Supplementary for 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

A Additional Ablation Studies

Sampling Techniques. We perform experiment to ablate different sampling techniques used in reconstruction network (R-Net) on KITTI dataset. In Table 1, we can see that the proposed adaptive sampling technique benefits the R-Net in learning better spatial and temporal representation of the feature volume $\hat{V}_{1,T}$ for given sequence of frames $I_{1,T}$. In Fig. **??** of the main paper, we can see adaptive sampling generates masks that preserve representations of important object information and exclude unnecessary back ground information, thus benefiting FutureDepth depth estimation performance. Note that in this experiment we do not use F-Net and refinement network in FutureDepth, *i.e.,* we perform experiments using Baseline (MF) + R-Net to understand the effect of sampling techniques more clearly.

Table 1: Using different masking techniques on KITTI (Eigen split) dataset. We perform this experiment using Swin-L for FutureDepth encoder. Here we set $T = 4$.

Masking Method	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE$_{log}$ ↓	$\delta < 1.25$ ↑
Random	0.051	0.148	2.040	0.077	0.976
Tube	0.050	0.146	2.035	0.077	0.977
Adaptive (ours)	0.048	0.136	1.999	0.073	0.980

Number of iteration (L) in F-Net. We perform experiment on KITTI dataset to ablate on the number of iterations (L) in the future prediction network (F-Net), which shows the impact of $Q_{motion,1,T}$ motion queries on FutureDepth performance. As shown in Table 2, as the iterations in F-Net increases, F-Net can predict the current motion and future of objects over the time frames from 1 to $L + T$ and generate beneficial $Q_{motion,1,T}$ queries that provide temporal information to FutureDepth decoder and benefits FutureDepth performance. Note, here we didn't use the refinement network in FutureDepth for this experiment.

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

Table 2: Ablation study for different number of iterations (L) in future prediction on KITTI (Eigen split) dataset. We perform this experiment using Swin-L for FutureDepth encoder. Here we set $T = 4$.

Metric	0	1	2	3	4
RMSE↓	1.999	1.976	1.962	1.944	1.931
Sq Rel↓	0.136	0.136	0.130	0.124	0.122
OPW↓	0.416	0.398	0.342	0.303	0.281

Number of refinement steps (N). We perform experiment on KITTI dataset to ablate on the number of refinement steps (N), in order to analyze the impact of refinement network using $Q_{scene,1,T}$ and $Q_{motion,1,T}$ queries on FutureDepth performance. The result is shown in Table 3.

Table 3: Ablation study for different of refinement steps (N) on KITTI (Eigen split) dataset. We perform this experiment using Swin-L for FutureDepth encoder. Here we set $L = 4$ and $T = 4$.

Metric	0	1	2	3
RMSE↓	1.931	1.924	1.922	1.920
Sq Rel↓	0.122	0.120	0.119	0.119

Comparison with auto-regressive. Existing auto-regressive methods require teacher forcing, which limits the model's motion understanding. For a concrete comparison, we replace F-Net with a ConvLSTM (from [7] and using their temporal loss). Table 4 shows that F-Net is more accurate, consistent, and efficient.

Table 4: Comparing our F-Net and auto-regressive method on KITTI.

Metric	Base+R-Net	Base+R-Net+ConvLSTM	Base+R-Net+F-Net
Sq Rel↓	0.136	0.133	0.122
OPW↓	0.416	0.394	0.311
GFlops↓	296	710	342

Number of frames (T). We perform experiment on KITTI dataset to ablate on the number of frames T in a video/multi-frame sequence. Here, Table 5 shows the performance of FutureDepth when processing T video frames in a batch simultaneously.

A.1 What happens when the adaptive sampler is used during inference?

We perform experiment on KITTI dataset to study the benefits of adaptive sampler during inference time. Note that we train FutureDepth with adaptive sampler where we choose the masking ratio from $r \in [0.6, 0.9]$ to train R-Net. Here,

Table 5: Ablation study on different of numbers of frames (T) in video sequence on KITTI (Eigen split) dataset. We perform this experiment using Swin-L for FutureDepth encoder. Here we set $L = 4$.

Metric	3	4	6	8
RMSE↓	1.932	1.920	1.906	1.911
Sq Rel↓	0.122	0.119	0.114	0.116

we set $L = 4$, $T = 4$, and $N = 3$ for FutureDepth training and inference. During inference we ablate on different values of masking ratio ($r = 0, 0.2, 0.4, 0.6, 0.9$) for adaptive sampler in R-Net. Table 6 demonstrates that the inference-time utilization of adaptive sampling in R-Net can benefit FutureDepth by assigning significance to critical attentions and enhancing the feature volume ($\hat{V}_{1,T}$), which is subsequently processed by the FutureDepth decoder.

