FutureDepth: Learning to Predict the Future Improves Video Depth Estimation

Rajeev Yasarla, Manish Kumar Singh, Hong Cai, Yunxiao Shi, Jisoo Jeong, Yinhao Zhu, Shizhong Han, Risheek Garrepalli, Fatih Porikli

Qualcomm AI Research*
{ryasarla, masi, hongcai, yunxshi, jisojeon, yinhaoz, shizhan, rgarrepa, fporikli}@qti.qualcomm.com

Abstract. In this paper, we propose a novel video depth estimation approach, FutureDepth, which enables the model to implicitly leverage multi-frame and motion cues to improve depth estimation by making it learn to predict the future at training. More specifically, we propose a future prediction network, F-Net, which takes the features of multiple consecutive frames and is trained to predict multi-frame features one time step ahead iteratively. In this way, F-Net learns the underlying motion and correspondence information, and we incorporate its features into the depth decoding process. Additionally, to enrich the learning of multi-frame correspondence cues, we further leverage a reconstruction network, R-Net, which is trained via adaptively masked auto-encoding of multi-frame feature volumes. At inference time, both F-Net and R-Net are used to produce queries to work with the depth decoder, as well as a final refinement network. Through extensive experiments on several benchmarks, i.e., NYUDv2, KITTI, DDAD, and Sintel, which cover indoor, driving, and open-domain scenarios, we show that FutureDepth significantly improves upon baseline models, outperforms existing video depth estimation methods, and sets new state-of-the-art (SOTA) accuracy. Furthermore, FutureDepth is more efficient than existing SOTA video depth estimation models and has similar latencies when comparing to monocular models.

Keywords: Depth estimation · Temporal consistency · Video prediction

1 Introduction

Depth plays a critical role for 3D perception, in applications like autonomous driving, augmented reality/virtual reality (AR/VR), camera image and video processing, and robotics. While depth can be measured using LiDAR or Time-of-Flight (ToF) sensors, they are expensive, incur substantial power consumption, need to be calibrated, and can fail to produce correct measurements for certain surfaces (e.g., specular surfaces). As such, inferring depth based on camera images has become a cost effective and promising alternative. Traditional approaches [12, 31, 38], such as stereo vision and structure-from-motion, have

* Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

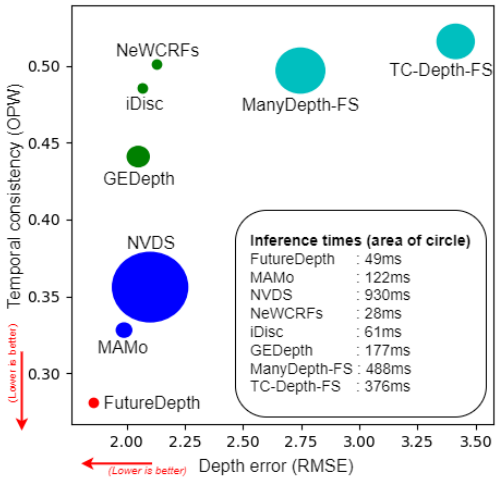

Fig. 1: FutureDepth vs. existing SOTA in terms of depth accuracy (RMSE), temporal consistency (OPW [48]), and runtime (on NVIDIA RTX-3080 GPU), on KITTI. We compare with monocular methods: NeWCRFs [53], iDisc [33], and GEDepth [51], cost-volume-based methods: ManyDepth [49] and TC-Depth [36] (both fully supervised here), and video-based methods: MAMo [52] and NVDS [48]. FutureDepth outperforms existing methods in terms of both accuracy and temporal consistency, and runs efficiently.

been utilized to calculate depth, but have limited accuracy. Recently, by utilizing deep learning, researchers have achieved significantly more accurate depth estimation [1, 3, 9, 11, 33, 51].

Using a neural network to predict depth based on a single image, or monocular depth estimation, has been extensively studied due to simplicity of the setup and model efficiency [1, 3, 9, 11, 33, 51, 53]. This, however, does not take into account the consecutive video frames that are almost always available in many practical applications like autonomous driving and AR/VR. More recently, researchers have proposed different ways to exploit multiple frames for depth estimation. One common approach involves the employment of a cost volume, which assesses depth hypotheses and can be trained end-to-end within a neural network. Cost volumes can enable significant accuracy improvement, at the expense of significantly higher computational complexity and memory usage. Other researchers look into auto-regressive approaches, which does not require cost volumes but instead leverage alternative mechanisms such as recurrent neural network [32, 54], optical flow [10, 50], and/or attention [6, 47, 48, 52]. While they can be more efficient than cost volume models, these methods still require costly operations to achieve SOTA accuracy. For instance, among latest works, MAMo [52] requires optical flow estimation, attention, and online gradient computation, while NVDS [48] requires pairwise cross-attention between the target frame and every source frame to compute depth. Moreover, they do not consider predicting depths for consecutively frames jointly, and unable to learn underlying

dynamic motion/trajectories of objects and corresponding spatial information, which results in suboptimal temporal consistency.

In this paper, we propose a novel video depth estimation approach, FutureDepth, which leverages future prediction and adaptive masked reconstruction to enable the model to learn and utilize key, multi-frame spatial and temporal features, while being computationally efficient. More specifically, we take a representation learning approach and propose a multi-step Future Prediction Network, F-Net, which is trained to predict features of future frames that are one time step ahead based on the current given set of frame features, in an iteratively manner. This allows F-Net to identify how pixel-wise scene and object features move across time, as it learns to predict the future. At inference time, F-Net works together with the main encoder-decoder depth network and extracts useful motion features to enhance depth computation in the decoder.

In order to further enrich multi-frame correspondence understanding, we additionally propose a Reconstruction Network, R-Net, which is trained to perform Masked Auto-Encoding (MAE) on features of a consecutive set of frames with a learnable, adaptive masking strategy. This encourages R-Net to leverage critical scene features distributed across frames for reconstruction and thus, understand the multi-view correspondences. After training, R-Net parses the multi-frame features and generates key scene features to support the depth prediction. Note that this is different from existing video MAE methods (e.g., [44, 45]). Video MAE methods are used to pretrain the main encoder-decoder network, whereas we propose additional networks (F-Net & R-Net) work with the main network at inference (which can be any base depth network) to improve video depth estimation while maintaining computation efficiency.

