Lightweight Self-Supervised Depth Estimation with few-beams LiDAR Data

Rizhao Fan
rizhao.fan@unibo.it
Fabio Tosi
fabio.tosi5@unibo.it
Matteo Poggi
m.poggi@unibo.it
Stefano Mattoccia
stefano.mattoccia@unibo.it

Department of Computer Science and Engineering (DISI),
University of Bologna,
Italy

Abstract

This paper proposes a lightweight yet effective self-supervised depth completion network trained on monocular videos and sparse raw LiDAR measurements only. Specifically, we utilize a multi-stage network architecture, which depends on cheap CNN layers. We introduce a novel guided sparse convolution operator combining sparse and dense data to extract depth features. To mitigate the impact of outliers commonly present in the sparse raw LiDAR data, we adopt a distance-dependent outlier mask that incorporates an elastic threshold mechanism to selectively discard such points. Our experimental results on the KITTI dataset show the favorable trade-off between accuracy and efficiency achieved by our model, reaching state-of-the-art performance on self-supervised depth estimation from few-beams LiDAR (4-beams), depth completion (64-beams) and a few hundred depth points, using a fraction of the parameters. Our code will be available on `https://github.com/franky-ciomp/GSCNN/`.

1 Introduction

Estimating the depth of a sensed scene is one of the tasks at the core of many high-level applications inherent to navigation and interaction with the surrounding environment. For this purpose, several sensors hit the market in the last decade, capable of measuring the distances at which objects are by emitting signals and deriving depth from their interaction with the environment – e.g., based on the Time-of-Flight (ToF) of a laser impulse, as in the case of LiDARs. Although accurate up to a hundred meters, these sensors provide only sparse depth measurements, depending on the number of physical emitters they deploy and other technological factors, with a resolution dramatically lower than that of standard cameras. For instance, the Velodyne HDL-64e sensor has been one of the most popular sensors in recent years and emits up to 64 laser beams simultaneously, yet it only provides about 6% of the number of pixels in a 0.3Mpx image [13]. Consequently, depth completion [40] emerged as a vivid research trend, leveraging deep learning to obtain dense depth maps from sparse LiDAR data and the guidance of a color image.

Despite the accurate results achieved over the years [26, 30, 52], two main shortcomings still limit the deployment of these approaches as a mature technology. On the one hand, dense ground truth depth annotation is needed to train completion networks. To obtain such data, manual labor [40] is necessary to aggregate several scans performed over time, possibly by means of a high density LiDAR sensor. Self-supervised depth completion approaches [6, 28, 46, 49] try to soften this constraint by learning from monocular videos and sparse LiDAR how to infer dense depth maps. On the other hand, there is the much higher cost of LiDAR sensors compared to conventional color cameras: indeed, this scales with the density of measurements the sensor can deliver, with 64-beams LiDARs – and more recent, 128-beams devices – costing tens of thousands of dollars. At the same time, cheaper solutions characterized by much fewer emitters (e.g. 4-beams) exist, at the expense of making the completion task even more challenging. On this track, estimating depth from *few-beams* LiDAR data [10, 16], possibly in a self-supervised manner, represents the cheapest chance to develop a framework capable of densifying sparse depth measurements and requiring low-cost depth sensors to deploy it. Nonetheless, solutions proposed so far [10, 16] still rely on very complex CNNs, counting tens of millions of parameters and thus putting some constraints on the hardware capabilities required for deployment.

In this paper, we take a further step toward inexpensive solutions for densifying sparse LiDAR data developing a lightweight yet effective self-supervised network for this task. Our proposal involves the use of a multi-stage architecture that is designed to effectively utilize the guidance provided by color images during the densification process. For this purpose, we revise Sparsity-invariant CNNs [40] and introduce a novel layer called Guided Sparsity-invariant CNNs, capable of effectively processing the contextual information provided by dense guidance. Moreover, to further improve the accuracy of our model, we implement a Distance-Dependent Outlier Mask, capable of mitigating the impact of outliers in the sparse data on the resultant dense depth map.

The main contributions of our work can be summarized as follows:

- We propose a lightweight yet effective self-supervised network processing few-beams LiDAR data and a single image. It counts as few as $\sim$600K parameters.
- At the core of our architecture, we propose a Guided Sparsity-invariant CNN block, which can deal with sparse data to produce depth features under the guidance of dense color images or depth maps.
- To cope with outliers in the sparse input data, we introduce a Distance-Dependent Outlier Mask to mitigate the impact on the final predictions.
- We evaluate our framework processing data from cheap (4-beams) and expensive (64-beams) LiDAR sensors, achieving state-of-the-art performance in the former case and yielding results equivalent to existing models in the latter case, despite utilizing only about 2% of the parameters required by those models.

