PixelStitch: Structure-Preserving Pixel-Wise Bidirectional Warps for Unsupervised Image Stitching

Anonymous ICCV submission

Paper ID 7311

A. Supplementary Material

In this document, we provide the following supplementary contents:

- Explanation on objective formulas (Appendix B).
- Details of network architecture (Appendix C).
- Experiment details (Appendix D).
- Further experiment on inference time (Appendix E).
- Difference between homography decomposition and multiple-perspective methods (Appendix F).
- Advantages and limitations of our method (Appendix G).
- More quantitative/qualitative results and comparisons (Appendix H).
- Comparisons with the full pipeline of UDIS++ (Appendix I).
- Multiple images stitching (Appendix J).

B. Objective Formulas

As we estimate inverse optical flows in our work, some formulas may seem confusing. To unify our expressions, we do not warp inversely in our formulas. Instead, we enumerate different sets in summation. When $p \in I_1^F$ is mentioned, it simply means we enumerate every integer coordinate across the panorama. If $p \in I_1^{mid}$ is mentioned, it means we treat p as $F^{-1} * p'$ and enumerate p'. The relation above can be formulated as:

$$\sum_{p \in I_1^{mid}} f(p) \iff \sum_{p' \in I_1^F} f(F^{-1} * p'). \tag{1}$$

C. Network Architecture

C.1. RAFT

We adopt RAFT [17] as a baseline model to estimate optical flows, along with minor modification. The network consists of a feature encoder, a context encoder, a correlation layer, and a GRU-based [2] recurrent update unit.

The ResNet-like [5] networks with a 7×7 convolution layer and 3 bottlenecks are leveraged to extract dense semantic features for the feature encoder and context encoder at 1/8 resolution ($h \times w$). The outputs of the feature encoder are input into the correlation layer to produce correlation volume. The outputs of the context encoder are used to produce primary hidden states for the recurrent unit and context feature maps involved in optical flow prediction.

Then in the correlation layer, we leverage cosine similarity [15] to measure the similarity between two feature vectors ($\vec{v_1}$ and $\vec{v_2}$) as:

$$cos(\vec{v_1}, \vec{v_2}) = \frac{\vec{v_1} \cdot \vec{v_2}}{\|\vec{v_1}\| \|\vec{v_2}\|}. \tag{2}$$

The vectors are considered to be similar only if the angle between them is small. Then a 4D correlation volume of $h \times w \times h \times w$ is formulated by densely computing the correlation between two feature maps. To reduce the subsequent computational burden, a lookup operation is applied to extract the local correlation between each point and its $k \times k$ neighborhood points from the complete correlation volume, yielding a $h \times w \times k \times k$ volume.

Next, a convolutional GRU is adopted to update the hidden state of the recurrent unit iteratively and the input of each iteration is a combination of flows, local correlations, and context features. The updated hidden state is then passed through another convolutional network to obtain the optical flows at 1/8 resolution, which are bilinearly upsampled to recover their original resolution.

C.2. Content Selection Mask Generator

The content selection mask generator consists of 4 convolution layers, all with kernel size 3, padding 1, and stride 1. The output dimensions of these layers are 16, 64, 64, 1. BN and ReLU are adopted between consecutive convolution layers.

D. Experiment Details

D.1. Metrics

We adopt mSSIM and mPSNR (masked SSIM [18] and masked PSNR) in quantitative experiments (mPSNR

Table 1. Details of inference time on the "temple" [4] and "cabin" [8] datasets.

Dataset	Resolution	Output Resolution	Device	Time (second) HE UDIS++ / VGG + RANSAC	HD	BHW	OFE	PN-Coef	BPW
temple	730 × 487	1153 × 558	CPU	0.89 / 3.50	< 0.01	0.19	6.39	0.04	0.07
			GPU	0.02 / 0.04	< 0.01	0.13	0.05	0.01	< 0.01
cabin	1280 × 720	1889 × 1111	CPU	0.83 / 12.46	< 0.01	0.56	16.30	0.09	0.24
			GPU	0.02 / 0.10	< 0.01	0.14	0.19	0.02	< 0.01

Table 2. Comparisons on inference time with other methods.

