Exploring Few-Beam LiDAR Assistance in Self-Supervised Multi-Frame Depth Estimation

Rizhao Fan[1,2], Matteo Poggi[2] and Stefano Mattoccia[2]

Abstract— **Self-supervised multi-frame depth estimation methods only require unlabeled monocular videos for training. However, most existing methods face challenges, including accuracy degradation caused by moving objects in dynamic scenes and scale ambiguity due to the absence of real-world references. In this field, the emergence of low-cost LiDAR sensors highlights the potential to improve the robustness of multi-frame depth estimation by exploiting accurate sparse measurements at the correct scale. Moreover, the LiDAR ranging points often intersect moving objects, providing more precise depth cues for them. This paper explores the impact of few-beam LiDAR data on self-supervised multi-frame depth estimation, proposing a method that fuses multi-frame matching and sparse depth features. It significantly enhances depth estimation robustness, particularly in scenarios involving moving objects and textureless backgrounds. We demonstrate the effectiveness of our approach through comprehensive experiments, showcasing its potential to address the limitations of existing methods and paving the way for more robust and reliable depth estimation based on this paradigm.**

I. INTRODUCTION

Accurate perception of dense depth maps is vital for various computer vision applications, including autonomous vehicles, robotic, and augmented reality. Among depth estimation techniques, monocular depth estimation has attracted notable attention in recent years, as witnessed by the significant body of literature in this field. Nonetheless, supervised monocular depth prediction networks require large-scale datasets with dense depth labels, which are highly expensive to collect [1]. Consequently, there has been a growing interest in self-supervised monocular depth estimation from unlabeled image sequences [2], [3], [4]. These latter methods, inspired by structure-from-motion (SfM), aim to simultaneously predict depth and camera pose, leveraging the geometric consistency between adjacent frames as supervision signals.

Early self-supervised monocular depth estimation approaches regress the dense depth map from a single frame image [3], [4], [5], [2]. Specifically, they utilized reprojection loss to encourage temporal depth consistency during training, ignoring the temporal frames available at test time in most practical applications. For instance, in real-world scenarios like autonomous driving, spatio-temporal adjacent frames are commonly accessible. Consequently, self-supervised monocular depth estimation evolved to a new paradigm utilizing multi-frame information for both training and inference [6], [7], [8], [9], [10], [11]. Hence, unlike single-frame depth estimation methods that predict depth per pixel utilizing a single individual image, multi-frame approaches achieve superior performance by incorporating temporal adjacent images at the testing time. These methods, based on *cost volumes*, learn multi-frame geometric features in addition to appearance-based features, achieving improved performance compared with single-frame methods.

However, single-frame and cost-volume-based methods rely upon the assumption of static environments, which conflicts with most real-world cases. Consequently, the reprojection loss often results in inaccuracies when predicting the depth values of moving objects. In order to address this issue, significant efforts have been made to improve depth prediction with moving objects [3], [6], [12], [13], [8], [11], [6], [10], [14]. Monodepth2 [3] uses auto-masking loss to disregard training pixels that violate camera motion assumptions. Manydepth [6] adopts monocular single-frame depth estimation to mask moving objects within a multi-frame framework. Guanghui *et al.* [10] utilize a motion-aware regularization loss to supervise regions with moving objects. These methods endeavor to devise specific loss functions tailored to address moving objects within dynamic scenarios and achieve some improvement.

Another issue affecting any self-supervised monocular depth estimation network concerns scale ambiguity, due to the absence of absolute depth cues. Since training relies on unlabeled monocular videos, the networks output a relative depth with an unknown scale factor due to the limited supervision of loss functions. The commonly used scale recovery method, median scaling [3], is unfeasible in practical applications since it requires ground truth depth data.

Although multi-beam sensors (with 64 beams or more) still come with a high cost, the price of few-beam LiDAR (e.g., 4 beams or fewer) has dropped to just a few hundred dollars [15], [16], [17]. Nonetheless, few-beam LiDAR can provide very sparse yet accurate depth measurements. Moreover, despite the limited vertical angle of LiDAR lasers, these sensors can capture ranging points on moving objects such as vehicles, pedestrians, and cyclists in self-driving scenarios. These ranging points are generated when the laser beam intersects objects, offering sparse distance data about the surroundings, which is crucial for environmental perception and decision-making in autonomous driving systems.

Considering the facts above, this paper proposes a novel self-supervised monocular multi-frame depth learning network assisted by sparse LiDAR depth measurements, as shown in Fig. 1. Our primary insight is to enhance the self-supervised depth estimation network with the prior knowl-

[1] Research Institute of Mine Artificial Intelligence, China Coal Research Institute

[2] University of Bologna

edge (depth and scale) from very sparse LiDAR data to effectively combine the advantage of the structured representation of matching costs, appearance-based features, and sparse LiDAR measurements.

This approach naturally solves the issues mentioned above: (1) depth maps are at the absolute scale, eliminating ambiguity. (2) the network accurately predicts depth for moving objects when the LiDAR hits them. We use a multi-stage depth estimation approach to fuse cost volume and sparse LiDAR data to predict depth maps in a coarse-to-fine manner. Additionally, a context network guides the depth propagation. Our proposed method significantly outperforms state-of-the-art methods that rely on or do not rely on sparse LiDAR data. To summarize, our key contributions are:

- We propose a novel self-supervised multi-frame depth estimation model assisted by sparse LiDAR data that combines the strengths of LiDAR and multi-view depth estimation.
- Our method significantly outperforms existing state-of-the-art methods on the KITTI datasets.

II. Related Work

In this section, we review self-supervised depth estimation approaches relevant to our proposal categorized into (1) single-frame, (2) multi-frame, and (3) self-supervised depth estimation with sparse depth measurements.