Finally, we propose a small refinement network after the decoder, which further enhances the details of the predicted depth map.

Our main contributions are summarized as follows:

- We propose a novel approach, FutureDepth, which leverages future prediction and adaptive masked reconstruction to enhance the model's ability to extract and exploit key, multi-frame motion and correspondence cues for video depth estimation.
- Our proposed Future Prediction Network, F-Net adopts a multi-frame/time step future prediction loss based on auto-regressive sampling of future frames. This forces F-Net to extract stronger motion, correspondence cues at inference time for better temporally consistent depth prediction. To the best of our knowledge, this is the first work to combine a multi-frame/time-step objective without teaching forcing and combine batch processing.
- We additionally propose a Reconstruction Network, R-Net, which is trained using learnable, adaptive masked auto-encoding of multi-frame features. Features generated by R-Net are also incorporated by the depth decoder to enrich scene understanding for better spatio-temporal aggregation. Furthermore, we propose a small refinement network to improve the details of the final depth maps.
- We conduct extensive experiments on several depth estimation datasets: NYUDv2 [42], KITTI [13], DDAD [15], and Sintel [4]. FutureDepth sets the

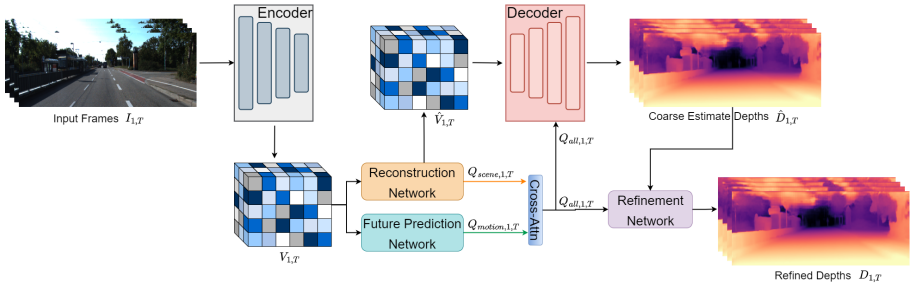

Fig. 2: Our proposed FutureDepth method. Features of consecutive frames are extracted by the encoder and fed to the Future Prediction Network (F-Net) and Reconstruction Network (R-Net), which are trained using iterative future prediction and adaptive masked auto-encoding, respectively. At inference, features generated by F-Net and R-Net, $Q_{motion,1,T}$ and $Q_{scene,1,T}$, which contain key motion and correspondence cues, are integrated into the depth decoding process. Furthermore, these features are also utilized in a refinement process to improve the final depth map quality.

new state-of-the-art (SOTA). As shown in Fig. 1, FutureDepth has lower depth errors, is more temporally consistent, and runs faster than existing video depth methods and has similar latencies as compared to SOTA monocular models.

2 Proposed Approach: FutureDepth

Consider a batch of consecutive video frames, $I_{1,T} = \{I_1, I_2, ..., I_T\}$, for which FutureDepth computes their depths jointly, $D_{1,T} = \{D_1, D_2, ..., D_T\}$, where T is the number of frames. The input frames are first fed into an image encoder individually. The image features are concatenated along the time and channel dimension to form a feature volume, $V_{1,T} = \{F_1, , F_2, ..., F_T\} \in \mathbb{R}^{H' \times W' \times (T \cdot C)}$, where F_t is feature map of I_t extracted by the encoder, H' and W' are the height and width of the feature maps, and C is the number of feature channels.

In the proposed FutureDepth framework, we propose a Future Prediction Network, F-Net, which is trained with multi-step prediction of future features based on features of a given set of consecutive frames. In this way, F-Net learns to capture the underlying motion information of the video frames and generates useful features to facilitate the depth prediction. To further enrich scene understanding, we additionally propose a Reconstruction Network, R-Net, which is trained via auto-encoding on the multi-frame features with a learnable masking scheme. In order to recover the original features, R-Net learns to seek available scene information distributed across frame and thus, implicitly identifies and utilizes multi-frame correspondences. After training, both networks work with the main encoder-decoder network, providing key, implicit motion and correspondence query features, $Q_{motion,1,T}$ and $Q_{scene,1,T}$, to enhance depth prediction. Fig. 2 provides an overview of our proposed FutureDepth approach. For conciseness, we drop the subscripts "1" and "T" on the query feature variables when the context is clear in the subsequent sections.

2.1 Future Prediction Network (F-Net)

The goal of F-Net is to capture useful, underlying motion cues to assist the video depth prediction process. To this end, we train F-Net based on multi-step future prediction within the auto-regressive paradigm, i.e., we iteratively predict multi-frame features one step ahead with *unrolling*. More specifically, F-Net takes $V_{1,T}$ as input and predicts $\tilde{V}_{2,T+1}$. It then continues to take the currently predicted volume $\tilde{V}_{i,T+i-1}$ and predicts the one that is one step ahead $\tilde{V}_{i+1,T+i}$, until a prescribed number of steps, L, is reached. Our approach does not use teacher forcing, and the model is unrolled in an auto-regressive way. This means that when the network is predicting future values $\tilde{V}_{2,T+1}, \tilde{V}_{3,T+2}...\tilde{V}_{L+1,T+L}$, it does not take the actual future values $V_{2,T+1}, V_{3,T+2}, ..., V_{L,T+L-1}$ as input, when calculating the multi-step/multi-frame loss. This requirement compels the network to minimize error accumulation across time-steps during unrolling, thereby extracting more robust motion and correspondence cues. This process is illustrated in Fig. 3. Our work builds upon research in model-based reinforcement learning, where previous studies have explored the distinction between 'observation-dependent' and 'prediction-dependent' unrolling strategies [7, 16, 17]. This also aligns with the concept of 'covariate shift' explored in the context of imitation learning and teacher forcing [2, 43]. To our knowledge, this is the first work to combine a multi-frame/time-step objective with batched processing, specifically aimed at improving the temporal consistency of dense perceptual tasks.