Time (s)	APAP	LPC	UDIS++	PixelAlign	Ours
temple	3.80	5.52	0.05	0.36	0.23
cabin	11.64	159.24	0.10	1.26	0.39

Table 3. Quantitative comparison of warp on UDIS-D dataset [13]. The best is marked in **red** and the second best is in ***blue***.

	mPSNR ↑			
	Easy	Moderate	Hard	Average
APAP[19]	26.77	22.88	18.75	22.39
ELA[8]	28.82	24.19	18.44	23.27
LPC[6]	25.01	21.27	17.34	20.82
GES-GSP[3]	25.34	19.37	14.79	20.72
UDIS++[14]	27.58	23.75	20.04	23.41
PixelAlign[7]	***29.69***	***26.30***	***21.12***	***25.24***
Ours	**31.58**	**27.39**	**23.18**	**26.96**

adopted in Appendix H), which are defined as:

$$mSSIM(I) = \frac{\sum_{p \in I} M_{olp}(p) \cdot SSIM(p)}{\sum_{p \in I} M_{olp}(p)}, \quad (3)$$

$$mPSNR(I) = 20 \times log_{10}(\frac{1}{mRMSE(I)}), \quad (4)$$

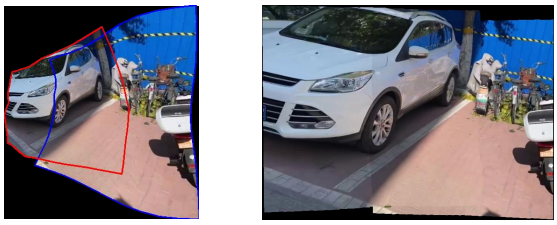
where $mRMSE$ is the RMSE of the overlapping area, and M_{olp} is a mask representing the overlapping area in the image, which has been mentioned in the paper. The window size of SSIM is 7.

D.2. Unidirectional Method

For the unidirectional solution mentioned in our ablation study, we substitute bidirectional consistency with cyclic consistency which is adopted in RANSAC-Flow [16] and PixelAlign [7] in our unidirectional method. Other settings remain the same.

D.3. Presentation

Our experiment presentation is supposed to emphasize three kind of defects: misalignments (artifacts), deformative distortions (bended line structures) and projective distortions (tilted or abnormally scaled objects). The captions give the respective types of defects marked by each kind of markers.



GES-GSP Ours

Figure 1. A failure case of GES-GSP on handling distortion. The case is from UDIS-D dataset. GES-GSP fails to determine the correct scale ratio, resulting in the obviously wrong scale of the left-view image.

E. Inference Time

We measure the elapsed time of every phase in our method, including homography estimation (HE), homography decomposition (HD), bidirectional homography warp (BHW), optical flow estimation (OFE), PN-Coef computation & optical flow refinement (PN-Coef), and bidirectional pixel-wise warp (BHW). We test our model on the "temple" [4] and "cabin" [8] datasets, with Intel i7-10875H 2.30GHz CPU and NVIDIA RTX 4090 GPU. Results are shown in Tab. 1. Our model is capable of producing results in practical time, and with the acceleration of GPU, we can even reach real-time image stitching. Note again that most of the time is used to estimate homography and optical flows. By choosing the proper homography and optical flow estimator, our method is able to achieve better performance. Tab. 2 presents comparisons on inference time between other methods and ours.

F. Homography Decomposition vs. Multiple-Perspective Warps

Existing multiple-perspective methods follow a pipeline of "create a warp W' with lower distortion from the original warp W, then apply $W' + W^{-1}$ to the reference image to compensate the alignment". However, there is no guarantee that $W' + W^{-1}$ is a warp with low distortion, which sometimes causes additional distortion, or even produces more distorted results than the original warp. Fig. 1 shows a failure case of GES-GSP [3] on handling distortion. In contrast, our homography decomposition is more comprehensive and does not bring extra distortion.

Table 4. Investigation of output pixel number and invalid pixel ratio on traditional datasets. The best is marked in **red** and the second best is in ***blue***.