Table 6: Ablation study on different masking ratios of adaptive sampler during inference on KITTI (Eigen split) dataset. We perform this experiment using Swin-L for FutureDepth encoder. Here we set $L = 4$, $T = 4$ and $N = 3$.

Metric	0.0	0.2	0.4	0.6	0.9
RMSE↓	1.920	1.906	1.892	1.911	1.956
Sq Rel↓	0.119	0.114	0.108	0.111	0.133

B Zero-Shot Evaluation

In Table 7, we assess the performance of KITTI-trained models on DDAD to evaluate their generalization capabilities. The results indicate that our proposed FutureDepth surpasses existing state-of-the-art methods. The evaluation of KITTI-trained models on DDAD demonstrates that FutureDepth exhibits superior generalizability compared to other models.

C Limitations

There are a few aspects of video depth estimation not addressed in this work, which can be interesting for future research. For instance, we do not propose specific treatment for cases where an object becomes occluded and then reappears in a set of consecutive frames. Proper handling of occlusion can lead to better motion and correspondence understanding and as a result, more accurate depth estimation.

D Training and Infernece Algorithms

Algorithm 1 outlines the steps involved in the pre-training phase of FutureDepth, while Algorithm 2 details the steps of the main training phase of FutureDepth.

Table 7: Quantitative results on DDAD dataset for distances up to 150 meters. The input frame resolution is 1216×1936.

Method	Encoder	Sq Rel↓	RMSE↓	$\delta < 1.25 \uparrow$
ManyDepth-FS [12]	ResNet50	5.471	16.123	0.744
ManyDepth-FS [12]	Swin-L	4.211	13.899	0.784
TC-Depth-FS [8]	ResNet50	5.285	15.121	0.777
AdaBins [3]	[9]	4.950	15.228	0.780
AdaBins [1]	[13]	4.791	14.595	0.789
NeWCRFs [15]	Swin-L	4.041	11.956	0.816
PixelFormer [2]	Swin-L	4.474	12.467	0.802
MAMo [14]	Swin-L	3.349	11.094	0.870
Baseline (ours)	Swin-L	4.506	12.841	0.804
FutureDepth (ours)	Swin-L	2.960	10.016	0.833

Additionally, Algorithm 3 provides the steps for evaluating FutureDepth during inference.

Algorithm 1 Pretraining of FutureDepth

Input: Train dataset $\mathcal{D}$ which consists of training videos and corresponding ground truth depths. Training video or sequence of frames, $I_{1,T} = \{I_1, ..., I_T\}$ and $D^{gt}_{1,T} = \{D^{gt}_0, ..., D^{gt}_T\}$
Model: $en(\cdot)$: encoder of FutureDepth, $de(\cdot)$: decoder of FutureDepth, $g(\cdot)$: reconstruction network and $h(\cdot)$: future prediction network, $rf(\cdot)$: refinement network, $s(\cdot)$: adaptive sampler
pretraining $en(\cdot)$, $de(\cdot)$
for epoch = 1→5 **do**
  for $I_{1,T}, D^{gt}_{1,T} \in \mathcal{D}$ **do**
    $D_{1,T} = de(en(I_{1,T}))$
    SILogLoss $(D_{1,T}, D^{gt}_{1,T})$
    Update parameters of $en(\cdot)$, $de(\cdot)$
  end for
end for
pretraining reconstruction network $g(\cdot)$######
for epoch = 1→3 **do**
  freeze $en(\cdot)$ weights
  for $I_{1,T}, D^{gt}_{1,T} \in \mathcal{D}$ **do**
    $V_{1,T} = en(I_{1,T})$
    generate random mask $M_{1,T}$
    $\hat{V}_{1,T} = g(M_{1,T} \odot V_{1,T})$
    L2-loss between $V_{1,T}$ and $\hat{V}_{1,T}$
    Update parameters of $g(\cdot)$
  end for
end for

Algorithm 2 Main Training of FutureDepth

Input: Train dataset $\mathcal{D}$ which consists of training videos and corresponding ground truth depths. Training video or sequence of frames, $I_{1,T} = \{I_1, ..., I_T\}$ and $D^{gt}_{1,T} = \{D^{gt}_0, ..., D^{gt}_T\}$
Model: $en(\cdot)$: encoder of FutureDepth, $de(\cdot)$: decoder of FutureDepth, $g(\cdot)$: reconstruction network and $h(\cdot)$: future prediction network, $rf(\cdot)$: refinement network, $s(\cdot)$: adaptive sampler
Training FutureDepth network
initialize weights of $h(\cdot)$ with $g(\cdot)$
for every epoch **do**
  for $I_{1,T}, D^{gt}_{1,T} \in \mathcal{D}$ **do**