By using the feature volume of multiple frames to predict the entire volume one step ahead, F-Net learns how the objects and scene move over time extracting stronger spatio-temporal information. Specifically, to predict the future feature at a certain pixel location, F-Net needs to find the corresponding features that are available from the current and previous time steps. This essentially enables F-Net to understand the underlying motion and multi-frame correspondences, as well as motion in longer contexts.

F-Net is trained with the following loss:

$$\mathcal{L}_F = \sum_{i=1}^{L} ||\tilde{V}_{1+i,T+i} - V_{1+i,T+i}||_2, \tag{1}$$

where $\tilde{V}_{1+i,T+i-1}$ is the predicted volume for time steps $1+i$ to $T+i$.

We use a transformer-based architecture for F-Net, which consists of convolutional and attention layers; more network details can be found in Sec. 3.1. To generate motion features that will be used for depth decoding, we use the last-layer features from all the prediction steps and average them to obtain Q_{motion}, which is fed into the depth decoder. Fig. 4 provides sample visualizations of Q_{motion}. We can see that moving objects are captured in Q_{motion}; particularly, extended motion understanding can be seen on the train and biker.

2.2 Reconstruction Network (R-Net)

We train R-Net using learnable, adaptive masked auto-encoding of the video feature volume. Suppose we have masks, $M_{1,T} = \{M_1, \ldots, M_T\}$, which are derived from the input image features. We element-wise multiply the masks with

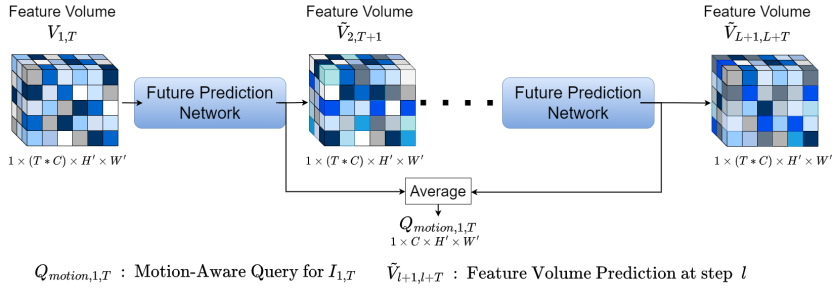

Fig. 3: Future Prediction Network (F-Net).

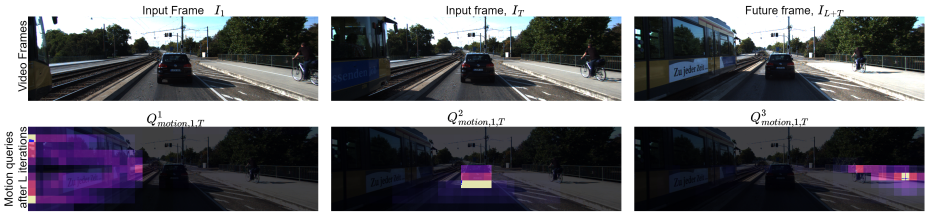

Fig. 4: Example motion query $Q_{motion,1,T}$ generated using future prediction network. Here we show three example channels in $Q_{motion,1,T}$, with $T = 4$, and $L = 6$.

the feature volume to generate the masked feature volume, $M_{1,T} \odot V_{1,T}$, which R-Net uses as as input and produces the reconstructed volume, $\hat{V}_{1,T}$. R-Net is trained using the following loss:

$$\mathcal{L}_R = ||(1 - M_{1,T}) \odot (\hat{V}_{1,T} - V_{1,T})||_2 + \mathcal{L}_D(D_{1,T}, D^{gt}_{1,T}), \tag{2}$$

which is the sum of the L_2 loss between the reconstructed and original feature volumes for masked locations, as well as a SILog loss [9], $\mathcal{L}_D$, between predicted depths $D_{1,T}$ based on the reconstructed features and ground-truth depths $D^{gt}_{1,T}$.

We adopt a learned scheme to generate the masks based on the input image features. Since no ground-truth masks are available, we train the mask generator using a SILog depth loss between predicted depths $D_{1,T}$ based on the masked features and ground-truth depths $D^{gt}_{1,T}$, with all the other network components frozen. We denote this loss as $\mathcal{L}_A$.

We observe that the mask generator learns to mask the frames in a way that encourages R-Net to exploit multi-view correspondences. In particular, it masks out different parts of the same object across frames; for instance, see the white truck and the masks over it in Fig. 5. As such, R-Net needs to utilize information distributed across frames in order to reconstruct the full feature volume.

We adopt the same architecture of F-Net for R-Net. Once R-Net is trained, we use it to generate features, Q_{scene}, which contain important features of the scene and can be used to enhance depth estimation. More specifically, the input feature volume (not masked at inference) goes through convolutional and attention layers, and we use the output features from the last layer as Q_{scene}. More details of R-Net architecture can be found in Sec. 3.1. Fig. 6 visualizes Q_{scene}. We see that in two sample channels, Q_{scene} captures important foreground information (e.g., truck, parked cars, nearby tree) at different time steps.

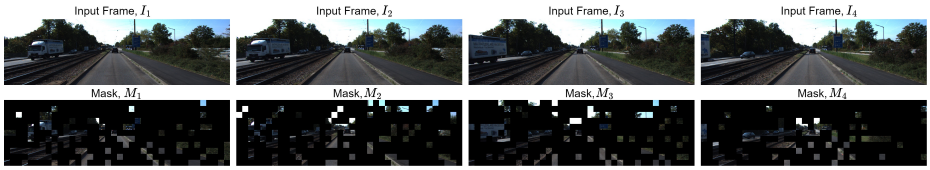

Fig. 5: Generated masks over a sample input set of frames. Masks generated by adaptive sampler focuses on important objects like van, cars, tram, bus, railway tracks, and road boundaries etc. across the frames.

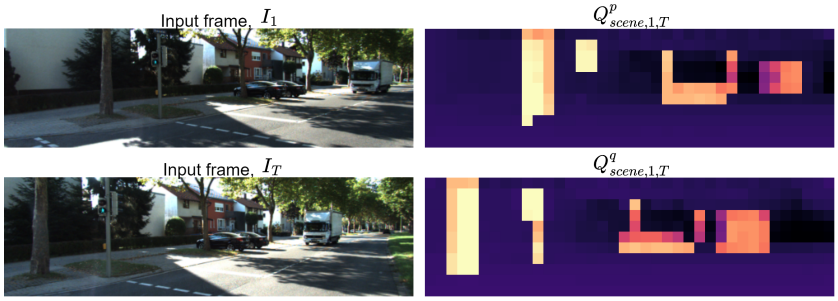

Fig. 6: Sample Q_{scene} generated by R-net. We show two sample channels p and q in $Q_{scene,1,T}$ for input frames $I_{1,T}$ ($T = 4$).