Dataset	Pixel Number ↓ (10^4) / Invalid Pixel Ratio ↓					
	APAP	GES-GSP	LPC	UDIS++	PixelAlign	Ours
train [19]	101.8 / 0.223	162.4 / 0.204	98.6 / 0.229	**83.6** / ***0.149***	125.2 / 0.271	***85.3*** / **0.148**
conssite [19]	654.4 / 0.230	784.3 / ***0.152***	***614.0*** / 0.208	599.0 / 0.200	756.0 / 0.246	**502.4 / 0.105**
building [10]	203.6 / 0.220	**115.7 / 0.101**	196.0 / 0.201	189.1 / 0.188	206.9 / 0.218	***180.3 / 0.184***
fence [10]	258.9 / 0.279	***206.1*** / **0.138**	223.3 / 0.207	**172.7** / ***0.176***	242.2 / 0.248	211.2 / 0.232
carpark [4]	100.0 / 0.268	**47.1 / 0.074**	77.0 / 0.218	86.2 / 0.227	91.9 / 0.249	***69.9 / 0.125***
temple [4]	85.8 / 0.251	**45.6 / 0.053**	80.7 / 0.207	79.4 / 0.221	85.7 / 0.209	***64.3 / 0.116***
campus [1]	412.7 / 0.349	**137.4 / 0.036**	354.2 / 0.347	239.8 / 0.237	-	***202.4 / 0.171***
garden [1]	814.2 / 0.227	**449.6 / 0.027**	805.7 / 0.235	620.2 / 0.139	839.6 / 0.240	***585.2 / 0.097***
parksquare [8]	953.8 / 0.292	**396.0 / 0.054**	1007.6 / 0.317	742.5 / 0.220	1028.0 / 0.331	***653.2 / 0.151***
lawn [8]	709.3 / 0.207	585.8 / **0.084**	690.6 / 0.203	**532.1** / ***0.112***	704.7 / 0.218	***581.2*** / 0.139
racetracks [8]	74.5 / 0.212	**21.8** / 0.235	69.7 / ***0.187***	***61.7*** / **0.140**	81.9 / 0.203	65.7 / 0.203
cabin [8]	271.2 / 0.203	**157.5 / 0.047**	259.2 / 0.196	234.3 / 0.158	269.8 / 0.208	***209.9 / 0.095***
theater [8]	512.4 / 0.253	**218.6 / 0.085**	485.6 / 0.242	***473.7*** / 0.230	811.8 / 0.252	574.1 / ***0.103***
footpath [8]	393.8 / 0.251	**189.1 / 0.053**	343.8 / 0.229	306.7 / 0.178	397.4 / 0.250	***285.0 / 0.138***
window [9]	110.7 / 0.204	**63.5 / 0.048**	134.3 / 0.242	114.5 / 0.209	120.2 / 0.224	***91.5 / 0.104***
door [9]	***54.7*** / **0.018**	81.5 / 0.022	**54.4** / ***0.020***	**54.4** / 0.021	60.6 / 0.033	57.7 / 0.033
four [9]	71.8 / 0.116	**38.5** / ***0.096***	70.9 / 0.110	71.5 / 0.116	***67.9*** / **0.071**	70.5 / 0.103
chessgirl [12]	***96.6*** / **0.134**	172.4 / 0.144	97.7 / 0.140	**95.9 / 0.134**	107.1 / 0.154	99.9 / ***0.136***

Figure 2. A failure case when the contents are inconsistent. The case is from [11].

G. Advantages and Limitations

Pixel-wise warps are able to give more accurate alignments comparing to mesh-base warps like TPS and multiple homography, because they are more flexible and fine-grained. We demonstrate that our warp performs the best in some challenging scenarios, such as low texture (line 2 and 5 of Fig. 5, line 2 of Fig. 6, line 3 of Fig. 7), low overlapping ratio (Fig. 5 in the main paper, line 2 of Fig. 8) and large parallax (Fig. 6 in the main paper, Fig. 9).

Nevertheless, limitations exist when the contents are inconsistent between both images, in which case the pixel-level method fails to align the contents or could produce distortions, as there is no reference for some objects and the model may mistakenly align them. A failure case is presented in Fig. 2.

H. More Experiments

Tab. 4 and Tab. 3 list more quantitative results. Tab. 3 shows mPSNR (masked PSNR) on UDIS-D dataset. Tab. 4 reveals the capability of homography decomposition to decrease output pixel number and invalid pixel proportion, which are beneficial for memory/storage efficiency and visual naturalness. Our method is second only to GES-GSP [3], whereas better than it on alignment accuracy by a large margin. The traditional datasets for quantitative experiments in the main paper and Tab. 4 are listed in Fig. 10.

