SQLdepth: Generalizable Self-Supervised Fine-Structured Monocular Depth Estimation

Youhong Wang[1, 2], **Yunji Liang**[1*], **Hao Xu**[2], **Shaohui Jiao**[2], **Hongkai Yu**[3]

[1]Northwestern Polytechnical University
[2]Bytedance Inc
[3]Cleveland State University

Abstract

Recently, self-supervised monocular depth estimation has gained popularity with numerous applications in autonomous driving and robotics. However, existing solutions primarily seek to estimate depth from immediate visual features, and struggle to recover fine-grained scene details. In this paper, we introduce SQLdepth, a novel approach that can effectively learn fine-grained scene structure priors from ego-motion. In SQLdepth, we propose a novel Self Query Layer (SQL) to build a self-cost volume and infer depth from it, rather than inferring depth from feature maps. We show that, the self-cost volume is an effective inductive bias for geometry learning, which implicitly models the single-frame scene geometry, with each slice of it indicating a relative distance map between points and objects in a latent space. Experimental results on KITTI and Cityscapes show that our method attains remarkable state-of-the-art performance, and showcases computational efficiency, reduced training complexity, and the ability to recover fine-grained scene details. Moreover, the self-matching-oriented relative distance querying in SQL improves the robustness and zero-shot generalization capability of SQLdepth. Code is available at https://github.com/hisfog/SfMNeXt-Impl.

1 Introduction

As one of the fundamental research topics in computer vision, monocular depth estimation (MDE) aims to predict the corresponding depth of each pixel within a single image, and is widely used in numerous applications, such as autonomous driving, 3D reconstruction, augmented reality and robotics. For the supervised solutions, depth sensors, such as LiDAR and Kinect camera are used to collect the depth measurements as ground truth. However, it is time-consuming and expensive to collect large-scale depth measurements from physical world. In addition, supervised depth estimation cannot be well-optimized under the sparse supervision and has limited generalization to unseen scenarios.

Lately, self-supervised solutions have gained popularity. Existing efforts have concentrated on training with self-distillation (Peng et al. 2021), leveraging depth hints (Watson et al. 2019), and inferring with multi-frames (Watson et al. 2021; Feng et al. 2022). However, they often fail to recover fine-grained scene details, as shown in Figure 1. How to learn the fine-grained scene structure priors effectively and efficiently in self-supervised setting is still challenging.

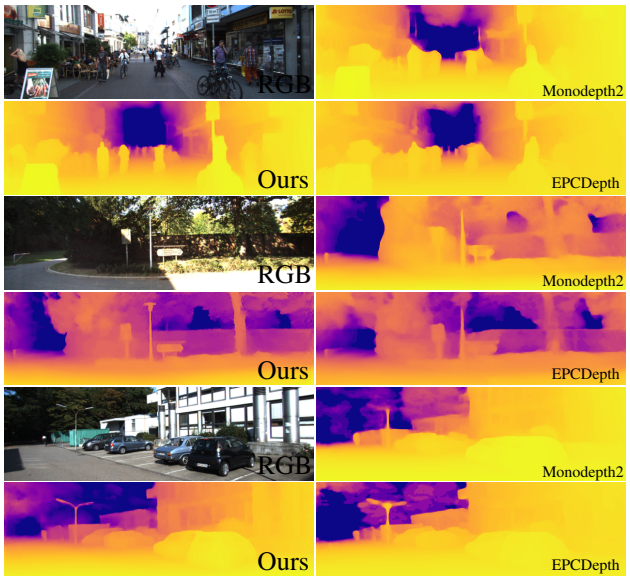

Figure 1: Visualized comparison on images from KITTI dataset (Geiger et al. 2013). Existing self-supervised solutions such as Monodepth2 (Godard et al. 2019) and EPCDepth (Peng et al. 2021) show degraded performance under the conditions of illumination change, or occlusion, and fail to estimate the depth of thin or small objects.

We observe that the depth of a pixel is strongly correlated with the depth of its adjacent pixels and related objects within the image. This suggests that a pixel's depth can be inferred from related contexts, which provide relative distance information. Therefore, in this paper, we aim to explore whether *relative distance* information can serve as an effective inductive bias, thereby improving the self-supervised learning of monocular depth estimation.

To achieve this, we propose to build a novel self-cost volume using a Self Query Layer (SQL), with the aim of accurately modeling the underlying geometry of the monocular scene. Specifically, we first model points and objects

*Corresponding Author.

in a latent space. For each point (pixel), we employ a fully convolutional U-Net with skip-connections to extract visual features with high-frequency details. For objects, we utilize a compact Vision Transformer (ViT) with large patch size to extract coarse-grained object queries. Secondly, within a novel Self Query Layer (SQL), we perform dot-product to compare each pixel with each object to build the self-cost volume. Finally, we propose a novel and effective decoding approach specifically tailored for compressing the self-cost volume to the final depth map.

Our discovery shows that, compared to directly estimating depth from visual feature maps, depth from a geometric self-cost volume is significantly more effective and robust. Visualization results further show that each slice in the self-cost volume can serve as a relative distance map, accurately modeling the scene geometry. Our main contributions are:

- Introducing SQLdepth, a novel self-supervised method empowered by the Self Query Layer (SQL) to construct a self-cost volume that effectively captures fine-grained scene geometry of a single image.
- Demonstrating through comprehensive experiments on KITTI and Cityscapes datasets that SQLdepth is simple yet effective, and surpasses existing self-supervised alternatives in accuracy and efficiency.
- Highlighting SQLdepth's improved generalization. This is demonstrated by applying a KITTI pre-trained model to other datasets, such as zero-shot transfer to Make3D.

2 Related Work

2.1 Supervised Depth Estimation

Eigen *et al.* (Eigen, Puhrsch, and Fergus 2014) was the first to propose a framework consisting of a multiscale convolutional neural network to directly predict depth from an RGB image under the supervision of a scale-invariant loss function. Since then, numerous solutions have been proposed. Generally, those methods formulated the depth estimation task as either a per-pixel regression problem (Eigen, Puhrsch, and Fergus 2014; Huynh et al. 2020; Ranftl, Bochkovskiy, and Koltun 2021), or a per-pixel classification problem (Fu et al. 2018; Diaz and Marathe 2019).

