Dual-Camera Smooth Zoom on Mobile Phones (Supplementary Material)

Renlong Wu, Zhilu Zhang(✉), Yu Yang, and Wangmeng Zuo

Harbin Institute of Technology, China
hirenlongwu@gmail.com,cszlzhang@outlook.com,
806224005qq@gmail.com,wmzuo@hit.edu.cn

The content of the supplementary material involves:

- More details of camera transition module in Sec. A.
- More details of Lipschitz regularization item in Sec. B.
- More details of synthetic and real-world datasets in Sec. C.
- More visual results of ZoomGS in Sec. D.
- More visual results of FI models in Sec. E.
- Limitation in Sec. F.

A More Details of Camera Transition Module

The architecture of MLP in Camera Transition (CamTrans) module is provided in Fig. A. It stacks three FC blocks as the main branch, where each block consists of an FC layer followed by an LeakyReLU operation. Then we deploy two heads to predict position offsets $\mathbf{\Delta x}$ and color offsets $\mathbf{\Delta c}$, respectively.

B More Details of Lipschitz Regularization Item

Lipschitz Continuous. A neural network f_θ with parameter θ is called Lipschitz continuous if there exist a constant $q \geq 0$ such that

$$\underbrace{\|f_\theta(e_0) - f_\theta(e_1)\|_p}_{\text{change in the output}} \leq q \ \underbrace{\|e_0 - e_1\|_p}_{\text{change in the input}} \tag{A}$$

for all possible inputs e_0 and e_1 under a p-norm choice. The parameter q is called the Lipschitz constant. In the CamTrans module, we hope a smooth change of camera encoding leads to a smooth change of 3D models, thus, we introduce the Lipschitz regularization item [7] to encourage it to be a Lipschitz continuous mapping.

Lipschitz Regularization Item. Denote $\mathcal{L}_{lipschitz}$ by the Lipschitz regularization item. Following previous work [7], we impose an Lipschitz weight normalization (WN) in each MLP layer of CamTrans module, as shown in Algorithm A. Then, we introduce $\mathcal{L}_{lipschitz}$ on per-layer Lipschitz bounds in WN, *i.e.*,

$$\mathcal{L}_{lipschitz} = \prod_{d=1}^{D} \texttt{Softplus}(q_d). \tag{B}$$

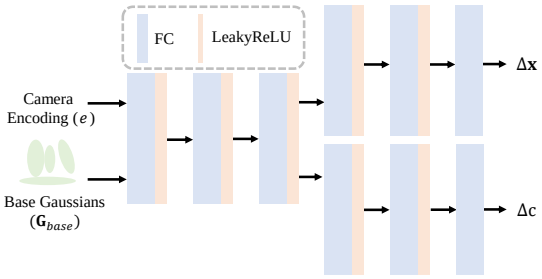

Fig. A: Illustration of MLP in CamTrans module.

Algorithm A Pseudo code about Lipschitz weight normalization (WN)

Require: $\mathbf{W}_d$: weight of d-th MLP layer,
q_d: trainable bound.

$\hat{\mathbf{w}}_d = \sum_{c=1}^{C_{in}} |\mathbf{W}_d^c|$ ▷ *C_{in}: input channel number of d-th MLP layer*

$\mathbf{s} = \texttt{min}(\texttt{Softplus}(q_d) \odot \frac{1}{\hat{\mathbf{w}}_d}, 1.0)$

return $\mathbf{s} \odot \mathbf{W}_d$

q_d is a trainable bound for d-th WN layer, `Softplus` is the softplus activation function, $\texttt{Softplus}(q_d)$ is the Lipschitz bound for d-th WN layer. D is the number of MLP layers.

C More Details of Synthetic and Real-World Datasets

Synthetic dataset. We generate 155 synthetic zoom sequences for DCSZ from 78 scenes, including 48 indoor ones captured in *classrooms*, *dining halls*, and *shopping malls*, and 30 outdoor ones collected in *campuses* and *companies*. Some examples of the synthetic dataset are shown in Fig. B. 127 sequences from 64 randomly sampled scenes are used for training, and the remaining 28 sequences from 14 scenes are used for testing.