1) Single-Frame Monocular Depth Estimation: Self-supervised monocular depth estimation is a highly active research field faced according to two principal training methodologies: exploiting stereo images [18] or monocular videos [2], [3]. Garg *et al.* [19] proposed the first self-supervised depth estimation framework, using an image reconstruction loss computed on stereo images to train a monocular depth model. Zhou *et al.* [2] employ a depth and a pose network to use a photometric loss in monocular videos. Many researchers followed these paths improving upon them [12], [3], [20], [5], [21], [22], [23]. ViP-DeepLab [24] uses a multi-task network for jointly learning self-supervised depth estimation and panoptic segmentation from videos, while PackNet [12] leverages 3D convolutions to learn geometric representations.

2) Multi-Frame Monocular Depth Estimation: In contrast to single-frame methods, multi-frame methods enhance depth estimation by using multiple consecutive frames at test time. Multi-frame depth estimation methods [9], [6], [10], [11], [8] warp reference frame features to the current image using depth hypotheses and create a *cost volume* by measuring the similarity between the two. MonoRec [8] handles dynamic objects with this setting but needs sparse supervision obtained by a visual odometry system and long sequences for pose estimation. ManyDepth [6] utilizes geometry constraints to construct a cost volume and adopts a separate, single-frame model as a teacher network to encourage the network to ignore unreliable dynamic regions encoded by the volume. Feng *et al.* [11] proposed to use the pre-trained segmentation model to disentangle object motions and solve the mismatch problem in dynamic scenarios. DepthFormer [9] used a Transformer to improve the quality of cost volume through a series of self- and cross-attention layers. Zhong *et al.* [14] rely on multi-scale feature aggregation that strengthens both the spatial-temporal and texture features to improve the robustness of depth estimation during larger camera ego-motion.

3) Self-supervised Depth Estimation with Sparse depth measurements: A very recent trend [25], [26], [27], [28], [29], [**?**], [13] consists of estimating dense depth from images and *few-beams* LiDAR sensors – e.g., single-beam and 4-beams, in a self-supervised manner – reducing deployment cost at its minimum. We formulate this problem at the intersection between self-supervised depth estimation and depth completion, given the minimal impact of the few LiDAR scans available concerning the usual standard 64-beam setup for outdoor depth completion. Ma *et al.* [25] proposed a self-supervised training framework on sequences of color and sparse depth images with pose estimation using the PnP method. Feng *et al.* [29] proposed a representative solution in this field using a two-stage network to infer dense depth maps. LiDARTouch [**?**] explored self-supervised depth estimation with 64-beam LiDAR data in multiple depth completion networks. Fan *et al.* [13] used a lightweight yet effective self-supervised network processing few-beam LiDAR data and a single image.

Nonetheless, the methods mentioned above still have severe drawbacks. (1) Self-supervised single-frame depth prediction methods cannot reason about the temporal domain during inference time. (2) Self-supervised methods need auxiliary modules or pre-processes to handle the dynamic objects. (3) Self-supervised methods struggle with scale ambiguity without a real-world reference. Our proposal acts at different levels to overcome the aforementioned limitations.

III. Method

We now present our proposal: Sec. III-A introduces our self-supervised multi-frame depth estimation framework. Then, in Sec. III-B we introduce how to build cost volume, and in Sec. III-C we describe the depth network that fuses the cost volume and LiDAR data. Finally, Sec. III-D and Sec. III-E describe how to soften the impact of LiDAR outliers and our loss function.

A. Framework

Given the current frame $\mathbf{I}_t \in \mathbb{R}^{W\times H\times 3}$, the aligned sparse LiDAR depth map $\mathbf{S}_t \in \mathbb{R}^{H\times W}$ captured by a few-beam LiDAR (e.g. 4-beam) and their previous temporal counterparts $\mathbf{I}_s \in \mathbb{R}^{W\times H\times 3}$, $S_s \in \mathbb{R}^{H\times W}$ in a video stream, our goal is to estimate a dense depth map $\mathbf{D}_t \in \mathbb{R}^{H\times W}$ of $\mathbf{I}_t$ by taking advantage of multi-frame matching and sparse LiDAR data, as shown in Fig. 1. Following [2], [6], [3], we employ the standard self-supervised depth estimation paradigm. Specifically, estimated depth $\mathbf{D}_t$ is used together with pose $\mathbf{T}_{t\to s}$ predicted by a PoseNet to synthesize the scene from the target viewpoint using pixels from neighboring source frames $\mathbf{I}_s$ – i.e., $\mathbf{I}_{s\to t}$:

$$\mathbf{I}_{s\to t} = \mathbf{I}_s\Big\langle \mathit{proj}(\mathbf{D}_t, \mathbf{T}_{t\to s}, \mathbf{K})\Big\rangle \tag{1}$$

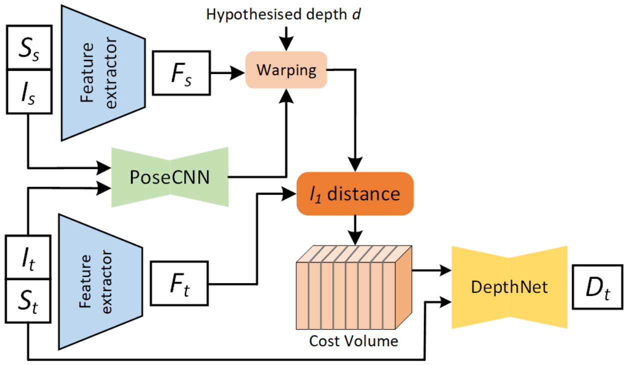


Fig. 1: **The main network architecture.** A shared feature extractor processes the input images. A PoseNet estimates the camera ego-motion from two frames during training, which is used to build a *Cost Volume*. The resulting volume, image, and sparse depth map are fed to DepthNet to generate D_t.

where $\langle\rangle$ is the bi-linear sampling operator and *proj()* returns the 2D coordinates of the depths in D_t when reprojected into the camera of $\mathbf{I}_s$. The PoseNet uses a modified ResNet18 [30], taking two stacked frames $\{\mathbf{I}_t, \mathbf{I}_s\}$ as input to infer their 6-DoF relative pose $\mathbf{T}_{t\to s}$. Following [6], for each pixel, we optimize the loss $\mathcal{L}_m$ for the best-matching source image by selecting the per-pixel minimum over the reconstruction loss $\mathcal{L}_p$

$$\mathcal{L}_m = \min_n \mathcal{L}_p(\mathbf{I}_t, \mathbf{I}_{s\to t}). \tag{2}$$

where $s \in \{t-1, t+1,\}$ is used for training.