The regression based methods can predict continuous depths, but are hard to optimize. The classification based methods can only predict discrete depths, but are easier to optimize. To combine the strengths of regression and classification tasks, studies in (Bhat, Alhashim, and Wonka 2021; Johnston and Carneiro 2020) reformulated depth estimation as a per-pixel classification-regression task. In this formulation, they proposed to first regress a set of depth bins and then perform per-pixel classification to assign each pixel to the depth bins. The ultimate depth was derived through a linear combination of bin centers, weighted by the probability logits. This approach has attained remarkable improvement in precision.

2.2 Self-supervised Depth Estimation

In the absence of ground truth, self-supervised methods are usually trained by either making use of the temporal scene consistency in monocular videos (Zhou et al. 2017a; Godard et al. 2019), or left-right scene consistency in stereo image pairs (Godard, Mac Aodha, and Brostow 2017).

Monocular Training. In monocular training, supervision information comes from the consistency between the synthesis scene view from referenced frame and the scene view of source frame. SfMLearner (Zhou et al. 2017a) jointly trained a DepthNet and a separate PoseNet under the supervision of a photometric loss. Following this classical joint-training pipeline, many advances were proposed to improve the learning process, e.g. more robust image reconstruction loss (Gordon et al. 2019), feature-metric reconstruction loss (Shu et al. 2020; Zhan et al. 2018), leveraging auxiliary information (Watson et al. 2019; Klodt and Vedaldi 2018), dealing with moving objects that break the static scene assumption (Feng et al. 2022; Gordon et al. 2019; Bian et al. 2019; Klingner et al. 2020; Li et al. 2020), and introducing extra constraints, such as scene flow (Jiang and Okutomi 2023), optical flow (Ranjan et al. 2019), semantics (Guizilini et al. 2020b; Chen et al. 2019), epipolar constraints (Chen, Schmid, and Sminchisescu 2019). Recently, lightweight neural architectures (Zhang et al. 2023) and orthogonal planes (Wang, Yu, and Gao 2023) were proposed.

Stereo Training. Stereo training requires stereo image pairs or synchronized stereo videos. In the setting of stereo training, the relative camera pose is known, and we only need to predict the disparity (or depth) map. Garg *et al.* (Garg et al. 2016) was the first to introduce photometric consistency loss between stereo pairs for self-supervised stereo training. Following this, a series of improvements were proposed, including left-right consistency (Godard, Mac Aodha, and Brostow 2017), and integrating monocular depth estimation via siamese neural architecture (Zhou and Dong 2023), predicting continuous disparity values (Garg et al. 2020). Stereo training has been further extended with semi-supervised data (Kuznietsov, Stuckler, and Leibe 2017; Luo et al. 2018), auxiliary information (Watson et al. 2019), and self-distillation training (Peng et al. 2021; Guo et al. 2018; Pilzer et al. 2019). Generally, stereo views are unaffected by moving objects, therefore, they can provide accurate supervision and be utilized to obtain the absolute depth scale.

However, existing methods often fall short in recovering scene details, and struggle to effectively learn fine-grained scene structure priors. These approaches typically either estimate depth from immediate visual feature maps or from Transformer-enhanced high-level visual representations (Zhao et al. 2022). They overlook the importance of pixel-level geometric cues that might be beneficial to model performance as well as generalization capability. This is attributed to the intrinsic geometry understanding nature of monocular depth estimation.

3 Problem Setup

The goal of self-supervised monocular depth estimation is to predict the depth map from a single RGB image without ground truth, which can also be viewed as learning structure from motion (SfM). As illustrated in Figure 2, given a single source image I_t as input, first, the DepthNet predicts its corresponding depth map D_t. And the PoseNet takes both

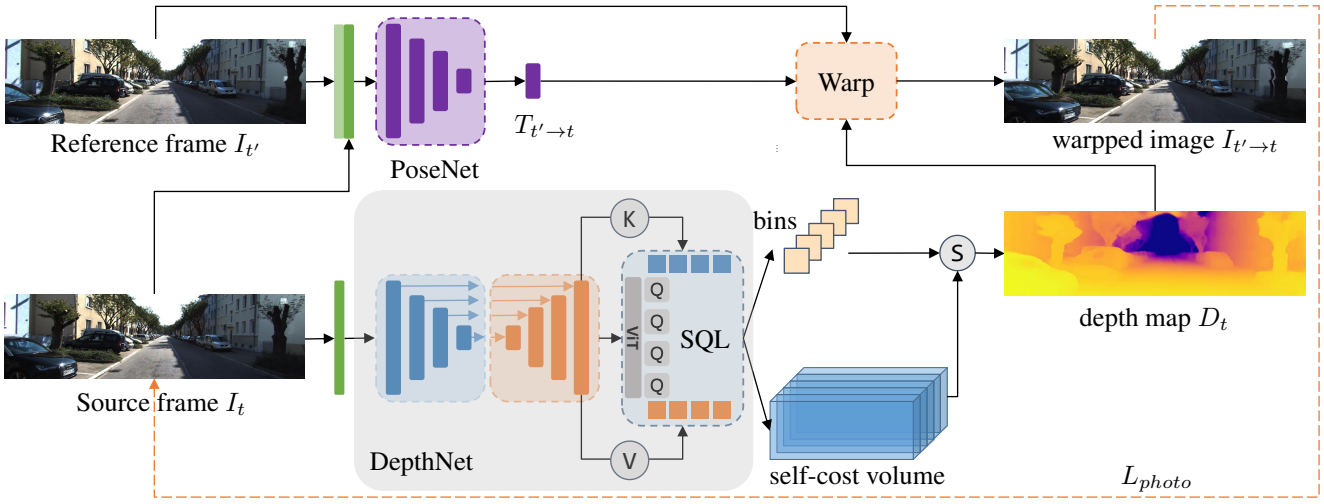

Figure 2: Framework Overview: (1) DepthNet: A fully convolutional encoder-decoder is used to extract immediate visual features of frame I_t. Then these visual features are passed into a Self Query Layer (See Figure 3 for more details) to obtain the depth map D_t of current frame. (2) PoseNet: Relative pose between the current frame I_t and the reference frame $I_{t'}$ is predicted with a PoseNet. This relative pose is only needed for training. (3) Differentiable warpping: Following many previous works, we perform differentiable warpping (Jaderberg et al. 2015) to reconstruct frame I_t, using D_t, $T_{t'\to t}$ and $I_{t'}$. The loss function is built upon the differences of the warpped image $I_{t'\to t}$ and current frame I_t.