2 Related Work

This section provides an overview of the literature that is pertinent to the task we face.

Supervised depth completion. Approaches for predicting depth out of a single image include depth estimation and completion, with the latter being the most relevant to our work.

Nonetheless, for these tasks, high-performance LiDAR sensors are used for annotating data for both tasks in outdoor scenarios. Depth completion recovers dense depth maps from sparse measurements and a high-resolution image. Uhrig *et al.* [40] propose a sparsity-invariant convolution layer to consider the location of missing data while addressing data sparsity within deep networks. The work by Ma *et al.* [27] represents the first attempt to combine sparse depth and color images through an encoder-decoder CNN. Multi-stage network architectures [9, 20, 23, 34, 41] are effective tools to fuse the multi-modal color and depth data, while Spatial Propagation Networks (SPN) [3, 4, 26, 30] are popular approaches for iterative depth refinement. Graph representations have been used [2, 48, 54] for better modeling the relationships between sparse point clouds, while transformers [35, 52] have been deployed to model long-range relationships.

Self-supervised depth estimation. In recent years, self-supervised monocular depth estimation has gained significant attention, with two primary training methods being explored using either stereo images [12] or monocular videos [55]. Garg *et al.* [12] propose the first framework that uses an image reconstruction loss on stereo images to train a monocular depth model. In contrast, Zhou *et al.* [55] leverage a framework that jointly estimates depth and pose by utilizing video sequences and a photometric loss at training time. Subsequent works followed both paths [14, 15, 19, 21, 32, 33, 37, 53], significantly improving the accuracy of self-supervised solutions and shrinking the gap with supervised ones.

Self-supervised depth completion. The popularity of self-supervised methodologies for depth estimation has led to the emergence of several related studies in the depth completion literature. Some of them [6, 28, 46, 49] construct depth prediction network by minimizing the photometric error across monocular sequences, as well as minimizing the discrepancy between the sparse inputs and the dense outputs. Ma *et al.* [28] proposed a self-supervised training framework on sequences of color and sparse depth images with pose estimation using the PnP method. Choi *et al.* [6] designed a self-supervised network leveraging sparsity-invariant CNNs [40] to extract sparse depth features and pixel-adaptive convolutions to fuse image and depth features for challenging indoor environments.

Self-supervised depth estimation with few-beams LiDAR. A very recent trend consists of estimating dense depth from images and *few-beams* LiDAR sensors, e.g., 4-beams, in a self-supervised manner, reducing deployment costs at the minimum. We position this task at the intersection between self-supervised depth estimation and completion, given the minimal impact of the few LiDAR scans available with respect to the usual standard 64-beams setup for outdoor depth completion. LidarStereoNet [5] proposed a Lidar-stereo fusion network in an unsupervised learning scheme. Feng *et al.* [10] proposed a representative solution in this field using a two-stage network to infer dense depth maps. LidarTouch [1] explored self-supervised depth estimation with few LiDAR data in multiple depth completion networks and pose estimation methods.

Works approaching the last two tasks deploy complex architectures but often fail to account for outliers in the input raw LiDAR data. Purposely, we propose a lightweight network capable of masking out such outliers and delivering accurate depth maps.

3 Method

The proposed self-supervised framework aims at predicting a dense depth map $\hat{D} \in \mathbb{R}^{H \times W \times 3}$ from monocular image $I \in \mathbb{R}^{H \times W \times 3}$ and the sparse 4-beams LiDAR depth map $S \in \mathbb{R}^{H \times W}$, which is aligned with I. We formulate this task as a self-supervised learning problem, obtain-

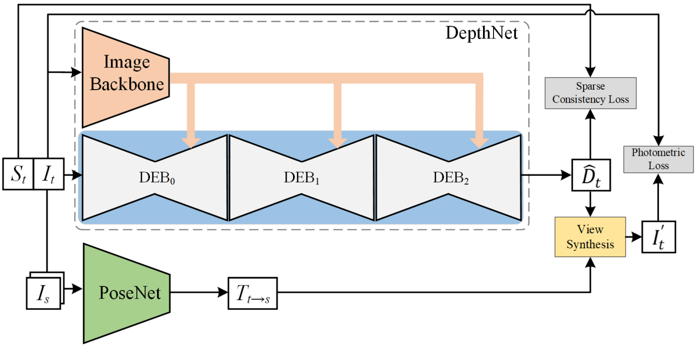


Figure 1: **Overview of our framework:** A DepthNet processes a single image and corresponding LiDAR data to predict a dense depth map. A PoseNet estimates the camera ego-motion from two images during training.

ing supervision from color images in a video and the very same input depth points. Figure 1 provides an overview of the architecture of our framework.