Real-World dataset. We additionally capture dual camera images from 100 scenes that are non-overlapped with the synthetic dataset to evaluate FI model in the real world. It includes diverse scenes, like *desks*, *chairs*, *debris piles*, *billboards*, *buildings*, *vegetation*, *etc*. Some examples of the real-world dataset are shown in Fig. C.

D More Visual Results of ZoomGS

We provide the visual comparison results between 3DGS [4] and the proposed ZoomGS in Fig. D. First, when applying a 3DGS model trained with one camera data to the other camera, the rendered dual camera images keep the imaging characteristics of the trained camera, as shown in Fig. D(a) and (b). Second, when naively mixing dual-camera data to train a 3DGS model, it easily produces

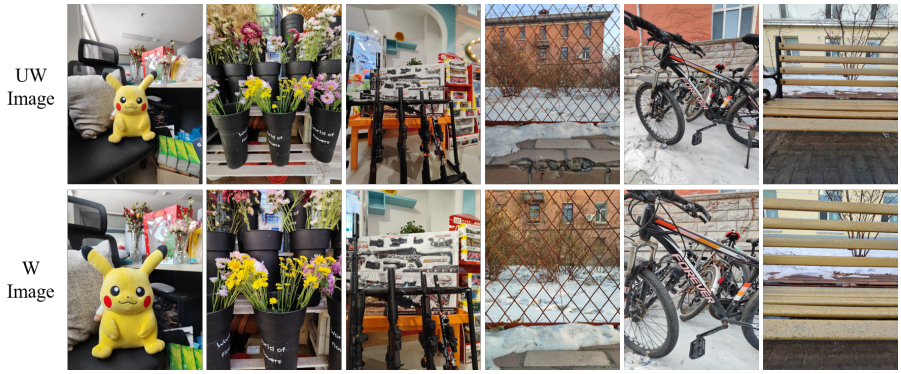

Fig. B: Some examples from synthetic datasets.

Fig. C: Some examples from real-world datasets.

some visual artifacts, as shown in Fig. D(c). Third, by constructing 3D models for each camera, ZoomGS renders dual images that are more consistent with the GT, as shown in Fig. D(d). Besides, we provide some examples of zoom sequences rendered from ZoomGS at the https://dualcamerasmoothzoom.github.io.

E More Visual Results of FI models

We provide more visual comparisons of FI models on the synthetic and real-world datasets, as shown in Fig. E and Fig. F respectively. The fine-tuned FI models produce more photo-realistic details and fewer visual artifacts on both datasets. It indicates the effectiveness of the proposed data factory.

F Limitation

This work is still limited in the FI model generalization between two mobile devices (*e.g.*, apply a model trained with images from an Xiaomi mobile phone