source image and reference image (I_t, $I_{t'}$) as input and predicts the relative pose $T_{t\to t'}$ between the source image I_t and reference image $I_{t'}$. Finally, the predicted depth D_t and relative pose $T_{t\to t'}$ are used to perform view synthesis by Eq. 1.

$$I_{t'\to t} = I_{t'} \left\langle \text{proj}\left(D_t, T_{t\to t'}, K\right)\right\rangle \tag{1}$$

where $\langle\rangle$ is the sampling operator and proj returns the 2D coordinates of the depths in D_t when reprojected into the camera view of $I_{t'}$. At training time, both the DepthNet and PoseNet are optimized jointly by minimizing the photometric reconstruction loss, which is widely used in prior works (Garg et al. 2016; Zhou et al. 2017a,b). For each pixel, we optimize the photometric loss L_{photo} for the best matching across reference views, by selecting the per-pixel minimum over the photometric error pe, as defined in Eq. 2, where $t' \in \{t-1, t+1\}$. More details about pe are provided in Section 4.3.

$$L_{photo} = \min_{t'} pe\left(I_t, I_{t'\to t}\right) \tag{2}$$

4 Method

In this section, we elaborate the design details of the two core components of SQLdepth: the fully convolutional U-Net for extracting immediate visual representations, and the Self Query Layer (SQL) for effective depth estimation.

4.1 Extract Immediate Visual Representations

Given an input RGB image of shape $\mathbb{R}^{3\times H\times W}$, the convolutional U-Net first extracts image features and decodes with upsampling them into high resolution immediate features $\mathbf{S}$ of shape $\mathbb{R}^{C\times h\times w}$. Considering of the hardware limitation,

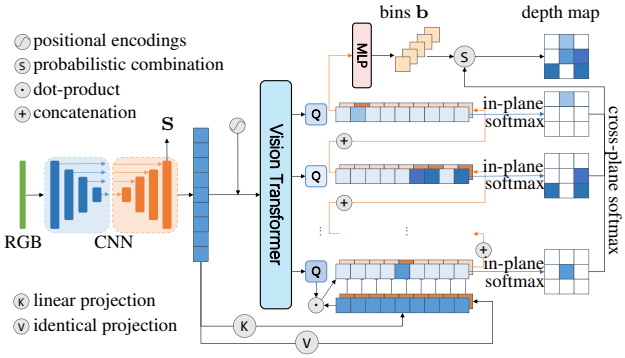

Figure 3: Details of DepthNet with the Self Query Layer.

we set $h = \frac{H}{2}$ and $w = \frac{W}{2}$. Benefiting from the encoder-decoder architecture with skip connections, we can extract visual representations with fine-grained visual cues.

4.2 Self Query Layer

Building a self-cost volume V. Exploring geometric cues, such as relative distance cues, is a key factor in achieving accurate monocular depth estimation. But how to obtain the relative distance remains unclear. Drawing inspiration from prior works, where a cost volume is built upon two different images to capture cross image geometric cues for other geometric tasks (e.g. MVS (Watson et al. 2021) and optical flow (Teed and Deng 2020)), we can build a cost volume upon an image and itself, and call it as a self-cost volume, to capture the relative distance between points and points. However, the time complexity of this procedure is $O(h^2 \times w^2)$, which makes it infeasible to build the self-cost volume di-

rectly upon the high resolution feature map $\mathbf{S}$.

Coarse-grained queries Q. Instead of calculating the relative distance between points and points, we show that it is more advantageous and computationally efficient to compute it between points and objects. To achieve this, we introduce the coarse-grained queries to represent objects in the image. We split the feature map $\mathbf{S}$ into large patches, and utilize a transformer to enhance the patch embeddings, enabling them to represent objects in the image implicitly. Afterwards, we use these coarse-grained object queries to perform per-pixel relative distance querying (dot product) to get the relative distance representations. Specifically, we start by applying a convolution with a $p \times p$ kernel and stride of p (e.g., $p = 16$) to $\mathbf{S}$, yielding a feature map $\mathbf{F}$ of shape $C \times \frac{h}{p} \times \frac{w}{p}$. Next, we reshape $\mathbf{F}$ to $\mathbb{R}^{C \times N}$ and add positional embeddings, where $N = \frac{h*w}{p^2}$ denotes the number of patches. Subsequently, these patch embeddings are input into a compact transformer comprising 4 layers, producing a set of coarse-grained queries $\mathbf{Q}$ of shape $\mathbb{R}^{C \times Q}$, where Q is a hyperparameter and $Q \leq N$. Finally, these coarse-grained queries are applied to the per-pixel immediate visual representations in $\mathbf{S}$ to build the self-cost volume $\mathbf{V}$ of shape $h \times w \times Q$. where $V_{i,j,k}$ is calculated as Eq. 3.

