Multi-Frame Self-Supervised Depth with Transformers

Vitor Guizilini Rareș Ambruș Dian Chen Sergey Zakharov Adrien Gaidon
Toyota Research Institute (TRI), Los Altos, CA
{first.lastname}@tri.global

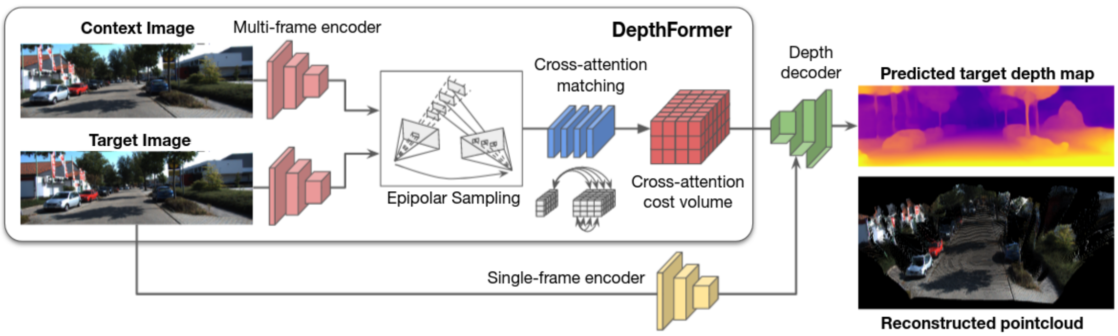


Figure 1. **Our DepthFormer architecture** achieves state-of-the-art multi-frame self-supervised monocular depth estimation by **improving feature matching** across images during cost volume generation.

Abstract

Multi-frame depth estimation improves over single-frame approaches by also leveraging geometric relationships between images via feature matching, in addition to learning appearance-based features. In this paper we revisit feature matching for self-supervised monocular depth estimation, and propose a novel transformer architecture for cost volume generation. We use depth-discretized epipolar sampling to select matching candidates, and refine predictions through a series of self- and cross-attention layers. These layers sharpen the matching probability between pixel features, improving over standard similarity metrics prone to ambiguities and local minima. The refined cost volume is decoded into depth estimates, and the whole pipeline is trained end-to-end from videos using only a photometric objective. Experiments on the KITTI and DDAD datasets show that our DepthFormer architecture establishes a new state of the art in self-supervised monocular depth estimation, and is even competitive with highly specialized supervised single-frame architectures. We also show that our learned cross-attention network yields representations transferable across datasets, increasing the effectiveness of pre-training strategies. Project page: https://sites.google.com/tri.global/depthformer.

1. Introduction

Feature matching is a fundamental component of Structure-from-Motion (SfM). By establishing correspondences between points across frames, a wide range of tasks can be performed, including depth estimation [5, 15, 16, 18], ego-motion estimation [33, 34, 58], keypoint extraction [58, 59], calibration [17, 66], optical flow [30, 51, 77], and scene flow [24, 25]. Within these tasks, self-supervision enables learning without explicit ground-truth [15, 82], by using view synthesis losses obtained via the warping of information from one image onto another, obtained from multiple cameras or a single moving camera. While more challenging from a training perspective [16, 18, 72], self-supervised methods can leverage arbitrarily large amounts of unlabeled data, which has been shown to achieve performance comparable to supervised methods [18, 72], while enabling new applications such as test-time refinement [17, 56, 72] and unsupervised domain adaptation [20].

Single-frame self-supervised methods use multi-view information only at training time, as part of the loss calculation [15, 16, 18, 56, 82]. In contrast, multi-frame methods use multi-view information at *inference time*, traditionally by building cost volumes [32, 57, 72, 74] or correlation layers [24, 61, 62]. These methods learn geometric features in addition to appearance-based ones, which leads to better performance relative to single-frame methods [61, 72, 74]. However, multi-frame calculation relies heavily on feature matching to establish correspondences between frames, using only image information. Because of that, correspondences will be noisy and often inaccurate [16, 18, 74] due to ambiguities and local minima caused by lack of texture, repetitions, luminosity changes, dynamic objects, and so forth.

In this paper we introduce a novel architecture designed to improve self-supervised feature matching (Figure 1), focusing on the task of monocular depth estimation. We build a cost volume between target and context image features using differentiable depth-discretized epipolar sampling, and propose a novel attention-based mechanism to refine per-pixel matching probabilities. We show that the refined probabilities are sharper and more representative of the underlying 3D structure than traditional similarity metrics [70]. The resulting multi-frame cost volume is converted into depth estimates directly, via high-response window filtering, and in combination with single-frame features from a separate network, to account for failure cases in cost volume generation. Through extensive experiments, we show that our feature matching refinement module leads to a new state of the art in self-supervised depth estimation, and that it can be directly transferred between datasets with minimal degradation thanks to its strong geometric grounding. Our main contributions are:

- We introduce a novel architecture, the *DepthFormer*, that **improves multi-view feature matching via cross- and self-attention combined with depth-discretized epipolar sampling**.
- Our architecture leads to **state-of-the-art depth estimation results**. It outperforms other self-supervised multi-frame methods by a large margin, and even **surpasses supervised single-frame architectures**.
- Our learned attention-based matching function is **transferable across datasets**, which can significantly improve convergence speed while decreasing memory.

2. Related Work

2.1. Self-Supervised Depth Estimation

The work of Godard *et al*. [15] introduced self-supervision to the task of depth estimation by framing it as a view synthesis problem, and minimizing an image reconstruction objective [70]. Originally proposed for stereo pairs, the same self-supervised framework was later extended to the monocular setting [82], with the addition of a pose network to estimate camera motion between frames. Although more challenging and restrictive, due to limitations such as scale ambiguity [18] and inability to model dynamic objects [16], monocular self-supervision enables learning from raw videos, which makes it much more scalable to large amounts of data from different sources. Further improvements in the past few years, in terms of view synthesis [16, 56], camera geometry modeling [17, 66], network architectures [18], domain adaptation [20, 21, 49, 81], scale disambiguation [18], and other sources of supervision [17, 19], have led to performance comparable to or even surpassing supervised approaches [18, 37, 72].

