

DS-Depth: Dynamic and Static Depth Estimation via a Fusion Cost Volume

Xingyu Miao, Yang Bai, Haoran Duan, Yawen Huang, Fan Wan, Xinxing Xu,
Yang Long, *Senior Member IEEE*, Yefeng Zheng, *Fellow IEEE*

Abstract—Self-supervised monocular depth estimation methods typically rely on the reprojection error to capture geometric relationships between successive frames in static environments. However, this assumption does not hold in dynamic objects in scenarios, leading to errors during the view synthesis stage, such as feature mismatch and occlusion, which can significantly reduce the accuracy of the generated depth maps. To address this problem, we propose a novel dynamic cost volume that exploits residual optical flow to describe moving objects, improving incorrectly occluded regions in static cost volumes used in previous work. Nevertheless, the dynamic cost volume inevitably generates extra occlusions and noise, thus we alleviate this by designing a fusion module that makes static and dynamic cost volumes compensate for each other. In other words, occlusion from the static volume is refined by the dynamic volume, and incorrect information from the dynamic volume is eliminated by the static volume. Furthermore, we propose a pyramid distillation loss to reduce photometric error inaccuracy at low resolutions and an adaptive photometric error loss to alleviate the flow direction of the large gradient in the occlusion regions. We conducted extensive experiments on the KITTI and Cityscapes datasets, and the results demonstrate that our model outperforms previously published baselines for self-supervised monocular depth estimation. *Code:* <https://github.com/xingyu038/DS-Depth>.

Index Terms—Cost volume, Depth estimation, Monocular

I. INTRODUCTION

Currently, depth information plays a significant role in several fields, including autonomous vehicles [1], robots [2], AR/VR applications [3], and 3D reconstructions [4]. Although professional hardware can provide relatively accurate depth information, its high cost precludes widespread use. An alternative approach is to use RGB cameras, which generate a sequence of RGB images that can be leveraged by self-supervised monocular depth prediction methods [5]–[9]. While this approach addresses the expensive cost of professional hardware, its performance still falls short of that of professional hardware or deep multi-view methods. Nevertheless, self-supervised monocular depth prediction methods show promise and are gaining popularity in both research and industrial communities. However, estimating depth from a

Xingyu Miao, Yang Long, Haoran Duan and Fan Wan are with the Department of Computer Science, Durham University(E-mail: {xingyu.miao; yang.long; fan.wan}@durham.ac.uk; haoran.duan@ieee.org).

Yang Bai and Xinxing Xu are with the Institute of High Performance Computing (IHPC), ASTAR, Singapore 138632, Singapore (E-mail: {bai.yang; xuxinx}@ihpc.a-star.edu.sg).

Yawen Huang and Yefeng Zheng are with Tencent Jarvis Lab, Shenzhen, China (E-mail: {yawenhuang; yefengzheng}@tencent.com).

Yang Long and Haoran Duan are the corresponding authors.

Manuscript received April, 2023;

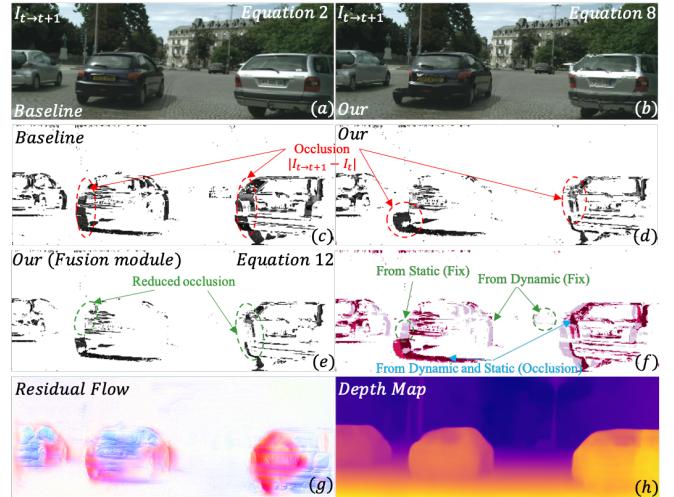


Fig. 1. Our baseline [10] is based on a static environment to generate a cost volume. However, this approach is limited when there are dynamic objects in the scene, which can cause errors such as mismatches and occlusions (Figures (a) and (c)). To handle this issue, we propose a novel dynamic cost volume that incorporates residual optical flow (Figure (g)) to capture object motion. While the dynamic cost volume can reduce occlusions, it may also introduce new occlusions and noise (Figures (b) and (d)). To overcome this challenge, we design an adaptive fusion module that makes the static and dynamic cost volumes complement each other. This approach leads to further occlusion reduction and significant improvements in performance (Figures (e) and (f)).

monocular image is an ill-posed problem due to the numerous plausible depth values that can exist in the same 3D scene, with the depth information able to project countless identical 2D scenes.

Previous self-supervised depth estimation methods rely on multi-frame information during training but only use the current frame as input during inference [11]–[16]. In contrast, multi-frame self-supervised depth estimation methods employ multi-frame information during both training and inference stages, typically by constructing a cost volume [10], [17]–[23] or utilizing related layers [24] to learn additional geometric features for improving the performance of the model. Although multi-frame methods seem to perform better, it heavily relies on feature matching to establish geometric relationships between frames. Therefore, multi-frame methods will fail in some cases, especially encountering moving objects. Currently, both the single-frame method and the multi-frame method use photometric error loss to train the model. This kind of loss is based on the static environment, once there is a moving object between consecutive frames, which will mislead

the model and bring some wrong information. Additionally, despite the fact that the existing work on self-unsupervised depth estimation [10], [19], [22], [25], [26] provides excellent solutions for unlabeled data, they are incapable of effectively handling dynamic objects, which limits their applicability to real-world scenarios.

In this work, we present DS-Depth, a self-supervised depth estimation framework designed to achieve general applicability in depth estimation. To mitigate occlusions caused by dynamic objects during view synthesis, as depicted in Figure 1, we incorporate residual optical flow to refine the cost volume generated by ego-motion, resulting in a dynamic cost volume that accurately captures the scene's dynamics. However, it is worth noting that dynamic cost volume alone is insufficient to resolve the occlusion problem while bringing extra occlusion and noise. Therefore, we propose a fusion module to combine the dynamic and static cost volumes and mitigate the occlusion and noise during view synthesis. In order to make the fused cost volume obtain the more correct gradient, we design an adaptive photometric error loss to alleviate the large gradient in the occlusion region. Additionally, during the training process, we introduce a pyramid distillation loss to alleviate the inaccuracies of photometric errors at low resolutions, leading to more accurate predicted depth maps. In summary, the contributions of our work are as follows:

- 1) We propose a novel dynamic cost volume using camera ego-motion and residual optical flow to construct, which improves upon the cost volume constructed from camera ego-motion to handle occlusions caused by moving objects.
- 2) We investigate the occlusion and noise in dynamic cost volume approaches and propose an adaptive fusion module that makes static and dynamic cost volumes compensate for each other. Our experimental results show a significant improvement in the estimated depth.
- 3) We design a pyramid error loss to improve the photometric error at low resolution and an adaptive photometric error loss to make the fused cost volume get a more accurate gradient. Experiments show that our design is effective.
- 4) We achieve state-of-the-art depth estimation results on two challenging datasets KITTI and Cityscapes.

II. RELATED WORKS

A. Self-supervised monocular depth estimation

More recently, self-unsupervised monocular depth estimation has been a kind of promising method for the limited labeled depth data, which aims to from a single image to predict a pixel-level depth map. The original self-supervised depth estimation framework is proposed by Zhou et al. [15], which leverages a DpethNet and PostNet to predict the geometry relationship between frames. This framework originally is used for stereo pairs, and then it was extended to the monocular. Additionally, Godard et al. [11] consider the depth estimation task as a view synthesis problem and minimize the image reconstruction objective. In terms of the view synthesis, Monodepth2 [12] propose a minimal reprojection

error to address occlusion and reduce visual artifacts using full-resolution multiscale sampling. In terms of the additional losses, FeatDepth [14] designs a new reconstruction error metric, which improves depth prediction of the low-texture area. For camera geometry modeling, Gordon et al. [27] first explore learning the camera intrinsic parameters through the network so that the model can be applied in wild videos. In terms of the network architectures, PackNet [13] aims to solve the problem that traditional encoder (such as ResNet) leads to the resolution being reduced and thus losing some details.

B. Multi-frame Monocular Depth Estimation

Single-frame depth estimation is based on the depth cues, such as motion information, linear perspective, occlusion, texture, and shadow [28]–[30]. The accuracy of these cues determines the estimated depth [11]–[15], [27]. Early multi-frame depth estimation approaches use test-time refinement methods [3], [14], [31]–[34] and recurrent neural networks [35], [36] to improve the performance of the model.