The reconstruction loss $\mathcal{L}_p$ consists of a structure similarity (SSIM) term and absolute error ($\mathcal{L}_1$) term, and we minimize it over all the pixels at three output scales.

The smoothness loss is adopted to regularize the dense depth maps by utilizing texture information from the input color image [3]:

$$\mathcal{L}_{sm} = |\ \partial_x d^*\ | e^{-|\partial_x \mathbf{I}_t|} + |\ \partial_y d^*\ | e^{-|\partial_y \mathbf{I}_t|}, \tag{3}$$

with ∂_x, ∂_y being gradients along x and y direction, and d^* is the normalized inverse depth map.

Additionally, we used other loss functions during the model training, introduced in the remainder.

B. Cost Volume Computation

Given a current frame $\{\mathbf{I}_t, \mathbf{S}_t\}$ and its nearby frame $\{\mathbf{I}_s, \mathbf{S}_s\}$, we firstly leverage a shared encoder ResNet18 [30] Θ_{enc} to extract 2D features of these consecutive frames. We modify the initial layer of ResNet18 to accommodate the inputs – the concatenation of the image and the sparse depth map. The inputs are downscaled to lower resolution deep features $\mathbf{F}_t$, $\mathbf{F}_s$ at $1/4$ input size.

Unlike previous learning-based multi-frame methods, which get 'up to scale' depth prior provided by a coarse depth estimation from a single image [6], [14], [11], we obtain depth prior knowledge from the LiDAR depth range, $0.1 \sim 100m$. We define N = 96 depth planes uniformly sampled in depth space, and the hypothesized depths d_i as

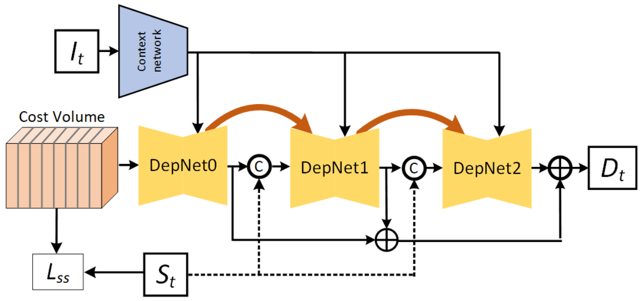


Fig. 2: **DepthNet architecture.** It comprises four main components: three cascade depth estimation networks, and a context network. The context network extracts contextual features from the input image. Three cascade DepthNet predict the final result in a coarse-to-fine way.

$$d_i = d_{\min} + \frac{i \times (d_{\max} - d_{\min})}{N-1}, \quad i = 0, 1, \cdots, N-1. \tag{4}$$

where d_{min} and d_{max} have a known scale consistent with the LiDAR depth.

Given the known camera intrinsic $\mathbf{K}$ and the estimated pose $\mathbf{T}_{t\to s}$, the source feature $\mathbf{F}_s$ are warped to the viewpoint of $\mathbf{I}_t$ and get a warped feature map

$$\mathbf{F}^i_{s\to t} = \mathbf{F}_s < proj(d_i, \mathbf{T}_{t\to s}, \mathbf{K}) >, \tag{5}$$

The cost volume $\mathbf{CV} \in \mathbb{R}^{D\times H/4\times W/4}$ is constructed as the absolute difference between the warped features $F^i_{s\to t}$ and the features $\mathbf{F}_t$. The cost volume $\mathbf{CV}$ is concatenated with features $\mathbf{F}_t$ and used as input to depth prediction network **DepthNet**, which regresses the depth map $\mathbf{D}_t$.

Cost volumes help the networks to use inputs effectively since they represent matching relationships between pixels. Indeed, this step is crucial in inferring depth from multi-frame images by measuring geometric compatibility between pixels of two nearby frames. From it, we can get a downsampled coarse depth map

$$\mathbf{D}_{coarse} = \arg\min(\mathbf{CV}), \tag{6}$$

We use a loss function L_{ss} for the cost volume to optimize the model, which we will describe later.

C. DepthNet

We build a multi-stage DepthNet, sketched in Fig 2, that integrates the image context, cost volume matching information, and sparse LiDAR data to regress the final depth map. However, fusing these cues directly is not trivial because each modality conveys different information. Sparse LiDAR data provides accurate metric depth and serve as a reference for scaled depth estimation, whereas the image context information plays a crucial role in providing visual features and guidance to estimate missing depth values accurately. We integrate them through the multiple stages of our model.

Context Network. We use a compact network to extract multi-scale features only from the input image $\mathbf{I}_t$. The extracted image features $\mathbf{F}_{cont}$ have cumulative strides of 1, 2, 4, 8, and 16, respectively. The multi-scale image features contain contextual and semantic cues that are added to depth features at the DepthNet multi-stage decoders.

Multi-stage Depth Network. We utilize three cascade depth estimation networks – DepNet0, DepNet1, and DepNet2 in Fig. 2 – to predict the depth map in a coarse-to-fine manner. All the DepNets modules share a similar structure but have different specific settings. Each encoder relies on CNN layers, having 64 output channels each. The decoder features of the DepNet are sent to the following DepNet encoder network by skip connection, which helps propagate the learning depth representation.