More qualitative results are shown in Fig. 5, Fig. 6, Fig. 8, Fig. 7 and Fig. 9. The former four figures respectively show general comparison, comparison with warps for better alignment (APAP [19], ELA [8]), comparison with warps for less distortion (LPC [6], GES-GSP [3]) and comparison with deep methods (UDIS++ [14], PixelAlign [7]). Fig. 9 lists some large-parallax cases from [11] and [20] to show our superior performance on large-parallax scenarios.

I. Comparisons with UDIS++

Specifically, UDIS++ [14] includes an additional post-processing procedure to compose warped images. Here we adopt the same post-processing on our method and compare it with the full pipeline of UDIS++. Results are presented in Fig. 3.

J. Multiple Image Stitching

Single-perspective methods have difficulties in stitching more than two images, as projective distortion will accumulate in a single direction. To deal with the problem, they

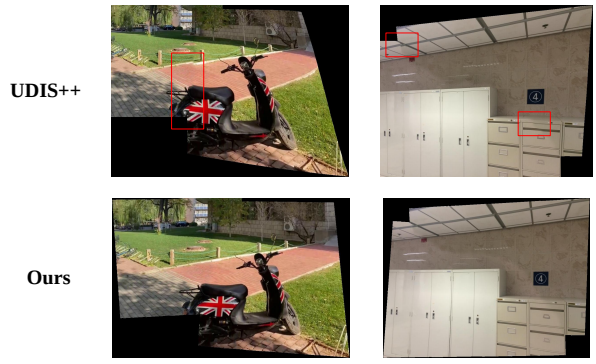


Figure 3. Comparisons between our method and UDIS++ using the same post-processing.

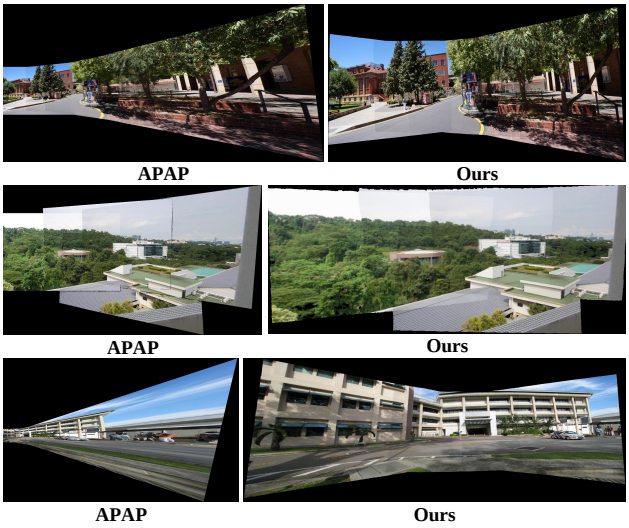


Figure 4. Comparison between APAP [19] and *PixelStitch* on stitching multiple images from left to right. The cases are "garden" [19], "forest" [4] and "carpark" [4].

need to partition images into several groups and merge them in the stitching process. In contrast, we do not care much about stitching order in our method, since homography decomposition is capable of autonomously distributing projective distortion to each view. We compare the results from APAP [19] and *PixelStitch* by naively stitching multiple images from left to right, as shown in Fig. 4.

References

[1] Che-Han Chang, Yoichi Sato, and Yung-Yu Chuang. Shape-preserving half-projective warps for image stitching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3254–3261, 2014. 3, 9

[2] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017. 1

[3] Peng Du, Jifeng Ning, Jiguang Cui, Shaoli Huang, Xinchao Wang, and Jiaxin Wang. Geometric structure preserving warp for natural image stitching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3688–3696, 2022. 2, 3

[4] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 49–56, 2011. 2, 3, 4, 6, 7, 9

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[6] Qi Jia, ZhengJun Li, Xin Fan, Haotian Zhao, Shiyu Teng, Xinchen Ye, and Longin Jan Latecki. Leveraging line-point consistence to preserve structures for wide parallax image stitching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12186–12195, 2021. 2, 3

[7] Qi Jia, Xiaomei Feng, Yu Liu, Xin Fan, and Longin Jan Latecki. Learning pixel-wise alignment for unsupervised image stitching. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1392–1400, 2023. 2, 3

[8] Jing Li, Zhengming Wang, Shiming Lai, Yongping Zhai, and Maojun Zhang. Parallax-tolerant image stitching based on robust elastic warping. *IEEE Transactions on multimedia*, 20(7):1672–1687, 2017. 2, 3, 6, 7, 9