The stereo matching method performs feature matching by correcting the image with a known baseline, thus the method has been transformed into a pixel-by-pixel disparity estimation in the horizontal direction [37], [38]. The test-time refinement method employs a monocular approach to use temporal information at test time, while the recurrent neural network combines a monocular depth estimation network to process continuous frame sequences. However, models using recurrent neural networks are often computationally expensive and have no explicit geometric inference method. Multi-view stereo (MVS) matches any number of views [21], [37]–[41], however, most methods are supervised, and some recent self-supervised methods exploit the cost volume in stereo matching combined with single-frame features for geometric inference [10], [18], [19], [22], [42]. Similar to MVS, these works first pre-define a set of depths through which the reference frame is warped to the target frame, and then compute the difference between this frame and the target frame, stacking this difference to form a cost volume. In the cost volume space, the hypothesized depth with the lowest value is closest to the true depth. By using cost volume, the performance of the model has been greatly improved, however, these methods also use reprojection error as training loss, the model fails when encountering scenes with dynamic objects. These methods either try to avoid dynamic objects or use a single-frame model as a teacher-guided cost volume to alleviate this problem. We propose to use optical flow to describe dynamic objects to synthesize correct views, thereby building a credible cost volume to improve the performance of the model.

In addition, we use cost volume to solve the ambiguity of multiple depths, while [43] utilizes the consistency of point clouds for the same purpose. Additionally, they address the matching problem caused by photometric errors, while we tackle the issue of occlusions. [44] employs a layered method to refine the camera pose and generate a depth map, whereas we apply a similar approach by manipulating the cost volume to adjust inaccurate information and obtain a more precise depth map. Although [45] and [29] enhance the quality

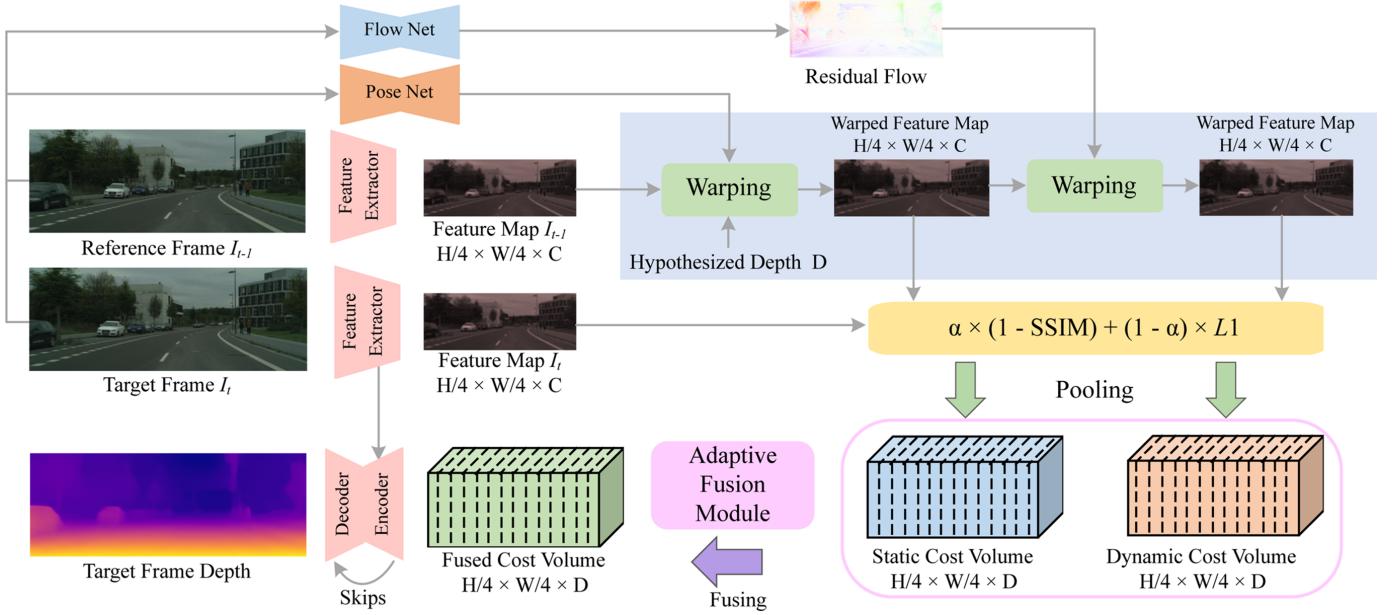


Fig. 2. **The pipeline of our network** is depicted and comprises four main components: a multi-frame depth estimation network, a single-frame depth estimation network, a pose network, and an optical flow module. Specifically, we adopt a ResNet18 [46] as the backbone. We employ the same pose network as used in [12]. In contrast to [10], our model generates two cost volumes, which are fused using a specialized fusion module. Additionally, we generate the dynamic cost volume by combining ego-motion with the residual flow, where the residual flow is obtained from FlowNet [16].

of the depth map through depth bin and iterative features, respectively, they do not consider the impact of moving objects on the results.

C. Dynamic Objects in Self-supervised Depth Prediction

Currently, due to self-supervised depth prediction using reprojection error is not suitable for dynamic scenes. Thus, the key to solving this problem is how to separate static objects and dynamic objects. Additionally, separating static environments and dynamic objects can also improve the robustness of depth estimation in dynamic situations. There have been some works [10], [18], [22], [27], [47]–[49] that propose solutions to train static and moving objects separately, which aim to solve the problem that moving objects in assumed static environment cannot be reprojected well to the correction position.

Although, above all methods have some good performance, there are still some limitations. For example, the input frame is a single frame that cannot clarify the temporal information, and disentangling moving objects and static objects will increase the complexity of the model. In addition, for methods using cost volume, dynamic object regions can mislead the model to build the wrong cost volume, this cost volume with negative information will affect the gradient of the model, causing the model to incorrectly learn the motion region. When the model finally predicts depth, it mistakenly thinks that the motion region is infinite, which reduces the accuracy of the model and affects the performance of the model. Although the method using the cost volume is better than the single-frame method, it still needs to consider the influence of moving objects.

In addition, several approaches have been proposed for jointly learning optical flow and depth using multi-task networks [16], [24], [50]–[52], where the optical flow can be

indirectly recovered for capturing dynamic objects. While our method uses the residual optical flow to fix the reprojection inaccurate regions in the cost volume.

III. METHOD

A. Preliminaries

In this section, we briefly introduce the use of view synthesis to construct similar images and then compute their correlations to construct cost volumes.

1) *View Synthesis*: Following [10], [12], we use view synthesis as supervision signals. Let two frames I_{t-1} and I_t from an input video as the source image and target image where pixel from target image can be expressed as:

$$zp_{s \rightarrow t} = K[R|T]_{t \rightarrow s} D_t K^{-1} p_s \quad (1)$$

where K denotes the camera intrinsic and $[R|T]$ is the camera extrinsic, p denotes the 2D pixel coordinates, and we use the estimated depth map of the target image, relative pose from the target image to the source image, and source image to synthesize target image.

2) *Cost Volume Construction*: Based on the Equation (1), which can be transformed into view synthesis between two features F_t and F_{t+i} in temporal. Thus, The synthesized frame \hat{F}_{t+i} can be expressed as:

$$\hat{F}_{t+i} = F_t \langle \text{proj}(D_t, T_{t \rightarrow t+i}, K) \rangle. \quad (2)$$

$F \langle \cdot \rangle$ is bilinear sampling operation and proj is the resulting 2D coordinates of the projected depths D_t , which is equal Equation (1). Note that when building the cost volume, the predicted depth is replaced by a predefined set of depth values ($D_t = \{D_1, \dots, D_{i-1}, D_i, i \in [1, N]\}$) to generate a synthetic voxel (V^s) to calculate the

correlation (The commonly used methods are SSIM [53], difference of absolute values and dot product.) with the target frame (I_t), and the final correlation cost volume (CV) is obtained. As we describe in Section III-C, a cost volume constructed in this way can produce erroneous information in dynamic scenarios. Therefore, we introduce residual optical flow to refine the moving object information in cost volume.

B. Architecture

Previous works [10], [22] construct the cost volume of two successive frames solely based on camera ego-motion learned from pose networks, which is less capable of handling dynamic regions/objects as explained in Section III-C. To address this problem, we propose dynamic cost volumes, which can effectively handle the dynamic regions/objects from the static cost volume, while also bringing some noise. Thus, based on the static and dynamic cost volumes, we carefully design an adaptive fusion module to handle static and dynamic scenarios pixel-wise to alleviate this case. The overview of our DS-Depth is shown in Figure 2.

C. Dynamic Cost Volume Construction

The differences that need to be captured when constructing the cost of two frames essentially originate from 1) the motion of moving objects and 2) changes in the relative camera pose between the two frames. This difference creates occlusions that affect the performance of the model. To better model the effect of moving objects, which was ignored in previous cost volume based on depth estimation networks [10], we first define the relationship between two-frame correspondences in a 3D scene as follows.

There is a 3D point X in the space, the 2D point projected by X in the frame I at time t is u . We define $X_{t \rightarrow t+1}^{sen}$ as the motion of the X from time t to time $t+1$ in the 3D scene. When using a known intrinsic camera to observe point X_t , we define $\mathcal{P}(X_t)$ as the projection of the X_t to the image coordinate u_t . Additionally, we define the ego-motion of the camera as $u_{t \rightarrow t+1}^{cam}$, and the move of optical flow as $u_{t \rightarrow t+1}^{opt}$.