2.2. Multi-Frame Depth Estimation

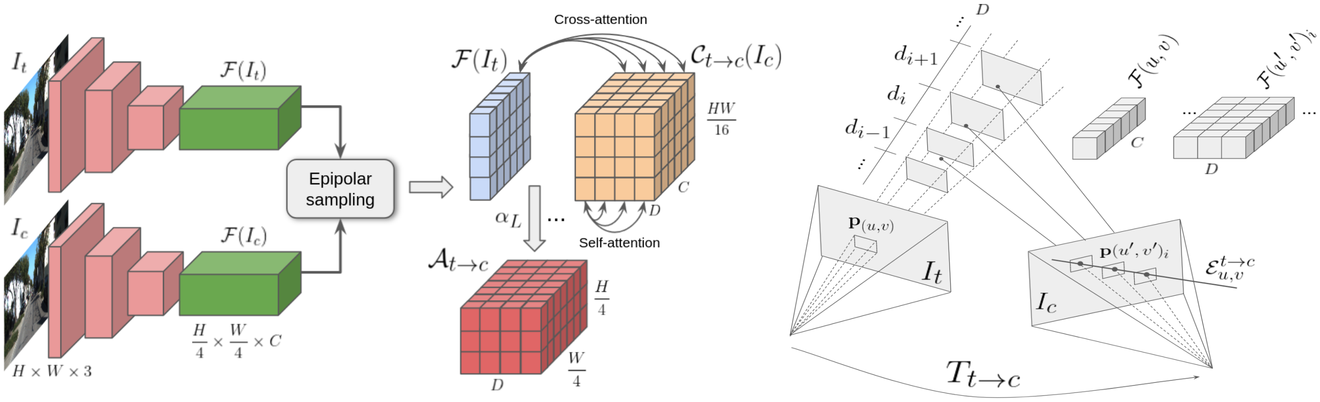
Depth estimation from a single image is inherently an ill-posed problem, since an infinite number of 3D scenes could result in the same 2D projection [22]. Single-frame networks learn appearance-based cues that are suitable for depth estimation (e.g., vanishing point distance, location relative to the ground plane), however these cues are usually based on strong assumptions and will fail with the right adversarial attacks [65]. Multi-frame depth estimation methods circumvent this limitation by using multiple images at test time, which enables the learning of additional geometric cues from feature matching across frames. Although other frameworks for multi-view depth estimation are available, e.g., test-time refinement [5, 44, 56] and recurrent neural networks [38, 48, 79], here we focus on methods that explicitly reason about geometry during inference.

Stereo methods simplify this feature matching process by considering fronto-parallel rectified image pairs with known baseline [2, 35, 41, 45, 75]. Multi-view stereo (MVS) is a generalization of the rectified setting, that operates on images with arbitrary overlaps [23, 27, 31, 43, 76]. Most MVS approaches, however, are supervised and assume known camera poses (either as ground-truth or obtained through COLMAP [55]). Similarly, recently implicit representation methods have also enabled multi-view self-supervised learning [28, 71, 78, 80], including extensions to depth estimation [8, 73]. However, such methods focus on over-fitting to simple scenes with static objects and surrounding high-overlapping views, which limits their generalization to large-scale datasets [4, 7, 13, 18].

Importantly, the use of known camera poses, stereo pairs, supervision and/or static scenes, side-steps some of the main limitations of monocular self-supervised learning. A few methods [72, 74] have recently enabled depth and ego-motion estimation in this setting by combining a multi-frame cost volume with single-frame features. However, they still rely on hand-crafted similarity metrics: ManyDepth [72] uses sum of absolute differences (SAD); and MonoREC [74] uses structural similarity (SSIM). As we shown in our experiments, these metrics are prone to ambiguity and local minima, leading to sub-optimal correspondences. Our attention-based mechanism is designed to improve multi-frame matching for cost volume generation.

2.3. Attention for Depth Estimation

After transforming the field of natural language processing [67], attention-based architectures are becoming increasingly popular in computer vision [9, 40, 42, 50]. In [26], a depth-attention volume is used to guide the learning of indoor planar surfaces, while [53] uses attention for depth decoding. Similarly, [39] uses patch-wise attention over convolutional features, and [50] eliminates convolutional encoding by proposing a fully attention-based backbone. In [29] a self-attention mechanism is used to process a convolutional feature embedding, and depth is decoded via integration over a discretized disparity cost volume. More



(a) **Cross-attention cost volume generation** from two images. (b) **Depth-discretized epipolar sampling** for cost volume generation.

Figure 2. **Diagram for our proposed cross-attention cost volume generation.** Two images (target I_t and context I_c) are processed by a feature extraction network $\mathcal{F}$ and, for each target feature $\mathcal{F}(u,v)$, D matching candidates $\mathcal{F}(u'_i, v'_i)$ are sampled from the depth-discretized epipolar line $\mathcal{E}^{t\rightarrow c}_{u,v}$. A series of L self- and cross-attention layers is then used to refine this initial matching distribution. The output is a *cross-attention cost volume* $\mathcal{A}_{t\rightarrow c}$, containing the matching probability of each target feature relative to its epipolar candidates, given by the corresponding estimated cross-attention value $\alpha_L(u_i, v_i)$.

related to our work, [40] proposes self- and cross-attention over rectified images, followed by cost volume decoding into depth estimates. Their approach, however, is supervised and operates on the simpler stereo setting. A self-supervised monocular attention-based method is proposed in [52], using a spatio-temporal module to leverage both geometric and appearance information. However, by focusing on 3D points for attention, they forego the epipolar constraints we use to determine matching candidates.

3. Self-Supervised Depth with Transformers

3.1. Monocular Depth Estimation

The standard self-supervised monocular depth and ego-motion architecture consists of (i) a depth network $f_D(I_t;\theta_D)$, that produces depth maps $\hat{D}_t$ for a target image I_t; and (ii) a pose network $f_T(I_t, I_c;\theta_T)$, that predicts the relative transformation for pairs of target I_t and context I_c images. This pose prediction is a rigid transformation $\hat{\mathbf{T}}_{t\rightarrow c} = \begin{pmatrix} \hat{\mathbf{R}}_{t\rightarrow c} & \hat{\mathbf{t}}_{t\rightarrow c} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \in \mathrm{SE}(3)$. We train these two networks jointly by minimizing a photometric reprojection error [15, 82] between the original target image I_t and the synthesized target image $\hat{I}_t$, obtained by projecting pixels from I_c onto I_t using predicted depth and pose. The synthesized image is obtained via grid sampling with bilinear interpolation [82], and is thus differentiable, which enables gradient back-propagation for end-to-end training.