DepNet0 Θ_{D_0} takes cost volume $\mathbf{CV}$ as input and predicts a quarter-sized depth map $\mathbf{D}_0$.

$$\mathbf{D}_0 = \Theta_{\mathbf{D}_0}(\mathbf{CV}, \mathbf{F}_{cont}) \tag{7}$$

DepNet1 $\Theta_{\mathbf{D}_1}$ takes the combination of a half-sized down-sampled depth input $\mathbf{S}_t^{'}$ and the upsampled $\mathbf{D}_0^{'}$ as input and predicts a half-sized depth map $\mathbf{D}_1$.

$$\mathbf{D}_1 = \Theta_{\mathbf{D}_1}(\mathbf{S}_t^{'}, \mathbf{D}_0^{'}, \mathbf{F}_{cont}, \mathbf{F}_{dec}^{\mathbf{D}_0}) \tag{8}$$

Similarly, DepNet2 $\Theta_{\mathbf{D}_2}$ works as:

$$\mathbf{D}_2 = \Theta_{\mathbf{D}_2}(\mathbf{S}_t, \mathbf{D}_1^{'}, \mathbf{F}_{cont}, \mathbf{F}_{dec}^{\mathbf{D}_1}) \tag{9}$$

Residual connections integrate the three upsampled results at full resolution.

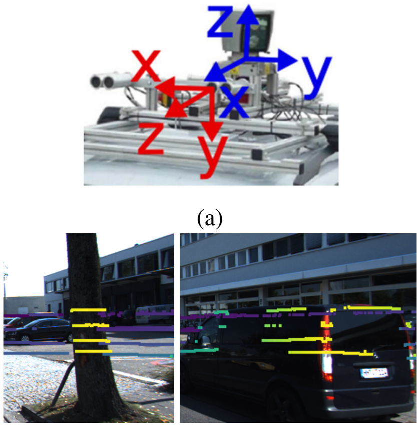
$$\mathbf{D} = \mathbf{D}_2 + \mathbf{UP}(\mathbf{D}_1 + \mathbf{UP}(\mathbf{D}_0)) \tag{10}$$

where $\mathbf{UP}$ is the upsample operation. This way, the network generates the final prediction in a coarse-to-fine manner.

D. LiDAR Outlier Removal

In the KITTI recording platform, the displacement between the LiDAR and the camera maps some occluded points from the background onto the foreground object, as shown in Fig. 3. This is common to any acquisition setup: since the two observe the world from distinct perspectives, certain occluded areas may be detected solely by the LiDAR sensor, remaining unseen by the camera [45]. In the supervised setting, the network can weaken the impact of outliers under supervision [46], [47], [1]. However, mapping those points on the image plane in our self-supervised setting would create incorrect depth values without handling outliers in LiDAR data. Outlier points typically exhibit larger depth values compared to nearby accurate depth values, as they are from a more distant background. Due to the displacement between the camera and LiDAR, this occurs in particular near depth discontinuities and objects edges. For the sparse LiDAR depth map $\mathbf{S}$, we first get the inverted depth map d

$$d = 100.0m - \mathbf{S} \tag{11}$$



(a)

(b)

Fig. 3: **Outliers in LiDAR data.** (a) KITTI setup with displaced sensors; (b) examples of LiDAR measurements, with overlapping background and foreground points. We can observe that points with significant differences in depth values interlace in the edge area of the same object.

and then we max-pool it to get d_{max}

$$d_{max} = \max(d) \tag{12}$$

This operation can remove the points with smaller values, namely those with larger values in the original depth map. Then, we compare d_{max} and d and get a mask $\mathbf{M}_o$ for the outliers $\mathbf{O}$

$$\mathbf{M}_o(x) = \begin{cases} 0 & \text{if } |d_{max} - d| \geq \tau \\ 1 & \text{if } |d_{max} - d| < \tau \end{cases} \tag{13}$$

where $< \tau$ is set $2.0m$. Accordingly, 0 indicates that the point is considered an outlier.

E. Depth Consistency Loss Function

Apart from the photometric loss function described in Sec. III-A and commonly used in self-supervised depth estimation, we use the sparse depth-consistency-loss map to optimize our model. Following [29], [13], we enforce consistency between the predicted and sparse depth using the scale-invariant [48] depth loss:

$$\begin{aligned} \mathcal{L}_{si}(\mathbf{D}, \mathbf{S}) = \frac{1}{2n^2} \sum_{i,j} ((\log \mathbf{D}_i - \log \mathbf{D}_j) \\ -(\log \mathbf{S}_i - \log \mathbf{S}_j))^2 \end{aligned} \tag{14}$$

with n being the number of pixels belonging to Ω – i.e., the set for which sparse LiDAR measurements are available.

The overall depth consistency loss $\mathcal{L}_{sd}$ is then defined as:

$$\begin{aligned} \mathcal{L}_{sd} = \omega(& \sum_{x \in \Omega} \mathcal{L}_{si}(\mathbf{D}_0^{''}(x), \mathbf{S}(x)) \\ & + \sum_{x \in \Omega} \mathcal{L}_{si}(\mathbf{D}_1^{'}(x), \mathbf{S}(x))) \\ & + \sum_{x \in \Omega} \mathcal{L}_{si}(\mathbf{D}_2(x), \mathbf{S}(x)) \end{aligned} \tag{15}$$