3.1 Self-Supervised Depth Estimation from few-beams LiDAR

We now introduce our framework for self-supervised completion, which consists of two main networks to predict depth and ego-motion [55].

Lightweight Multi-stage DepthNet: The DepthNet takes a single image I, and the corresponding sparse depth map S as inputs to progressively recover a dense depth map $\hat{D}$. It follows a multi-stage design, common in literature [9, 18, 20, 23, 34, 38], consisting of one image backbone and three cascade depth estimation networks. The former extracts multi-scale color features through convolutions and downsampling operators, encoding semantics and texture as guidance to recover dense depth. These features are then fed to the three cascade Depth Estimation Blocks (DEB), namely DEB_0, DEB_1, DEB_2, from left to right in Fig. 1, respectively. The three blocks are compact encoder-decoder networks, sharing the same architecture for the decoder, processing sparse depth points at a quarter (S_0), half (S_1), and full resolution (S_2) respectively – with S_0, S_1 being downsampled from S_2, i.e., S points on the image plane – as well as color features. Each encoder relies on Guided Sparse Convolutions – introduced in the remainder – having 32 output channels each. The decoders predict outputs at the exact resolution as the original input to the specific DEB block, processing image features from the image backbone. Residual connections [9, 17] integrate the results by the three after upsampling to full resolution.

PoseNet: Inferring camera ego-motion is essential for learning depth estimation from videos in a self-supervised manner. Thus, following [14], our PoseNet uses an ImageNet pre-trained ResNet18, taking two stacked color images as input to infer their 6-DoF relative pose. This network is needed at training time only.

3.2 Guided Sparsity-Invariant Convolution

Although most of the existing approaches [9, 18, 20, 23, 34, 38] rely on standard CNNs to extract sparse depth features through dedicated branches, Uhrig *et al.* [40] demonstrated that this approach is sub-optimal when dealing with highly-sparse data, and proposed Sparsity-

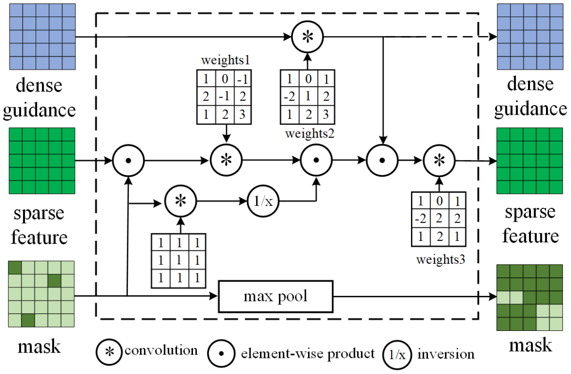


Figure 2: **Guided Sparsity-invariant CNN (GSCNN).**

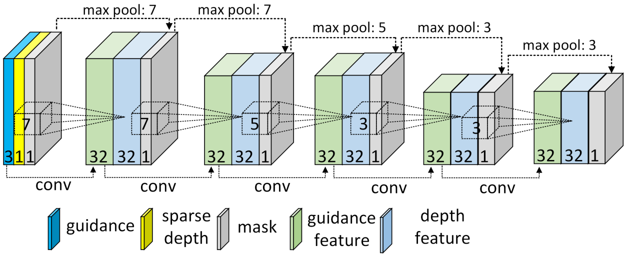


Figure 3: **DEB encoder.** Five GSCNN layers extract sparse depth features each with decreasing kernel sizes from 7×7 to 3×3.

invariant CNNs (SCNNs) to handle it better. However, SCNNs do not exploit any guide from color images usually coupled with the sparse data we aim at densifying, a powerful cue seldom ignored when available [9, 18, 20, 23, 34, 38]. To overcome this lack of the original SCNNs, we revise it to exploit additional dense guidance, as shown in Fig. 2.