3.2. Cross-Attention Cost Volumes

3.2.1 Monocular Epipolar Sampling

A diagram of our proposed cross-attention cost volume generation procedure is shown in Figure 2a. Two $H \times W \times 3$ input images, target I_t and context I_c, are encoded to produce C-dimensional features $\mathcal{F}_t$ and $\mathcal{F}_c$ at 1/4 the original resolution. For each feature $\boldsymbol{f}^{uv}_t \in \mathcal{F}_t$, corresponding to pixel $\mathbf{p}_t = \{u, v\}$, matching candidates are sampled from $\mathcal{F}_c$ along its epipolar line $\mathcal{E}^{uv}_{t\rightarrow c}$, as shown in Figure 2b. We use spatial-increasing discretization (SID) [11] to uniformly sample depth values in *log* space. Assuming D bins ranging from d_{min} to d_{max}, each depth value d_i is given by:

$$\log(d_i) = \log(d_{min}) + \frac{\log(d_{max}/d_{min}) * i}{D} \tag{1}$$

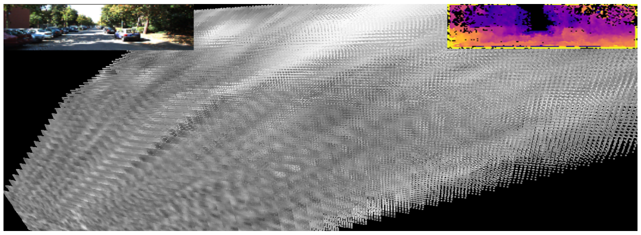
A $H/4 \times W/4 \times D \times C$ *feature volume* $\mathcal{C}_{t\rightarrow c}$ is generated from these matching candidates. Each (u, v, i) cell receives sampled features $\mathcal{F}^{uv}_{t\rightarrow c} = \mathcal{F}_c\langle u'_i, v'_i\rangle$, for $i \in [0, \cdots, D]$, where $\langle\rangle$ is the bilinear sampling operator and (u', v') are projected pixel coordinates such that:

$$z'_i \begin{bmatrix} u'_i \\ v'_i \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}_{t\rightarrow c} \left(\mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} d_i + \mathbf{t}_{t\rightarrow c} \right) \tag{2}$$

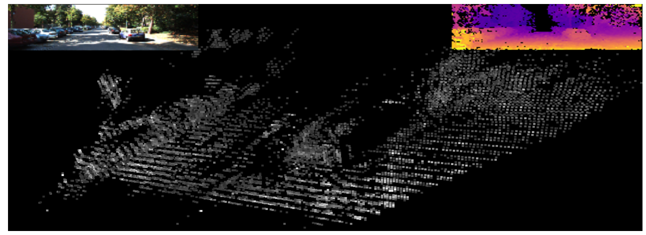
where $\mathbf{R}_{t\rightarrow c}$ and $\mathbf{t}_{t\rightarrow c}$ are relative rotation and translation between frames, and $\mathbf{K} \in \mathbb{R}^{3\times 3}$ are pinhole camera intrinsics. In practice, relative rotation and translation are predicted by the pose network, and $\mathbf{K}$ is assumed known and constant, although this assumption can be relaxed [17, 66].

3.2.2 Cross-Attention Matching

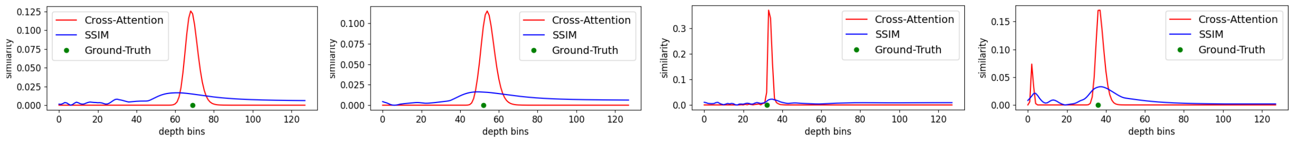
An *attention* module [67] is then used to compute the similarity between $\mathcal{F}_t$ and $\mathcal{C}_{t\rightarrow c}$. More specifically, we use L multi-head attention layers, splitting the C feature channel dimensions into N_h groups such that $C_h = C/N_h$. Feature updates are computed per head and each may have different representations, which increases expressiveness. For each attention head h, a set of linear projections are used to compute queries $\mathcal{Q}_h$ from the target features $\mathcal{F}_t$, and keys $\mathcal{K}_h$



(a) SSIM



(b) Cross-Attention



(c) Matching probability distribution along depth bins for different pixels relative to their depth-discretized epipolar candidates. The blue line shows SSIM values (Equation 9), the red line shows cross-attention values α_L after refinement (Section 3.2), and the green dot marks the corresponding ground-truth depth value (used only for comparison).

Figure 3. **Cost volume visualization.** In (a) and (b), each of the $H \times W \times D$ cells is colored based on its corresponding normalized SSIM or cross-attention value. Even though the decoded depth maps (top right) look similar, our proposed cross-attention cost volume produces sharper distributions, as further evidenced in (c) where we present various per-pixel matching distributions over depth bins. Our proposed attention-based similarity significantly increases the sharpness of these distributions, eliminating ambiguities and local minima.

and values $\mathcal{V}_h$ from the feature volume $\mathcal{C}_{t\to c}$:

$$\begin{aligned} \mathcal{Q}_h &= \mathcal{F}_t W_{\mathcal{Q}_h} + b_{\mathcal{Q}_h} \\ \mathcal{K}_h &= \mathcal{C}_{t\to c} W_{\mathcal{K}_h} + b_{\mathcal{K}_h} \\ \mathcal{V}_h &= \mathcal{C}_{t\to c} W_{\mathcal{V}_h} + b_{\mathcal{V}_h} \end{aligned} \tag{3}$$

with $W_{\mathcal{Q}_h}, W_{\mathcal{K}_h}, W_{\mathcal{V}_h} \in \mathbb{R}^{C_h \times C_h}$, and $b_{\mathcal{Q}_h}, b_{\mathcal{K}_h}, b_{\mathcal{V}_h} \in \mathbb{R}^{C_h}$. Similarities are normalized per-bin using *softmax* to obtain the attention values $\alpha_h \in \mathbb{R}^{N_h \times D}$:

$$\alpha_h = \mathit{softmax}\left(\frac{\mathcal{Q}_h^T \mathcal{K}_h}{\sqrt{C_h}} \right)\Bigg|_D \tag{4}$$