Method	Test frames	LiDAR	W×H	Abs. Rel.	Sq. Rel.	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Guizilini et al. [31]	1	✗	640×192	0.102	0.698	4.381	0.178	0.896	0.964	0.984
Johnston et al. [32]	1	✗	640×192	0.106	0.861	4.699	0.185	0.889	0.962	0.982
Monodepth2 [3]	1	✗	640×192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SFM [12]	1	✗	640×192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
RM-Depth [33]	1	✗	640×192	0.108	0.710	4.513	0.183	0.884	0.964	0.983
RA-depth [34]	1	✗	640×192	0.096	0.632	4.216	0.171	0.903	0.968	0.985
DIFFNet [35]	1	✗	640×192	0.102	0.764	4.483	0.180	0.896	0.965	0.983
MonoViT [36]	1	✗	640×192	0.099	0.708	4.372	0.175	0.900	0.967	0.984
CoMoDA [37]	N	✗	640×192	0.103	0.862	4.594	0.183	0.899	0.961	0.981
TC-Depth [38]	3(-1, 0, +1)	✗	640×192	0.103	0.746	4.483	0.185	0.894	-	0.983
DRO [39]	2 (-1, 0)	✗	640×192	0.099	0.813	4.478	0.192	0.881	0.957	0.980
DepthFormer [9]	2 (-1, 0)	✗	640×192	0.090	0.661	4.149	0.175	0.905	0.967	0.984
ManyDepth [6]	2 (-1, 0)	✗	640×192	0.098	0.770	4.459	0.176	0.900	0.965	0.983
Long et al. [40]	2 (-1, 0)	✗	640×192	0.097	0.731	4.392	0.176	0.901	0.965	0.983
DynamicDepth [11]	2 (-1, 0)	✗	640×192	0.096	0.720	4.458	0.175	0.897	0.964	0.984
MGDepth [41]	2 (-1, 0)	✗	640×192	0.091	0.650	4.263	0.171	0.908	0.967	0.984
MOVEDepth [42]	2 (-1, 0)	✗	640×192	0.089	0.663	4.216	0.169	0.904	0.966	0.984
CAT-Net [10]	2 (-1, 0)	✗	480×160	0.086	0.681	4.246	0.180	0.902	0.960	0.983
Xiang et al. [43]	2 (-1, 0)	✗	640×192	0.086	0.613	4.096	0.165	0.915	0.969	0.985
Guizilini et al. [44]	1	✓	640×192	0.082	0.424	3.73	**0.131**	0.917	-	-
FusionDepth (Initial) [29]	1	✓	640×192	0.078	0.515	3.67	0.154	0.935	0.973	0.986
FusionDepth (Refined) [29]	1	✓	640×192	0.074	0.423	3.61	0.150	0.936	0.973	0.986
GSCNN [13]	1	✓	640×192	0.069	0.476	3.31	0.144	0.943	**0.975**	**0.987**
Ours	2 (-1, 0)	✓	640×192	**0.058**	**0.407**	**3.154**	0.142	**0.948**	**0.975**	0.986

TABLE I: **Comparison with existing self-supervised methods on the KITTI [1] Eigen split.** We report: at the top, monocular methods using one frame at test time; in the middle, multi-frame methods with multiple frame inputs at test time; at the bottom, self-supervised methods using LiDAR data. The best results are in **bold**.
L: LiDAR, all the input frames have **640×192** resolution.

where $\mathbf{D}_0^{''}$, $\mathbf{D}_1^{'}$ is the result of upsampling $\mathbf{D}_0$, $\mathbf{D}_1$ to full size, respectively. It is worth noting that the sparse depth map $\mathbf{S}$ is filtered according to the outlier mask $\mathbf{M}_o$. Furthermore, ω is a hyper-parameter to control the impact of the loss on intermediate predictions. Specifically, we use a multi-stage training scheme by setting ω = 1 for the first 10 epochs, then reducing it to 0.5 for the following 10 epochs, and finally reducing it to 0.1 until convergence.

We also use the scale-invariant loss to construct the sparse supervised loss function $\mathcal{L}_{ss}$ to optimize the cost volume.

$$\mathcal{L}_{ss} = \mathcal{L}_{si}(\mathbf{D}_{coarse}^{''}, \mathbf{S}) \tag{16}$$

with $\mathbf{D}_{coarse}^{''}$ being the result of upsampling to the input size.

IV. Experiments

We now discuss the outcomes of our evaluation.

A. Dataset

We conduct experiments on the widely-used KITTI [1] dataset, a standard benchmark for evaluating depth estimation networks. It is an outdoor dataset framing driving scenarios and includes raw LiDAR data, as well as refined, ground truth depth. For what concerns the few-beam LiDAR setting, we sample four beams from the raw LiDAR data, following [29], [13], and evaluate our method on the Eigen split [48] – both with data preprocessing from [2], as well as with the improved ground truth [49]. The data is divided into 39 810, 4424, and 697 (652 when using the improved ground truth) training, validation, and test images respectively.

B. Implementation Details

Following the literature [6], [42], [3], we use color-jitter and flip as training-time augmentations, and resize input frames to 640×192 resolution. We adopt two consecutive image-LiDAR frames $\{(\mathbf{I}_s, \mathbf{S}_s), (\mathbf{I}_t, \mathbf{S}_t)\}$ for cost volume construction, both at training and testing time, while we use $\{\mathbf{I}_s$ and $\mathbf{I}_t\}$ images to compute the reprojection loss, and the LiDAR data $\mathbf{S}_t$ for the depth consistency loss. We train our model on 1 NVIDIA RTX 3090 GPU with batch size 16, using Adam [50] for 30 epochs with an initial learning rate of 0.0001, dropped by a factor of 10 every 10 epochs.

C. KITTI results

We compare our method with three different state-of-the-art, self-supervised strategies for inferring depth from images: (1) self-supervised monocular single-frame methods [31], [32], [3], [12], [33], [34], [35], (2) self-supervised monocular multi-frame methods [37], [38], [39], [9], [6], [11], [41], [42], (3) self-supervised single-frame methods with few-beam LiDAR data as guidance [44], [29], [13].