As shown in Figure 3, for ease of understanding here, we set t and $t+1$ to 0 and 1, respectively. The projection of optical flow in a 3D perspective can be expressed as:

$$u_{0 \rightarrow 1}^{opt} = \mathcal{P}(T_1(X_0 + X_{0 \rightarrow 1}^{sen})) - \mathcal{P}(T_0 X_0) \quad (3)$$

where T denotes the known camera extrinsic matrix for I_t . Intuitively, Equation (3) could represent projected scene flow $X_{0 \rightarrow 1}^{sen}$ on 2D plane. However, since the camera is moving, the view motion should be taken into account. To be specific, the 2D correspondences moves I_0 to I_1 can be expressed as:

$$u_{0 \rightarrow 1}^{cam} = \mathcal{P}(T_1 X_0) - \mathcal{P}(T_0 X_0) \quad (4)$$

Thus, the u_0 observed on I_0 should be represented as u'_0 on I_1 , thereby scene flow $X_{0 \rightarrow 1}^{sen}$ can be derived as:

$$X_{0 \rightarrow 1}^{sen} = T_1^{-1} \mathcal{P}^{-1}(u'_0 + u_{0 \rightarrow 1}^{opt}, D_1) - T_0^{-1} \mathcal{P}^{-1}(u'_0, D_0) \quad (5)$$

where D indicates i_{th} depth level. Based on Equation (5), we need to know two different depths (D_0 and D_1) to represent

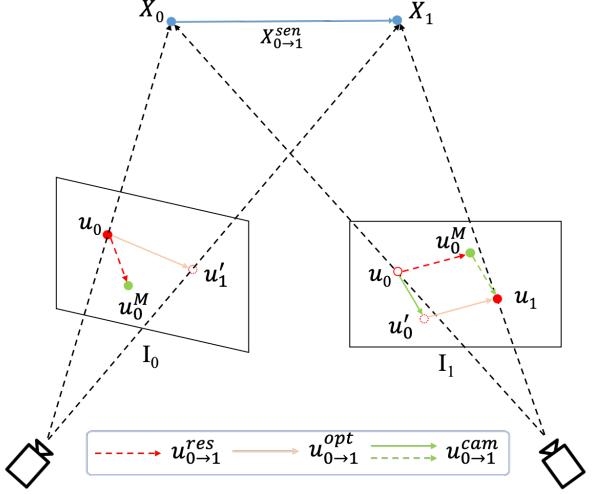


Fig. 3. **Geometric relationship of points on two frames.** Consider two frames, denoted I_0 and I_1 , where the camera moves from I_0 to I_1 . Let X_0 and X_1 be two points in 3D space, and let their corresponding 2D coordinates be projected onto the frames as u_0 and u_1 , respectively. Note that u'_0 is the projection of X_0 onto I_1 , and if X_0 is observed in I_1 , its location can be calculated by reprojecting the camera, and its motion can be expressed as $u_{0 \rightarrow 1}^{cam}$. If the camera is static and observes X_0 and X_1 at the same time, the motion from X_0 to X_1 can be expressed as $X_{0 \rightarrow 1}^{sen}$. Then, the motion of the 2D coordinates u'_0 and u_1 on I_1 can be expressed as a projected optical flow, denoted $u_{0 \rightarrow 1}^{opt1}$. Similarly, on I_0 , this projected optical flow can be expressed as $u_{0 \rightarrow 0}^{opt0}$. Finally, $u_{0 \rightarrow 1}^{res0}$ represents the residual flow learned by our optical flow module, and u_0^M is obtained by moving u_0 by residual optical flow.

the scene flow $X_{0 \rightarrow 1}^{sen}$ in space, but in our framework, we can only predict D_0 . Therefore, we use an optical flow module to learn the residual flow $u_{0 \rightarrow 1}^{res0}$ on the frame I_0 combined with the camera ego-motion to represent the scene flow. As shown in Figure 3, the 2D point u_0 move to u_1 can be expressed as:

$$(u_0 \rightarrow u_1) = u_{0 \rightarrow 1}^{cam} + u_{0 \rightarrow 1}^{opt1} \quad (6)$$

However, $u_{0 \rightarrow 1}^{opt1}$ is not equal to $u_{0 \rightarrow 1}^{opt0}$ because the camera is moving, which causes the corresponding motion of u_0 . Therefore, we utilize an optical flow network to learn a residual flow $u_{0 \rightarrow 1}^{res0}$ that approximates $u_{0 \rightarrow 1}^{opt1}$. In this case, we can substitute $u_{0 \rightarrow 1}^{opt1}$ with $u_{0 \rightarrow 1}^{res0}$, resulting in the following relationship:

$$(u_0 \rightarrow u_1) = u_{0 \rightarrow 1}^{cam} + u_{0 \rightarrow 1}^{res0} \quad (7)$$

thus, similar to Equation (2), new synthesized feature \hat{F}_t^N is:

$$\hat{F}_{t+i}^N = \hat{F}_t(\langle proj(D_t, T_{t \rightarrow t+i}, K) \rangle + u_{t \rightarrow t+i}^{rest}). \quad (8)$$

To construct dynamic cost volume, we follow a similar approach to Manydepth, but with the added consideration of dynamic object movement. Specifically, when generating a target view from a reference view, we incorporate residual optical flow and camera transformation information into the synthesis process. Furthermore, we have enhanced the construction of our cost volume by utilizing a projection error consistent method. While Manydepth employs an L_1 approach, we adopt a formulation that combines structural similarity index measure (SSIM) and L_1 , which is:

$$\mathcal{E}(\hat{F}, F) = \alpha(1 - \text{SSIM}(\hat{F}, F)) + (1 - \alpha) \|\hat{F} - F\|_1 \quad (9)$$

TABLE I

DEPTH ESTIMATION RESULTS ON KITTI AND CITYSCAPES. THE METHODS IN THIS TABLE ARE CLASSIFIED ACCORDING TO MULTI-FRAME AND SINGLE-FRAME, ABS.REL. ERROR DESCENDING ORDER, WHERE THE BEST METHOD IS IN **BOLD** AND THE SECOND BEST IS UNDERLINED. ADDITIONALLY, THE TABLE ONLY SHOWS THE RESULTS OF SPLITTING THE KITTI WITH EIGEN ET. AL. [54] AND THE CITYSCAPES SPLIT WITH [15]. **K** – KITTI **C** – CITYSCAPES

Method	Test frames	Dataset	WxH	Lower is better				Higher is better		
				AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ranjan et al. [55]	1	K	832x256	0.148	1.149	5.464	0.226	0.815	0.935	0.973
EPC++ [56]	1	K	832x256	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2depth (M) [31]	1	K	416x128	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Li et al. [57]	1	K	416x128	0.130	0.950	5.138	0.209	0.843	0.948	0.978
Videos in the wild [27]	1	K	416x128	0.128	0.959	5.230	0.212	0.845	0.947	0.976
Monodepth2 [12]	1	K	640x192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Packnet-SFM [13]	1	K	640x192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
Johnston et al. [58]	1	K	640x192	0.106	0.861	4.699	0.185	0.889	0.962	0.982
Guizilini et al. [59]	1	K	640x192	0.102	0.698	4.381	0.178	0.896	0.964	0.984
Patil et al. [36]	N	K	640x192	0.111	0.821	4.650	0.187	0.883	0.961	0.982
Wang et al. [60]	2 (-1,0)	K	640x192	0.106	0.799	4.662	0.187	0.889	0.961	0.982
ManyDepth [10]	2 (-1,0)	K	640x192	<u>0.098</u>	<u>0.770</u>	4.459	<u>0.176</u>	<u>0.900</u>	<u>0.965</u>	<u>0.983</u>
Our	2 (-1,0)	K	640x192	0.095	0.698	4.329	0.173	0.905	0.966	0.984
Pilzer et al. [61]	1	C	512x256	0.240	4.264	8.049	0.334	0.710	0.871	0.937
Struct2depth 2 [31]	1	C	416x128	0.145	1.737	7.280	0.205	0.813	0.942	0.976
Monodepth2 [12]	1	C	416x128	0.129	1.569	6.876	0.187	0.849	0.957	0.983
Videos in the wild [27]	1	C	416x128	0.127	1.330	6.960	0.195	0.830	0.947	0.981
Li et al. [57]	1	C	416x128	0.119	1.290	6.980	0.190	0.846	0.952	0.982
Lee et al. [49]	1	C	832x256	0.116	1.213	6.695	0.186	0.852	0.951	0.982
Struct2Depth 2 [31]	3 (-1,0,1)	C	416x128	0.151	2.492	7.024	0.202	0.826	0.937	0.972
ManyDepth [10]	2 (-1,0)	C	416x128	<u>0.114</u>	<u>1.193</u>	<u>6.223</u>	<u>0.170</u>	<u>0.875</u>	<u>0.967</u>	<u>0.989</u>
Our	2 (-1,0)	C	416x128	0.100	1.055	5.884	0.155	0.899	0.974	0.991

where α is 0.4. The yielded cost volume is expressed as:

$$CV_S = \mathcal{E}(\hat{F}, F) \quad (10)$$

$$CV_D = \mathcal{E}(\hat{F}^N, F) \quad (11)$$

where CV_S is the static cost volume that is constructed with camera movement, and CV_D is the dynamic cost volume that uses residual optical flow to refine static cost volume, feature map \hat{F} is generated by Equation (2) and feature map \hat{F}^N is generated by Equation (8).