2.3 Using the F-Net and R-Net Features

The features provided by F-Net and R-Net summarize key motion and scene information from the set of input frames. Given Q_{motion} and Q_{scene}, we first perform cross-attention between them to produce Q_{all}. We expand the channel dimension of Q_{all} by T, by repeating its C channels.

In the decoder, we employ transformer layers, where we combine Q_{all} and the queries generated from the previous decoder layer's output, and process them at every decoder layer (more details are provided in the supplementary). In this way, we incorporate useful motion and multi-frame correspondence information into the depth prediction process.

In order to further improve the predicted depths from the decoder, we additionally design a progressive refinement network, consisting of self- and cross-attention layers. The refinement network takes depth maps as input and at the cross-attention layers, leverages Q_{all} again to refine the depth map features. The refined version of the depth maps can be fed to the refinement network again to generate even further refined ones; we repeat this N times in our pipeline. As we shall see, this helps improve the details of the depth maps.

2.4 Training

We first pretrain the FutureDepth encoder and decoder to perform depth prediction, without using F-Net, R-Net, and the refinement network. This training is supervised with a SILog loss between the predicted and ground-truth depths. After the encoder and decoder are trained, we train R-Net to perform feature volume reconstruction with random masking. In this step, we only use the L_2

loss to train R-Net, and do not update the encoder or decoder. The pretraining provides reasonable initial weights for several components in FutureDepth, which is useful for a stable training of the entire system.

In the main training phase, we initialize both F-Net and R-Net with the pretrained R-Net weights, as they share the same architecture, as well as initialize the encoder and decoder with their pretrained weights. First, we freeze encoder and decoder, and train the adaptive mask generator, F-Net, and R-Net simultaneously. We compute the $\mathcal{L}_F$, $\mathcal{L}_A$, and $\mathcal{L}_R$ losses and learn the weights for F-Net, mask generator, and R-Net, respectively. Finally, we freeze F-Net, mask generator, and R-Net, and train the encoder, decoder, and refinement network using the following loss:

$$\mathcal{L}_{D,final} = \frac{1}{NT} \sum_{i=0}^{N} \sum_{t=1}^{T} \mathcal{L}_D(D_t^i, D_t^{gt}) \tag{3}$$

where L_D is the SILog loss [9] and the superscript i indicates the refinement step. In this final step, Q_{scene} and Q_{motion} are cross-attended to generate Q_{all}, which is used in the decoding and refinement parts of the pipeline.

We refer readers to Algorithm 2 in the supplementary material for a more detail description of the training process. Note that after training, when performing inferences, we no longer need the mask generator and the inference pipeline is shown in Fig. 2. Detailed description of the inference can be found in Algorithm 3 the supplementary.

3 Experiments

We conduct extensive experiments to evaluate our proposed FutureDepth approach on large-scale public benchmarks and compare with existing state-of-the-art methods. We also perform ablation studies to analyze different aspects of our proposed approach.

3.1 Implementation and Experiment Setup

Networks. In FutureDepth, we have a main an encoder-decoder depth network architecture, where the encoder can be any image backbones like ResNet [18], ViT [8], and Swin [25], and the decoder consists of four Skip Attention Modules (SAM, modified from [1]). We start from a single-frame baseline (designed by us) containing only an encoder and a decoder to compute depths, with 3 input channels and a single depth output channel. Then, we create a multi-frame baseline that processes T frames as a batch, where their encoder outputs are concatenated to create a single feature volume with $T \times C$ channels. The feature volume is consumed by the decoder in its entirety, based on which the decoder predicts T depth outputs in a batch manner. FutureDepth shares a similar base encoder-decoder architecture as the multi-frame baseline, but additionally includes our proposed F-Net, R-Net, and refinement network. In all our experiments, we use Swin-Large (Swin-L) as the encoder unless specified otherwise.

Both F-Net and R-Net consists of a self-attention layer, a convolutional layer that reduces the channel dimension from $T \cdot C$ to C, and four SAM layers. For R-Net, we use the queries from the last SAM layer as Q_{scene}. For F-Net, we use the last-layer queries from all prediction steps and average them to generate Q_{motion}. The mask generator consists of fully-connected layers and a softmax layer. Based on the softmax scores, we keep the top $r \times P$ patches and mask out the rest, where r is the masking ratio and P is total number of patches. The refinement network consists of one self-attention layer and two cross-attention layers. It takes depth maps as input and at the cross-attention layers, incorporates Q_{all} to predict the improved depths.

Hyperparameters. We set number of video frames in a batch $I_{1,T}$ to be $T = 4$. For each frame batch during training, we sample from $1, 2, 3, 4$ to determine the interval between two consecutive frames. For instance, if the interval is 2, we sub-sample every other frame from the original sequence to form the batch. This allows the network to see more diverse motion ranges. We set the number of iterations in future prediction to $L = T$ unless otherwise specified. We set the number of refinement step to $N = 3$. In training, we sample the masking ratio from $r \in [0.6, 0.9]$ to train R-Net. We set the initial learning rate to 4×10^{-5} and then linearly decrease it to 4×10^{-6}. In the pretraining stage, we train the encoder and decoder for 5 epochs, and subsequently, the R-Net for 3 epochs. In main training part, we train all the components of FutureDepth for 15 epochs. The total training takes about 2 days on 2 Nvidia A100 GPUs.

Evaluation. We use standard depth estimation metrics defined in [9]. In addition, we evaluate the temporal consistency of the predicted depths, using aTC (lower is better) and rTC (higher is better) from [21, 52], and OPW (lower is better) from [48]. These metrics assess the prediction consistency across two frames by warping using optical flow.[1]

3.2 Datasets

NYUDv2 [42]. This is a standard benchmark for indoor depth estimation tasks, containing 120K RGB-D videos captured from 464 indoor scenes. We follow the official Eigen training and test splits to evaluate our method, where 249 scenes are used for training and 654 images from 215 scenes are used for testing.