Tab. I and Tab. II present a quantitative comparison between the performance of our method and state-of-the-art networks on KITTI, with respect to raw 64-line LiDAR [1] and improved ground truth [49] respectively. In both cases, our method achieves the best performance among all the competitors, single-frame, multi-frame, and other self-supervised methods with LiDAR on most metrics. This outcome highlights the notable advantage brought by our method, which effectively leverages the very sparse guidance obtained from few-beam LiDAR data together with the richer scene understanding enabled by monocular videos.

Method	Test frames	LiDAR	W×H	Abs. Rel.	Sq. Rel.	RMSE	RMSE$_{\log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Johnston et al. [32]	1	✗	640×192	0.081	0.484	3.716	0.126	0.927	0.985	0.996
Monodepth2 [3]	1	✗	640×192	0.090	0.545	3.942	0.137	0.914	0.983	0.995
PackNet-SFM [12]	1	✗	640×192	0.078	0.420	3.485	0.121	0.931	0.986	0.996
RA-depth [34]	1	✗	640×192	0.074	0.362	3.345	0.113	0.940	0.990	0.997
DIFFNet [35]	1	✗	640×192	0.076	0.412	3.494	0.119	0.935	0.988	0.996
MonoViT [36]	1	✗	640×192	0.075	0.389	3.419	0.115	0.938	0.989	0.997
DepthFormer [9]	2 (-1, 0)	✗	640×192	0.055	0.271	2.917	0.095	0.955	0.991	**0.998**
ManyDepth [6]	2 (-1, 0)	✗	640×192	0.070	0.399	3.455	0.113	0.941	0.989	0.997
DynamicDepth [11]	2 (-1, 0)	✗	640×192	0.068	0.362	3.454	0.111	0.943	0.991	**0.998**
MOVEDepth [42]	2 (-1, 0)	✗	640×192	0.065	0.377	3.449	0.112	0.942	0.988	0.996
Long et al. [40]	2 (-1, 0)	✗	640×192	0.068	0.366	3.338	0.110	0.946	0.989	0.997
Xiang et al. [43]	2 (-1, 0)	✗	640×192	0.058	0.302	3.070	0.098	0.955	0.992	**0.998**
GSCNN [13]	1	✓	640×192	0.058	0.257	2.526	0.096	0.972	0.992	0.997
LiDARTouch [?]	1	✓	640×192	**0.044**	0.242	2.504	0.086	0.974	0.991	0.996
Ours	2 (-1, 0)	✓	640×192	**0.044**	**0.170**	**2.071**	**0.078**	**0.982**	**0.995**	**0.998**

TABLE II: **Comparison with existing self-supervised methods on the KITTI [1] Eigen split – improved ground truth [49].** We report: at the top, monocular methods using one frame at test time; in the middle, multi-frame methods with multiple frame inputs at test time; at the bottom, self-supervised methods using LiDAR data. The best results are in **bold**.

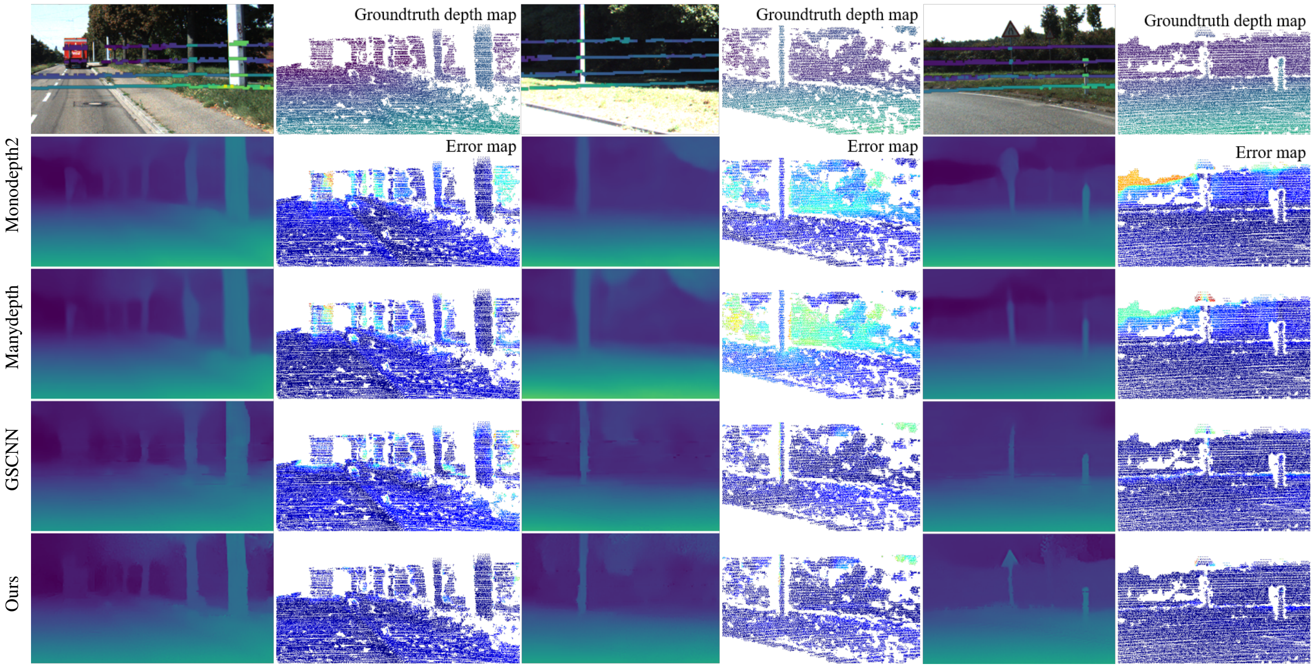


Fig. 4: **Qualitative results on KITTI.** For three cases, the first row reports (left) the input image $\mathbf{I}_t$ with superimposed the few-beam LiDAR measurements $\mathbf{S}_t$, and (right) the ground truth. For each method, we report the depth map and the Abs. Rel. error map [49] using the same colormap as [6]. The figure is best viewed in color and by zooming in.

