vo. In *Conference on Robot Learning*, 1761–1772.

Wang, W.; Zhu, D.; Wang, X.; Hu, Y.; Qiu, Y.; Wang, C.; Hu, Y.; Kapoor, A.; and Scherer, S. 2020a. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4909–4916.

Wang, Z.; Li, S.; Howard-Jenkins, H.; Prisacariu, V.; and Chen, M. 2020b. Flownet3d++: Geometric losses for deep scene flow estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 91–98.

Wu, W.; Wang, Z.; Li, Z.; Liu, W.; and Fuxin, L. 2019. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*.

Xu, H.; Yang, J.; Cai, J.; Zhang, J.; and Tong, X. 2021. High-resolution optical flow from 1d attention and correlation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10498–10507.

Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; and Tao, D. 2022. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8121–8130.

Yang, G.; and Ramanan, D. 2020. Upgrading optical flow to 3d scene flow through optical expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1334–1343.

Yang, G.; and Ramanan, D. 2021. Learning to segment rigid motions from two frames. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1266–1275.

Zhang, F.; Prisacariu, V.; Yang, R.; and Torr, P. H. 2019. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 185–194.

Zhang, F.; Woodford, O. J.; Prisacariu, V. A.; and Torr, P. H. 2021. Separable flow: Learning motion cost volumes for optical flow estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10807–10817.