$$V_{i,j,k} = Q_i^T \cdot S_{j,k} \tag{3}$$

Depth bins estimation with the self-cost volume. Prior works (Bhat, Alhashim, and Wonka 2021; Li et al. 2022) show that depth bins are useful for estimating continuous depth. Bhat *et. al* (Bhat, Alhashim, and Wonka 2021) employed a brute force regression method to calculate depth bins from a token extracted by a Vision Transformer (Dosovitskiy et al. 2020), and utilized a chamfer loss as supervision. However, this brute force approach fails in self-supervised setting (see ablation in Table 6). Therefore, we rethink the essence of depth bins: Depth bins essentially represent the distribution of depth, that is, the *countings* of different depth values. Therefore, we propose to estimate depth bins $\mathbf{b}$ by counting the latent depths in the self-cost volume. Since we perform in a latent depth space, we can view the counting process as a information aggregation process, and use two basic operations for information aggregation: softmax and weighted sum to achieve the counting operation. Specifically, for every plane (slice) in the self-cost volume $\mathbf{V}$, we first apply a pixel-wise softmax to convert the volume-plane into a pixel-wise probabilistic map. Then we perform a weighted sum of per-pixel visual representations in $\mathbf{S}$ using this map. After this procedure, we get Q vectors of dimension C, representing Q depth countings in Q planes. Finally we concat them and feed it into a MLP to regress the depth bins $\mathbf{b}$ as shown in Eq. 4.

$$\mathbf{b} = MLP\left(\bigoplus_{i=1}^{Q} \sum_{(j,k)=(1,1)}^{(h,w)} softmax(V_i)_{j,k} \cdot S_{j,k}\right) \tag{4}$$

Probabilistic combination. We compress the self-cost volume $\mathbf{V}$ to get the final depth map, using the depth bins $\mathbf{b}$ we extracted. Firstly, in order to match the dimension of depth bins $\mathbf{b}$ of shape D, we apply a 1×1 convolution to the self-cost volume $\mathbf{V}$ to obtain a D-planes volume. Secondly, we apply a plane-wise softmax operation to convert the volume into plane-wise probabilistic maps as shown in Eq. 5.

$$p_{i,j,k} = softmax(V)_{i,j,k}, 1 \leq i \leq Q \tag{5}$$

Finally, for each pixel, the depth is calculated by aggregating the centers of bins using a probabilistic linear combination (Bhat, Alhashim, and Wonka 2021) as shown in Eq. 6:

$$\tilde{d} = \sum_{i=1}^{N} c\left(b_i\right) p_{i,j,k}, 1 \leq j \leq h, 1 \leq k \leq w \tag{6}$$

where $c\left(b_i\right)$ is the center depth of the i^{th} bin and it is determined by Eq 7.

$$c\left(b_i\right) = d_{\min} + \left(d_{\max} - d_{\min}\right)\left(b_i/2 + \sum_{j=1}^{i-1} b_j\right) \tag{7}$$

4.3 Loss Functions

Objective Functions. Following (Godard, Mac Aodha, and Brostow 2017; Godard et al. 2019), we use the standard photometric error pe combined by the L1 and SSIM (Wang et al. 2004) as the main objective, as denoted in Eq. 8.

$$pe\left(I_a, I_b\right) = \frac{\alpha}{2}\left(1 - \text{SSIM}\left(I_a, I_b\right)\right) + (1 - \alpha)\left\|I_a - I_b\right\|_1 \tag{8}$$

In order to regularize the depths in textureless regions, we use edge-aware smooth loss as regularization, as follows:

$$L_s = \left|\partial_x d_t^*\right| e^{-\left|\partial_x I_t\right|} + \left|\partial_y d_t^*\right| e^{-\left|\partial_y I_t\right|} \tag{9}$$

Masking Strategy. In real-world scenarios, a stationary camera or moving objects will break down the assumptions of a moving camera and a static scene, and hurt the training process of self-supervised depth estimation. To address this issue, several prior studies integrated motion mask to deal with moving objects with the help of a scene specific instance segmentation model (especially in Cityscapes). This approach, however, is not applicable to all scenarios. To keep scalable, we do not use motion masks to deal with moving objects. We only use auto-masking strategy in (Godard et al. 2019) to filter out stationary pixels and low-texture regions that remain with the same appearance between two frames in a sequence. The mask μ is computed in Eq. 10, where [] is the Iverson bracket.

$$\mu = \left[\min_{t'} pe\left(I_t, I_{t' \to t}\right) < \min_{t'} pe\left(I_t, I_{t'}\right)\right] \tag{10}$$

Final Training Loss. We combine the edge-aware smooth loss, photometric loss in Eq. 2 and auto-mask μ as the final training loss, as shown in Eq. 11.

$$L = \mu L_{photo} + \lambda L_s \tag{11}$$

5 Experiments

We evaluate SQLdepth on three datasets including KITTI, Cityscapes and Make3D, and quantify the model performance in terms of 7 widely used metrics from (Eigen and Fergus 2015). In addition, we investigate the generalization of our model by zero-shot evaluating or fine-tuning on unseen datasets with KITTI-pretrained weights.

	Method	Train	Test	AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
640×192	PackNet-SfM (Guizilini et al. 2020a)	M	1	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	HR-Depth (Lyu et al. 2021)	MS	1	0.107	0.785	4.612	0.185	0.887	0.962	0.982
	Monodepth2 (*34M*) (Godard et al. 2019)	MS	1	0.106	0.818	4.750	0.196	0.874	0.957	0.979
	DynamicDepth (Feng et al. 2022)	M	2	0.096	0.720	4.458	0.175	0.897	0.964	**0.984**
	ManyDepth (MR, *36M*) (Watson et al. 2021)	M	2*	0.090	0.713	4.261	0.170	0.914	0.966	0.983
	SQLdepth (Efficient-b5, *34M*)	M	1	0.094	**0.697**	4.320	0.172	0.904	0.967	**0.984**
	SQLdepth (ResNet-50, *31M*)	M	1	0.091	0.713	4.204	0.169	0.914	0.968	**0.984**
	SQLdepth (ResNet-50, *31M*)	MS	1	**0.088**	**0.697**	**4.175**	**0.167**	**0.919**	**0.969**	**0.984**
1024×320	Monodepth2 (*34M*) (Godard et al. 2019)	MS	1	0.106	0.806	4.630	0.193	0.876	0.958	0.980
	HR-Depth (Lyu et al. 2021)	MS	1	0.101	0.716	4.395	0.179	0.899	0.966	0.983
	DIFFNet (Zhou, Greenwood, and Taylor 2021)	M	1	0.097	0.722	4.345	0.174	0.907	0.967	0.984
	Depth Hints (Watson et al. 2019)	S	1	0.096	0.710	4.393	0.185	0.890	0.962	0.981
	CADepth-Net (Yan et al. 2021)	MS	1	0.096	0.694	4.264	0.173	0.908	0.968	0.984
	EPCDepth (ResNet-50) (Peng et al. 2021)	S+D	1	0.091	0.646	4.207	0.176	0.901	0.966	0.983
	ManyDepth (Res-50, *37M*) (Watson et al. 2021)	M	2*	0.087	0.685	4.142	0.167	0.920	0.968	0.983
	SQLdepth (Efficient-b5, *37M*)	M	1	0.087	0.649	4.149	0.165	0.918	0.969	0.984
	SQLdepth (ResNet-50, *37M*)	M	1	0.087	0.659	4.096	0.165	0.920	0.970	0.984
	SQLdepth (ResNet-50, *37M*)	MS	1	**0.082**	**0.607**	**3.914**	**0.160**	**0.928**	**0.972**	**0.985**