The output values $\mathcal{V} \in \mathbb{R}^C$ are obtained as a weighted concatenation of per-head output values:

$$\mathcal{V} = \left(\alpha_1 \mathcal{V}_1 \oplus \cdots \oplus \alpha_{N_h} \mathcal{V}_{N_h} \right) W_{\mathcal{O}} + b_{\mathcal{O}} \tag{5}$$

where $W_{\mathcal{O}} \in \mathbb{R}^{C \times C}$ and $b_{\mathcal{O}} \in \mathbb{R}^C$, and $\oplus$ is the concatenation operation. Similarly, per-bin attention values $\alpha = \frac{1}{N_h} \sum_h \alpha_h$ are obtained by averaging over the number of heads. This process is repeated L times, each using the output values to update the feature volume for key and value calculation, such that $\mathcal{C}_{t\to c}^{l+1} = \mathcal{V}^l$. The final attention values are used to populate a *cross-attention cost volume* $\mathcal{A}_{t\to c}$, a $H/4 \times W/4 \times D$ structure encoding the similarity between each feature in $\mathcal{F}_t$ and its matching candidates in $\mathcal{C}_{t\to c}$. Each (u, v, i) cell of $\mathcal{A}_{t\to c}$ receives the corresponding attention value $\alpha(u'_i, v'_i)$ from the last cross-attention layer L as the similarity metric for feature matching.

In Figure 3 we show the impact of our proposed cross-attention matching refinement procedure. In Figure 3a the input features are used directly to build a similarity cost volume using SSIM (Equation 9), similar to [72, 74], and in Figure 3b we use the refined cross-attention weights generated from the same features. After refinement the matching distributions are sharper (see Figure 3c for per-pixel examples), resulting in a more robust cost volume without the ambiguities and local minima found in other non-learned appearance-based similarity metrics.

3.2.3 Self-Attention Refinement

Similar to [54], we alternate *cross-attention* between target $\mathcal{F}_t$ and sampled context features $\mathcal{C}_{t\to c}$ with *self-attention* among epipolar-sampled context features. In this setting, queries $\mathcal{Q}'_h$ are also calculated from $\mathcal{C}_{t\to c}$, such that:

$$\mathcal{Q}'_h = \mathcal{C}_{t\to c} W'_{\mathcal{Q}_h} + b'_{\mathcal{Q}_h} \tag{6}$$

The self-attention refinement step takes place after each cross-attention layer, and is repeated $L-1$ times. It is omitted from the last iteration because cross-attention weights α from the last layer L are used to populate $\mathcal{A}_{t\to c}$, not output values $\mathcal{V}$, so self-attention updates are not required.

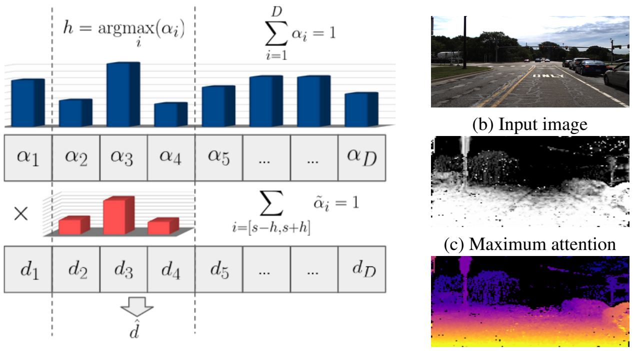
3.3. Cost Volume Decoding

3.3.1 High-Response Depth Decoding

We use a localized high-response window [64] to estimate continuous depth values from discretized bins, thus increasing robustness to multi-modal distributions [40]. A diagram is shown in Figure 4a, and below we describe each step. For each pixel $\mathbf{p}_{uv}$, the argmax operation is used to find the index h_{uv} of the most probable α alongside its sampled epipolar line $\mathcal{E}_{t\to c}^{uv}$. A 1-dimensional $2s+1$ window is placed around h_{uv}, and a re-normalization step is applied:

$$\tilde{\alpha}_i = \frac{\alpha_i}{\sum_i a_i}, \quad \text{for} \quad i \in [h-s, h+s] \tag{7}$$

such that its sum is 1. The depth value for $\mathbf{p}_{uv}$ is calculated by multiplying this re-normalized distribution with the cor-



(a) High-response window (b) Input image (c) Maximum attention (d) High-response depth

Figure 4. **High response depth estimation** from a cross-attention cost volume. Instead of a weighted sum of all candidates, we use a window centered on the most probable candidate.

responding depth bins:

$$\hat{d}_H = \sum_{i \in [h-s, h+s]} d_i \tilde{\alpha}_i \tag{8}$$

The normalized attention values can also be used as a measure of matching confidence, as shown in Figure 4c. In particular, maximum attention values have a clear tendency to decrease at longer depth ranges and particularly towards the vanishing point, which is expected due to resolution degradation and small motion between frames. We leverage this novel matching confidence metric by masking out pixels with maximum attention value below a certain threshold λ_{min}, both from the high response loss calculation and the decoded features (Figure 4d). Evaluation for these intermediate depth maps are provided in Table 2.

3.3.2 Context-Adjusted Depth Decoding

Because our proposed cross-attention cost volume is regressed over epipolar lines, it lacks surrounding context information. To address this limitation, we use a context adjustment layer similar to [40], where estimated depth values are adjusted via conditioning with input images. This adjustment is *residual*, with the output being added to the normalized high-response depth map $\hat{D}_H$ before it is restored using the same statistics. For more details, including qualitative examples, please refer to the supplementary material.

3.3.3 Multi-Scale Depth Decoding

Generating cost volumes from monocular information has two main limitations: (i) it requires ego-motion, and will fail if the camera is static between frames; (ii) it assumes a static world, and will fail in the presence of dynamic objects. To circumvent these limitations, recent methods [72, 74] have proposed combining multi-frame cost volumes with features from a single-frame depth network. These features are then decoded jointly, which makes predicted depth maps robust to multi-frame failure cases.