KITTI [13]. KITTI is one of the most commonly used benchmarks for outdoor depth estimation. We follow the Eigen training and test splits [9], with 23,488 training images and 697 test images. We use the video (sub)sequences that correspond to the training and test image. These video frames are of size 375×1241 and depth estimation is evaluated up to 80 meters.

DDAD [15]. Dense Depth for Autonomous Driving (DDAD) is a more recently introduced dataset featuring diverse urban driving scenarios with extended depth ranges. This dataset contains 12,650 samples for training and 3,950 samples for validation. We use the corresponding video (sub)sequences for training and

[1] Detailed mathematical definitions of the metrics can be found in the supplementary file.

Table 1: Comparison with SOTA video-based models on NYUDv2 Eigen split and Sintel. OPW measures temporal consistency, as proposed in NVDS paper. FS means method is trained in fully-supervised fashion using ground-truth depth. ↑ (↓) means higher (lower) is better.

Method	NYUDV2			Sintel		
	$\delta < 1.25$ ↑	Abs Rel↓	OPW↓	$\delta < 1.25$ ↑	Abs Rel↓	OPW↓
ST-CLSTM [54]	0.833	0.131	0.645	0.351	0.517	0.585
FMNet [47]	0.832	0.134	0.387	0.357	0.513	0.521
R-CVD [19]	0.886	0.103	0.394	0.521	0.422	0.475
Many-Depth-FS [49]	0.865	0.096	0.428	0.492	0.487	0.540
NVDS [48]	0.950	0.072	0.364	0.591	0.335	0.424
MAMo [52]	0.942	0.074	0.388	0.579	0.358	0.493
Baseline (ours)	0.917	0.093	0.480	0.477	0.504	0.611
FutureDepth (ours)	**0.981**	**0.063**	**0.303**	**0.623**	**0.296**	**0.392**

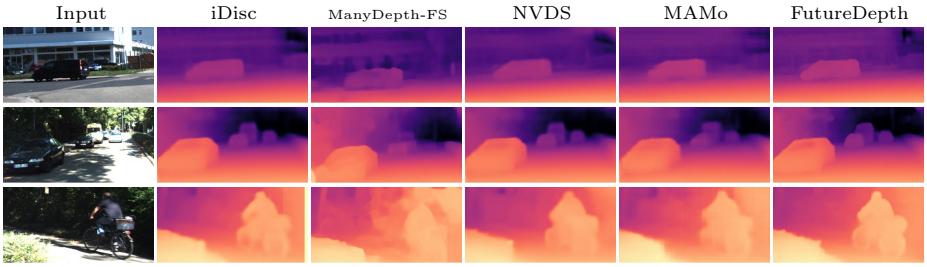

Fig. 7: Qualitative results on KITTI. FutureDepth predicts more accurate depths, for instance, for building and car (first row), far-away cars and van (second row), and biker (third row).

running inferences. We follow the same setting introduced by [33], where images are cropped to 870×1920 and the maximum depth is set to 150 meters.

Sintel [4]. MPI Sintel [4] consists of 23 synthetic sequences of open source animated films and captures open-domain scenarios. Following the protocol introduced by [19, 35], we conduct zero-shot evaluation of our proposed FutureDepth and compare with state-of-the-art video depth estimation methods, which assesses model generalizability.

3.3 Main Evaluation Results

On NYUDv2. In Table 1 (left), we evaluate our proposed FutureDepth approach and compare with existing SOTA on NYUDv2. It can be seen that FutureDepth outperforms latest existing SOTA video depth estimation models (e.g., MAMo, NVDS) and sets the new SOTA accuracy. In particular, we reduce the depth error (in terms of Abs Rel) by more than 12% when comparing to NVDS and MAMo. It is also noteworthy that we improve temporal consistency (measured by OPW) by more than 16%.

On KITTI. The depth estimation accuracy results and comparison with existing methods are presented in Table 2. Our proposed FutureDepth approach sets the new SOTA accuracy on KITTI, outperforming both latest monocular methods, e.g., iDisc [33], and video depth methods, e.g., [52]. For ManyDepth [49]

Table 2: Quantitative results on KITTI (Eigen split). † indicates methods using multiple networks to estimate depth. The methods are ordered in each group based on Abs Rel. MF means multi-frame methods. SF means single-frame methods.

Type	Method	Encoder	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE_{log} ↓	$\delta < 1.25$ ↑
SF	AdaBins [3]	EfficientNet	0.058	0.190	2.360	0.088	0.964
	BinsFormer [23]	Swin-L	0.052	0.151	2.098	0.079	0.975
	NeWCRFs [53]	Swin-L	0.052	0.155	2.129	0.079	0.974
	PixelFormer [1]	Swin-L	0.051	0.149	2.081	0.077	0.976
	iDisc [33]	Swin-L	0.050	0.145	2.067	0.077	0.977
	GEDepth [51]	[22]	0.048	0.142	2.050	0.076	0.976
MF	FlowGRU [10]	[10]	0.112	0.700	4.260	0.184	0.881
	RDE-MV [32]	ResNet18†	0.111	0.821	4.650	0.187	0.821
	STAD [20]	[24]†	0.109	0.594	3.312	0.153	0.889
	Patil *et.al.* [32]	ConvLSTM†	0.102	–	4.148	–	0.884
	ST-CLSTM [54]	ResNet18	0.101	–	4.137	–	0.890
	NeuralRGB [24]	CNN-based†	0.100	–	2.829	–	0.931
	Cao *et.al.* [6]	–	0.099	–	3.832	–	0.886
	FMNet [47]	ResNeXt-101	0.099	–	3.744	0.129	0.888
	Flow2Depth [50]	[28]†	0.081	0.488	3.651	0.146	0.912
	TC-Depth-FS [36]	ResNet50	0.071	0.330	3.222	0.108	0.922
	ManyDepth-FS [49]	ResNet50	0.069	0.342	3.414	0.111	0.930
	ManyDepth-FS [49]	Swin-L	0.060	0.248	2.747	0.099	0.955
	NVDS [48]	DPT-L [34]	0.052	0.159	2.101	0.077	0.976
	MAMo [52]	Swin-L	0.049	0.130	1.989	0.072	0.977
MF (Ours)	Baseline	ResNet34	0.063	0.219	2.521	0.098	0.957
		Swin-B	0.055	0.162	2.163	0.082	0.973
		Swin-L	0.053	0.154	2.094	0.079	0.975
		Dinov2 (ViT-L)	0.051	0.141	2.064	0.076	0.979
	FutureDepth	ResNet34	0.054	0.179	2.016	0.087	0.965
		Swin-B	0.049	0.129	1.998	0.077	0.976
		Swin-L	0.044	0.119	1.920	0.068	0.983
		Dinov2 (ViT-L)	**0.041**	**0.117**	**1.856**	**0.066**	**0.984**