---

step-1 updating $h(\cdot)$, $s(\cdot)$, $g(\cdot)$ weights###
freeze $en(\cdot)$, $de(\cdot)$ weights
$V_{1,T} = en(I_{1,T})$; $M_{1,T} = s(V_{1,T})$
$\tilde{V}_{1,T} = V_{1,T}$
for i=1→ L **do**
    $\tilde{V}_{i+1,i+T} = h(\tilde{V}_{i,i+T-1})$
end for
$\hat{V}_{1,T}$, $Q_{scene,1,T} = g(M_{1,T} \odot V_{1,T})$
$Q_{all,1,T}$ = cross-attn($Q_{scene,1,T}, Q_{motion,1,T}$)
$D_{1,T} = de(\hat{V}_{1,T}, Q_{all,1,T})$
compute loss $\mathcal{L}_F$ in Eq. 1 (main paper)
compute loss $\mathcal{L}_A$ (refer section 2.2)
compute loss $\mathcal{L}_R$ in Eq. 2 (main paper)
Update parameters of $h(\cdot)$, $s(\cdot)$, $g(\cdot)$

---

step-2 updating FutureDepth's $en(\cdot)$, $de(\cdot)$, $rf(\cdot)$ weights###
freeze $s(\cdot)$, $g(\cdot)$, $h(\cdot)$ weights
$V_{1,T} = en(I_{1,T})$
$\hat{V}_{1,T}$, $Q_{scene,1,T} = g(V_{1,T})$
get $Q_{motion,1,T}$ from future prediction F-Net $h(\cdot)$
$Q_{all,1,T}$ = cross-attn($Q_{scene,1,T}, Q_{motion,1,T}$)
$D^0_{1,T} = de(\hat{V}_{1,T}, Q_{all,1,T})$
for i=1→ N **do**
    $D^i_{1,T} = rf(D^{i-1}_{1,T}, Q_{all,1,T})$
end for
compute loss $\mathcal{L}_{D,final}$ in Eq. 3 (main paper)
Update parameters of FutureDepth 's $en(\dot{)}$, $de(\cdot)$, $rf(\cdot)$ weights
  end for
end for

E Details on FutureDepth Networks

Fig. 1 shows the detailed network architecture of FutureDepth. PPM head used in FutureDepth encoder is similar to [15]. Note, cross-attention and self-attention layers used in FutureDepth are similar to [10]. For example we use [10] cross-attention layer to perform cross-attention between Q_{scene} and Q_{motion} to pro-

Algorithm 3 Inference of FutureDepth

Input: Test dataset $\mathcal{D}^{test}$ which consists of inference videos. Inference video or sequence of frames, $I_{1,T} = \{I_1, ..., I_T\}$
Model: $en(\cdot)$: encoder of FutureDepth , $de(\cdot)$: decoder of FutureDepth , $g(\cdot)$: reconstruction network and $h(\cdot)$: future prediction network, $rf(\cdot)$: refinement network, $s(\cdot)$: adaptive sampler
######Inference FutureDepth network ######
for $I_{1,T} \in \mathcal{D}^{test}$ **do**
  $V_{1,T} = en(I_{1,T})$
  $\hat{V}_{1,T}$, $Q_{scene,1,T} = g(V_{1,T})$
  get $Q_{motion,1,T}$ from future prediction F-Net $h(\cdot)$
  $Q_{all,1,T}$ = cross-attn($Q_{scene,1,T}, Q_{motion,1,T}$)
  $D^0_{1,T} = de(\hat{V}_{1,T}, Q_{all,1,T})$
  for i=1→ N **do**
    $D^i_{1,T} = rf(D^{i-1}_{1,T}, Q_{all,1,T})$
  end for
  $D_{1,T} = \hat{D}^N_{1,T}$
end for

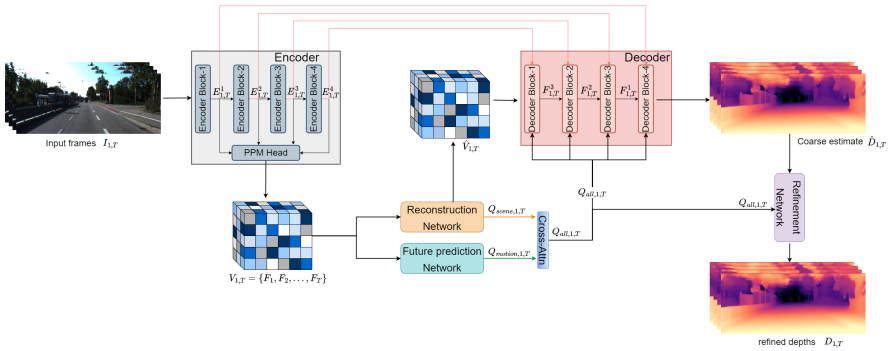

Fig. 1: Our proposed FutureDepth method.