Table 1: Performance comparison on KITTI eigen benchmark (Geiger et al. 2013). In the *Train* column, S: stereo training, M: monocular training, MS: stereo and monocular training, D: self-distillation training, In the *Test* column, 1: one single frame as input, 2: two frames (the previous and current) as input, *: TTR (Test-Time Refinement) used by ManyDepth (Watson et al. 2021). The best results are in bold, and second best are underlined. The #param is highlighted in *italic*. All self-supervised methods use median-scaling in (Eigen and Fergus 2015) to estimate the absolute depth scale.

	Method	Train	Test	AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
640×192	Monodepth2 (*34M*) (Godard et al. 2019)	MS	1	0.080	0.466	3.681	0.127	0.926	0.985	0.995
	DynamicDepth (Feng et al. 2022)	M	2	0.068	0.362	3.454	0.111	0.943	0.991	**0.998**
	ManyDepth (MR, *36M*) (Watson et al. 2021)	M	2*	0.058	0.334	3.137	0.101	0.958	0.991	0.997
	SQLdepth (Efficient-b5, *34M*)	M	1	0.066	0.356	3.344	0.107	0.947	0.989	0.997
	SQLdepth (ResNet-50, *31M*)	M	1	0.061	0.317	3.055	0.100	0.957	0.992	0.997
	SQLdepth (ResNet-50, *31M*)	MS	1	**0.054**	**0.276**	**2.819**	**0.092**	**0.964**	**0.993**	**0.998**
1024×320	Monodepth2 (*34M*) (Godard et al. 2019)	MS	1	0.091	0.531	3.742	0.135	0.916	0.984	0.995
	CADepth-Net (Yan et al. 2021)	MS	1	0.070	0.346	3.168	0.109	0.945	0.991	0.997
	ManyDepth (Res-50, *37M*) (Watson et al. 2021)	M	2*	0.055	0.305	2.945	0.094	0.963	0.992	0.997
	SQLdepth (Efficient-b5, *37M*)	M	1	0.058	0.287	3.039	0.096	0.959	0.992	**0.998**
	SQLdepth (ResNet-50, *37M*)	M	1	0.058	0.289	2.925	0.095	0.962	0.993	**0.998**
	SQLdepth (ResNet-50, *37M*)	MS	1	**0.052**	**0.223**	**2.550**	**0.084**	**0.971**	**0.995**	**0.998**

Table 2: Performance comparison using KITTI improved ground truth from (Uhrig et al. 2017).

5.1 Datasets and Experimental Protocol

KITTI (Geiger et al. 2013) is a dataset that provides stereo image sequences, which is commonly used for self-supervised monocular depth estimation. We use Eigen test split with 697 images for testing (raw ground-truth), and we also provide results using improved ground-truth (Uhrig et al. 2017). We train SQLdepth from scratch on KITTI under the simplest setting: auto-masking (Godard et al. 2019) only, no auxiliary information, and no self-distillation. For testing, we also keep the simplest setting: only one single frame as input, while the other methods may use multiple frames and test-time refinement to improve accuracy.

Cityscapes (Cordts et al. 2016) is a challenging dataset which contains numerous moving objects. In order to evaluate the generalization of SQLdepth, we fine-tune (self-supervised) and perform zero-shot evaluation on Cityscapes using the KITTI-pretrained model. Note that we do not use motion mask while most of the other baselines do. We use the data preprocessing scripts from (Zhou et al. 2017a) as others baselines do, and preprocess the image sequences into triples. In addition, we also train SQLdepth from scratch on Cityscapes under the same setting with other baselines.

Make3D (Saxena, Sun, and Ng 2008) To evaluate the generalization ability of SQLdepth on unseen images, we use the KITTI-pretrained SQLdepth to perform zero-shot evaluation on the Make3D dataset, and provide additional depth map visualizations.

5.2 Implementation Details

Our method is implemented using Pytorch framework (Paszke et al. 2019). The model is trained on 3 NVIDIA V100 GPUs, with a batch size of 16. Following the settings from (Godard et al. 2019), we use color and flip augmentations on images during training. We jointly train both DepthNet and PoseNet with the Adam Optimizer (Kingma and Ba 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The initial learning rate is set to $1e-4$ and decays to $1e-5$ after 15 epochs. We set the SSIM weight to $\alpha = 0.85$ and smooth loss term weight to $\lambda = 1e-3$. We use the ResNet-50 (He et al. 2016) with ImageNet (Russakovsky et al. 2015) pretrained weights as