[9] Shiwei Li, Lu Yuan, Jian Sun, and Long Quan. Dual-feature warping-based motion model estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4283–4291, 2015. 3, 6, 9

[10] Chung-Ching Lin, Sharathchandra U Pankanti, Karthikeyan Natesan Ramamurthy, and Aleksandr Y Aravkin. Adaptive as-natural-as-possible image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1163, 2015. 3, 6, 7, 9

[11] Kaimo Lin, Nianjuan Jiang, Loong-Fah Cheong, Minh Do, and Jiangbo Lu. Seagull: Seam-guided local alignment for parallax-tolerant image stitching. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 370–385. Springer, 2016. 3, 8

[12] Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. Smoothly varying affine stitching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 345–352. IEEE, 2011. 3, 9

[13] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Unsupervised deep image stitching: Reconstructing stitched features to images. *IEEE transactions on image processing*, 30:6184–6197, 2021. 2, 5

[14] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Parallax-tolerant unsupervised deep image stitching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7399–7408, 2023. 2, 3

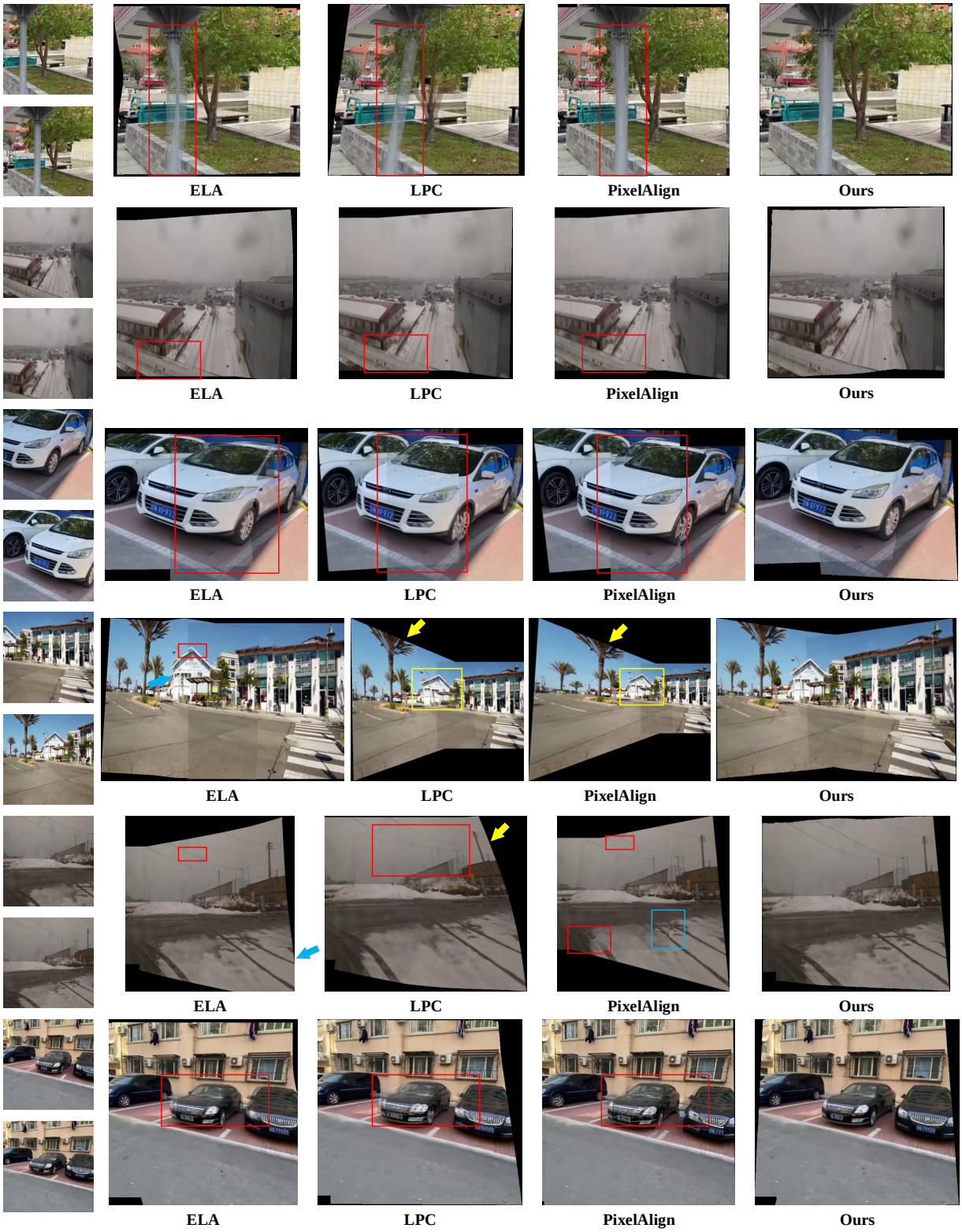


Figure 5. General comparison on UDIS-D dataset [13]. Red boxes indicate misalignments. Blue boxes and arrows indicate deformative distortion. Yellow boxes and arrows indicate projective distortion.