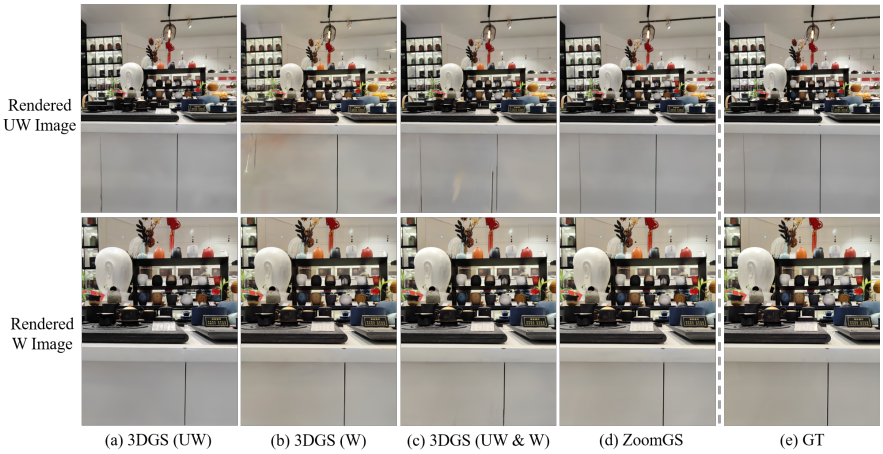

Fig. D: Visual comparisons between 3DGS [4] and the proposed ZommGS. (a) Images rendered from a 3DGS model trained with ultra-wide-angle (UW) images. (b) Images rendered from a 3DGS model trained with wide-angle (W) images. (c) Images rendered from a 3DGS model trained with UW and W images. (d) Images rendered from a ZoomGS model. (e) GT images.

to an OPPO mobile phone). When the relative positions of the dual cameras on two mobile phones are greatly different, the model trained with one mobile phone data may not generalize well to the other one. It may need to fine-tune the FI model with the data from the other mobile phone.

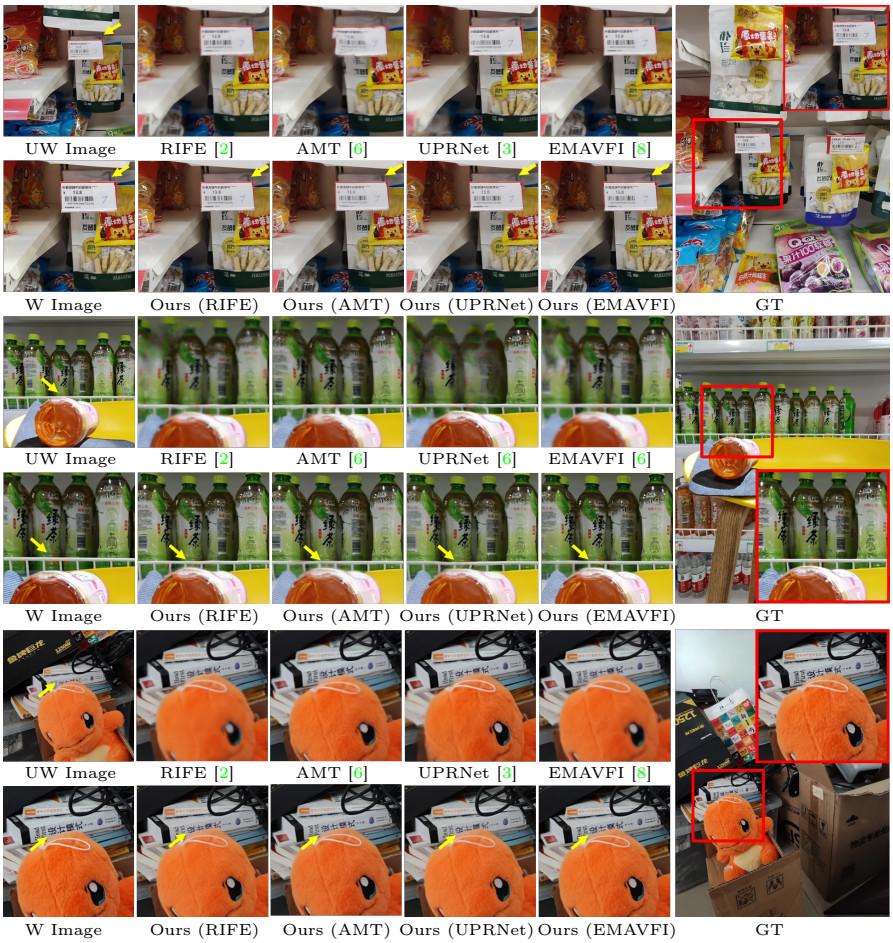

Fig. E: Visual comparisons on the synthetic dataset. The FI models synthesize the intermediate geometry content between dual cameras, as indicated with yellow arrows.