	Method	Train	Test	AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
416×128	Struct2Depth2 (Casser et al. 2019b)	MMask, C	1	0.145	1.737	7.280	0.205	0.813	0.942	0.976
	Monodepth2 (Godard et al. 2019)	–, C	1	0.129	1.569	6.876	0.187	0.849	0.957	0.983
	Videos in the Wild (Gordon et al. 2019)	MMask, C	1	0.127	1.330	6.960	0.195	0.830	0.947	0.981
	SQLdepth (Zero-shot)	–, K	1	0.125	1.347	7.398	0.194	0.834	0.951	0.985
	Li *et al.* (Li et al. 2020)	MMask, C	1	0.119	1.290	6.980	0.190	0.846	0.952	0.982
	Lee *et al.* (Lee et al. 2021b)	MMask, C	1	0.116	1.213	6.695	0.186	0.852	0.951	0.982
	ManyDepth (Watson et al. 2021)	MMask, C	2	0.114	1.193	**6.223**	0.170	0.875	0.967	0.989
	InstaDM (Lee et al. 2021a)	MMask, C	1	0.111	1.158	6.437	0.182	0.868	0.961	0.983
	SQLdepth (From scratch)	MMask, C	1	0.110	**1.130**	6.264	0.165	0.881	0.971	**0.991**
	SQLdepth (Fine-tuned)	–, K→C	1	**0.106**	1.173	6.237	**0.163**	**0.888**	**0.972**	0.990

Table 3: Performance comparison on Cityscapes dataset (Cordts et al. 2016). We present results of zero-shot, fine-tuning (self-supervised) and training from scratch on Cityscapes. All the other baselines are trained from scratch on Cityscapes. K: KITTI, C: Cityscapes, K→C: pretrained on KITTI and fine-tuned on Cityscapes. MMask: use motion mask, –: no motion mask.

backbone, as the other baselines do. We also provide results of ImageNet pretrained Efficient-net-b5 (Tan and Le 2019) backbone, which has similar params with ResNet-50.

5.3 KITTI Results

As shown in Table 1, following prior studies, we conduct experiments under two resolutions: the top half for input image resolution of 640×192, while the bottom half for high resolution of 1024×320. We observe that SQLdepth outperforms all existing self-supervised methods by significant margins, and also outperforms counterparts trained with self distillation, or use multi-frames for testing. We conduct a comparative study with Monodepth2 (Godard et al. 2019) and EPCDepth (Peng et al. 2021). As shown in Figure 1, SQLdepth produces impressive depth maps with sharp boundaries, especially for fine-grained scene details, such as traffic signs and pedestrians. Due to the low quality of ground truth in KITTI, we also provide evaluation results with KITTI improved ground-truth in Table 2. Compared with ManyDepth (Watson et al. 2021), which uses multiple frames for testing, SQLdepth still presents better results across all metrics, and achieves a 5.5% error reduction in terms of AbsRel in 1024×320 resolution, and 6.9% error reduction in 640×192 resolution. In addition, We provide visualizations in Figure 4 to show the effectiveness of depth from a self-cost volume. As for efficiency comparison, details are present in Figure 6.

5.4 Cityscapes Results

In order to evaluate the generalization of SQLdepth, we present results of zero-shot evaluation, fine-tuning (self-supervised), and training from scratch on Cityscapes. We used the KITTI pre-trained model for zero-shot evaluation and fine-tuning. The results are reported in Table 3. We have to emphasize that although most of the baselines use motion mask to deal with moving objects, SQLdepth still presents superior performance without motion mask, and only requires 2 epochs of self-supervised fine-tuning to achieve the best. The zero-shot evaluation result in Table 3 shows competitive performance. In addition, the result of training from scratch also shows SOTA performance.

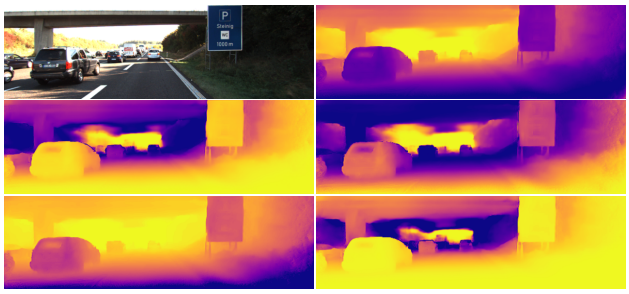

Figure 4: Visualization of planes in the self-cost volume. It's noteworthy that each slice of the self-cost volume maintains clear scene structures. This demonstrates that the self-cost volume essentially serves as a beneficial inductive bias that captures useful geometrical cues for depth estimation.

5.5 Make3D Results

To further evaluate the generalization capacity of SQLdepth, we conducted zero-shot transfer experiments where the pre-trained model on KITTI is directly applied to Make3D (Saxena, Sun, and Ng 2008) dataset. Following the same evaluation setting in (Godard, Mac Aodha, and Brostow 2017), we tested on a center crop of 2×1 ratio. As shown in Table 4 and Figure 5, SQLdepth shows superior performance on the unseen images. This demonstrates the improved generalization capability of SQLdepth, which is attributed to the effectiveness of self-matching-oriented SQL.

Method	AbsRel	SqRel	RMSE
Zhou (Zhou et al. 2017b)	0.383	5.321	10.470
DDVO (Wang et al. 2017)	0.387	4.720	8.090
MD2 (Godard et al. 2019)	0.322	3.589	7.417
CADepthNet (Yan et al. 2021)	0.312	3.086	7.066
SQLdepth (Ours)	**0.285**	**2.202**	**5.582**

Table 4: Make3D results.

5.6 Ablation Study

In this section, we conduct ablation studies to investigate the effects of designs in SQLdepth, including coarse-grained

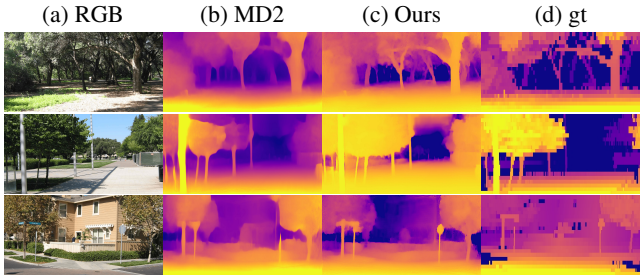

Figure 5: Qualitative results on Make3D (Zero-shot).

queries, plane-wise depth counting, and probabilistic combination, respectively.