duce Q_{all}. Fig. 2 shows the decoder block used in FutureDepth decoder. Fig. 2 clearly show Q_{all} queries which contains critical scenes and temporal cues are utilized by FutureDepth decoder. Fig. 3 shows overview architecture of R-Net. The mask generator consists of two 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. Fig 4 shows the overview of refinement network, where we utilize Q_{all} and improve the coarse estimate depths predicted by FutureDepth decoder. Refinement network contains a self-attention layer and a cross-attention layers as shown in Fig. 4, where we first perform self-attention between the initial coarse depth predictions $D^0_{1,T}$, which are further cross-attended with the Q_{all} to obtain $D^1_{1,T}$. We perform this progressive refinement process for N steps to obtain final depth prediction $D_{1,T}$ $(=D^N_{1,T})$.

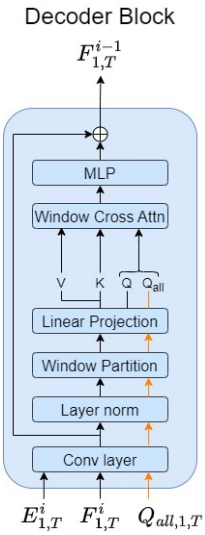

Fig. 2: Skip attention module used as a building block for each decoder layer in FutureDepth .

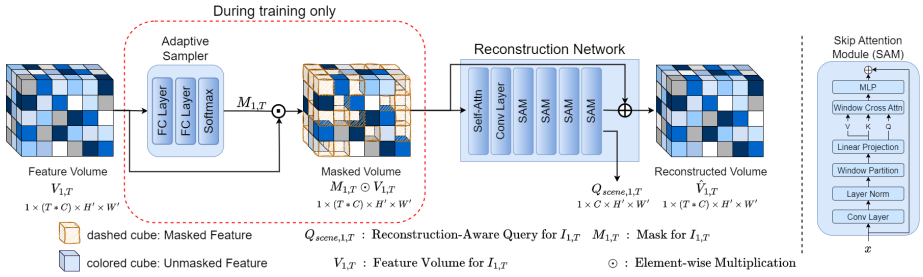

Fig. 3: Reconstruction network (R-Net) in our proposed FutureDepth framework.

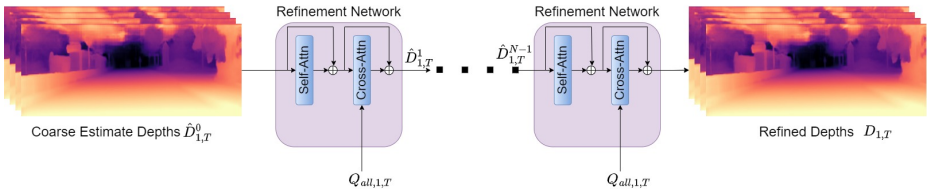

Fig. 4: Overview of refinement network.

F Details on Evaluation Metrics

We follow [4] to use the following metrics to evaluate the performance of predicted depth outputs of different methods,

$$\begin{aligned}
&\text{Abs Relative: } \frac{1}{\sum(K_t==1)} \sum_{k_t \in K, d_t \in D_t} k_t \| \frac{d_t - d_t^{gt}}{d_t^{gt}} \| \\
&\text{Squared Relative: } \frac{1}{\sum(K_t==1)} \sum_{k_t \in K, d_t \in D_t} \frac{\| d_t - d_t^{gt} \|^2}{d_t^{gt}} \\
&\text{RMSE (linear)} : \sqrt{\frac{1}{\sum(K_t==1)} \sum_{k_t \in K, d_t \in D_t} \| d_t - d_t^{gt} \|^2} \\
&\text{RMSE (log): } \sqrt{\frac{1}{\sum(K_t==1)} \sum_{k_t \in K, d_t \in D_t} \| \log d_t - \log d_t^{gt} \|^2} \\
&\delta < \text{thr: } \frac{1}{\sum(K_t==1)} K_t \left[\text{Max} \left(\frac{D_t}{D_t^{gt}}, \frac{D_t^{gt}}{D_t} \right) < \text{thr} \right]
\end{aligned} \tag{1}$$

where K_t is a depth validity mask, D_t is predicted depth for image I_t and D_t^{gt} is ground-truth depth.