Figure 6. Comparison with warps for better alignment. The datasets are "building" [10], "four" [9] and "lawn" [8]. Red boxes indicate misalignments. Yellow boxes and arrows indicate projective distortion.

Figure 7. Comparison with deep methods. The datasets are "carpark" [4], a image pair from [20] and "roof" [9]. Red and green boxes indicate misalignments.

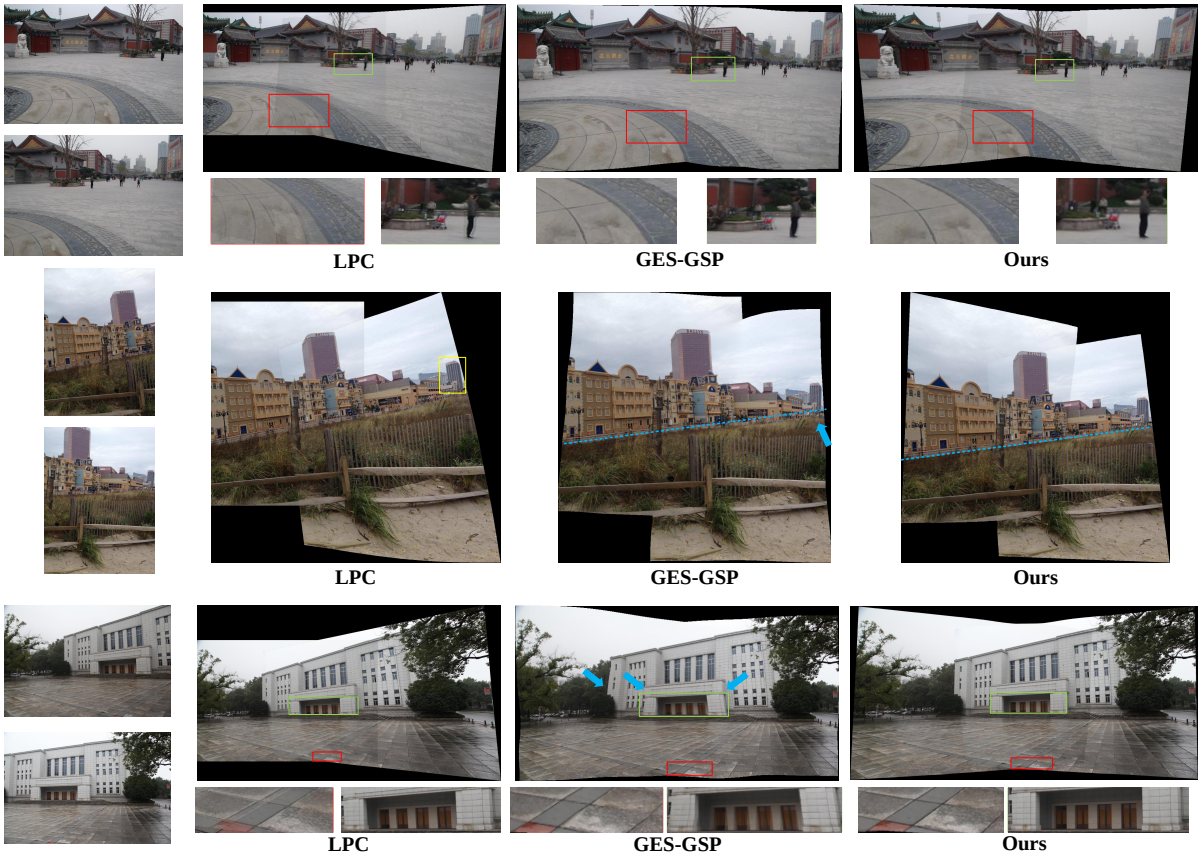


Figure 8. Comparison with warps for less distortion. The datasets are "temple" [4], "fence" [10] and "theater" [8]. Red and green boxes indicate misalignments. Blue arrows indicate deformative distortion. Yellow boxes indicate projective distortion.

