Parallax-Tolerant Unsupervised Deep Image Stitching
SUPPLEMENTARY MATERIAL

Lang Nie[1,2] Chunyu Lin[1,2*] Kang Liao[1,2] Shuaicheng Liu[3] Yao Zhao[1,2]
[1]Institute of Information Science, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Advanced Information Science and Network, Beijing, China
[3]University of Electronic Science and Technology of China, Chengdu, China
{nielang, cylin, kang_liao, yzhao}@bjtu.edu.cn, liushuaicheng@uestc.edu.cn

1. Supplemental Material

In this document, we provide the following supplementary contents:

- Details of warp (Section 2).
- Details of composition (Section 3).
- Analysis on robustness and distortion (Section 4).
- More results (Section 5).

Regarding the network architecture, we have not provided specific details such as layers, channels, etc., as we would like readers to focus more on the motivations behind our approach to solving the problem. For the details, we promise to release the code for reference.

2. More Details of Warp

2.1. Physicality of TPS

The thin plate spline (TPS) method can simulate arbitrary 2D deformation through the use of a deformable thin plate, which is more general than using homography. When all control points are correctly matched, we aim to use a thin plate with minimal curvatures. We then formulate an energy optimization problem that involves both alignment and distortion, as described in [1]:

$$\varepsilon = \varepsilon_{alignment} + \lambda \varepsilon_{distortion}, \tag{1}$$

where λ is a balancing factor to control the smoothness of the warp. The alignment energy and distortion energy are defined as follows:

$$\begin{aligned} \varepsilon_{alignment} &= \sum_{i=1}^{N} \| p' - \mathscr{T}(p) \|^2, \\ \varepsilon_{distortion} &= \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 \mathscr{T}}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 \mathscr{T}}{\partial x \partial y} \right)^2 \right. \\ &\quad \left. + \left(\frac{\partial^2 \mathscr{T}}{\partial y^2} \right)^2 \right) dx dy, \end{aligned} \tag{2}$$

where $\mathscr{T}(\cdot)$ is the warp function. When $\lambda > 0$, the control points are allowed to be slightly misaligned in order to produce a warp with less distortion. However, in our implementation, we set $\lambda = 0$ to strongly constrain the motion of the control points. This means that our network predicts the motions of the control points, and enforces the real motions $(\mathscr{T}(p) - p)$ to be equal to the predicted motions. By minimizing Eq. 1, we are able to determine the warp function, which is derived as follows (see Eq. 2 in the manuscript):

$$p' = \mathscr{T}(p) = C + Mp + \sum_{i=1}^{N} w_i O(\| p - p_i \|_2). \tag{3}$$

2.2. Discussion of Alignment and Distortion

In the previous section, we explained that aligning all control points causes distortion in the warp function. To mitigate this issue, we assume that control points are evenly distributed in the target image, and their motions are smooth. We form a mesh by connecting control points and introduce the intra-grid/inter-grid constraint for content preservation.

To summarize, the proposed warp yields two improvements. (1) Our network architecture with TPS benefits the alignment in overlapping regions. (2) The distortion loss (Eq. 7,8 of the manuscript) benefits the distortion elimination in non-overlapping regions.

*Corresponding author.

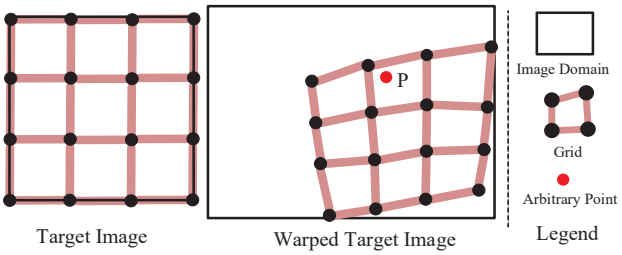


Figure 1: Backward interpolation.

2.3. Multiple Homography vs. TPS

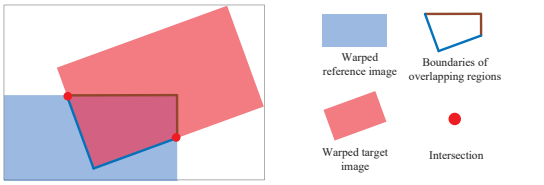
The TPS warp is more appropriate than the traditional mesh-based multi-homography warp [9] in deep stitching. Here, we discuss the reason in detail.

The multi-homography stitching methods warp the target image into a warped target image through mesh deformation as illustrated in Fig. 1. In the implementation, backward interpolation is commonly leveraged to avoid invalid pixels like holes. In backward interpolation, for an arbitrary point P in a warped target image, we need to calculate the corresponding location in the target image. Then bilinear interpolation is leveraged to obtain the pixel value of P. Therefore, how to calculate the corresponding position is the key problem. To make it, the first thing is to determine which grid dose P belong to in multi-homography warp. In the case of Fig. 1, it seems easy to find that P belongs to the second grid, so we could calculate the corresponding homography through the four pairs of vertices of this grid. *However, how to determine the belongings of all points in the warped target image in an efficient parallel manner makes a big difficulty.* Because the warped mesh has an irregular shape, in which even the non-convex grid might be produced. This process is hard to be parallelly accelerated, especially in GPUs, making the training time unbearable. (Empirically, the training process might take millions of iterations.)