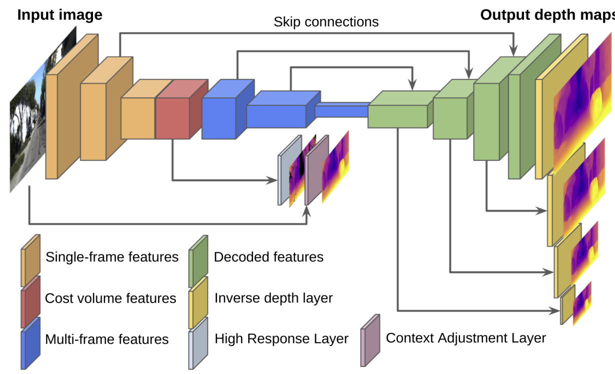


Figure 5. **Decoding architecture** used in our experiments. A single-frame depth network is augmented to include multi-frame cross-attention features (Section 3.2), generating depth maps at multiple scales in addition to those generated directly from multi-frame attention (Sections 3.3.1 and 3.3.2).

Our multi-scale decoding architecture is shown in Figure 5. The cross-attention cost volume (Figure 2a) is first masked out, removing pixels with low matching confidence, and then concatenated with single-frame features from I_t encoded by a separate network. A bottleneck convolutional layer is used to combine these two feature maps, and the output is decoded to produce S depth estimates at multiple increasing resolutions. Similar to [72], we use a teacher-student training procedure, improving the performance of multi-frame predictions via the supervision of a single-frame depth network in areas where cost volume generation fails. This single-frame depth network is trained jointly, sharing the same pose predictions, and discarded during evaluation.

3.4. Training Loss

We train our self-supervised depth and ego-motion architecture end-to-end using only the photometric reprojection loss, consisting of a weighted sum between a structure similarity (SSIM) [70] and absolute error (L1) terms:

$$\mathcal{L}_p = \alpha \frac{1 - \text{SSIM}(I_t, \hat{I}_t)}{2} + (1-\alpha) \, ||I_t - \hat{I}_t|| \tag{9}$$

Following standard procedure, we also use depth regularization [15] to enforce smoothness in low-textured regions:

$$\mathcal{L}_s = \frac{1}{HW} \sum_{u,v} |\delta_u \hat{d}_{uv}| e^{-||\delta_u I_{uv}||} + |\delta_v \hat{d}_{uv}| e^{-||\delta_v I_{uv}||} \tag{10}$$

These two terms are combined to produce the final training loss $\mathcal{L} = \mathcal{L}_p + \lambda_s \mathcal{L}_s$, which is aggregated across all predicted depth maps: $\hat{D}_H$ (high response, Section 3.3.1), $\hat{D}_C$ (context adjustment, Section 3.3.2), and $\hat{D}_M$ (multi-scale, Section 3.3.3) as follows:

$$\mathcal{L} = \lambda_H \mathcal{L}_H + \lambda_C \mathcal{L}_C + \sum_{i=1}^{S} \frac{1}{2^i} \mathcal{L}_{M_i} \tag{11}$$

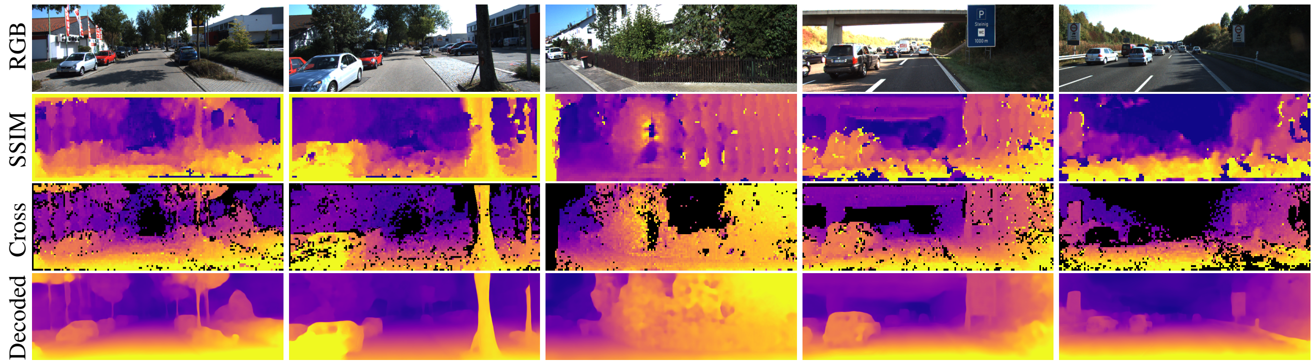
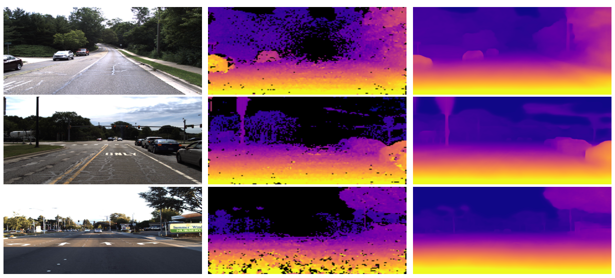


Figure 6. **Predicted depth maps** on the KITTI dataset, obtained from SSIM (argmin) and cross-attention (high response) cost volumes, as well as the decoded depth estimation output (full resolution). Corresponding quantitative results are reported in Table 2.



(a) Input target image (b) High-response depth (c) Decoded depth

Figure 7. **Predicted depth maps** on the DDAD dataset.

4. Experiments

4.1. Datasets

KITTI [13] The KITTI dataset is the standard benchmark for depth evaluation. To compare with other methods, we adopt the training protocol from Eigen *et al.* [10], with Zhou *et al.*'s [82] filtering of static frames, resulting in 39810/4424/697 training, validation, and test images.

DDAD [18] The DDAD dataset is a novel benchmark for depth evaluation, with denser ground-truth and longer ranges, which is particularly challenging for multi-frame methods. Following [18], we use only the front camera, resulting in 12560/3950 training and validation images.

Cityscapes [6] We use the Cityscapes dataset to test the generalization properties of our proposed cross-attention module. We use the 2975 training images with their 30-frame sequences, for a total of 89250 images.

VKITTI2 [3] The Virtual KITTI 2 dataset contains reconstructions of five sequences from the KITTI odometry benchmark [14], for a total of 12936 samples in varying weather conditions and time of day.