[15] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017. 1

[16] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 618–637. Springer, 2020. 2

[17] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 1

[18] 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*, 13(4):600–612, 2004. 1

[19] Julio Zaragoza, Tat-Jun Chin, Michael S Brown, and David Suter. As-projective-as-possible image stitching with moving dlt. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2339–2346, 2013. 2, 3, 4, 9

[20] Fan Zhang and Feng Liu. Parallax-tolerant image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3, 6, 8

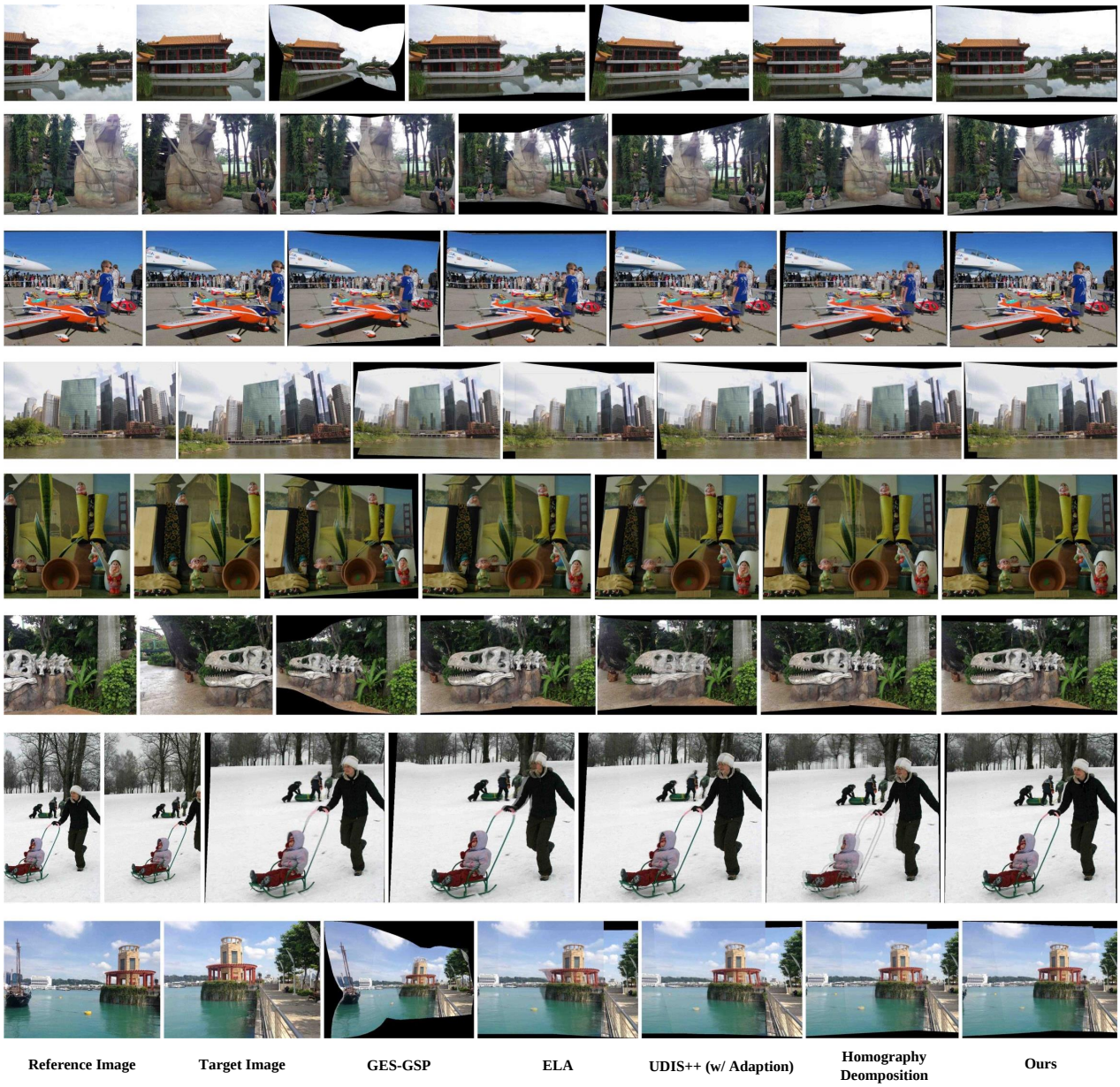


Figure 9. Large-parallax cases from [11] and [20].