and TC-Depth [36], since the original models are trained in a self-supervised setting, we use the numbers from their fully-supervised versions retrained in [52], referring them as ManyDepth-FS and TC-Depth-FS.

Fig. 7 shows a visual comparison of the depths predicted by FutureDepth and existing SOTA methods. It can be seen that our predicted depth is more accurate and captures more details of the scene. For instance, sharp depth boundaries are predicted for the biker in the last example, even though this is a challenging case where the contrast between the biker and the background is low.

In addition, we evaluate the temporal consistency in Table 3. It can be seen that the temporal consistency of FutureDepth is significantly better than existing monocular and video methods. Fig. 8 shows a visual comparison of predicted depths on consecutive frames by FutureDepth and SOTA video depth estimation methods. We see that the depth predictions by MAMo and NVDS are inconsistent and noisy across frames, whereas our prediction is more temporally consistent and accurate.

We further measure the average model runtime. We see that our proposed FutureDepth is significantly more efficient as compared to existing video depth estimation models, and has comparable or better runtime as compared to SOTA monocular models.

On DDAD. Table 4 shows the depth prediction results of models trained and evaluated on DDAD. All methods are trained using Swin-L or DPT-L encoder

Table 3: Temporal consistency and runtime on KITTI. All models use Swin-L as the encoder except for NVDS which uses DPT-L. Inference times are computed using NVIDIA RTX-3080 GPU with 11GB memory.

Type	Method	rTC ↑	aTC ↓	OPW ↓	Runtime (ms) ↓
SF	NeWCRFs	0.914	0.116	0.501	28
	iDisc	0.923	0.108	0.486	61
	GEDepth	0.919	0.133	0.441	177
MF	Many-Depth-FS	0.920	0.111	0.497	488
	TC-Depth-FS	0.901	0.122	0.516	376
	NVDS	0.951	0.096	0.356	930
	MAMo	0.963	0.088	0.328	122
MF (ours)	Baseline	0.900	0.126	0.540	32
	FutureDepth	**0.988**	**0.076**	**0.281**	49

Table 4: Quantitative results on DDAD. ↑ (↓) means higher (lower) is better. Best numbers are highlighted in bold. MF (SF) means multi-frame (single-frame) methods.

Type	Method	RMSE ↓	Sq Rel↓	Abs Rel↓	OPW↓
SF	NeWCRFs [53]	10.98	2.831	0.291	0.622
	iDisc [33]	8.99	1.854	0.163	0.596
	GEDepth [51]	10.60	2.119	0.157	0.571
MF	Many-Depth-FS [49]	12.35	3.946	0.292	0.544
	TC-Depth-FS [36]	12.11	3.788	0.283	0.699
	NVDS [48]	9.24	1.995	0.174	0.496
	MAMo [52]	8.45	1.772	0.150	0.464
MF (ours)	Baseline	11.36	3.216	0.232	0.681
	FutureDepth	**7.72**	**1.290**	**0.114**	**0.366**

for fair comparison. It can be seen that our proposed FutureDepth outperforms the existing SOTA methods. Additionally, we perform zero-shot testing by evaluating KITTI-trained models on DDAD and observe that FutureDepth has better generalizability as compared to existing models. The zero-shot testing results are included in the supplementary.

On Sintel. Table 1 (right) shows zero-shot evaluation results on Sintel. We see that FutureDepth significantly outperforms existing SOTA video depth estimation models, with better accuracy and temporal consistency. This demonstrates the strong generalization ability of our proposed FutureDepth approach.

3.4 Ablation Study

We conduct comprehensive experiments to investigate different components in our proposed FutureDepth pipeline. Table 5 summarizes the results. We assess two baselines: (1) single-frame (SF) baseline that operates on single input frames and uses the same encoder and decoder architecture as our main model, with channel numbers modified accordingly for the single-frame setting; (2) multi-frame (MF) baseline that is described in Sec. 3.1. We then incrementally add our proposed modules to evaluate their effects, including F-Net, R-Net with random masking and adaptive masking, and refinement network.

We can see that by naively extending a single-frame model to a multi-frame model only brings minimal gains; see the first two rows in Table 5. Using F-Net significantly improves the baseline in terms of both accuracy (e.g., 16% in Sq Rel) and temporal consistency (e.g., 42% in OPW). In addition, R-Net

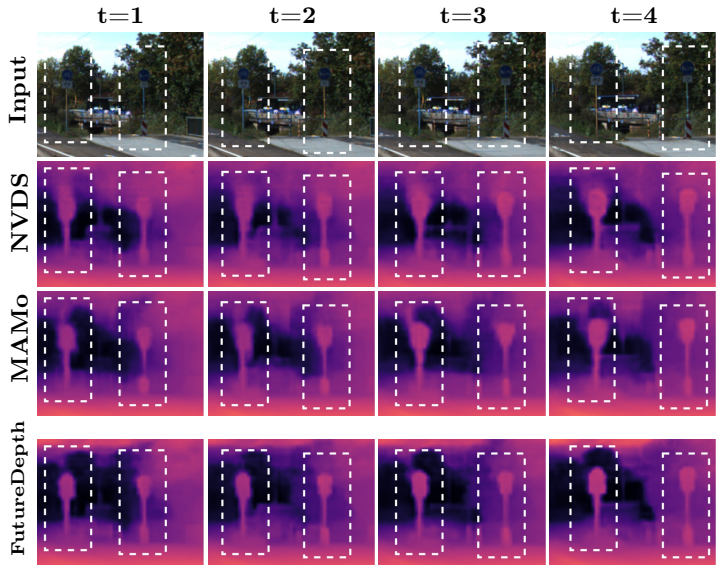

Fig. 8: Sample patches from 4 consecutive frames. FutureDepth is more temporally consistent and accurate than existing SOTA.