Specifically, we propose Guided Sparsity-Invariant CNNs (GSCNNs) to overcome one main limitation of SCNNs, which struggle to recover sharp object boundaries due to the lack of awareness of semantic and dense structural cues that are available on RGB images instead. For this purpose, we introduce dense guidance d as an additional input to SCNNs, which will guide the propagation process of sparse data s within the network. The dense guidance can be color image, depth maps, or multi-channel depth features, and the input sparse feature can be LiDAR data or multi-channel depth features. A standard convolution operator processes these features, then multiplied to sparse features processed according to the standard SCNN design – i.e., a binary validity mask m is used to identify the meaningful features of the sparse data from those extracted out of invalid inputs – and then a final convolution produces the enhanced, sparse features output of the GSCNN layer.

This revised design keeps the merits of SCNNs to deal with sparse data s more effectively than CNNs while complementing its lack of semantic knowledge with the dense guide d. GSCNNs can be formalized as follows:

$$f_i(d,s,m) = \sum_{j\in\Omega(i)} w_j^3[(\sum_{j\in\Omega(i)} w_j^2 d_j)(\frac{\sum_{j\in\Omega(i)} m_j s_j w_j^1}{\sum_{j\in\Omega(i)} m_j + \varepsilon} + b)] \tag{1}$$

with $\Omega(i)$ being the convolution window centered in i, w^1, b weights and bias already present in standard SCNNs and w^2, w^3 the additional weights used to process d.

We use GSCNNs to extract sparse depth features in DEB encoders. For DEB_0, we stack five GSCNN layers, as shown in Fig. 3, using the color image as guidance alone. In DEB_1 and DEB_2, we use GSCNN layer in the first layer only, this time guided by dense output predicted by the previous DEB block, i.e. DEB_0 and DEB_1 for the two, respectively.

3.3 Loss function

Following the literature [6, 10, 45, 47], our model is trained using three loss terms that are optimized jointly:

$$L_{total} = \alpha L_{ph} + \beta L_{sm} + \gamma L_{sd} \tag{2}$$

with L_{ph}, L_{sm}, L_{sd} denoting the photometric consistency, smoothness, and sparse depth consistency losses, weighted by α, β and γ, respectively. Following [14], we compute these terms on intermediate depth predictions, i.e., on the output of each DEB block upsampled to the original input resolution.

Photometric Consistency Loss. Given the camera intrinsic matrix K, we synthesize the target image I_t' by warping the source image I_s according to the estimated depth and relative poses. As in [12, 14], we evaluate the pixel-level similarity between I_t' and the real target image I_t using a combination of an L1 pixel-wise loss term and the Structural Similarity (SSIM) [44] term:

$$L_{ph}(I_t, I_t') = \alpha \frac{1 - \text{SSIM}(I_t, I_t')}{2} + (1-\alpha) \parallel I_t - I_t' \parallel, \tag{3}$$

We adopt auto-masking [14] to filter out static pixels and the occluded region.

Smoothness Loss. We enforce a smoothness constraint on the dense depth maps by utilizing texture information from the input color image [14]:

$$L_{sm} = |\ \partial_x d^*\ | e^{-|\partial_x I_t|} + |\ \partial_y d^*\ | e^{-|\partial_y I_t|}, \tag{4}$$

with ∂_x, ∂_y being gradients along x and y direction, and $d^* = \hat{d_t}/\overline{\hat{d_t}}$ normalized inverse depth.

Sparse Depth Consistency Loss. We enforce consistency between densified and sparse depth using the scale-invariant [8] depth loss:

$$L_{si} = \frac{1}{2n^2} \sum_{i,j} \left((\log y_i - \log y_j) - (\log y_i^* - \log y_j^*) \right)^2 \tag{5}$$

with y and y^* being the predicted and input depth over the whole depth map space Ω, respectively, and n the number of pixels.

The raw sparse depth data contains outliers primarily due to the displacement between the LiDAR and the color camera. This misalignment causes the projection of some background points to overlap with foreground objects, as shown in Fig. 4. This fact would yield background points to emerge on the foreground objects in the predicted dense maps, causing inaccuracy near the depth discontinuities.

To avoid this behavior, we design a Distance-Dependent Outlier Mask M by setting a threshold σ on the discrepancy D_δ between prediction $\hat{D}$ and sparse depth measurements S. Such a threshold is dynamic, it varies in the different distance ranges over the sparse depth domain Ω, since a relatively more significant error is tolerable when predicting a farther depth value [11, 36].

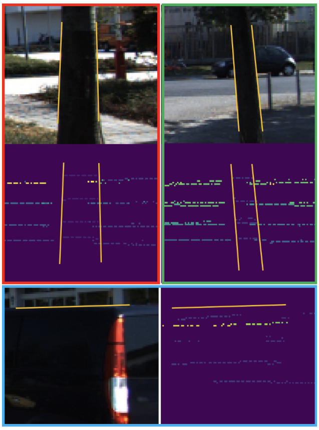


Figure 4: **Outliers on depth data.** Three images and corresponding LiDAR points, with overlapping background and foreground points.