In contrast, the TPS transformation has the advantage that all pixels share the same warp function (Eq. 3), eliminating the need to determine the belonging of each pixel to a particular grid. In the multi-homography scheme, the warp of a pixel is determined by only four pairs of vertices, while in TPS, it is influenced by all pairs of control points ($(U+1)\times(V+1)$ in our paper). As a result, the backward interpolation of all pixels in the warped target image can be efficiently achieved in a parallel manner for TPS, making the training process faster compared to multi-homography.

2.4. Difference to Stitching Methods using TPS

The existing stitching methods using TPS are all traditional feature-based solutions. For example, ELA [5] calculates TPS transformations using matched keypoints such as SIFT. This transformation is then processed to reduce computational cost and distortions.



(a) Left: a composition case. Right: the legend.

(b) Boundary masks: M_{br} (left) and M_{bt} (right).

(c) Warped masks: M_r (left) and M_t (right).

Figure 2: Details of the boundary term for the composition.

In contrast, the proposed method is the first deep learning-based stitching scheme that utilizes TPS transformations. The calculation of this warp is no longer reliant on matched keypoints. Instead, we initially define control points that are evenly distributed in the target image and then predict the motions of these points using the unsupervised network. Through the initial control points and the predicted motions, we obtain two sets of control points with one-to-one correspondence. We then formulate the warp and eliminate projective and structural distortions using intra-grid and inter-grid constraints as additional loss functions. Compared to ELA, our proposed method achieves superior alignment (Table 1 of the manuscript), fewer distortions (Fig. 5 of the manuscript), and better efficiency (Table 2 of the manuscript).

3. More Details of Composition

3.1. Boundary Term

Considering a composite case (Fig. 2a), we aim to fix the endpoints of a seam on the intersections. To achieve this, we define two boundary masks, as shown in Fig. 2b: M_{br} and M_{bt}. The two boundaries are located inside the warped reference image and the warped target image, respectively. In our boundary constraint, we encourage the boundary pixels of overlapping regions in S to be from either I_{wr} or I_{wt} using the following equation:

$$\mathcal{L}^{c}_{boundary} = \| (S - I_{wr}) \cdot M_{br} \|_1 + \| (S - I_{wt}) \cdot M_{bt} \|_1 . \quad (4)$$

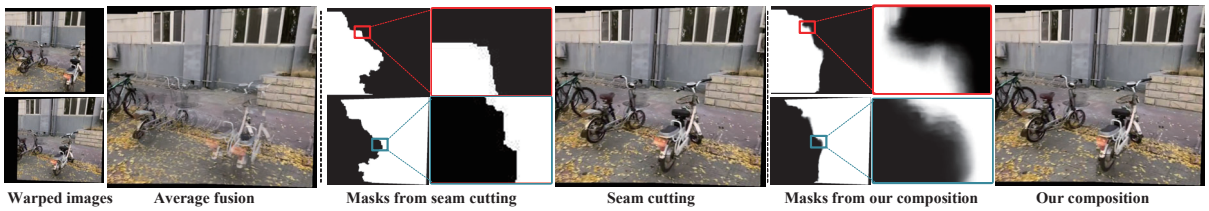


Figure 3: The difference between masks from seam cutting [6] and our composition.

By constraining the values of boundary pixels in a stitched image, we constrain that in composition masks indirectly. More importantly, M_{br} and M_{bt} share two common intersections as represented by the red circles in Fig.2a. These common pixels inevitably yield ambiguity for the belongs of intersections, and the ambiguity helps to determine the seam endpoints.

Next, we describe how to get the boundary masks. Given the warped masks M_r, M_t (as shown in Fig. 2c), we obtain boundary masks by the following formulation:

$$\begin{aligned} M_{br} &= M_r \cdot \mathscr{E}(M_t), \\ M_{bt} &= M_t \cdot \mathscr{E}(M_r), \end{aligned} \tag{5}$$

where $\mathscr{E}(\cdot)$ denotes the edge extraction operation that can be implemented by several convolutional layers with $SOBEL$ filters.

3.2. Difference to Seam Cutting

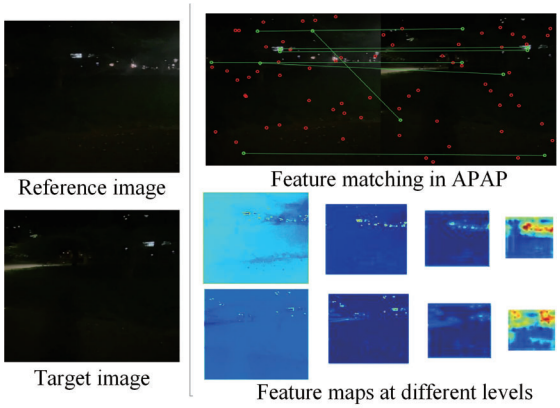
Traditional seam-cutting methods find the invisible seams by dynamic programming or assign composition labels by graph-cut optimization. The masks used for fusion in these methods only contain values of 0 or 1.