D. Adaptive Fusion Module

The cost volume construction involves leveraging the time t feature and the warped time $t - 1$ feature. Specifically, the feature F^w at time $t - 1$ is warped to time t along a hypothetical depth D , after which Equation (10) is employed to generate the cost volume. However, this process will occur occlusion, which can pollute the cost volume distribution. To address this issue, we employ a learned residual optical flow to simulate object motion and rectify the incorrectly warped pixels caused by dynamic objects, thereby guiding the gradient flow to the correct pixels. Although we use residual optical flow to refine the incorrect information in static cost volume caused by dynamic objects, it will inevitably cause some extra occlusions and noise (As shown in Figure 1). Thus, we carefully design a fusion module to alleviate this problem. This fusion module is divided into two branches, one of the branches is a simple concatenate operation, while another branch utilizes static and dynamic cost quantities to

adaptive complement each other. The adaptive fusion branch can be expressed as:

$$CV_{com} = \begin{cases} CV_D, & (F_o^w \in CV_S) \cup (F_v^w \in CV_D) \\ CV_S, & (F_o^w \in CV_D) \cup (F_v^w \in CV_S) \\ \min(CV_S, CV_D), & F_v^w \in (CV_D \cup CV_S) \end{cases} \quad (12)$$

where O/V are the set of occluded/visible areas in F^w . Specifically, in cases where a pixel in the static cost volume is occluded, we substitute it with the corresponding pixel in the dynamic cost volume and vice versa. Thus, the final fused cost volume is:

$$CV_f = CV_{com} + CV_{cat} \quad (13)$$

where CV_{cat} is obtained by concatenating the two cost quantities and passing through a simple convolution layer. After experiments, the fusion cost volume can effectively alleviate the partial occlusions and noise problem. The effectiveness of our design is confirmed in ablation studies. The module architecture is shown in Figure 4.

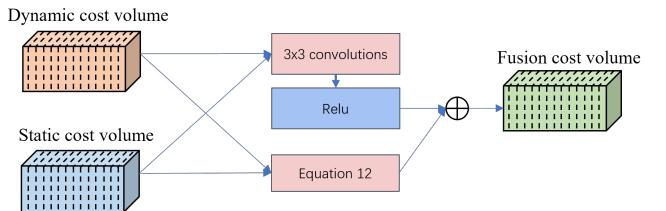


Fig. 4. **Adaptive Optical Flow Fusion Module.** Here, we describe our adaptive optical flow fusion module, which consists of two branches. The cost volume is a 4D tensor ($B \times D \times \frac{H}{4} \times \frac{W}{4}$). Where D is the depth level, using 96 depth bins.

TABLE II

DEPTH ESTIMATION RESULTS ON KITTI DATASET WITH IMPROVED GROUND TRUTH [62]. WE EVALUATED OUR METHOD USING THE KITTI DATASET WITH IMPROVED GROUND TRUTH AND FOLLOWED CONVENTION BY SORTING METHODS IN EACH CATEGORY BY THEIR ABSOLUTE RELATIVE ERROR WITH RESPECT TO THE GROUND TRUTH. THE BEST METHODS WERE HIGHLIGHTED IN **BOLD**. OUR METHOD SURPASSED ALL OTHER STATE-OF-THE-ART APPROACHES, INCLUDING SOME STEREO-BASED AND SUPERVISED METHODS.

SUP – SUPERVISED BY GROUND TRUTH DEPTH S – STEREO M – MONOCULAR

Method	Training	WxH	The lower the better				The higher the better		
			Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhan FullNYU [63]	Sup	608 x 160	0.130	1.520	5.184	0.205	0.859	0.955	0.981
Kuznetsov et al. [64]	Sup	621 x 187	0.089	0.478	3.610	0.138	0.906	0.980	0.995
DORN [65]	Sup	513 x 385	0.072	0.307	2.727	0.120	0.932	0.984	0.995
Monodepth [11]	S	512 x 256	0.109	0.811	4.568	0.166	0.877	0.967	0.988
3net [66] (VGG)	S	512 x 256	0.119	0.920	4.824	0.182	0.856	0.957	0.985
3net [66] (ResNet 50)	S	512 x 256	0.102	0.675	4.293	0.159	0.881	0.969	0.991
SuperDepth [67]	S	1024 x 384	0.090	0.542	3.967	0.144	0.901	0.976	0.993
Monodepth2 [12]	S	640 x 192	0.085	0.537	3.868	0.139	0.912	0.979	0.993
EPC++ [56]	S	832 x 256	0.123	0.754	4.453	0.172	0.863	0.964	0.989
SfMLearner [15]	M	416 x 128	0.176	1.532	6.129	0.244	0.758	0.921	0.971
Vid2Depth [68]	M	416 x 128	0.134	0.983	5.501	0.203	0.827	0.944	0.981
GeoNet [16]	M	416 x 128	0.132	0.994	5.240	0.193	0.833	0.953	0.985
DDVO [69]	M	416 x 128	0.126	0.866	4.932	0.185	0.851	0.958	0.986
Ranjan [55]	M	832 x 256	0.123	0.881	4.834	0.181	0.860	0.959	0.985
EPC++ [56]	M	832 x 256	0.120	0.789	4.755	0.177	0.856	0.961	0.987
Johnston et al. [58]	M	640 x 192	0.081	0.484	3.716	0.126	0.927	0.985	0.996
Monodepth2 [12]	M	640 x 192	0.090	0.545	3.942	0.137	0.914	0.983	0.995
Packnet-SFM [13]	M	640 x 192	0.078	0.420	3.485	0.121	0.931	0.986	0.996
Patil et al. [36]	M	640 x 192	0.087	0.495	3.775	0.133	0.917	0.983	0.995
Wang et al. [60]	M	640 x 192	0.082	0.462	3.739	0.127	0.923	0.984	0.996
ManyDepth [10]	M	640 x 192	0.070	0.399	3.455	0.113	0.941	0.989	0.997
Our	M	640 x 192	0.067	0.359	3.314	0.109	0.946	0.989	0.997

E. Loss Function

We train our self-unsupervised monocular depth architecture using only the photometric reprojection loss, which includes two parts, a structure similarity (SSIM) [53] and absolute error (L1) terms:

$$L_p = \frac{\alpha}{2} (1 - \text{SSIM}(I_t, \hat{I}_t)) + (1 - \alpha) \|I_t - \hat{I}_t\|_1 \quad (14)$$

Based on previous work we set $\alpha = 0.85$. In order to train the residual optical flow, we have changed the synthesis of \hat{I}_t . In the previous work, the synthesis of view is via Equation (2). Now we first obtain the residual optical flow and then use Equation (8) for the final view synthesis. We can obtain the new synthetic view \hat{I}_t^f . Although the new synthetic view captures the correct dynamic region, it may also introduce additional noise, which can cause larger gradients and ultimately lead to degraded module performance if the view is used directly without any preprocessing or regularization. Therefore, we design an adaptive photometric loss to alleviate this problem:

$$L_p = \frac{\alpha}{2} \mathcal{S}(I_t, \hat{I}_t, \hat{I}_t^f) + (1 - \alpha) \mathcal{L}(I_t, \hat{I}_t, \hat{I}_t^f) \quad (15)$$

where \mathcal{S} can be expressed as:

$$\mathcal{S} = 1 - \max(\text{SSIM}(I_t, \hat{I}_t), \text{SSIM}(I_t, \hat{I}_t^f)) \quad (16)$$

and \mathcal{L} can be expressed as:

$$\mathcal{L} = \min(\|I_t - \hat{I}_t\|_1, \|I_t - \hat{I}_t^f\|_1) \quad (17)$$

We also use edge-aware smoothness for depth regularization:

$$L_s = |\partial_x \hat{d}_t| e^{-|\partial_x I_t|} + |\partial_y \hat{d}_t| e^{-|\partial_y I_t|} \quad (18)$$

Moreover, the photometric consistency measurement is not accurate for low resolution [43], [70]. Direct unsupervised training at intermediate levels is not suitable, especially at low resolutions. In this case, we use the last output depth map as a pseudo-label to add supervised learning to the low-resolution output. We directly upsample the lower resolution output and evaluate its difference from the final output. Therefore, our pyramid distillation loss is:

$$L_d = \sum_{i=0}^N \frac{1}{HW} \mathcal{F}(D_f - S_\uparrow(s_i, D_f)) \quad (19)$$

where i is scale, S_\uparrow is upsample operate, and \mathcal{F} is the robust penalty function [71]: $\mathcal{F} = (|x| + \varepsilon)^q$, q, ε being 0.4 and 0.1. Finally, our training loss is $L = L_p + L_s + L_{consistency} + L_d$, where $L_{consistency}$ is consistency loss from [10] with no additional modifications.

IV. EXPERIMENTAL RESULTS

We evaluate our DS-Depth model on two challenging depth estimation datasets (KITTI [1] and Cityscapes [72]) and show SOTA results by comparison. Finally, the effectiveness of our model is verified by ablation experiments.