To ease convergence, we empirically first set $\sigma = 4.0$ for the first 2 epochs:

$$M(x) = \begin{cases} 1 & \text{if } D_\delta(x) < \sigma \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Then, we set multiple thresholds σ_i according to different depth ranges:

$$M(x) = \begin{cases} 1 & \text{if } D_\delta(x) < \sigma_1, \forall\ \hat{D}(x) < 5 \\ 1 & \text{if } D_\delta(x) < \sigma_2, \forall\ \ 5 \leq \hat{D}(x) < 10 \\ 1 & \text{if } D_\delta(x) < \sigma_3, \forall\ 10 \leq \hat{D}(x) < 20 \\ 1 & \text{if } D_\delta(x) < \sigma_4, \forall\ 20 \leq \hat{D}(x) < 30 \\ 1 & \text{if } D_\delta(x) < \sigma_5, \forall\ 30 \leq \hat{D}(x) \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

with $\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5$ set to 0.2, 0.4, 0.8, 1.0, 2.0. We will analyze the effect of a fixed threshold σ over the whole depth range in the ablation study.

The overall, sparse depth consistency loss L_{sd} is then defined as:

$$L_{sd} = \omega \sum_{x \in \Omega} L_{si}(M(\hat{D}_0(x), S(x))) + \omega \sum_{x \in \Omega} L_{si}(M(\hat{D}_1(x), S(x))) + \sum_{x \in \Omega} L_{si}(M(\hat{D}_2(x), S(x))) \tag{8}$$

with $\hat{D}_0(x), \hat{D}_1(x), \hat{D}_2(x)$ being the predicted depth maps, S the sparse input depth, M an outlier mask used to ignore them – described in the remainder – and ω a hyper-parameter to control the impact of the loss on intermediate predictions. Specifically, we use a multi-stage training scheme by setting $\omega = 1$ for 10 epochs and then reducing it to 0.5 until convergence.

4 Experiments

We now introduce our experiments on two self-supervised tasks: 1) depth estimation from few-beam LiDAR and 2) depth completion.

Dataset. We focus on self-supervised depth estimation with LiDAR data in the outdoor environment. KITTI dataset [13] is popularly used in depth estimation. For what concerns the few-beams LiDAR setting, we follow [10] and evaluate our method on the Eigen split

	Method	Input	Train	Parameters	The lower the better				The higher the better		
					Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
(1)	Dorn [11]	M	M+Sup	99M	0.099	0.593	3.714	0.161	0.897	0.966	0.986
	BTS [22]	M	M+Sup	52M	0.091	0.555	4.033	0.174	0.904	0.967	0.984
(2)	MonoDepth2 [14]	M	S	14M	0.109	0.873	4.960	0.209	0.864	0.948	0.975
	MonoDepth2 [14]	M	M+S	14M	0.107	0.849	4.764	0.201	0.874	0.953	0.977
(3)	LEGO [50]	M	M	-	0.162	1.352	6.276	0.252	0.783	0.921	0.969
	PackNet-SfM [15]	M	M	>50M	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	MonoDepth2 [14]	M	M	14M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
(4)	Guizilini *et al.* [16]	M+L	M+L	>50M	0.082	0.424	3.73	**0.131**	0.917	-	-
	FusionDepth[10]	M+L	M+L	26M	0.078	0.515	3.67	0.154	0.935	0.973	0.986
	FusionDepth*[10]	M+L	M+L	26M	0.076	0.490	3.63	0.149	0.934	0.974	0.986
	FusionDepth (Refined Depth)[10]	M+L	M+L	>26M	0.074	**0.423**	3.61	0.150	0.936	0.973	0.986
	Ours	M+L	M+L	**628.53K**	**0.069**	0.476	**3.31**	0.144	**0.943**	**0.975**	**0.987**

Table 1: **Depth prediction on KITTI Eigen Split.** All methods process 640×192 images. *M*, *S*, and *L* respectively indicate Monocular, Stereo, and Sparse LiDAR data, with *Sup* referring to supervised training with accurate ground truth. Results for existing methods are directly taken from [10]. * means retrained by ourselves (with better results).

[7] of the KITTI original dataset[13] by uniformly sampling the sparse 4-beams data from original 64-beams LiDAR data [10, 51]. Regarding the standard depth completion setting – i.e., with 64-beams LiDAR – we test on the KITTI Depth Completion validation set [40].