Benefits of the SQL layer. As shown in Table 5, SQLdepth without SQL layer shows a massive performance downgrade with a 17.54% decrease from 0.091 to 0.114 in terms of AbsRel. This demonstrates the effectiveness of the SQL layer.

Ablation	$AbsRel \downarrow$	$SqRel \downarrow$	$RMSE \downarrow$
No queries	0.114	0.805	4.816
Learned queries (static)	0.102	0.738	4.512
Fine-grained queries	0.094	0.727	4.437
Coarse-grained queries	**0.091**	**0.713**	**4.204**

Table 5: Ablation study for SQL and coarse-grained queries.

Benefits of coarse-grained queries. To investigate the effectiveness of coarse-grained queries, we compare it with two variants of queries: learned queries (static) and fine-grained queries. Learned queries are learnable vectors (Carion et al. 2020) during training and then frozen for testing. Fine-grained queries are generated from patch embeddings at runtime, but with smaller patch size (e.g. 4×4). The results are summarized in Table 5. We find that runtime queries (fine-grained queries or coarse-grained queries) are better than static queries. This is due to that static queries are not able to adaptively represent the context in different images. For runtime queries, we notice that coarse-grained queries produce better results than fine-grained queries. This is due to that coarse-grained queries are able to capture the contexts within a larger receptive field.

Ablation	$AbsRel \downarrow$	$SqRel \downarrow$	$RMSE \downarrow$
Fixed bins (uniform)	0.114	0.894	4.659
Bins regression (AdaBins)	0.112	0.874	4.534
Bins from counting	**0.087**	**0.659**	**4.096**

Table 6: Ablation of different design choices for depth bins.

Benefits of estimating depth bins via counting. As shown in Table 6, we replace depth bins with fixed bins or depth bins directly regressed from a token extracted by a Transformer, as AdaBins (Bhat, Alhashim, and Wonka 2021) did. Compared with both variants, our proposed depth counting approach leads to better results. This is because that the depth counting can effectively make use of the latent depths in the self-cost volume.

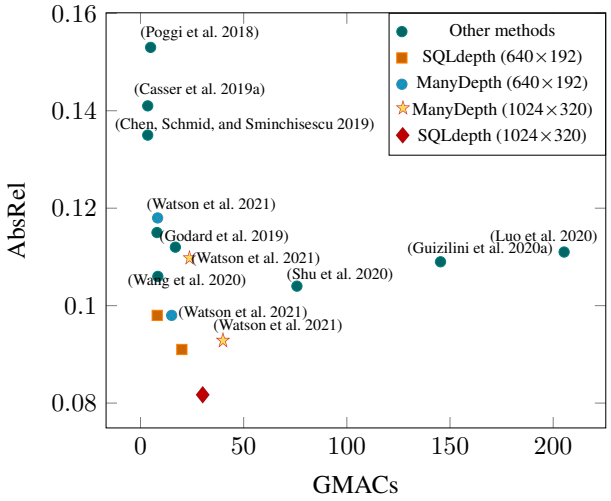

Figure 6: We compare AbsRel against Giga Multiply-Add Caculation per Second (GMACs) on the KITTI Eigen test set. Our model is efficient and also accurate at the same time.

Benefits of probabilistic combination. As shown in Table 7, we replace the probabilistic combination with either global average pooling or Conv1×1. We observe that both of them lead to a substantial performance drop. The potential reason could be that probabilistic combination can adaptively fuse all relative depth estimations provided by different contexts in the image.

Ablation	$AbsRel \downarrow$	$SqRel \downarrow$	$RMSE \downarrow$
Global average pooling	0.115	0.926	4.832
Conv1×1 as combination	0.106	0.826	4.592
Probabilistic combination	**0.087**	**0.659**	**4.096**

Table 7: Ablation study of different combination strategies.

6 Conclusion

In this paper, we have revisited the problem of self-supervised monocular depth estimation. We introduce a simple yet effective method, SQLdepth, in which we build a self-cost volume by coarse-grained queries, extract depth bins using plane-wise depth counting, and estimate depth map using probabilistic combination. SQLdepth attains remarkable SOTA results on KITTI, Cityscapes and Make3D datasets. Furthermore, we demonstrate the improved generalization capability of our model.

Acknowledgements

This work is supported by the Natural Science Foundation of China under Grant No. 72225011, 62372378 and 61960206008. Thanks to the anonymous reviewers, Dr. Bin Guo, Dr. Ying Zhang, Dr. Zhiwen Yu (Northwestern Polytechnical University) and Dr. Mingming Cheng (Nankai University) for their valuable suggestions.

References

Bhat, S. F.; Alhashim, I.; and Wonka, P. 2021. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4009–4018.

Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.-M.; and Reid, I. 2019. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. *arXiv: Computer Vision and Pattern Recognition*.

Casser, V.; Pirk, S.; Mahjourian, R.; and Angelova, A. 2019a. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 8001–8008.

Casser, V.; Pirk, S.; Mahjourian, R.; and Angelova, A. 2019b. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Chen, P.-Y.; Liu, A. H.; Liu, Y.-C.; and Wang, Y.-C. F. 2019. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2624–2632.

Chen, Y.; Schmid, C.; and Sminchisescu, C. 2019. 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*, 7063–7072.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Diaz, R.; and Marathe, A. 2019. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4738–4747.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Eigen, D.; and Fergus, R. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, 2650–2658.

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

Feng, Z.; Yang, L.; Jing, L.; Wang, H.; Tian, Y.; and Li, B. 2022. Disentangling Object Motion and Occlusion for Unsupervised Multi-frame Monocular Depth. *arXiv preprint arXiv:2203.15174*.

Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; and Tao, D. 2018. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2002–2011.

Garg, D.; Wang, Y.; Hariharan, B.; Campbell, M.; Weinberger, K. Q.; and Chao, W.-L. 2020. Wasserstein distances for stereo disparity estimation. *Advances in Neural Information Processing Systems*, 33: 22517–22529.