For evaluating temporal consistency, [6] introduces the following metrics,

$$\begin{aligned}
aTC_t &= \frac{1}{\sum(K_t == 1)} K_t \| \frac{D_t - D_t^w}{D_t} \|, \\
rTC_t &= \frac{1}{\sum(K_t == 1)} K_t \left[\text{Max} \left(\frac{D_t}{D_t^w}, \frac{D_t^w}{D_t} \right) < \text{thr} \right],
\end{aligned} \tag{2}$$

where K_t is a depth validity mask, D_t is predicted depth for I_t and D_t^w is warped from D_{t-1} using optical flow. Following the protocol introduced by [14] we use the optical flow generated by the latest SOTA FlowFormer [5]. Optical flow based warping metric (OPW) is introduced by [11],

$$\begin{aligned}
OPW_t &= \frac{1}{n} \sum_{i=1}^{n} W_{t+1 \Rightarrow t}^{(i)} \left\| D_{t+1}^{(i)} - \hat{D}_t^{(i)} \right\|_1 \\
OPW &= \sum_{t=0}^{T-1} OPW_t,
\end{aligned} \tag{3}$$

where, $W_{t+1 \Rightarrow t}^{(i)}$ is optical flow based visibility mask calculated from the warping discrepancy between subsequent frames as explained in [11].

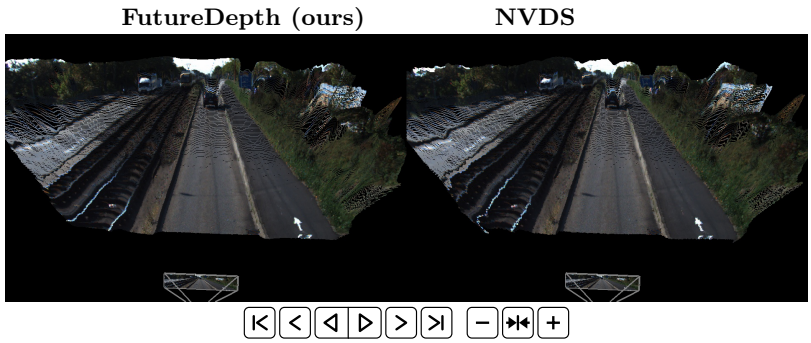

Fig. 5: Point cloud visualization of predicted depths by our proposed FutureDepth (left) and NVDS (right) on KITTI, respectively. This can be viewed as a video or frame-by-frame in the (free) Adobe Acrobat Reader.

G Additional Qualitative Results

G.1 More Qualitative Temporal Consistency Comparisons

Fig. 7 and 8 show visual comparisons 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.

G.2 More Visualization Results of Q_{scene}

Fig. 6 shows samples Q_{scene} generated by R-net. We can clearly observe that important fore-ground and back-ground objects are captured as queries in $Q_{scene,1,T}$ that helps FutureDepth decoder in computing high quality depth maps.

G.3 Point cloud visualization

Fig. 5 shows that our depths lead to better visual quality and consistency of the 3D point cloud, e.g., smoother and straighter railway tracks.

G.4 More Qualitative Results on Depth Estimation Quality

Fig. 9, 10, and 11 provide more visual comparisons on depth quality between FutureDepth and existing SOTA methods. We see that our depth predictions are more accurate and better capture fine details.

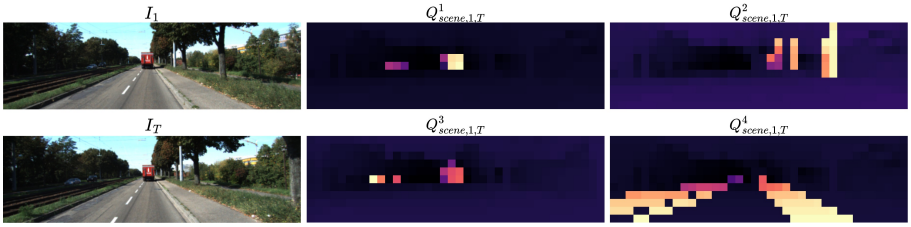

Fig. 6: Sample Q_{scene} generated by R-net. We show four sample channels in $Q_{scene,1,T}$ for input frames $I_{1,T}$ ($T = 4$). We can clearly observe that important fore-ground and back-ground objects are captured as queries in $Q_{scene,1,T}$ that helps FutureDepth decoder in computing high quality depth maps.