Fig. 4 facilitates a qualitative analysis between our framework and representative methods on the KITTI dataset [49], reporting three cases characterized by different challenges: the presence of a dynamic object at the distance (left), a textureless scene (middle), and a completely static scene (right). We can notice how our method exhibits remarkably superior performance with respect to any of the competitors. Although all methods get qualitatively similar results in static areas, such as the ground regions and closer objects, our method preserves the shape of the objects far in the scene, and achieves accurate predictions across all regions. GSCNN [13] and ours, using few-beam LiDAR data, can predict more accurate results on dynamic objects and when dealing with textureless regions compared to methods relying uniquely on images – Monodepth2 [3] and Manydepth [6].

Benefiting from the semantic information provided by the context network and the coarse depth information provided by the cost volume, our method can better preserve fine details in comparison to the competitors – e.g., as when dealing with flagpoles and traffic signs.

D. Ablation Study

We conclude with some ablation studies aimed at measuring the impact of the different design choices in our framework. These experiments are carried out on the KITTI eigen split [48], using raw LiDAR as ground truth.

Methods	Abs. Rel.	Sq. Rel.	RMSE
Multi-stage	**0.058**	**0.407**	**3.154**
Single-stage	0.114	0.655	3.953

TABLE III: **Ablation study – single/multi-stage DepthNet.**

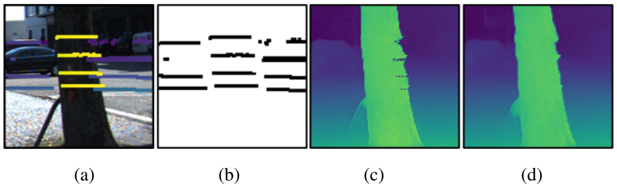


Fig. 5: **Qualitative result – impact of outliers on predictions.** (a) Image with few-beam LiDAR data, (b) filtered LiDAR according to $\mathbf{M}_o$, predictions by our model using (c) raw LiDAR affected by noise, or (d) filtered LiDAR.

Firstly, we analyze the improvement that the multi-stage design brings. Then, we evaluate the impact of outliers in the raw LiDAR data on our self-supervised framework, and finally the impact of the loss functions we enforced.

Multi-Stage DepthNet. We compare our multi-stage design with the commonly used single-stage network, e.g. the depth network in Manydepth [6]. In our method, DepthNet takes cost volume, image, and few-beam LiDAR data as inputs, while others only take the constructed cost volume. Accordingly, we replace the multi-stage DepthNet with the single encoder-decoder depth network from Manydepth [6]. To ensure a fairer comparison, we use a simple CNN layer to extract features from the concatenation of image and LiDAR data and then set it as inputs of the DepthNet. Table III shows that the multi-stage design is better than the single-stage one at fusing multiple modality data in the current setting.

Outliers in LiDAR data. The presence of outliers in the LiDAR measurements can weaken our model and cause *holes* in the final prediction, both during training – if not properly masked during $\mathcal{L}_{sd}$ – or when processed as the input at test time. We now measure the impact of our filtering strategy – detailed in Sec. III-D – on the final results. Tab. IV collects the results obtained by using the outlier mask $\mathbf{M}_o$ when applied to $\mathbf{S}_t$ when used as input to the network or to compute $\mathcal{L}_{ds}$ to neglect the effect of outliers on self-supervision. Interestingly, filtering outliers only when computing $\mathcal{L}_{ds}$ gives the best results. Fig. 4 shows this effect qualitatively, highlighting the holes appearing in the final prediction when the mask is not used (c), whereas it gets free of them when outliers are filtered out (d).

Loss functions. The sparse depth consistency loss is essential in our model since it helps the model learn knowledge from the prior LiDAR signal. Purposely, we compare four different settings for our baseline method to evaluate the impact of our loss functions. Tab. V reports the outcome, showing that the depth consistency loss functions significantly improve the network accuracy when used, especially when employed to supervise the final prediction – $\mathcal{L}_{sd}$.

Input	$\mathcal{L}_{sd}$	Abs. Rel.	Sq. Rel.	RMSE
✓	✓	**0.058**	0.451	3.319
	✓	**0.058**	**0.407**	**3.154**
		0.065	0.430	3.231

TABLE IV: **Ablation study – outlier mask.** ✓indicates where the outlier mask is adopted.

$\mathcal{L}_{sd}$	$\mathcal{L}_{ss}$	Abs. Rel.	Sq. Rel.	RMSE
✓	✓	**0.058**	**0.407**	**3.154**
✓		0.059	0.452	3.250
	✓	0.089	0.531	3.507
		0.090	0.568	3.616

TABLE V: **Ablation study – loss functions.** ✓indicates where a specific loss term is adopted.

V. CONCLUSIONS

We presented a self-supervised multi-frame depth estimation method assisted by few-beam LiDAR data. It enjoys the benefits of both multi-frame and cheap LiDAR depth measurements in synergy, by exploiting geometry from multi-frame feature matching and the prior knowledge (depth and scale) provided by LiDAR data, making it more robust with dynamic objects and yielding more accurate predictions. Our framework achieves state-of-the-art results on the KITTI dataset, outperforming any existing self-supervised solution either processing single/multiple frames only or being assisted by few-beam LiDAR data.

Acknowledgment. This study was carried out within the MOST – Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU – PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. This study was also supported by the National Key Research and Development Program of China (Grant No. 2023YFC2907600), Key Science and Technology Innovation Project of CCTEG (No. 2021-TD-ZD002), and China Scholarship Council.