Parallel Domain [1] The Parallel Domain dataset, recently introduced in [20], contains procedurally-generated and fully annotated renderings of urban driving scenes. It contains 42000/8000 training and validation samples.

TartanAir [69] TartanAIR is a synthetic photo-realistic dataset for visual SLAM. We train on monocular videos, and following [63] select context images only if the average optical flow magnitude is between 8 and 96 pixels. Our total training set consists of 189696 images.

4.2. Implementation Details

Our models are implemented using PyTorch [47] and trained across 8 Titan V100 GPUs. We use the Adam optimizer [36], with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a batch size of 1 per GPU. Our networks are trained for 50 epochs, with an initial learning rate of $2 \cdot 10^{-4}$ that is halved every 20 epochs. Following [72], we freeze the pose and single-frame teacher network for the final 5 epochs. We use frame $t-1$ as context for cost volume calculation, and frames $t-1$ and $t+1$ for loss calculation. Our training and network parameters are: SSIM weight $\alpha = 0.85$, smoothness weight $\lambda_s = 10^{-4}$, high-response and context-adjusted weights $\lambda_H = \lambda_C = 0.5$, minimum attention $\lambda_{min} = 0.1$, high-response window size $s = 1$, epipolar depth bins $D = 128$, attention dimension $C = 128$, attention heads $h = 8$, attention layers $N = 6$, number of output scales $S = 4$. For more details, please refer to the supplementary material.

4.3. Depth Evaluation

To validate our DepthFormer architecture, we conducted a thorough comparison of its performance relative to other published methods. Our findings targeting the KITTI dataset, considered the standard benchmark for this task, are summarized in Table 1. We consistently outperform all other considered methods by a large margin, including single-frame and multi-frame methods, and even those that leverage additional information in the form of semantic labels [5,17,19] or synthetic data [20,21,29,49,81]. In particular, we significantly improve upon ManyDepth [72], that uses a similar depth decoding strategy but relies directly on the sum of absolute differences (SAD) as the similarity metric, without any feature matching refining strategy. Our architecture also compares favourably to single-frame supervised methods, outperforming the current state of the art (more details in the supplementary material).

In Table 2 we show intermediate depth estimation results

Method	Multi-Fr.	Synthetic	Semantic	*Lower is better*				*Higher is better*		
				AbsRel	SqRel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Struct2Depth [5]			✓	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Gordon *et al.* [17]			✓	0.128	0.959	5.230	0.212	0.845	0.947	0.976
GASDA [81]		✓		0.120	1.022	5.162	0.215	0.848	0.944	0.974
SharinGAN [49]		✓		0.116	0.939	5.068	0.203	0.850	0.948	0.978
Monodepth2 [16]				0.115	0.903	4.863	0.193	0.877	0.959	0.981
Patil *et al.* [48]	✓			0.111	0.821	4.650	0.187	0.883	0.961	0.982
PackNet-SFM [18]				0.111	0.785	4.601	0.189	0.878	0.960	0.982
GUDA [20]		✓		0.107	0.714	4.421	—	0.883	—	—
Johnston *et al.* [29]				0.106	0.861	4.699	0.185	0.889	0.962	0.982
Wang *et al.* [68]	✓			0.106	0.799	4.662	0.187	0.889	0.961	0.982
MonoDEVSNet [21]		✓		0.104	0.721	4.396	0.185	0.880	0.962	<u>0.983</u>
TC-Depth [52]	✓			0.103	0.746	4.483	0.185	0.894	—	<u>0.983</u>
Guizilini *et al.* [19]			✓	0.102	<u>0.698</u>	<u>4.381</u>	0.178	0.896	0.964	**0.984**
ManyDepth [72]	✓			<u>0.098</u>	0.770	4.459	<u>0.176</u>	<u>0.900</u>	<u>0.965</u>	<u>0.983</u>
DepthFormer	✓			**0.090**	**0.661**	**4.149**	**0.175**	**0.905**	**0.967**	**0.984**

Table 1. **Depth estimation results** on the KITTI *Eigen* test split [10], for distances up to 80m with the *Garg* crop [12] and half resolution (640 × 192 or similar). *Multi-Fr.* indicates the use of multiple frames at test time; *Synthetic* the use of additional synthetic training data; and *Semantic* the use of additional semantic information.

Method	AbsRel↓	RMSE↓	$\delta < 1.25$↑
SAD Depth (argmin)	0.647	17.662	0.575
SSIM Depth (argmin)	0.632	17.124	0.598
High-Response Depth	0.264	10.919	0.714
Context-Adjusted Depth	0.167	6.367	0.808
Decoded Depth (1/8)	0.095	4.336	0.892
Decoded Depth (1/4)	<u>0.091</u>	4.201	0.900
Decoded Depth (1/2)	**0.090**	**4.146**	<u>0.904</u>
Decoded Depth (Full)	**0.090**	<u>4.149</u>	**0.905**

Table 2. **Intermediate depth estimation results** of our architecture on the KITTI dataset (Table 1, Figure 6).

from the various outputs of our architecture, with qualitative examples in Figure 6. By replacing SAD or SSIM cost volumes with our cross-attention cost volume with high-response depth self-supervision, we already significantly improve performance, from an Abs.Rel. of 0.647 and 0.632 to 0.264. These results are further improved after context adjustment, to account for low confidence matches, occlusions and inaccuracies in epipolar projection, achieving 0.167. Finally, by combining multi-frame cross-attention with single-frame features for joint decoding, to reason over multi-frame failure cases, we achieve the reported result of 0.090. Interestingly, decoded depth maps at lower resolutions perform almost as well as the full resolution output. We attribute this behavior to the cross-attention cost volume, that is calculated at a lower resolution (1/4) and connected to the decoder via skip connections. Although high resolution decoding is beneficial, it is not necessary for our reported state-of-the-art performance.

We also performed experiments on the DDAD dataset, which is a more challenging benchmark due to its longer depth ranges and larger number of dynamic objects.

Method	AbsRel↓	SqRel↓	RMSE↓	$\delta < 1.25$↑
Monodepth2 [16]	0.213	4.975	18.051	0.761
PackNet-SFM [18]	0.162	3.917	13.452	0.823
GUDA† [20]	0.147	**2.922**	14.452	0.809
ManyDepth [72]	0.146	**3.258**	14.098	0.822
DepthFormer	**0.135**	<u>2.953</u>	**12.477**	**0.836**

Table 3. **Depth estimation results** on the DDAD validation split [18], for distances up to 200m without any cropping (Figure 7). The symbol † indicates supervision from synthetic data.