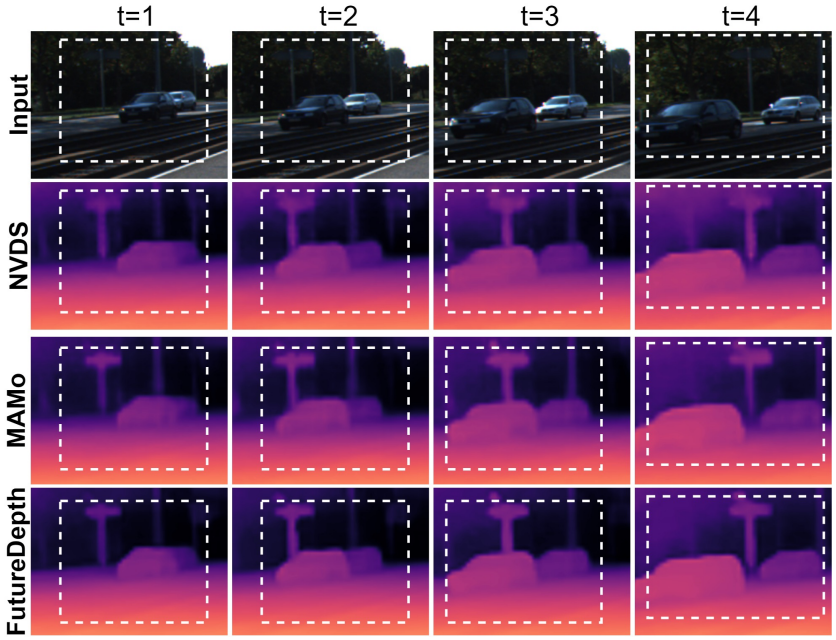

Fig. 7: Sample patches from 4 consecutive frames. FutureDepth is more temporally consistent and accurate than existing SOTA (e.g., see that depths over the traffic sign are less accurately predicted in the cases of NVDS and MAMo, in the 4th frame after it is unoccluded).

References

1. Agarwal, A., Arora, C.: Depthformer: Multiscale vision transformer for monocular depth estimation with global local information fusion. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 3873–3877 (2022) 4
2. 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) 4

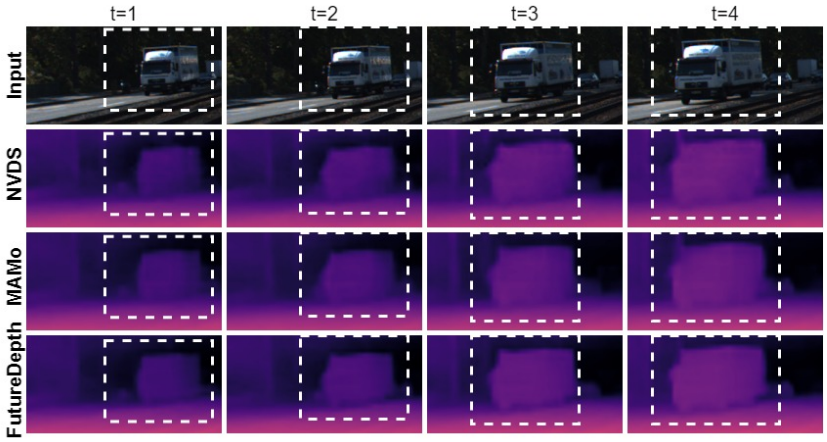

Fig. 8: Sample patches from 4 consecutive frames. FutureDepth is more temporally consistent and accurate than existing SOTA (e.g., see that the front of the truck is more blurry in the cases of NVDS and MAMo).

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) 4
4. 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) 8
5. Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K.C., Qin, H., Dai, J., Li, H.: Flowformer: A transformer architecture for optical flow. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII. pp. 668–685. Springer (2022) 8
6. 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) 8
7. 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
8. 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) 4
9. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 6105–6114. PMLR (2019) 4
10. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020) 5
11. 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) 8
12. Watson, J., Mac Aodha, O., Prisacariu, V., Brostow, G., Firman, M.: The temporal opportunist: Self-supervised multi-frame monocular depth. In: Proceedings of the