A. Datasets and Experimental Settings

1) **KITTI**: The KITTI is a widely-used dataset and is the standard benchmark for depth evaluation. We use Eigen et al. split form [54] with filtered static frames from Zhou et al. [15]. This segmentation method is mostly used for single-frame depth estimation, but it has also been used for multi-frame depth estimation recently [10], [18]. It includes 39,810 training images, 4,424 validation images, and 697 test images.

TABLE III

ABLATION STUDY ON KITTI. EVALUATE OUR DYNAMIC COST VOLUME, FUSION MODULE, ADAPTIVE PHOTOMETRIC LOSS (APM LOSS) AND PYRAMID DISTILLATION LOSS (PD LOSS) ON KITTI.

Static cost volume	Dynamic cost volume	Fusion module		APM loss	PD loss	Lower is better			
		Complementary	Concatenate			AbsRel	SqRel	RMSE	RMSElog
✓						0.101	0.784	4.559	0.183
	✓					0.102	0.761	4.557	0.180
✓	✓	✓				0.102	0.775	4.630	0.182
✓	✓			✓		0.101	0.757	4.500	0.178
✓	✓	✓	✓	✓		0.096	0.714	4.398	0.174
✓	✓	✓	✓	✓	✓	0.095	0.705	4.360	0.173
✓	✓	✓	✓	✓	✓	0.095	0.698	4.329	0.173

TABLE IV

ABLATION STUDY ON CITYSCAPES. HERE WE DEMONSTRATE THE EFFECTIVENESS OF OUR METHOD ON THE CITYSCAPES DATASET BY EVALUATING IT. OUR APPROACH LEADS TO SUBSTANTIAL IMPROVEMENTS ON CITYSCAPES, WHERE A LARGER NUMBER OF MOVING OBJECTS ARE PRESENT IN BOTH THE TRAINING AND TEST FOOTAGE, COMPARED TO KITTI.

Static cost volume	Dynamic cost volume	Fusion module		APM loss	PD loss	Lower is better			
		Complementary	Concatenate			AbsRel	SqRel	RMSE	RMSElog
✓						0.114	1.193	6.226	0.170
	✓					0.109	1.170	6.130	0.162
✓	✓	✓				0.104	1.159	6.012	0.159
✓	✓			✓		0.103	1.137	6.001	0.158
✓	✓	✓	✓	✓		0.102	1.140	5.940	0.156
✓	✓	✓	✓	✓	✓	0.101	1.051	5.883	0.156
✓	✓	✓	✓	✓	✓	0.100	1.055	5.884	0.155

TABLE V

THE MODEL PARAMETERS AND RUNNING TIME (UNIT: S/BATCH SIZE) OF THE BASELINE AND OUR METHOD.

Methods	Parameter	Inference time
Manydepth [10]	13.64M	0.020~0.034 s/b
Our	13.95M	0.035~0.045 s/b

2) *Cityscapes*: The cityscapes contains 150,000 images. Following [10], [15], [73], we train on 69,731 images, which are split according to how the script in [15] is split. We do not use any stereo image pairs or semantics. We evaluate our model on the 1,525 test images provided by SGM [74].

3) *Evaluation Metrics*: Following the state-of-the-art methods [10], [18], we use Absolute Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root Mean Squared Error (RMSE), Root Mean Squared Log Error (RMSElog), and δ_1 , δ_2 , δ_3 as the metrics to evaluate the performance of our model.

4) *Model Parameters and Inference Time*: We present the refined parameters of our improved encoder model along with the corresponding inference times in Table V. Despite the increase in parameters by 0.31M and the inference time ranging from 0.011 to 0.015 seconds per batch, the performance of our model has significantly improved on both databases, particularly on Cityscapes.

5) *Implementation Details*: Our model is implemented using PyTorch and trained on a single NVIDIA RTX3090 GPU. We adopt ResNet18 [46], which is pretrained on the ImageNet dataset [75], as our backbone. To optimize our model, we use the Adam optimizer [76] with an initial learning rate of 1e-4 for 30 epochs, a batch size of 12, and we reduce the learning rate by a factor of 10 every 10 epochs when training on the KITTI dataset. Following [10], we freeze the pose and single-frame teacher network for the last 5 epochs. To build the cost

volume, we only use the frame $t - 1$, and to calculate the loss, we use the $t - 1$ and $t + 1$ frames. For the Cityscapes dataset, we use a batch size of 8 and freeze network on the 5th epoch.

B. Comparison to State-of-the-art

1) *Results on KITTI*: In Table I we compare our method with other methods, e.g. single-frame methods [12], [13], [31], multi-frame methods [10], [36], [60] and dynamic region optimization method [49]. Our method focuses on dynamic region optimization, however, there are fewer moving objects in this database, and most of them are static scenarios. Thus, our methods and [18], [27], [31], [48], [49], [57], [77] have minor improvement for this database (Where [18] uses mask generated by a pretrained segmentation network, i.e. the predicted depth is closely related to the performance of this segmentation network. Furthermore, data splitting is different from most multi-frame methods, hence we did not compare with it.). Moreover, compared to our baseline, our SqRel error improves 9.35% which means our method predicts fewer depths with large errors (i.e. smaller errors in dynamic regions/objects.). The Abs.Rel. error statistics per pixel in Figure 7 also confirm that our method is better than multi-frame methods, the number of the Abs.Rel. error per pixel of our method in the interval [0, 0.5] is much larger than that of the baseline.

2) *KITTI benchmark scores*: The original Eigen [54] split of the KITTI [78] dataset employs re-projected single-frame raw LIDAR points as the ground truth for evaluation. However, it may contain outliers such as reflections on transparent objects. Thus, we reported results using the original ground truth since it is widely used.

Recently, Jonas et al. [62] introduced a set of high-quality ground truth depth maps for the KITTI dataset. They used a

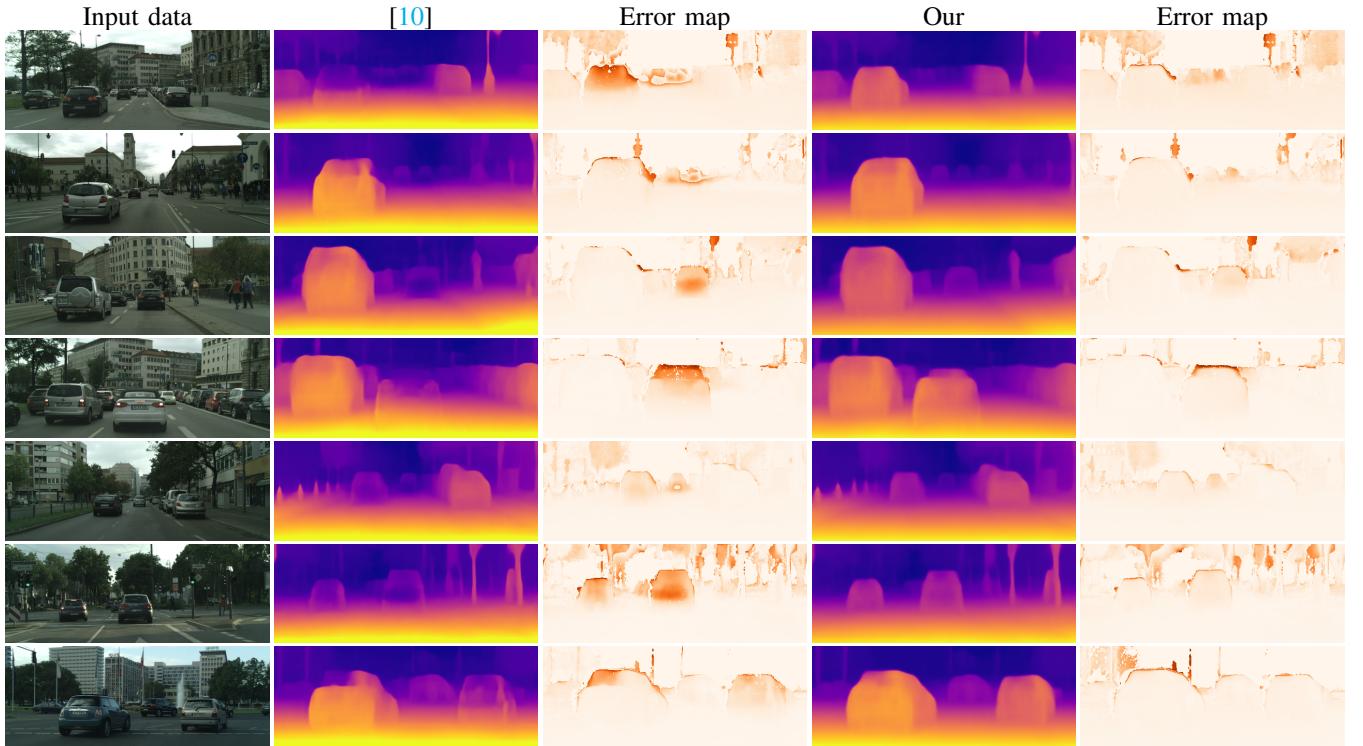


Fig. 5. **The qualitative results** on Cityscapes dataset. Here, the 3 and 5 columns show the Abs.Rel.error, and the 2 and 4 columns show the predicted depth maps. It can be observed from the error map that our method significantly outperforms our baseline, especially in dynamic regions.

