Buffer Anytime: Zero-Shot Video Depth and Normal from Image Priors

Supplementary Material

1. More Video Results

In addition to the qualitative comparisons in the paper, we provide more animated results in our website for better visualization of the prediction quality.

2. More Implementation Details

All models are implemented in PyTorch [?]. We utilize the official implementations of Depth Anything V2 [?] and Marigold-E2E-FT [?], adapting temporal blocks from the UnetMotion architecture in the Diffusers [?] library. Experiments are conducted on NVIDIA H100 GPUs with 80GB memory. Due to memory constraints, we limit the maximum sequence length to 110 frames for depth estimation and 32 frames for normal estimation.

For training, we use a dataset of approximately 200K videos, with each clip containing 128 frames. We employ the AdamW [?] optimizer with learning rates of 10^{-4} and 10^{-5} for depth and normal estimation, respectively. Training begins with a 1,000-step warm-up phase, during which the learning rate increases linearly from 0 to its target value. The training process runs on 24 H100 GPUs with a total batch size of 24 and incorporates Exponential Moving Average (EMA) with a decay coefficient of 0.999. The complete training cycle requires approximately one day to complete 15,000 iterations.

2.1. Details of the Deferred Back-Propagation

In our normal model, we employ deferred back-propagation as proposed by Zhang et al. [?] to reduce memory consumption. Algorithm 1 outlines the detailed implementation steps. Notably, the gradients obtained by back-propagating $\mathcal{L}_{def}$ are equivalent to those computed from the pixel-wise loss function $\mathcal{L}_{pix}$ across all decoded frames:

$$\frac{\partial \mathcal{L}_{def}}{\partial \theta} = \frac{\partial \frac{1}{K} \sum_k \mathtt{Sum}(\mathtt{SG}(\boldsymbol{g_k}) \cdot \boldsymbol{z_k})}{\partial \theta} \tag{1}$$

$$= \frac{1}{K} \sum_k \boldsymbol{g_k} \cdot \frac{\partial \boldsymbol{z_k}}{\partial \theta} \tag{2}$$

$$= \frac{1}{K} \sum_k \frac{\partial \mathcal{L}_{pix}(\mathcal{D}(\boldsymbol{z}_k))}{\partial \boldsymbol{z}_k} \cdot \frac{\partial \boldsymbol{z_k}}{\partial \theta} \tag{3}$$

$$= \frac{1}{K} \frac{\partial \sum_k \mathcal{L}_{pix}(\mathcal{D}(\boldsymbol{z}_k))}{\partial \theta} \tag{4}$$

2.2. Details of the Optical Flow Based Stabilization

Algorithm 2 presents the pseudo-code for our optical flow based stabilization loss calculation. The loss is computed separately for forward optical flow (previous frame to next frame) and backward flow (next frame to previous frame), then combined together. This stabilization algorithm is applied to both depth and normal models. In our experiments, we set the threshold τ_c to $\frac{\log 2}{2} = 0.34$.

```
Algorithm 1: Deferred Back-Propagation
  Parameter: Trained model $f_\theta$, image decoder $\mathcal{D}$,
             frame number $K$, chunk size $C$,
  Input: Input frames $\boldsymbol{I}_{1,\dots,K}$, loss function defined
         on the decoded frames $\mathcal{L}_{pix}$.
  Output: Deferred back-propagation loss $\mathcal{L}_{def}$
  $\mathcal{L}_{def} \leftarrow 0$;
  $\boldsymbol{z}_{1,\dots,K} \leftarrow f_\theta(\boldsymbol{I}_{1,\dots,K})$;
  for ch in Range(start=1, end=K, step=C) do
      /* Generate chunk prediction   */
      $\boldsymbol{z}^{ch} \leftarrow \boldsymbol{z}_{ch,\dots,ch+C-1}$;
      $\boldsymbol{\mathcal{G}}^{ch} \leftarrow \mathcal{D}(\boldsymbol{z}^{ch})$;
      /* Loss on decoded frames      */
      $l \leftarrow \mathcal{L}_{pix}(\boldsymbol{\mathcal{G}}^{ch})$;
      $\boldsymbol{g}^{ch} \leftarrow \mathtt{Autograd}(l, \boldsymbol{z}^{ch})$;
      /* SG means stop gradient      */
      $\mathcal{L}_{def} \leftarrow \mathcal{L}_{def} + \frac{1}{K}\mathtt{Sum}(\mathtt{SG}(\boldsymbol{g}^{ch}) \cdot \boldsymbol{z}^{ch})$ ;
  end
  return $\mathcal{L}_{def}$
```

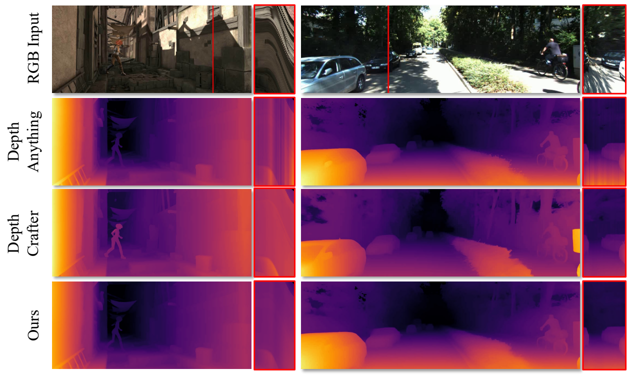

Figure 1. **Additional qualitive comparison**. We show comparison between our model and Depth Anything V2 [?], DepthCrafter [?] in the Sintel [?] dataset (*Left*) and the KITTI [?] dataset (*Right*).

3. Additional Qualitative Comparison

In Fig. 1, we show additional qualitative comparisons in the benchmark datasets used in the quantitative experiment (i.e. Tab. 1) in the paper.