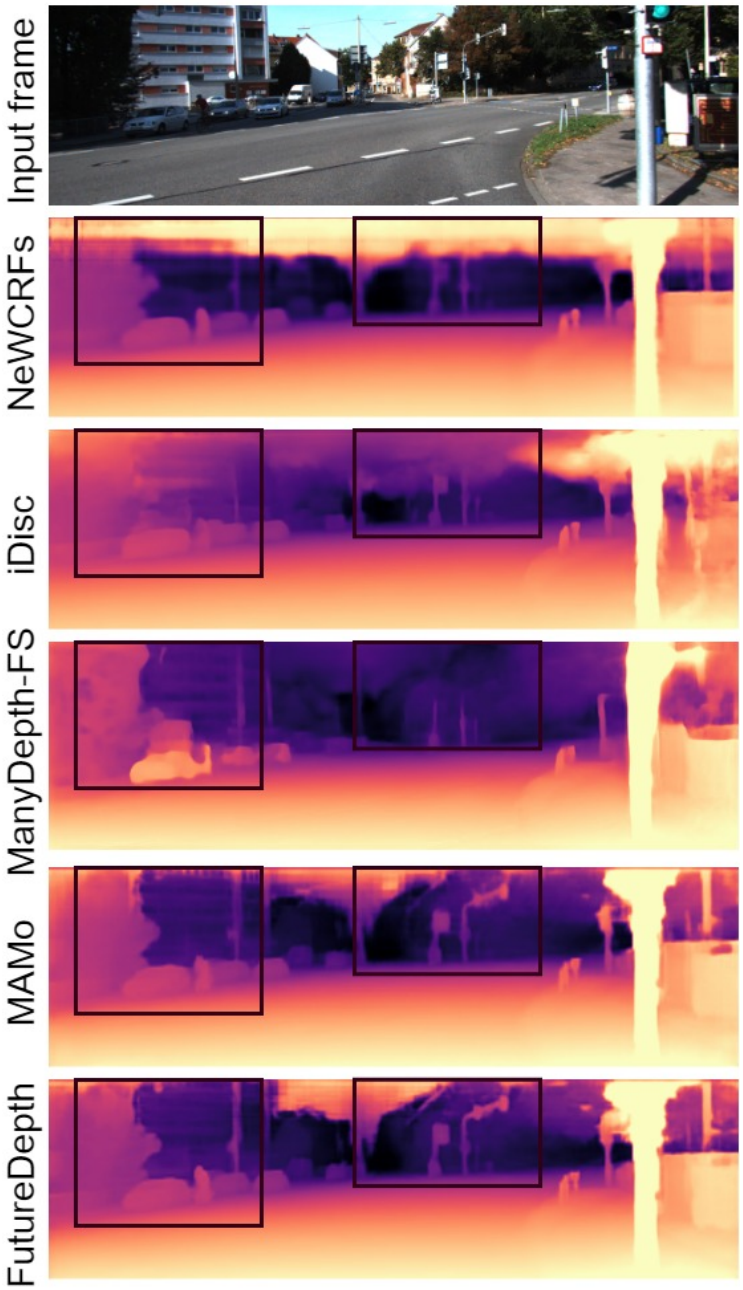

Fig. 9: Qualitative results on KITTI.