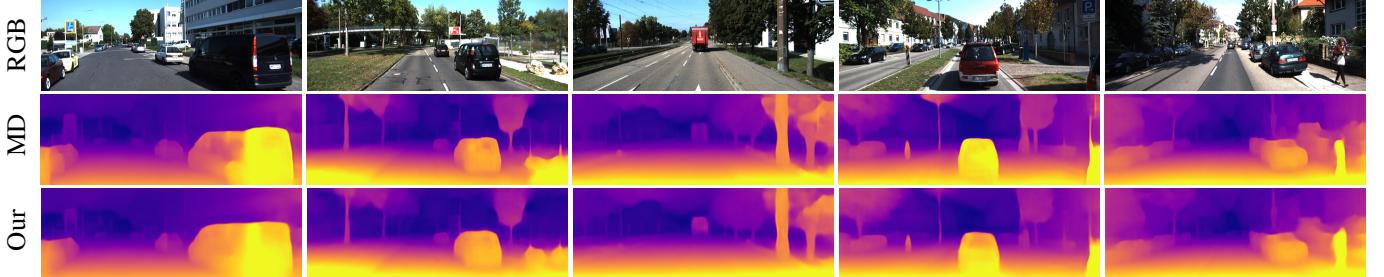


Fig. 6. **The qualitative results** on KITTI dataset. Here we show our prediction results and our baseline (MD) [10] results. Since we focus on dynamic objects, we can see from the figure that our prediction results have better contour and texture information in dynamic regions. Furthermore, the corresponding qualitative results show in Figure 7.

denser ground truth depth map obtained by accumulating 5 consecutive frames and removing the outliers. This improved ground truth depth is provided for 652 of the 697 test frames in the Eigen test split [54]. In this study, we evaluated our method using these 652 improved ground truth frames and compared the results with existing state-of-the-art published methods in Table II. To adhere to convention, we clipped the predicted depths to 80 meters to match the Eigen evaluation.

Our method was ranked by the Absolute Relative Error and outperformed all existing state-of-the-art methods, including some stereo-based and supervised methods.

3) Results on Cityscapes: Below the Table I shows each method score in the Cityscapes dataset, this dataset contains a large number of dynamic scenes. Our method currently outperforms all methods and compare to our baseline, the performance of our model achieves 12.3% improvement.

Moreover, Figure 10a provides consistency error (The difference between the depth of teacher network predictions and the lowest disparity map.), we get this error with the same parameters trained for 10 epochs and without any pretrained model, which improves by 7.43% compared to the baseline. In addition, in Table VI we show the evaluation of the depth only for the dynamic region, our method improves 24.85% in the dynamic region compared to the baseline.

C. Qualitative Results

The qualitative results are reported in Figure 6 and Figure 5. In Figure 6, our method performance is better in dynamic regions. We can see that the vehicles in columns 1, 3, and 4, the stones in column 2, and the people in column 5 have clearer texture, and contours information compared to the baselines.

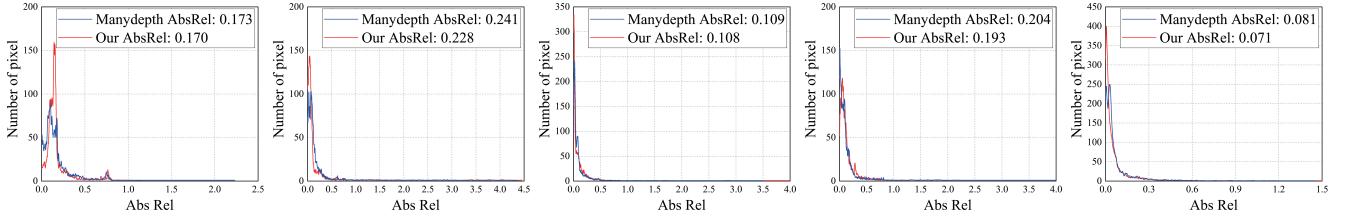


Fig. 7. **The quantitative results** on KITTI dataset. Here we show the Abs.Rel. error statistics for each valid pixel of our method and baseline final depth map corresponding to Figure 6.

TABLE VI
THE DEPTH EVALUATION RESULTS OF THE DYNAMIC OBJECTS (E.G. VEHICLES, BIKES, AND PEDESTRIANS) ON THE CITYSCAPES DATASET. DYNAMIC OBJECT MASKS ARE GENERATED BY THE PRE-TRAINED MODEL EFFICIENTPS [79].

Method	H x W	The lower the better				The higher the better		
		AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [12]	416 x 128	0.159	1.937	6.363	0.201	0.816	0.950	0.981
InstaDM [48]	832 x 256	0.139	1.698	5.760	0.181	0.859	0.959	0.982
Manydepth [10]	416 x 128	0.169	2.175	6.634	0.218	0.789	0.921	0.969
Our (W/o PD Loss)	416 x 128	0.130	1.163	5.953	0.183	0.801	0.955	0.986
Our (W PD Loss)	416 x 128	0.127	1.047	5.604	0.179	0.827	0.960	0.988

TABLE VII
THE RESULTS OF USING DIFFERENT BACKBONES FOR OUR OPTICAL FLOW MODULE. KITTI* IS KITTI DATASET WITH IMPROVED GROUND TRUTH.

Method	Dataset	The lower the better				The higher the better		
		AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Our	KITTI	0.095	0.698	4.329	0.173	0.905	0.966	0.984
Our (ResNet18)	KITTI	0.095	0.705	4.326	0.174	0.905	0.966	0.983
Our	KITTI*	0.067	0.359	3.314	0.109	0.946	0.989	0.997
Our (ResNet18)	KITTI*	0.068	0.360	3.284	0.109	0.946	0.989	0.997
Our	Cityscapes	0.100	1.055	5.884	0.155	0.899	0.974	0.991
Our (ResNet18)	Cityscapes	0.102	1.129	5.961	0.157	0.899	0.973	0.990

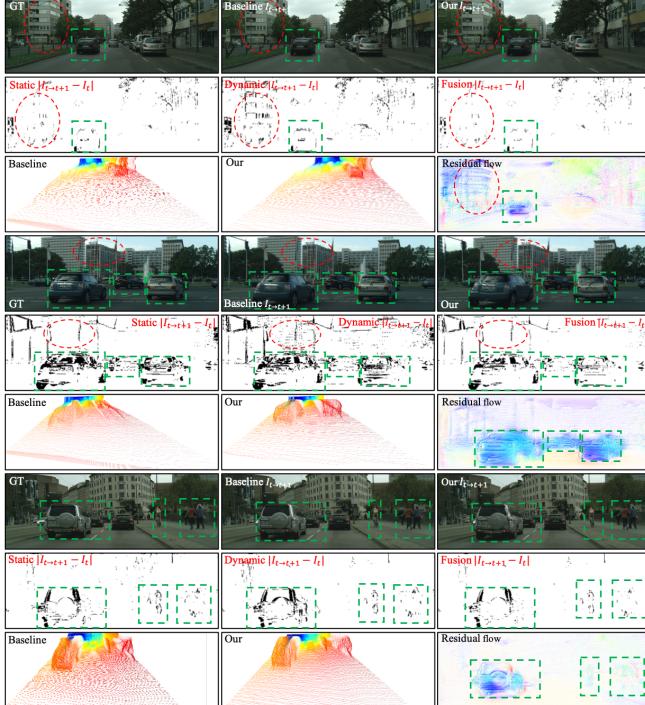


Fig. 8. **The qualitative results** on Cityscapes. We provide the view synthesis, residual flow, occlusion map, and depth map converted to point cloud results.

As shown in Figure 5, the depth predicted by our model significantly outperforms our baseline method, especially in dynamic regions, and the vehicles of our method are clearer and have no mismatched regions. The results, from the error map, indicate that the error of our method is smaller than that of our baseline method in the dynamic regions.

In addition, in Figure 8, only using dynamic will bring more occlusion and noise, while the fusion module can effectively reduce the occlusion, compared to our baseline. The residual flow results demonstrate our model's ability to accurately capture moving objects and handle inaccurate pixels without the use of priors, such as pre-trained segmentation models. Notably, the point cloud image converted from the depth map shows that our model performs better on moving objects. In Figure 9, we provide more optical flow visualization results.

D. Ablation Study

In Table III and Table IV, we provide an analysis of different components in our DS-Depth architecture on KITTI and Cityscapes, we validate the effectiveness of dynamic cost volume, three different forms of fusion modules, and pyramid distillation loss. In Table VII, we provide the results of the optical flow network using different backbones.