Even under these conditions, our DepthFormer architecture achieves state-of-the-art results, as shown in Table 3, with qualitative examples in Figure 7.

4.4. Ablation Analysis

In Table 4 we provide an analysis of the different components used in our DepthFormer architecture, including depth estimation results and memory requirements. Firstly, we analyze the impact of our proposed cross-attention module, showing that it is crucial for the reported state-of-the-art performance. We also show that optimizing the cross-attention cost volume itself, via self-supervision on the high-response and context-adjust depth maps, is key to our reported performance as well. This is expected, since without them the cross-attention features are only used in the context of joint single-frame decoding, rather than optimized to generate multi-frame-only depth estimates as well. Similarly, removing self-attention calculation from context features also degrades results. We also ablated different high-response and context-adjusted weights, achieving similar results between $\lambda_H = \lambda_C = [0.1, 1.0]$.

We also experimented with different variations of our architecture, obtained by modifying the number of atten-

Method	Depth Evaluation			GPU (GB)	
	AbsRel↓	RMSE↓	$\delta < 1.25\uparrow$	train	test
W/o cross-attn.	0.099	4.430	0.900	3.8	2.9
W/o cross-attn. loss	0.103	4.581	0.892	12.1	5.3
W/o self-attn.	0.094	4.259	0.901	12.2	5.3
16 depth bins	0.101	4.595	0.894	6.6	3.2
48 depth bins	0.095	4.330	0.900	8.9	4.8
96 depth bins	0.092	4.181	0.903	12.5	5.4
32 attn. channels	0.104	4.761	0.885	8.7	3.7
48 attn. channels	0.098	4.332	0.894	9.6	4.3
96 attn. channels	0.093	4.207	0.899	12.5	5.5
2 attn. layers	0.094	4.388	0.901	11.4	6.1
4 attn. layers	0.093	4.321	0.901	13.3	6.2
DepthFormer	0.090	4.149	0.905	15.2	6.4

Table 4. **Ablation analysis** of the various components of our architecture, including depth evaluation and GPU requirements.

tion layers N, attention feature channels C, and depth bins D. These show a clear overall trend that increasing cross-attention network complexity leads to improved results. This is further evidence that better feature matching is beneficial to depth estimation, but also shows that competitive results can still be obtained with simpler configurations. We leave further exploration of more complex architectures, as well as efficiency improvements [42, 60], to future work.

4.5. Cost Volume Generalization

Our proposed architecture is *modular*, in the sense that the cross-attention network can be separated from the joint single-frame decoding architecture. In this section we explore to which extent we can re-utilize cross-attention cost volumes between datasets, building on the well-studied intuition [40, 46, 52, 61, 74] that geometric features are more transferable than appearance-based ones. To this end, we design three experiments, considering the KITTI dataset as target and multiple other datasets as source. In *hot swap*, we replace the cross-attention network trained on the target dataset with one trained on a source dataset, maintaining the same single-frame and pose networks, without further training. In *fine-tune (mono)*, we train the single-frame and pose networks from scratch, and use a frozen cross-attention network pre-trained on a source dataset. In *fine-tune (all)* we follow the same setting, but also jointly optimize the pre-trained cross-attention network on the target dataset. To fully leverage synthetic data, the VKITTI2, PD, and TartanAir models are pre-trained with depth supervision (using a Smooth L1 loss) and use ground-truth relative poses. Real-world datasets (DDAD and Cityscapes) are pre-trained using the self-supervised loss described in Section 3.4.

Results for these experiments are reported in Table 5. Interestingly, swapping the cross-attention network between datasets results in only a small degradation in performance, of around 5%. This indicates that the learned matching function is robust to distribution shifts between datasets. In fact, we achieved nearly identical results when only training the single-frame and pose networks from scratch, using a frozen cross-attention network pre-trained on a source dataset. However, because the cross-attention network is not optimized (i.e., it is kept frozen), training iterations are both faster (around 100%, from 7.3 to 14.2 FPS) and require less memory (around 20%, from 15.3 to 12.4 GB). Once convergence in this setting is achieved, we can reproduce the reported state-of-the-art results by fine-tuning all networks for only 5 epochs, instead of the 50 required when training the entire architecture from scratch.

Dataset	Variation	AbsRel↓	RMSE↓	$\delta < 1.25\uparrow$
DDAD	Hot swap	0.098	4.364	0.899
	Fine-tune (mono)	0.099	4.336	0.902
	Fine-tune (all)	0.091	4.187	0.904
Cityscapes	Hot swap	0.097	4.339	0.897
	Fine-tune (mono)	0.096	4.291	0.899
	Fine-tune (all)	0.090	4.138	0.905
VKITTI2	Hot swap	0.094	4.302	0.898
	Fine-tune (mono)	0.094	4.232	0.899
	Fine-tune (all)	0.091	4.192	0.904
P. Domain	Hot swap	0.102	4.432	0.888
	Fine-tune (mono)	0.097	4.295	0.897
	Fine-tune (all)	0.090	4.110	0.904
TartanAir	Hot swap	0.102	4.532	0.886
	Fine-tune (mono)	0.095	4.397	0.897
	Fine-tune (all)	0.091	4.187	0.905
KITTI	—	0.090	4.149	0.905

Table 5. **Cross-attention cost volume generalization results** on the KITTI dataset, for different pre-trained source datasets.

5. Conclusion

This paper proposes a novel attention-based cost volume generation procedure for multi-frame self-supervised monocular depth estimation. Our key contribution is a cross-attention module designed to refine feature matching between images, improving upon traditional appearance-based similarity metrics that are prone to ambiguity and local minima. We show that our cross-attention module leads to more robust matching, that is decoded into depth estimates and trained end-to-end using only a photometric objective. We establish a new state of the art on the KITTI and DDAD datasets, outperforming other single- and multi-frame self-supervised methods, and our results are even comparable to state-of-the-art single-frame supervised architectures. We also show that our learned cross-attention module is highly transferable, and can be used without fine-tuning across datasets to speed up convergence and decrease memory requirements at training time.