Algorithm 2: Calculating Stabilization Loss

```
Parameter: Video optical flow model $\mathcal{O}$, frame number $K$, cycle-validation threshold $\tau_c$
Input: Predicted geometric buffers $\mathcal{G}^{pred}_{1,\ldots,K}$, input frames $\boldsymbol{I}_{1,\ldots,K}$
Output: Stabilization loss $\mathcal{L}_{stable}$

/* Calculate Optical Flow Maps                                                        */
$\boldsymbol{O}_{fwd} \leftarrow \mathcal{O}(\texttt{src} = \boldsymbol{I}^{pred}_{1,\ldots,K-1}, \texttt{dst} = \boldsymbol{I}^{pred}_{2,\ldots,K})$ ;      /* Shape: $(K-1) \times 2 \times H \times W$ */
$\boldsymbol{O}_{bwd} \leftarrow \mathcal{O}(\texttt{src} = \boldsymbol{I}^{pred}_{2,\ldots,K}, \texttt{dst} = \boldsymbol{I}^{pred}_{1,\ldots,K-1})$ ;      /* Shape: $(K-1) \times 2 \times H \times W$ */

/* Calculate Cycle-Validation Masks                                                   */
$\mathcal{M}^{cyc}_{fwd} \leftarrow \texttt{Where}_{\boldsymbol{x} \in \boldsymbol{I}_{2,\ldots,K}}(||\boldsymbol{O}_{fwd}(\boldsymbol{O}_{bwd}(\boldsymbol{x})) - \boldsymbol{x}||_2 < \tau_c)$ ;      /* Shape: $(K-1) \times H \times W$ */
$\mathcal{M}^{cyc}_{bwd} \leftarrow \texttt{Where}_{\boldsymbol{x} \in \boldsymbol{I}_{1,\ldots,K-1}}(||\boldsymbol{O}_{bwd}(\boldsymbol{O}_{fwd}(\boldsymbol{x})) - \boldsymbol{x}||_2 < \tau_c)$ ;      /* Shape: $(K-1) \times H \times W$ */

/* Calculate Edge-Based Masks                                                         */
$\boldsymbol{E} \leftarrow \texttt{CannyEdge}(\mathcal{G}^{pred}_{1,\ldots,K})$ ;      /* Shape: $K \times H \times W$ */
$\boldsymbol{E} \leftarrow \texttt{Dilate}(\boldsymbol{E}, \texttt{kernel\_size} = 3)$;
$\mathcal{M}^{edge} \leftarrow \texttt{Where}_{\boldsymbol{x} \in \boldsymbol{I}_{1,\ldots,K}}(\boldsymbol{E}(\boldsymbol{x}) = 0)$ ;      /* Shape: $K \times H \times W$ */

/* Calculate Stabilization Loss                                                       */
$\mathcal{M}^{fwd} \leftarrow \mathcal{M}^{fwd}_{cyc} \wedge \mathcal{M}^{edge}_{2,..,K}$;
$\mathcal{M}^{bwd} \leftarrow \mathcal{M}^{bwd}_{cyc} \wedge \mathcal{M}^{edge}_{1,..,K-1}$;
$\mathcal{L}^{fwd}_{stable} \leftarrow \frac{1}{(K-1)HW} \cdot |(\texttt{Warp}(\mathcal{G}^{pred}_{1,\ldots,K-1}, \boldsymbol{O}^{fwd}) - \mathcal{G}^{pred}_{2,\ldots,K}) \cdot \mathcal{M}^{fwd}|_1$;
$\mathcal{L}^{bwd}_{stable} \leftarrow \frac{1}{(K-1)HW} \cdot |(\texttt{Warp}(\mathcal{G}^{pred}_{2,\ldots,K}, \boldsymbol{O}^{bwd}) - \mathcal{G}^{pred}_{1,\ldots,K-1}) \cdot \mathcal{M}^{bwd}|_1$;
$\mathcal{L}_{stable} \leftarrow \frac{1}{2}(\mathcal{L}^{fwd}_{stable} + \mathcal{L}^{bwd}_{stable})$;
return $\mathcal{L}_{stable}$.
```

Method	AbsRel ↓	δ_1 ↑	OPW ↓
Ours $\mathcal{L}_1$	0.123	0.856	0.043
Ours w/o fine-tuning	0.121	0.859	0.040
Ours	**0.119**	**0.865**	**0.038**
Ours with DepthCrafter	0.112	**0.884**	**0.062**
DepthCrafter [?]	**0.110**	0.881	0.111

Table 1. **Additional Ablation Study on KITTI depth estimation.** Our model outperforms both variants (*Model with $\mathcal{L}_1$* and *Model w/o fine-tuning*), and when trained on DepthCrafter frames (*Model with DepthCrafter*), achieves comparable performance to DepthCrafter itself.

4. Additional Ablation Studies

We extend our ablation studies beyond the main paper by comparing our model with additional variants: *Model with $\mathcal{L}_1$* replaces $\mathcal{L}_2$ with $\mathcal{L}_1$ for the affine-invariant relative loss in the depth model; *Model w/o fine-tuning* maintains a fixed refinement network from the backbone model while training only the temporal layers. Additionally, we evaluate an enhanced version utilizing ”oracle” knowledge: *Model with DepthCrafter* incorporates a single frame from DepthCrafter [?] prediction per iteration as regularization guidance.

As shown in Table 1, our model demonstrates superior performance compared to the first two variants, validating the effectiveness of both our architectural and loss function designs. The *Model with DepthCrafter* achieves better results that comparable to DepthCrafter itself, suggesting potential for future improvements through enhanced image priors.