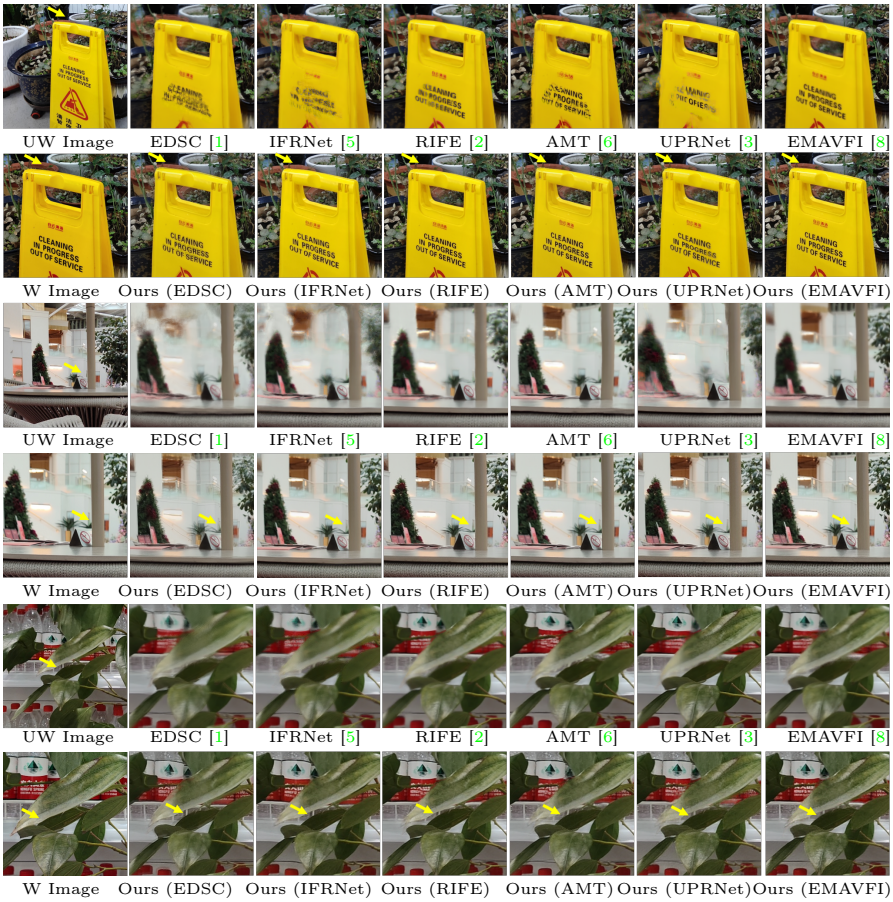

Fig. F: Visual comparisons on the real-world dataset. The FI models still synthesize the intermediate geometry content in the real world, as indicated with yellow arrows.

References

1. Cheng, X., Chen, Z.: Multiple video frame interpolation via enhanced deformable separable convolution. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 7029–7045 (2021) 6
2. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: European Conference on Computer Vision. pp. 624–642. Springer (2022) 5, 6
3. Jin, X., Wu, L., Chen, J., Chen, Y., Koo, J., Hahm, C.h.: A unified pyramid recurrent network for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1578–1587 (2023) 5, 6
4. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023) 2, 4
5. Kong, L., Jiang, B., Luo, D., Chu, W., Huang, X., Tai, Y., Wang, C., Yang, J.: Ifrnet: Intermediate feature refine network for efficient frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1969–1978 (2022) 6
6. Li, Z., Zhu, Z.L., Han, L.H., Hou, Q., Guo, C.L., Cheng, M.M.: Amt: All-pairs multi-field transforms for efficient frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9801–9810 (2023) 5, 6
7. Liu, H.T.D., Williams, F., Jacobson, A., Fidler, S., Litany, O.: Learning smooth neural functions via lipschitz regularization. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–13 (2022) 1
8. Zhang, G., Zhu, Y., Wang, H., Chen, Y., Wu, G., Wang, L.: Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5682–5692 (2023) 5, 6