1) *Only Using Dynamic Cost Volume*: In this setting, we only use the dynamic cost volume to train our model. In KITTI, the Abs.Rel. error does not change significantly,

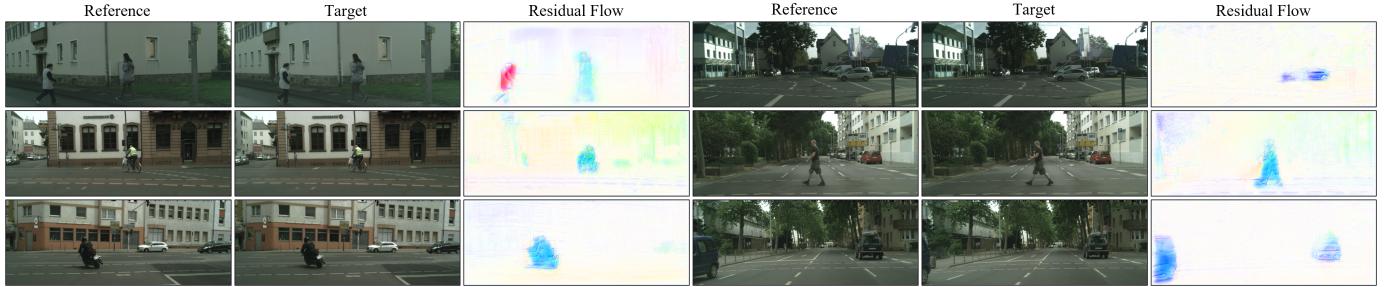


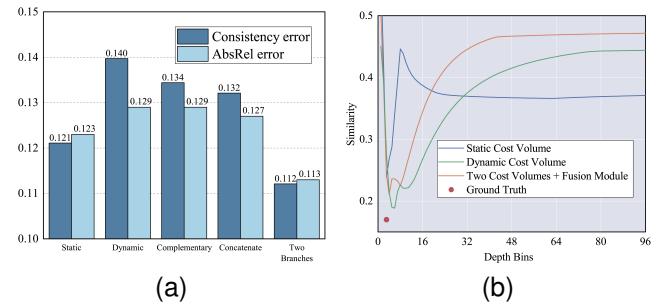
Fig. 9. The qualitative result of the residual flow on Cityscapes dataset. It is obvious that our optical flow network has captured moving objects.

TABLE VIII
EVALUATE OUR METHOD, L1, AND SSIM ON KITTI AND CITYSCAPES
DATASET TO BUILD COST VOLUME.

Method	KITTI			Cityscapes		
	AbsRel	SqRel	RMSE	AbsRel	SqRel	RMSE
L1	0.099	0.756	4.460	0.102	1.133	5.985
SSIM	0.099	0.778	4.507	0.102	1.136	5.964
Our	0.095	0.698	4.329	0.100	1.055	5.884

which is as we expected. Because the dynamic cost volume refines part of the occlusion areas but the same brings some extra occlusions and noise and the low number of dynamic objects in the KITTI dataset. As shown in Figure 10b, in the occlusion region, only using dynamic cost volume will degrade the performance of the model, and the predicted depth has already deviated from the ground truth. While in Cityscapes, the Abs.Rel. error improvement is significant, with a 4.39% improvement. This improvement over KITTI is very significant, as a large number of moving objects are included in Cityscapes, and our residual flow succeeds in warping the moving objects to relatively correct positions.

2) *Two Cost Volumes with Fusion Module*: We tried three fusion modules 1) complementary, 2) simple concatenate, and 3) two-branch. In KITTI, compared to only using the dynamic cost volume, the fusion module increases the performance of the model (0.9%-6.86%) and also alleviates the occlusion problem to a certain extent. Our results on Cityscapes also support this conclusion (4.59%-6.42%). Figure 10a illustrates that if the dynamic cost volume only is used, the performance of our model will decrease due to the extra occlusion areas and noise. When we leverage the fusion module, the error is significantly reduced and this suggests that our fusion module alleviates this problem to a certain extent, please see the effect of our model in the bottom half of Figure 1. In Figure 10b, when using the fusion module, the predicted depths are closer to the ground truth than using only the dynamic cost volume, which confirms that our fusion layer is effective in occluded regions. After our observation, it became evident that the complementary module's functionality on the KITTI dataset is rather limited. This arises from the dataset's distinctive trait of having a lower occurrence of moving objects, with a significant majority of them being stationary. As a result, the effectiveness of the complementary module appears to be diminished when applied to the KITTI dataset. This is because



(a)

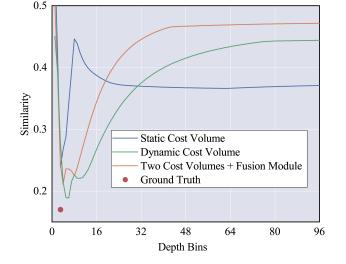


Fig. 10. (a) Consistency error and Abs.Rel. error between a teacher network and the lowest cost volume trained for 10 epochs with no pretrained model in the Cityscape dataset. (b) The matching probability along depth bins in the occlusion area. The blue line is our baseline, the orange line only uses dynamic cost volume and the green line uses the fusion module.

there are two cases in the complementary module: one of the same region in the dynamic and static cost volume is occluded, then the module will select the unoccluded cost volume, and if both cost volumes are unoccluded in the same region then the module will select the cost volume with the lower error. However, our cost volume is composed jointly by L1 and ssim errors, so there are cases of mismatching in some regions. If the error of dynamic cost volume is small and there is a lot of noise, and our module incorrectly selects this part of the dynamic cost volume, then the situation shown in our ablation experiment will occur. In contrast, for the concatenate module, because this module is learnable, the above problem does not exist.

Moreover, the effect of the two-branch fusion module is better than the complementary and simply concatenate fusion module, because the complementary fusion module obtains a small error but some regions are greatly affected by artifacts. For the simple concatenate fusion module, it cannot directly get the most correct complementary error, hence the effect is not significantly improved.

3) *Pyramid Distillation Loss*: The main purpose of this loss is to fix the inaccurate photometric error caused by low resolution. Based on our observations, the photometric error of our model is only less accurate at 1/8 resolution, this component has little effect on the performance of our model. However, as shown in Table VI, the effect of this loss is relatively obvious in dynamic regions.

In addition, we also provide three different ways to build cost volume results. Although constructing the cost volume in

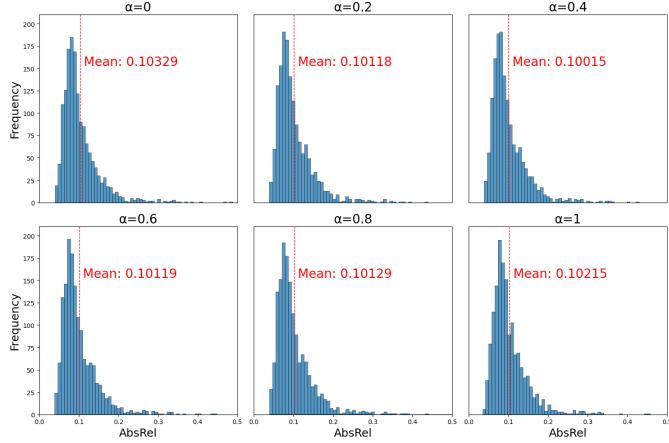


Fig. 11. The comparison of the cost volume constructed by different α for the performance of the model in the Cityscapes dataset. When $\alpha = 0.4$, the model performance is the best, and the Abs.Rel. error distribution is relatively smoother.

the L1 way can alleviate the dynamic problem to a certain extent, this way sacrifices the surface details, while SSIM can provide more surface details, thus we try to construct the cost volume in the way of photometric error. Table VIII presents to construct cost volume using the photometric error method, which is better than L1 and SSIM. As shown in Figure 11, we use Equation (9), we explore the impact of different α on model performance on cityscapes. Among them, when $\alpha = 0.4$, the model has the best performance and has a greater improvement compared to other hyperparameters.

V. CONCLUSION

In this work, we proposed DS-Depth, a general self-supervised depth estimation model framework. Specifically, we proposed a dynamic cost volume construction by combining camera ego-motion and residual optical flow to optimize the cost volume occlusion problem in dynamic regions. To alleviate the extra occlusion and noise caused by dynamic cost volume, the adaptive fusion module is designed to effectively improve the contour and texture information of dynamic objects. Moreover, we proposed a pyramid distillation loss to address the inaccuracy of photometric error at low resolutions and an adaptive photometric error loss to alleviate large gradients in the occlusion region. During our experiments, we found that the accuracy of the single-frame teacher network will have a great impact on the performance of the model in the later stage of training, especially on the Cityscapes dataset, which is very obvious and will bring some negative effects. Therefore, exploring how to optimize single-frame networks may further improve the performance of multi-frame methods.

ACKNOWLEDGMENTS

This work is supported by the UK Medical Research Council (MRC) Innovation Fellowship under Grant MR/S003916/2, International Exchanges 2022 IEC\NSFC\223523 and Securing the Energy/Transport Interface EP/X037401/1.