Implementation Details. We use PyTorch [31] and train our model with a single NVIDIA RTX 3090 GPU, implemented starting from [10] code base. The sparse depth is normalized in the range $[0,1]$ before being processed by our model, which predicts multi-scale dense disparity maps, and then brings them back to the metric scale. All the parameters are optimized using Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The learning rate is initialized to 0.004 and multiplied by 0.5 every 8 epochs. We set the weight decay factor to 0.0002, and the network is trained for 40 epochs using a batch size of 20 samples with input images downsampled to 640 × 192.

4.1 Depth Estimation from few-beams LiDAR

Following [10], we compare our model with methods representative of four main categories: (1) supervised monocular networks [11, 22]; self-supervised monocular networks trained on (2) stereo pairs [14, 24, 39] or (3) monocular videos [14, 42, 55]; (4) self-supervised monocular methods from few-beams LiDAR[10, 16]. Table 1 collects the outcome of our experiments. In the case of FusionDepth, the authors employed GDC (post-processing) results as a form of supervision, leading to further improvements in performance. However, it is important to note that GDC requires additional parameters and computational resources. To ensure a fairer comparison, we have deliberately chosen to adopt the FusionDepth results without GDC and prioritize the evaluation of pure self-supervised models. Not surprisingly, methods processing even the few depth points from 4-beams LiDARs notably outperform the others. Among them, our model achieves the best results on the Eigen split, with extremely few parameters.

Moreover, we evaluate the accuracy achieved by our model on another *very-sparse* setting, i.e. by randomly sampling only a few hundred depth points from the sparse LiDAR [10, 25, 28]. Table 2 (a) collects the outcome of this experiment. Even in this very challenging scenario, our model yields results close to existing methods while being much more compact.

Methods	Params.	Samples	Abs. Rel.	RMSE
Sparse-to-dense [28]	26.1M	100	0.074	4.11
FusionDepth [10]	26M	100	0.074	**4.11**
Ours	**628.53K**	100	**0.072**	4.13
Liao *et al.* [25]	-	225	0.113	4.50
Sparse-to-dense [28]	26.1M	200	0.069	3.92
FusionDepth [10]	26M	200	0.069	**3.92**
Ours	**628.53K**	200	**0.066**	4.01

(a)

Methods	Params.	RMSE	iRMSE	iMAE
Sparse-to-dense [28]	26M	1342.33	4.28	1.64
DPP [49]	$\approx$ 18.8M	1310.03	-	-
VOICED [46]	$\approx$ 6.4M	1230.85	3.84	1.29
SelfDeco [6]	-	1212.89	3.54	1.29
FusionDepth [10]	26M	**1193.92**	3.39	**1.28**
Ours	**628.53K**	1234.75	**3.25**	1.29

(b)

Table 2: **Self-supervised depth estimation.** Experiments with (a) randomly sampled LiDAR and (b) 64-beams LiDAR.

# DEB blocks	GFLOPs	Params.	FPS	latency	Abs Rel
2	44.3	335K	160.16	0.006s	0.072
3	46.8	468K	123.79	0.008s	0.069
4	47.5	475K	100.97	0.010s	0.075

(a)

Layers	Params.	Abs Rel	Sq Rel	RMSE
CNN	363.971K	0.072	0.463	3.450
SCNN	446.611K	0.070	**0.450**	3.466
GSCNN	628.531K	**0.069**	0.476	**3.312**

(b)

Methods	Abs Rel	Sq Rel	RMSE
w/o mask	0.071	0.504	3.492
$\sigma = 4.0$	0.070	0.489	3.429
$\sigma = 0.2$	0.070	0.538	3.434
confidence	0.071	**0.465**	3.472
ours	**0.069**	0.476	**3.312**

(c)

Table 3: **Ablation studies.** (a) Comparison between models deploying different numbers of DEBs. (b) Comparison between CNNs, SCNNs[40] and GSCNNs. (c) Comparison with different masking techniques.

4.2 Depth Completion

Finally, we also evaluate our method when processing denser depth maps provided by a more expensive HDL-64 LiDAR sensor – the one used by the standard KITTI depth completion dataset, yet training in a self-supervised manner. The performance of our model and existing self-supervised solutions on the KITTI completion validation set are reported in Table. 2 (b). Despite the much fewer parameters, our lightweight network achieves results comparable with those yielded by state-of-the-art models.