Garg, R.; Bg, V. K.; Carneiro, G.; and Reid, I. 2016. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, 740–756. Springer.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research*, 32(11): 1231 – 1237.

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

Godard, C.; Mac Aodha, O.; Firman, M.; and Brostow, G. J. 2019. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3828–3838.

Gordon, A.; Li, H.; Jonschkowski, R.; and Angelova, A. 2019. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8977–8986.

Guizilini, V.; Ambrus, R.; Pillai, S.; Raventos, A.; and Gaidon, A. 2020a. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2485–2494.

Guizilini, V.; Hou, R.; Li, J.; Ambrus, R.; and Gaidon, A. 2020b. Semantically-Guided Representation Learning for Self-Supervised Monocular Depth. *international conference on learning representations*.

Guo, X.; Li, H.; Yi, S.; Ren, J.; and Wang, X. 2018. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 484–500.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huynh, L.; Nguyen-Ha, P.; Matas, J.; Rahtu, E.; and Heikkilä, J. 2020. Guiding monocular depth estimation using depth-attention volume. In *European Conference on Computer Vision*, 581–597. Springer.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. *Advances in neural information processing systems*, 28.

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

Johnston, A.; and Carneiro, G. 2020. 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*, 4756–4765.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *arXiv: Learning*.

Klingner, M.; Termöhlen, J.-A.; Mikolajczyk, J.; and Fingscheidt, T. 2020. Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. *european conference on computer vision*.

Klodt, M.; and Vedaldi, A. 2018. Supervising the new with the old: learning SFM from SFM. *european conference on computer vision*.

Kuznietsov, Y.; Stuckler, J.; and Leibe, B. 2017. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6647–6655.

Lee, S.; Im, S.; Lin, S.; and Kweon, I. S. 2021a. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1863–1872.

Lee, S.; Rameau, F.; Pan, F.; and Kweon, I. S. 2021b. Attentive and contrastive learning for joint depth and motion field estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4862–4871.

Li, H.; Gordon, A.; Zhao, H.; Casser, V.; and Angelova, A. 2020. Unsupervised Monocular Depth Learning in Dynamic Scenes. *CoRL*.

Li, Z.; Wang, X.; Liu, X.; and Jiang, J. 2022. BinsFormer: Revisiting Adaptive Bins for Monocular Depth Estimation.

Luo, X.; Huang, J.-B.; Szeliski, R.; Matzen, K.; and Kopf, J. 2020. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4): 71–1.

Luo, Y.; Ren, J.; Lin, M.; Pang, J.; Sun, W.; Li, H.; and Lin, L. 2018. Single View Stereo Matching. *computer vision and pattern recognition*.

Lyu, X.; Liu, L.; Wang, M.; Kong, X.; Liu, L.; Liu, Y.; Chen, X.; and Yuan, Y. 2021. Hr-depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2294–2301.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *neural information processing systems*.

Peng, R.; Wang, R.; Lai, Y.; Tang, L.; and Cai, Y. 2021. Excavating the potential capacity of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15560–15569.

Pilzer, A.; Lathuilière, S.; Sebe, N.; and Ricci, E. 2019. Refine and Distill: Exploiting Cycle-Inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation. *computer vision and pattern recognition*.

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

Ranftl, R.; Bochkovskiy, A.; and Koltun, V. 2021. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12179–12188.

Ranjan, A.; Jampani, V.; Balles, L.; Kim, K.; Sun, D.; Wulff, J.; and Black, M. J. 2019. 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*, 12240–12249.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Saxena, A.; Sun, M.; and Ng, A. Y. 2008. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5): 824–840.

Shu, C.; Yu, K.; Duan, Z.; and Yang, K. 2020. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision*, 572–588. Springer.

Tan, M.; and Le, Q. V. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *international conference on machine learning*.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, 402–419. Springer.

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

Wang, C.; Buenaposada, J. M.; Zhu, R.; and Lucey, S. 2017. Learning Depth from Monocular Videos using Direct Methods. *computer vision and pattern recognition*.

Wang, J.; Zhang, G.; Wu, Z.; Li, X.; and Liu, L. 2020. Self-supervised joint learning framework of depth estimation via implicit cues. *arXiv preprint arXiv:2006.09876*.

Wang, R.; Yu, Z.; and Gao, S. 2023. PlaneDepth: Self-Supervised Depth Estimation via Orthogonal Planes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21425–21434.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Watson, J.; Firman, M.; Brostow, G. J.; and Turmukhambetov, D. 2019. Self-supervised monocular depth hints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2162–2171.

Watson, J.; Mac Aodha, O.; Prisacariu, V.; Brostow, G.; and Firman, M. 2021. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1164–1174.

Yan, J.; Zhao, H.; Bu, P.; and Jin, Y. 2021. Channel-Wise Attention-Based Network for Self-Supervised Monocular Depth Estimation. In *2021 International Conference on 3D Vision (3DV)*, 464–473. IEEE.

Zhan, H.; Garg, R.; Weerasekera, C. S.; Li, K.; Agarwal, H.; and Reid, I. 2018. 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*, 340–349.

Zhang, N.; Nex, F.; Vosselman, G.; and Kerle, N. 2023. Lite-Mono: A Lightweight CNN and Transformer Architecture for Self-Supervised Monocular Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18537–18546.

Zhao, C.; Zhang, Y.; Poggi, M.; Tosi, F.; Guo, X.; Zhu, Z.; Huang, G.; Tang, Y.; and Mattoccia, S. 2022. MonoViT: Self-Supervised Monocular Depth Estimation with a Vision Transformer. In *International Conference on 3D Vision*.

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

Zhou, T.; Brown, M.; Snavely, N.; and Lowe, D. G. 2017a. Unsupervised Learning of Depth and Ego-Motion from Video. *IEEE*.

Zhou, T.; Brown, M.; Snavely, N.; and Lowe, D. G. 2017b. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1851–1858.

Zhou, Z.; and Dong, Q. 2023. Two-in-One Depth: Bridging the Gap Between Monocular and Binocular Self-Supervised Depth Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9411–9421.