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*, 2012.

[2] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858.

[3] 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*, 2019, pp. 3828–3838.

[4] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[5] 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*. Springer, 2020, pp. 582–600.

[6] 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*, 2021, pp. 1164–1174.

[7] 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.

[8] 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, 2021, pp. 6112–6122.

[9] V. Guizilini, R. Ambruș, 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*, 2022, pp. 160–170.

[10] G. Wu, H. Liu, L. Wang, K. Li, Y. Guo, and Z. Chen, "Self-supervised multi-frame monocular depth estimation for dynamic scenes," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[11] Z. Feng, L. Yang, L. Jing, H. Wang, Y. Tian, and B. Li, "Disentangling object motion and occlusion for unsupervised multi-frame monocular depth," in *European Conference on Computer Vision*. Springer, 2022, pp. 228–244.

[12] 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*, 2020, pp. 2485–2494.

[13] R. Fan, F. Tosi, M. Poggi, S. Mattoccia *et al.*, "Lightweight self-supervised depth estimation with few-beams lidar data," in *The 34th British Machine Vision Conference Proceedings*. British Machine Vision Association's (BMVA), 2023, pp. 1–15.

[14] J. Zhong, X. Huang, and X. Yu, "Multi-frame self-supervised depth estimation with multi-scale feature fusion in dynamic scenes," *arXiv preprint arXiv:2303.14628*, 2023.

[15] Hokuyo, https://www.hokuyo-aut.jp, 2021.

[16] Quanergy, https://quanergy.com/products/, 2023.

[17] Hesaitech, https://www.hesaitech.com/product/, 2023.

[18] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, "On the synergies between machine learning and binocular stereo for depth estimation from images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5314–5334, 2022.

[19] 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*. Springer, 2016, pp. 740–756.

[20] 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*. Springer, 2020, pp. 572–588.

[21] H. Jung, E. Park, and S. Yoo, "Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 642–12 652.

[22] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 5848–5854.

[23] M. Poggi, F. Tosi, F. Aleotti, and S. Mattoccia, "Real-time self-supervised monocular depth estimation without gpu," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2022.

[24] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3997–4008.

[25] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3288–3295.

[26] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (ddp) from single image and sparse range," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3353–3362.

[27] A. Wong, X. Fei, S. Tsuei, and S. Soatto, "Unsupervised depth completion from visual inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1899–1906, 2020.

[28] J. Choi, D. Jung, Y. Lee, D. Kim, D. Manocha, and D. Lee, "Selfdeco: Self-supervised monocular depth completion in challenging indoor environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 467–474.

[29] Z. Feng, L. Jing, P. Yin, Y. Tian, and B. Li, "Advancing self-supervised monocular depth learning with sparse lidar," in *Conference on Robot Learning*. PMLR, 2022, pp. 685–694.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] 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.

[32] 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*, 2020, pp. 4756–4765.

[33] T.-W. Hui, "Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1675–1684.

[34] M. He, L. Hui, Y. Bian, J. Ren, J. Xie, and J. Yang, "Ra-depth: Resolution adaptive self-supervised monocular depth estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 565–581.

[35] H. Zhou, D. Greenwood, and S. Taylor, "Self-supervised monocular depth estimation with internal feature fusion," *arXiv preprint arXiv:2110.09482*, 2021.

[36] C. Zhao, Y. Zhang, M. Poggi, F. Tosi, X. Guo, Z. Zhu, G. Huang, Y. Tang, and S. Mattoccia, "Monovit: Self-supervised monocular depth estimation with a vision transformer," in *2022 international conference on 3D vision (3DV)*. IEEE, 2022, pp. 668–678.

[37] Y. Kuznietsov, 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*, 2021, pp. 2907–2917.

[38] P. Ruhkamp, D. Gao, H. Chen, N. Navab, and B. Busam, "Attention meets geometry: Geometry guided spatial-temporal attention for consistent self-supervised monocular depth estimation," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021.

[39] X. Gu, W. Yuan, Z. Dai, S. Zhu, C. Tang, Z. Dong, and P. Tan, "Dro: Deep recurrent optimizer for video to depth," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2844–2851, 2023.

[40] Y. Long, H. Yu, and B. Liu, "Two-stream based multi-stage hybrid decoder for self-supervised multi-frame monocular depth," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 291–12 298, 2022.

[41] K. Zhou, J.-X. Zhong, J.-W. Bian, Q. Xie, J.-Q. Zheng, N. Trigoni, and A. Markham, "Mgdepth: Motion-guided cost volume for self-supervised monocular depth in dynamic scenarios," *arXiv preprint arXiv:2312.15268*, 2023.

[42] X. Wang, Z. Zhu, G. Huang, X. Chi, Y. Ye, Z. Chen, and X. Wang, "Crafting monocular cues and velocity guidance for self-supervised multi-frame depth learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 2689–2697.

[43] J. Xiang, Y. Wang, L. An, H. Liu, and J. Liu, "Exploring the mutual influence between self-supervised single-frame and multi-frame depth estimation," *IEEE Robotics and Automation Letters*, 2023.

[44] V. Guizilini, J. Li, R. Ambrus, S. Pillai, and A. Gaidon, "Robust semi-supervised monocular depth estimation with reprojected distances," in *Conference on robot learning*. PMLR, 2020, pp. 503–512.

[45] A. Conti, M. Poggi, F. Aleotti, and S. Mattoccia, "Unsupervised confidence for lidar depth maps and applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, iROS.

[46] R. Fan, Z. Li, M. Poggi, and S. Mattoccia, "A cascade dense connection fusion network for depth completion," in *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022.

[47] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4796–4803.

[48] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[49] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *2017 international conference on 3D Vision (3DV)*. IEEE, 2017, pp. 11–20.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.