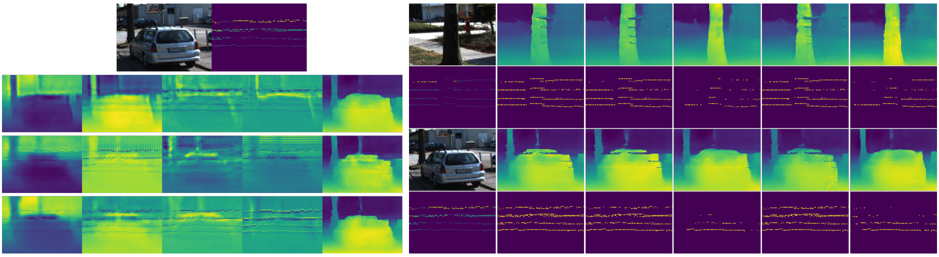
4.3 Ablation Study

We conclude with ablation studies to assess the effectiveness of the proposed modules, DEBs, GSCNNs, and the outlier mask. All experiments are conducted with the few-beams LiDAR setting on the Eigen split.

Cascade Depth Estimation Blocks. Our network progressively recovers dense depth map block by block, with each DEB predicting a dense depth map. To better study the effectiveness of this module, we conducted a study on the impact of the number of blocks regarding performance, speed, and computation cost, as presented in Table 3 (a). All the results indicate that the models run on a single RTX 3090, with an input resolution of 1216×352. Through experimentation with the number of DEBs ranging from 2 to 4, we observed that increasing the number of blocks leads to higher computation costs without consistent performance improvement beyond three blocks. Consequently, an architecture comprising three cascade DEBs is the best suited for our purposes.

Guided Sparsity-Invariant Convolution. To validate the effectiveness of GSCNN, we compared the performance of three different variants of our framework, obtained by using the proposed GSCNNs, the original SCNNs, or the standard CNNs to build the DEB encoders. From Table 3 (b), we can observe that using GSCNNs yields better results on two out of three metrics compared to alternative methods, in particular in terms of RMSE, with only a limited increase in the number of parameters.

To further validate this finding, we visualize the feature maps extracted from images and sparse data in the last block and the predicted results by the three methods in Fig. 5 (a).



(a) (b)

Figure 5: **Ablation studies – qualitative results.** (a) comparison between outputs by using, from top to bottom, by GSCNNs, SCNNs, and CNNs. The top two inputs consist of image and LiDAR data. The first four columns represent the extracted feature maps, while the remaining columns display the predicted results. (b) depth maps predicted, from left to right, without using any outlier mask, a fixed $\sigma = 4.0$ or 0.2, confidence predicted over the input depth or our strategy.

Using GSCNNs (2nd row) allows for extracting much more detailed features, already allowing for distinguishing foreground objects from the background and prediction with clearer boundaries and details. In contrast, SCNNs and CNNs (3rd and 4th rows) extract features exposing grid artifacts and scarce semantic information. Based on the above analysis, GSCNN demonstrates superiority over SCNN and CNN in both quantitative and qualitative comparisons, despite having limited additional parameters.

Distance-Dependent Outlier Mask. To validate the effectiveness of the proposed outlier mask, we compare its performance with alternative approaches [29, 43] – by adding a binary confidence layer in our model to identify the outlier in the sparse input – as well as to the use of a fixed threshold σ for any depth range. Results are reported in Table 3 (c). From it, we can notice that our strategy is the only one yielding consistent improvements on any metric. Figure 5 (b) shows a qualitative comparison between the dense depth maps predicted according to the different strategies. The absence of any outlier mask (2nd column) produces holes in the foreground objects, like using a fixed threshold $\sigma = 4.0$ (3rd column). A stricter threshold equal to 0.2 (4th column) can alleviate this behaviour, yet without significant improvements on the final accuracy according to Table 3 (b), while using confidence still cannot prevent holes from appearing in the densified maps (5th column). Our strategy (rightmost column) can remove holes and improve results quantitatively.

5 Conclusion

We have proposed a lightweight architecture for self-supervised depth estimation from sparse depth points and color images. Thanks to the revised Guided Sparsity-Invariant CNNs design, our model can accomplish accurate predictions without the need for over-parametrized layers. Moreover, the proposed Distance-Dependent Outlier Mask prevents outliers in the sparse data from irremediably damaging the predicted dense depth map. Experimental results with multiple settings, i.e. 4-beams, 64-beams, and a few hundred depth points, LiDAR data show that our model yields state-of-the-art accuracy with a minimal fraction of the parameters used by existing frameworks.

Acknowledgment. We sincerely thank the scholarship supported by China Scholarship Council (CSC).