ICCV
#7311

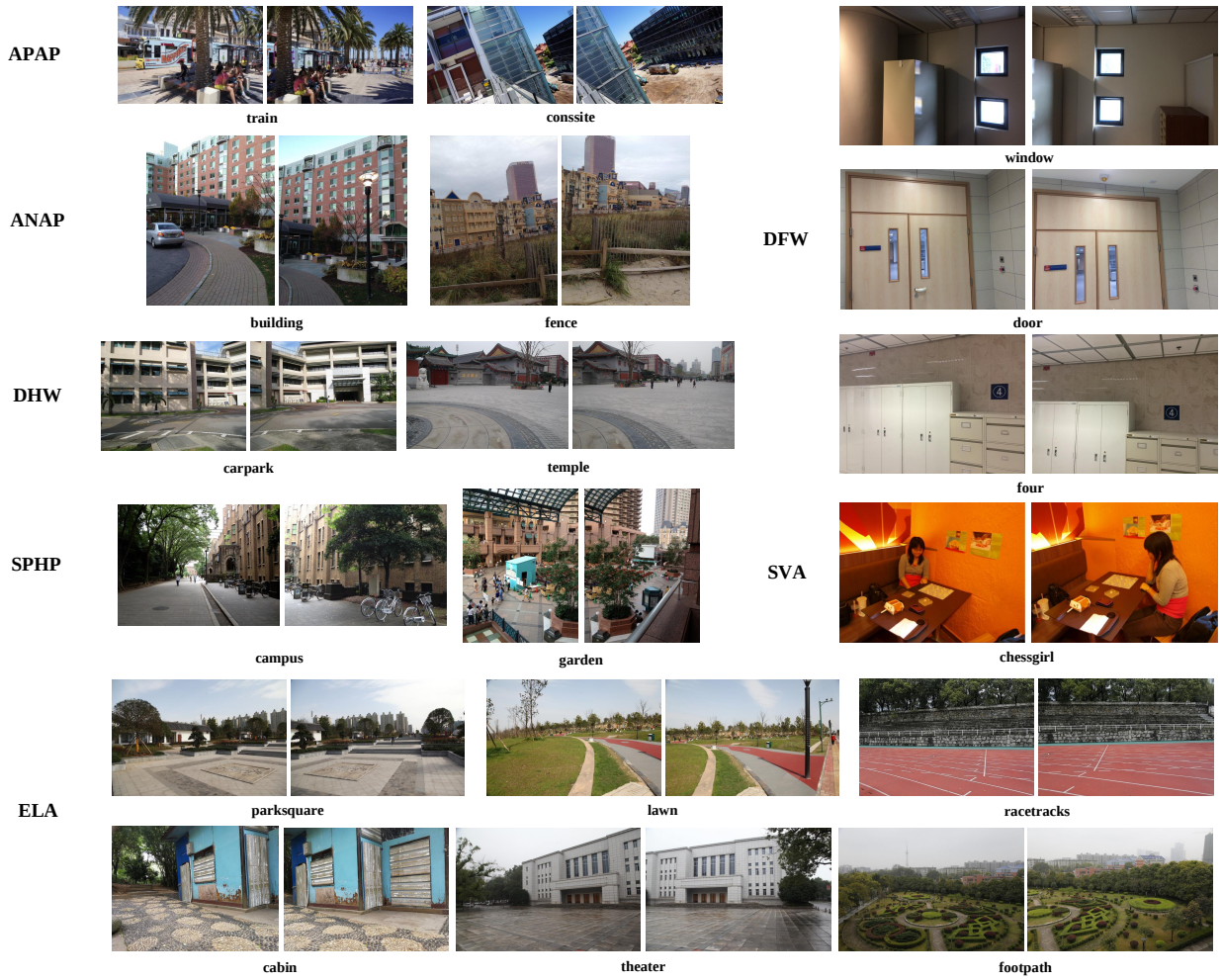


Figure 10. Image inputs of quantitative experiments. Cases are from APAP [19], ANAP [10], DHW [4], SPHP [1], ELA [8], DFW [9] and SVA [12].