REFERENCES

- [1] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.
- [2] B. Choi, C. Mericli, J. Biswas, and M. Veloso, “Fast human detection for indoor mobile robots using depth images,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 1108–1113, IEEE, 2013.
- [3] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf, “Consistent video depth estimation,” *ACM Transactions on Graphics (ToG)*, vol. 39, no. 4, pp. 71–1, 2020.
- [4] R. A. Newcombe, “Real-time dense surface mapping and tracking,” *IEEE ISMAR, IEEE, October 2011*, 2011.
- [5] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, “Unsupervised cnn for single view depth estimation: Geometry to the rescue,” in *European conference on computer vision*, pp. 740–756, Springer, 2016.
- [6] M. Song, S. Lim, and W. Kim, “Monocular depth estimation using laplacian pyramid-based depth residuals,” *IEEE transactions on circuits and systems for video technology*, vol. 31, no. 11, pp. 4381–4393, 2021.
- [7] Y. Cao, T. Zhao, K. Xian, C. Shen, Z. Cao, and S. Xu, “Monocular depth estimation with augmented ordinal depth relationships,” *IEEE Transactions on Image Processing*, 2018.
- [8] H. Mohaghegh, N. Karimi, S. R. Soroushmehr, S. Samavi, and K. Najarian, “Aggregation of rich depth-aware features in a modified stacked generalization model for single image depth estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 683–697, 2018.
- [9] X. Ye, Z. Li, B. Sun, Z. Wang, R. Xu, H. Li, and X. Fan, “Deep joint depth estimation and color correction from monocular underwater images based on unsupervised adaptation networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3995–4008, 2019.
- [10] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman, “The temporal opportunist: Self-supervised multi-frame monocular depth,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1164–1174, 2021.
- [11] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.
- [12] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3828–3838, 2019.
- [13] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, “3d packing for self-supervised monocular depth estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2485–2494, 2020.
- [14] C. Shu, K. Yu, Z. Duan, and K. Yang, “Feature-metric loss for self-supervised learning of depth and egomotion,” in *European Conference on Computer Vision*, pp. 572–588, Springer, 2020.
- [15] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858, 2017.
- [16] Z. Yin and J. Shi, “Geonet: Unsupervised learning of dense depth, optical flow and camera pose,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992, 2018.
- [17] T. Ke, T. Do, K. Vuong, K. Sartipi, and S. I. Roumeliotis, “Deep multi-view depth estimation with predicted uncertainty,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9235–9241, IEEE, 2021.
- [18] Z. Feng, L. Yang, L. Jing, H. Wang, Y. Tian, and B. Li, “Disentangling object motion and occlusion for unsupervised multi-frame monocular depth,” *arXiv preprint arXiv:2203.15174*, 2022.
- [19] V. Guizilini, R. Ambrus, D. Chen, S. Zakharov, and A. Gaidon, “Multi-frame self-supervised depth with transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 160–170, 2022.
- [20] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8934–8943, 2018.
- [21] X. Long, L. Liu, W. Li, C. Theobalt, and W. Wang, “Multi-view depth estimation using epipolar spatio-temporal networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8258–8267, 2021.

- [22] F. Wimbauer, N. Yang, L. Von Stumberg, N. Zeller, and D. Cremers, "Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6112–6122, 2021.
- [23] Z. Teed and J. Deng, "Deepv2d: Video to depth with differentiable structure from motion," *arXiv preprint arXiv:1812.04605*, 2018.
- [24] J. Hur and S. Roth, "Self-supervised monocular scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7396–7405, 2020.
- [25] V. Guizilini, K.-H. Lee, R. Ambrus, and A. Gaidon, "Learning optical flow, depth, and scene flow without real-world labels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3491–3498, 2022.
- [26] X. Meng, C. Fan, Y. Ming, and H. Yu, "Cornet: Context-based ordinal regression network for monocular depth estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4841–4853, 2021.
- [27] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8977–8986, 2019.
- [28] H. Kumar, A. S. Yadav, S. Gupta, and K. Venkatesh, "Depth map estimation using defocus and motion cues," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1365–1379, 2018.
- [29] Y. Wei, H. Guo, J. Lu, and J. Zhou, "Iterative feature matching for self-supervised indoor depth estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3839–3852, 2021.
- [30] T.-K. Lee, Y.-L. Chan, and W.-C. Siu, "Adaptive search range for heve motion estimation based on depth information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2216–2230, 2016.
- [31] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proceedings of the AAAI conference on artificial intelligence*, pp. 8001–8008, 2019.
- [32] Y. Chen, C. Schmid, and C. Sminchisescu, "Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7063–7072, 2019.
- [33] R. McCraith, L. Neumann, A. Zisserman, and A. Vedaldi, "Monocular depth estimation with self-supervised instance adaptation," *arXiv preprint arXiv:2004.05821*, 2020.
- [34] Y. Kuznetsov, M. Proesmans, and L. Van Gool, "Comoda: Continuous monocular depth adaptation using past experiences," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2907–2917, 2021.
- [35] A. CS Kumar, S. M. Bhandarkar, and M. Prasad, "Depthnet: A recurrent neural network architecture for monocular depth prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 283–291, 2018.
- [36] V. Patil, W. Van Gansbeke, D. Dai, and L. Van Gool, "Don't forget the past: Recurrent depth estimation from monocular video," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6813–6820, 2020.
- [37] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the IEEE international conference on computer vision*, pp. 66–75, 2017.
- [38] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, "Learning for disparity estimation through feature constancy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2811–2820, 2018.
- [39] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2821–2830, 2018.
- [40] S. Im, H.-G. Jeon, S. Lin, and I. S. Kweon, "Dpsnet: End-to-end deep plane sweep stereo," *arXiv preprint arXiv:1905.00538*, 2019.
- [41] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 767–783, 2018.
- [42] S.-J. Hwang, S.-J. Park, J.-H. Baek, and B. Kim, "Self-supervised monocular depth estimation using hybrid transformer encoder," *IEEE Sensors Journal*, 2022.
- [43] S. Chen, Z. Pu, X. Fan, and B. Zou, "Fixing defect of photometric loss for self-supervised monocular depth estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1328–1338, 2021.
- [44] G. Wang, J. Zhong, S. Zhao, W. Wu, Z. Liu, and H. Wang, "3d hierarchical refinement and augmentation for unsupervised learning of depth and pose from monocular video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 4, pp. 1776–1786, 2022.
- [45] J. Jin, J. Liang, Y. Zhao, C. Lin, C. Yao, and A. Wang, "A depth-bin-based graphical model for fast view synthesis distortion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1754–1766, 2018.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [47] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt, "Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance," in *European Conference on Computer Vision*, pp. 582–600, Springer, 2020.
- [48] S. Lee, S. Im, S. Lin, and I. S. Kweon, "Learning monocular depth in dynamic scenes via instance-aware projection consistency," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1863–1872, 2021.
- [49] S. Lee, F. Rameau, F. Pan, and I. S. Kweon, "Attentive and contrastive learning for joint depth and motion field estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4862–4871, 2021.
- [50] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding," in *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.
- [51] J. Hur and S. Roth, "Self-supervised multi-frame monocular scene flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2684–2694, 2021.
- [52] L. Liu, G. Zhai, W. Ye, and Y. Liu, "Unsupervised learning of scene flow estimation fusing with local rigidity," in *IJCAI*, pp. 876–882, 2019.
- [53] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [54] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, pp. 2650–2658, 2015.
- [55] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12240–12249, 2019.
- [56] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2624–2641, 2019.
- [57] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," *arXiv preprint arXiv:2010.16404*, 2020.
- [58] A. Johnston and G. Carneiro, "Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 4756–4765, 2020.
- [59] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically-guided representation learning for self-supervised monocular depth," *arXiv preprint arXiv:2002.12319*, 2020.
- [60] J. Wang, G. Zhang, Z. Wu, X. Li, and L. Liu, "Self-supervised joint learning framework of depth estimation via implicit cues," *arXiv preprint arXiv:2006.09876*, 2020.
- [61] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, "Unsupervised adversarial depth estimation using cycled generative networks," in *2018 international conference on 3D vision (3DV)*, pp. 587–595, IEEE, 2018.
- [62] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *2017 international conference on 3D Vision (3DV)*, pp. 11–20, IEEE, 2017.
- [63] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 340–349, 2018.
- [64] Y. Kuznetsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proceedings of the IEEE*

- conference on computer vision and pattern recognition*, pp. 6647–6655, 2017.
- [65] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2002–2011, 2018.
- [66] M. Poggi, F. Tosi, and S. Mattoccia, “Learning monocular depth estimation with unsupervised trinocular assumptions,” in *2018 International conference on 3d vision (3DV)*, pp. 324–333, IEEE, 2018.
- [67] S. Pillai, R. Amburuş, and A. Gaidon, “Superdepth: Self-supervised, super-resolved monocular depth estimation,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9250–9256, IEEE, 2019.
- [68] R. Mahjourian, M. Wicke, and A. Angelova, “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5667–5675, 2018.
- [69] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, “Learning depth from monocular videos using direct methods,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2022–2030, 2018.
- [70] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova, “What matters in unsupervised optical flow,” in *European Conference on Computer Vision*, pp. 557–572, Springer, 2020.
- [71] P. Liu, I. King, M. R. Lyu, and J. Xu, “Ddfow: Learning optical flow with unlabeled data distillation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8770–8777, 2019.
- [72] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.
- [73] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, “Lego: Learning edge with geometry all at once by watching videos,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 225–234, 2018.
- [74] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2007.
- [75] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
- [76] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [77] F. Gao, J. Yu, H. Shen, Y. Wang, and H. Yang, “Attentional separation-and-aggregation network for self-supervised depth-pose learning in dynamic scenes,” *arXiv preprint arXiv:2011.09369*, 2020.
- [78] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3061–3070, 2015.
- [79] R. Mohan and A. Valada, “Efficientps: Efficient panoptic segmentation,” *International Journal of Computer Vision*, vol. 129, no. 5, pp. 1551–1579, 2021.