References

[1] Parallel domain. https://paralleldomain.com/, November 2021. 6

[2] Filippo Aleotti, Fabio Tosi, Li Zhang, Matteo Poggi, and Stefano Mattoccia. Reversing the cycle: self-supervised deep stereo through enhanced monocular distillation. In *ECCV*, 2020. 2

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv:2001.10773*, 2020. 6

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2

[5] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019. 1, 2, 6, 7

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop (Future of Datasets in Vision)*, volume 2, 2015. 2

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, , and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv:2107.02791*, 2021. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction using a multi-scale deep network. *arXiv:1406.2283*, 2014. 6, 7

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 3

[12] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 7

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 2, 6

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 6

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1, 2, 3, 5

[16] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019. 1, 2, 7

[17] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *CVPR*, 2019. 1, 2, 3, 6, 7

[18] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, 2020. 1, 2, 6, 7

[19] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *ICLR*, 2020. 2, 6, 7

[20] Vitor Guizilini, Jie Li, Rares Ambrus, and Adrien Gaidon. Geometric unsupervised domain adaptation for semantic segmentation. In *ICCV*, 2021. 1, 2, 6, 7

[21] Akhil Gurram, Ahmet Faruk Tuna, Fengyi Shen, Onay Urfalioglu, and Antonio M López. Monocular depth estimation through virtual-world supervision and real-world sfm self-supervision. *arXiv:2103.12209*, 2021. 2, 6, 7

[22] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[23] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018. 2

[24] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *CVPR*, 2020. 1

[25] Junhwa Hur and Stefan Roth. Self-supervised multi-frame monocular scene flow. In *CVPR*, 2021. 1

[26] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and J. Heikkila. Guiding monocular depth estimation using depth-attention volume. In *ECCV*, 2020. 2

[27] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv:1905.00538*, 2019. 2

[28] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Animashree Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 2

[29] Adrian Johnston and G. Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *CVPR*, 2020. 2, 6, 7

[30] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. *arXiv:2006.04902*, 2020. 1

[31] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *arXiv:1708.05375*, 2017. 2

[32] Tong Ke, Tien Do, Khiem Vuong, Kourosh Sartipi, and Stergios I Roumeliotis. Deep multi-view depth estimation with predicted uncertainty. *arXiv:2011.09594*, 2020. 1

[33] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 1

[34] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1

[35] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, and Peter Henry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 2

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 6

[37] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *ECCV*, 2020. 2

[38] Aran Kumar, Suchendra Bhandarkar, and Mukta Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *CVPR Workshops*, 2018. 2

[39] Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. Patch-wise attention network for monocular depth estimation. In *AAAI*, 2021. 2

[40] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. *arXiv:2011.02910*, 2020. 2, 3, 4, 5, 8

[41] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *CVPR*, 2018. 2

[42] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 8

[43] Keyang Luo, Tao Guan, Lili Ju, Yuesong Wang, Zhuo Chen, and Yawei Luo. Attention-aware multi-view stereo. In *CVPR*, 2020. 2

[44] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. In *SIGGRAPH*, 2020. 2

[45] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 2

[46] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 2021. 8

[47] 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 *NeurIPS*, 2017. 6

[48] Vaishakh Patil, Wouter Gansbeke, Dengxin Dai, and Luc Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 2020. 2, 7

[49] Koutilya PNVR, Hao Zhou, and David Jacobs. Sharingan: Combining synthetic and real data for unsupervised geometry estimation. In *CVPR*, 2020. 2, 6, 7

[50] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv:2103.13413*, 2021. 2

[51] Anurag Ranjan, Varun Jampani, Lukas Balles, Deqing Sun, Kihwan Kim, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, 2019. 1

[52] Patrick Ruhkamp, Daoyi Gao, Hanzhi Chen, Nassir Navab, and Benjamin Busam. Attention meets geometry: Geometry guided spatial-temporal attention for consistent self-supervised monocular depth estimation. In *3DV*, 2021. 3, 7, 8

[53] Assem Sadek and Boris Chidlovskii. Self-supervised attention learning for depth and ego-motion estimation. In *IROS*, 2020. 2

[54] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 4

[55] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[56] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020. 1, 2

[57] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1

[58] Jiexiong Tang, Rares Ambrus, Vitor Guizilini, Sudeep Pillai, Hanme Kim, Patric Jensfelt, and Adrien Gaidon. Self-Supervised 3D Keypoint Learning for Ego-Motion Estimation. In *CoRL*, 2020. 1

[59] Jiexiong Tang, Hanme Kim, Vitor Guizilini, Sudeep Pillai, and Rares Ambrus. Neural outlier rejection for self-supervised keypoint learning. In *ICLR*, 2020. 1

[60] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ArXiv:2009.06732*, 2020. 8

[61] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *ICLR*, 2020. 1, 8

[62] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1

[63] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *arXiv:2108.10869*, 2021. 6

[64] Stepan Tulyakov, Anton Ivanov, and François Fleuret. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. In *NeurIPS*, 2018. 4

[65] Tom Van Dijk and Guido De Croon. How do neural networks see depth in single images? In *ICCV*, 2019. 2

[66] Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Wolfram Burgard, Greg Shakhnarovich, and Adrien Gaidon. Neural ray surfaces for self-supervised learning of depth and ego-motion. In *3DV*, 2020. 1, 2, 3

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[68] Jianrong Wang, Ge Zhang, Zhenyu Wu, Xuewei Li, and Li Liu. Self-supervised joint learning framework of depth estimation via implicit cues. *arXiv:2006.09876*, 2020. 7

[69] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020. 6

[70] 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*, 2004. 2, 5

[71] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv:2102.07064*, 2021. 2

[72] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7

[73] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 2

[74] F. Wimbauer, N. Yang, L. von Stumberg, N. Zeller, and D Cremers. Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *CVPR*, 2021. 1, 2, 4, 5, 8

[75] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Spatial correspondence with generative adversarial network: Learning depth from monocular videos. In *ICCV*, 2019. 2

[76] Youze Xue, Jiansheng Chen, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, and Jiayu Bao. Mvscrf: Learning multi-view stereo with conditional random fields. In *ICCV*, 2019. 2

[77] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 1

[78] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2

[79] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *ICCV*, 2019. 2

[80] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 2

[81] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *ICCV*, 2019. 2, 6, 7

[82] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 1, 2, 3, 6