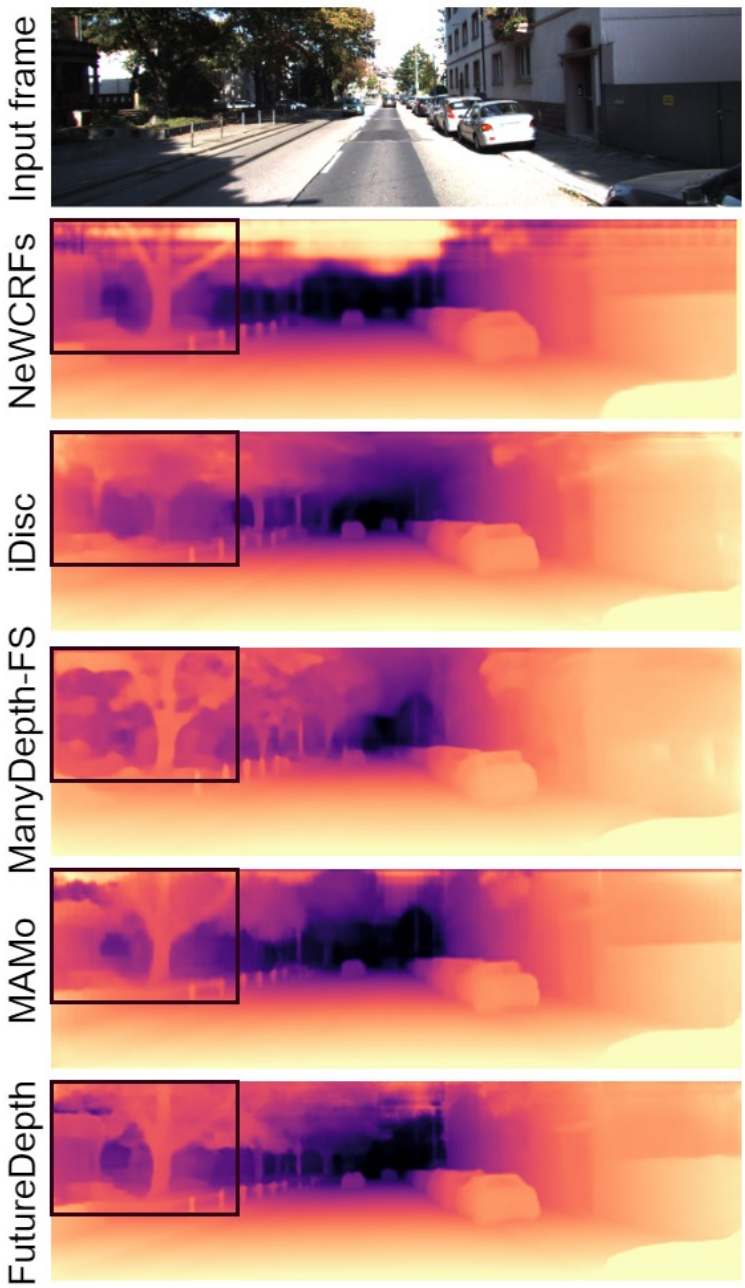

Fig. 10: Qualitative results on KITTI.

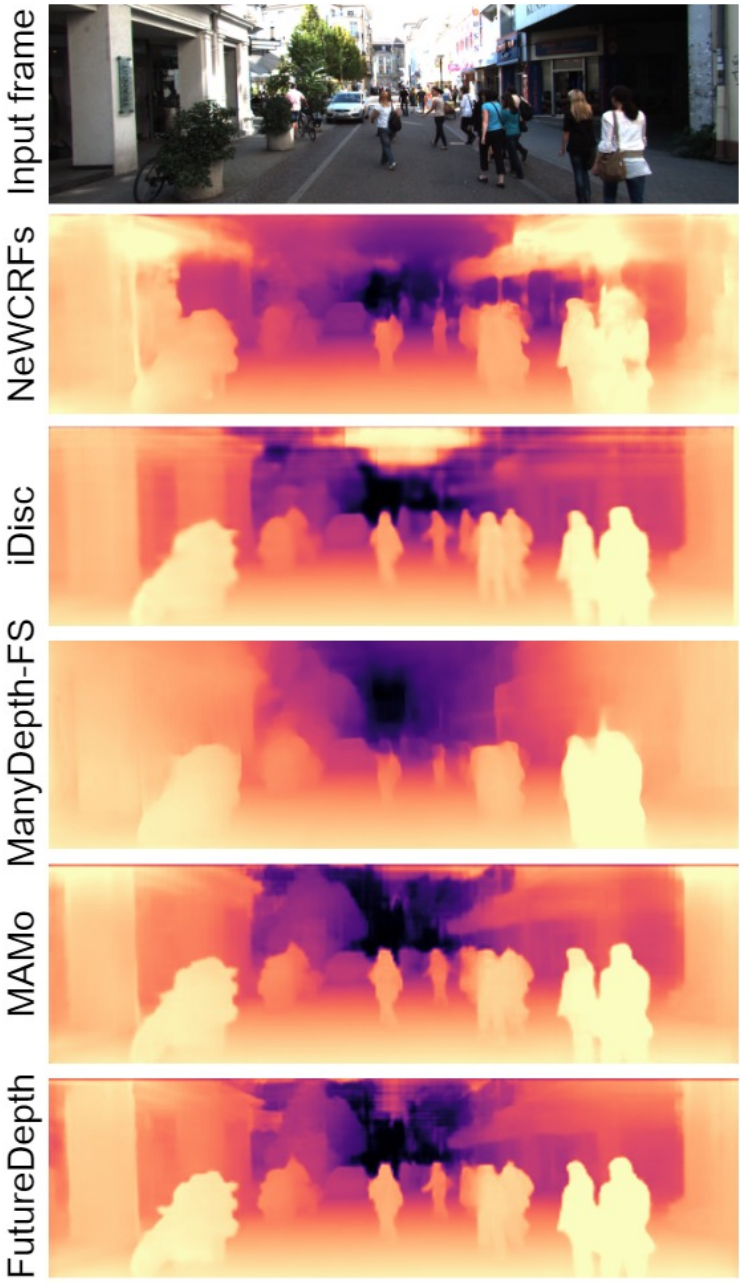

Fig. 11: Qualitative results on KITTI.

13. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021) 4
14. Yasarla, R., Cai, H., Jeong, J., Shi, Y., Garrepalli, R., Porikli, F.: Mamo: Leveraging memory and attention for monocular video depth estimation (2023) 4, 8
15. 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) 4, 5