Table 5: Ablation study on KITTI. We use Swin-L encoder for all variants. ↑ (↓) means higher (lower) is better. AM means adaptive masking. RM means random masking.

Model	Type	R-Net	AM	F-Net	Refine	Sq Rel↓	RMSE↓	$\delta < 1.25 \uparrow$	OPW↓
Baseline	SF					0.156	2.098	0.974	0.544
Baseline	MF					0.154	2.094	0.975	0.540
				✓		0.129	1.978	0.981	0.311
		✓ (RM)				0.148	2.040	0.976	0.478
		✓	✓			0.136	1.999	0.980	0.416
		✓	✓	✓		0.122	1.931	**0.983**	0.284
FutureDepth	MF	✓	✓	✓	✓	**0.119**	**1.920**	**0.983**	**0.281**

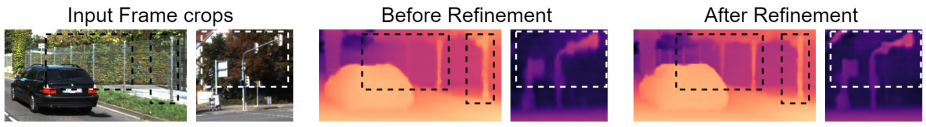

Fig. 9: Depth estimation quality is considerably improved after refinement, e.g., fence (first sample), traffic light (second sample).

provides visible improvements on top of the baseline, and together with adaptive masking brings more performance gains. Using both F-Net and R-Net jointly generates significant improvements as compared to using either of them alone. Finally, using the refinement network can further reduce prediction errors. Fig. 9 provides visual examples of how the refinement network improves the details of the predicted depth maps.

4 Related work

Monocular Depth Estimation (MDE). MDE is the task of estimating depth based on a single image. Early approaches utilize conventional or hand-crafted

features [29, 30, 37, 39, 46]. More recently, deep-learning-based methods have shown significant improvements. One general approach treats depth estimation as a continuous regression task [5, 14, 41, 53, 55]. Other works, e.g., [3, 11, 23], consider depth prediction as a classification or ordinal regression task, and put the depth values of a scene into discrete bins. While there has been extensive research on MDE, this approach inherently ignores the temporal information in video data, which is usually available in practical applications.

Video Depth Estimation (VDE). Recently, researchers have looked into utilizing multiple frames for depth estimation in a deep learning framework. [49] introduces a cost-volume-based method to leverage consecutive frames for depth estimation, which is further extended by [26] and [40] to include additional frames through cost-volume aggregation. Cost volume architectures, however, incur high computation and memory costs, making them challenging to run resource-constrained platforms and extend to more frames. Other works explore the use of recurrent neural networks, but only obtain sub-optimal accuracy [10, 32, 54]. Recently, researchers have started to adopt attentions in video depth estimation. Early attempts do not achieve SOTA performance, even when compared to the latest MDE models [6, 47]. Some researchers leverage optical flow in video depth estimation [10, 50, 52] and the latest of them [52] achieves significantly better accuracy. However, in addition to requiring optical flow estimation, [52] requires backpropagation-based feature updates on the fly, which is computationally expensive. Another latest attention-based work [48] achieves SOTA accuracy, but also incurs significant computational costs. Another line of work focuses on test-time training [19, 27]. While they can achieve temporally consistent depths by overfitting to one test video, such training-based approaches are too slow for real-time applications, computationally infeasible for resource-constrained devices, and not generalizable.

5 Conclusion

In this paper, we proposed a novel and efficient video depth estimation method, FutureDepth. Specifically, we proposed a Future Prediction Network (F-Net), which is trained using iterative future feature prediction, to capture key motion cues to enhance depth prediction. We also proposed a Reconstruction Network (R-Net), which is trained via masked auto-encoding on multi-frame features with a learnable masking. In this way, R-Net learns to identify multi-frame correspondences, which benefits depth estimation. At inference, F-Net and R-Net generates query features containing key motion and scene information, which is incorporated into the depth decoding process through cross-attention. Moreover, we use these queries in a refinement stage to further improve accuracy. Extensive results on benchmark datasets such as NYUDv2, KITTI, DDAD, and Sintel, have demonstrated the efficacy of our proposed FutureDepth method. FutureDepth sets the new state-of-the-art accuracy and at the same time, is more efficient than latest video and monocular depth models.

Acknowledge. We would like to thank Hanno Ackermann for the insightful feedback and discussion.

References

1. Agarwal, A., Arora, C.: Attention attention everywhere: Monocular depth prediction with skip attention. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 5861–5870 (January 2023) 2, 8, 11
2. ALIAS PARTH GOYAL, A.G., Sordoni, A., Côté, M.A., Ke, N.R., Bengio, Y.: Z-forcing: Training stochastic recurrent networks. Advances in neural information processing systems **30** (2017) 5
3. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4009–4018 (2021) 2, 11, 14
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Proceedings of the European Conference on Computer Vision. pp. 611–625 (2012) 3, 10
5. Cai, H., Matai, J., Borse, S., Zhang, Y., Ansari, A., Porikli, F.: X-distill: Improving self-supervised monocular depth via cross-task distillation. In: British Machine Vision Conference (2021) 14
6. Cao, Y., Li, Y., Zhang, H., Ren, C., Liu, Y.: Learning structure affinity for video depth estimation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 190–198 (2021) 2, 11, 14
7. Chiappa, S., Racaniere, S., Wierstra, D., Mohamed, S.: Recurrent environment simulators. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=B1s6xvqlx 5
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (2021) 8
9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems **27** (2014) 2, 6, 8, 9
10. Eom, C., Park, H., Ham, B.: Temporally consistent depth prediction with flow-guided memory units. IEEE Transactions on Intelligent Transportation Systems **21**(11), 4626–4636 (2019) 2, 11, 14
11. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2002–2011 (2018) 2, 14
12. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1434–1441 (2010) 1
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361 (2012). https://doi.org/10.1109/CVPR.2012.6248074 3, 9
14. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 14
15. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3D Packing for Self-Supervised Monocular Depth Estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 3, 9

16. Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. In: International conference on machine learning. pp. 2555–2565. PMLR (2019) 5
17. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104 (2023) 5
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016) 8
19. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1611–1621 (2021) 10, 14
20. Lee, H., Park, J.: Stad: Stable video depth estimation. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 3213–3217. IEEE (2021) 11
21. Li, S., Luo, Y., Zhu, Y., Zhao, X., Li, Y., Shan, Y.: Enforcing temporal consistency in video depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1145–1154 (2021) 9
22. Li, Z., Chen, Z., Liu, X., Jiang, J.: Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. arXiv preprint arXiv:2203.14211 (2022) 11
23. Li, Z., Wang, X., Liu, X., Jiang, J.: Binsformer: Revisiting adaptive bins for monocular depth estimation. arXiv preprint arXiv:2204.00987 (2022) 11, 14
24. Liu, C., Gu, J., Kim, K., Narasimhan, S.G., Kautz, J.: Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10986–10995 (2019) 11
25. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021) 8
26. Long, X., Liu, L., Li, W., Theobalt, C., Wang, W.: Multi-view depth estimation using epipolar spatio-temporal networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8258–8267 (June 2021) 14
27. Luo, X., Huang, J.B., Szeliski, R., Matzen, K., Kopf, J.: Consistent video depth estimation. ACM Transactions on Graphics (ToG) **39**(4), 71–1 (2020) 14
28. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4040–4048 (2016) 11
29. Michels, J., Saxena, A., Ng, A.Y.: High speed obstacle avoidance using monocular vision and reinforcement learning. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 593–600 (2005) 14
30. Nagai, T., Naruse, T., Ikehara, M., Kurematsu, A.: Hmm-based surface reconstruction from single images. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). vol. 2, pp. II–II. IEEE (2002) 14
31. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: Proceedings of the International Conference on Computer Vision (ICCV). pp. 2320–2327. IEEE (2011) 1

32. Patil, V., Van Gansbeke, W., Dai, D., Van Gool, L.: Don't forget the past: Recurrent depth estimation from monocular video. IEEE Robotics and Automation Letters **5**(4), 6813–6820 (2020) 2, 11, 14
33. Piccinelli, L., Sakaridis, C., Yu, F.: idisc: Internal discretization for monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 2, 10, 11, 12
34. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12179–12188 (2021) 11
35. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE transactions on pattern analysis and machine intelligence **44**(3), 1623–1637 (2020) 10
36. Ruhkamp, P., Gao, D., Chen, H., Navab, N., Busam, B.: Attention meets geometry: Geometry guided spatial-temporal attention for consistent self-supervised monocular depth estimation. In: Proceedings of the International Conference on 3D Vision (3DV). pp. 837–847 (2021) 2, 11, 12
37. Saxena, A., Chung, S., Ng, A.: Learning depth from single monocular images. Advances in Neural Information Processing Dystems **18** (2005) 14
38. Saxena, A., Schulte, J., Ng, A.Y., et al.: Depth estimation using monocular and stereo cues. In: Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI). vol. 7, pp. 2197–2203 (2007) 1
39. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(5), 824–840 (2008) 14
40. Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., Godard, C.: Simplerecon: 3d reconstruction without 3d convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022) 14
41. Shi, Y., Cai, H., Ansari, A., Porikli, F.: Ega-depth: Efficient guided attention for self-supervised multi-camera depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 119–129 (2023) 14
42. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Proceedings of the European Conference on Computer Vision (ECCV). vol. 7576, pp. 746–760 (2012) 3, 9
43. Spencer, J., Choudhury, S., Venkatraman, A., Ziebart, B., Bagnell, J.A.: Feedback in imitation learning: The three regimes of covariate shift. arXiv preprint arXiv:2102.02872 (2021) 5
44. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in Neural Information Processing Systems **35**, 10078–10093 (2022) 3
45. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.: Videomae v2: Scaling video masked autoencoders with dual masking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14549–14560 (2023) 3
46. Wang, X., Hou, C., Pu, L., Hou, Y.: A depth estimating method from a single image using foe crf. Multimedia Tools and Applications **74**, 9491–9506 (2015) 14
47. Wang, Y., Pan, Z., Li, X., Cao, Z., Xian, K., Zhang, J.: Less is more: Consistent video depth estimation with masked frames modeling. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 6347–6358 (2022) 2, 10, 11, 14

48. Wang, Y., Shi, M., Li, J., Huang, Z., Cao, Z., Zhang, J., Xian, K., Lin, G.: Neural video depth stabilizer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9466–9476 (2023) 2, 9, 10, 11, 12, 14
49. Watson, J., Mac Aodha, O., Prisacariu, V., Brostow, G., Firman, M.: The temporal opportunist: Self-supervised multi-frame monocular depth. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1164–1174 (2021) 2, 10, 11, 12, 14
50. Xie, J., Lei, C., Li, Z., Li, L.E., Chen, Q.: Video depth estimation by fusing flow-to-depth proposals. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 10100–10107 (2020) 2, 11, 14
51. Yang, X., Ma, Z., Ji, Z., Ren, Z.: Gedepth: Ground embedding for monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12719–12727 (2023) 2, 11, 12
52. Yasarla, R., Cai, H., Jeong, J., Shi, Y., Garrepalli, R., Porikli, F.: Mamo: Leveraging memory and attention for monocular video depth estimation (2023) 2, 9, 10, 11, 12, 14
53. Yuan, W., Gu, X., Dai, Z., Zhu, S., Tan, P.: Newcrfs: Neural window fully-connected crfs for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 11, 12, 14
54. Zhang, H., Shen, C., Li, Y., Cao, Y., Liu, Y., Yan, Y.: Exploiting temporal consistency for real-time video depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1725–1734 (2019) 2, 10, 11, 14
55. Zhu, J., Shi, Y., Ren, M., Fang, Y.: Mda-net: memorable domain adaptation network for monocular depth estimation. In: British Machine Vision Conference (2020) 14