References

[1] Florent Bartoccioni, Éloi Zablocki, Patrick Pérez, Matthieu Cord, and Karteek Alahari. Lidartouch: Monocular metric depth estimation with a few-beam lidar. *Computer Vision and Image Understanding*, 227:103601, 2023.

[2] Hu Chen, Hongyu Yang, and Yi Zhang. Depth completion using geometry-aware embedding. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8680–8686. IEEE, 2022.

[3] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018.

[4] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, AAAI 2020 - 34th AAAI Conference on Artificial Intelligence, pages 10615–10622, 2020.

[5] Xuelian Cheng, Yiran Zhong, Yuchao Dai, Pan Ji, and Hongdong Li. Noise-aware unsupervised deep lidar-stereo fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6339–6348, 2019.

[6] Jaehoon Choi, Dongki Jung, Yonghan Lee, Deokhwa Kim, Dinesh Manocha, and Donghwan Lee. Selfdeco: Self-supervised monocular depth completion in challenging indoor environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 467–474. IEEE, 2021.

[7] David Eigen and Rob 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*, pages 2650–2658, 2015.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

[9] Rizhao Fan, Zhigen Li, Matteo Poggi, and Stefano 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.

[10] Ziyue Feng, Longlong Jing, Peng Yin, Yingli Tian, and Bing Li. Advancing self-supervised monocular depth learning with sparse lidar. In *Conference on Robot Learning*, pages 685–694. PMLR, 2022.

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[12] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[14] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[15] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2485–2494, 2020.

[16] Vitor Guizilini, Jie Li, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. Robust semi-supervised monocular depth estimation with reprojected distances. In *Conference on robot learning*, pages 503–512. PMLR, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13656–13662. IEEE, 2021.

[19] Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12642–12652, 2021.

[20] Yanjie Ke, Kun Li, Wei Yang, Zhenbo Xu, Dayang Hao, Liusheng Huang, and Gang Wang. Mdanet: Multi-modal deep aggregation network for depth completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4288–4294. IEEE, 2021.

[21] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision*, pages 582–600. Springer, 2020.

[22] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[23] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. A multi-scale guided cascade hourglass network for depth completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 32–40, 2020.

[24] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[25] Yiyi Liao, Lichao Huang, Yue Wang, Sarath Kodagoda, Yinan Yu, and Yong Liu. Parse geometry from a line: Monocular depth estimation with partial laser observation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 5059–5066. IEEE, 2017.

[26] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. *arXiv preprint arXiv:2202.09769*, 2022.

[27] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018.

[28] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019.

[29] Robert McCraith, Eldar Insafutdinov, Lukas Neumann, and Andrea Vedaldi. Lifting 2d object locations to 3d by discounting lidar outliers across objects and views. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2411–2418. IEEE, 2022.

[30] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision*, pages 120–136. Springer, 2020.

[31] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017.

[32] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5848–5854. IEEE, 2018.

[33] Matteo Poggi, Fabio Tosi, Filippo Aleotti, and Stefano Mattoccia. Real-time self-supervised monocular depth estimation without gpu. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2022. doi: 10.1109/TITS.2022.3157265.

[34] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019.

[35] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *CVPR*, pages 6250–6259, 2022.

[36] Suman Saha, Anton Obukhov, Danda Pani Paudel, Menelaos Kanakis, Yuhua Chen, Stamatios Georgoulis, and Luc Van Gool. Learning to relate depth and semantics for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8197–8207, 2021.

[37] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision*, pages 572–588. Springer, 2020.

[38] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30: 1116–1129, 2020.

[39] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9809, 2019.

[40] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

[41] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th international conference on machine vision applications (MVA)*, pages 1–6. IEEE, 2019.

[42] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2022–2030, 2018.

[43] Yufei Wang, Yuchao Dai, Qi Liu, Peng Yang, Jiadai Sun, and Bo Li. Cu-net: Lidar depth-only completion with coupled u-net. *IEEE Robotics and Automation Letters*, 7 (4):11476–11483, 2022.

[44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[45] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12747–12756, 2021.

[46] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5 (2):1899–1906, 2020.

[47] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2):1495–1502, 2021.

[48] Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In *European Conference on Computer Vision*, pages 682–699. Springer, 2020.

[49] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3353–3362, 2019.

[50] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 225–234, 2018.

[51] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019.

[52] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. CVPR.

[53] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *International Conference on 3D Vision*, 2022.

[54] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30:5264–5276, 2021.

[55] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.