However, for a learning system, the predicted masks with strict integers would prevent gradients from back-propagation. Moreover, the masks with strict integers could easily produce discontinuous contents in the composited results. Therefore, we define the values of the masks to be float and propose a smoothness constraint on the stitched image (Eq. 12 of the manuscript) to encourage the smooth transition on both sides of this "seam". Fig. 3 shows the masks from seam cutting [6] and ours, where our "seam" is significantly wider. That is why we cannot quantitatively evaluate our composition in traditional metrics.

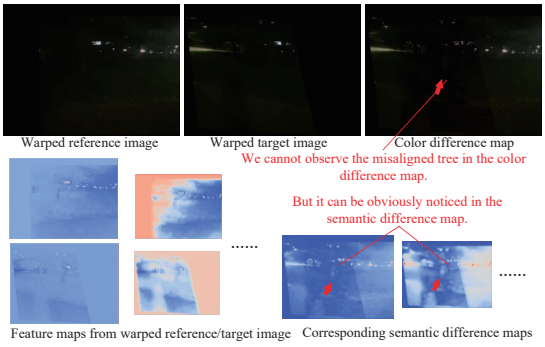
4. Analysis

4.1. Analysis on Robustness

Warp: We argue that the proposed method is more robust than traditional solutions, especially in challenging cases. To illustrate this, we compare our method with APAP [9], which represents traditional solutions. In Fig. 4a, we show a challenging case with extremely low light. APAP extracts SIFT keypoints, which are marked using red or green circles. RANSAC is then used to remove the outliers (red circles), and the green line indicates matched keypoints. As shown in Fig. 4a, the keypoints are very sparse, and some keypoints are even mismatched, which can easily lead to stitching failure. In contrast, our solution extracts semantic feature maps, which become increasingly evident with the increase of network layers, contributing to our robustness.



(a) Robustness analysis of warp.



(b) Robustness analysis of composition.

Figure 4: Robustness analysis.

Composition: Regarding composition, existing seam-cutting methods mainly rely on color difference or other pixel-level energy maps. However, these maps often lose some essential content in challenging cases, such as low light. Fig. 4b displays an example where the tree (highlighted by red arrows) is missing in the color difference map. The proposed deep composition method overcomes

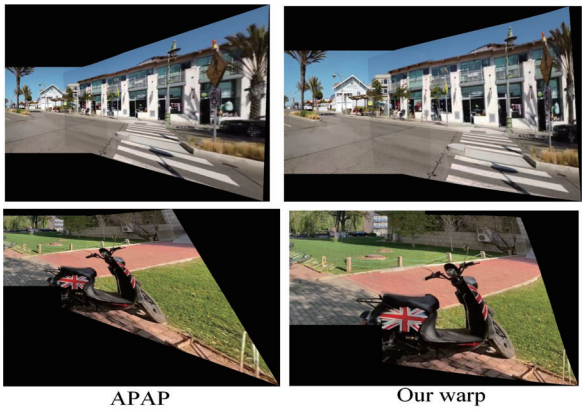


Figure 5: Projective distortion: APAP vs. ours. These instances are from UDIS-D dataset[8].

this issue by extracting semantic difference maps, even though it is trained with color difference. Through training with extensive samples (both simple cases and challenging cases), the composition network is capable of perceiving the semantic difference even in low-light scenes. We illustrate the extracted feature maps and semantic residuals of the composition network in Fig. 4b, where the tree can be obviously noticed in semantic difference maps.

4.2. Analysis on Projective Distortion

Compared with other warps, our warp produces fewer projective distortions. We analyze the phenomenon from two perspectives:

i) Traditional methods estimate the warp from matched features. However, these features are usually distributed in some texture-rich local areas, so that the warp aligns well with these regions and overlooks other overlapping areas. Compared with them, our objective goal is to align all the pixels in overlapping regions (Eq. 6 of the manuscript). Therefore, our warp produces less projective distortions.

ii) To further eliminate projective distortions, we design an intra-grid constraint (Eq. 7 of the manuscript) to prevent the deformed mesh from scaling dramatically.

5. More Results

5.1. Results of Warp

The Fig. 7,8 of this material are the inputs of Fig. 4, 5 in the manuscript. We demonstrate more results of warp on UDIS-D dataset and other datasets in Fig. 11 and Fig. 12.

5.2. Results of Composition

Here, we illustrate more comparative results of large-parallax composition in Fig. 13. To highlight the parallax artifacts intuitively, we use SIFT+RANSAC to align input images and blend the results with average fusion for reference. Then we compare our results with SoTA composition

Figure 6: User study of visual preferences with existing SoTA solutions. The results are presented in percentage and averaged on 20 participants.

Figure 7: The input images of Fig. 4 in the manuscript.

Figure 8: The input images of Fig. 5 in the manuscript.

methods (perception-based seam cutting [6] and reconstruction [8]) in UDIS-D [8] and other large-parallax datasets [7].

5.3. Results of Complete Solutions

Then, we compare our complete framework with other SoTA solutions (LPC [4] and UDIS [8]) with seam cutting

Table 1: Ablation studies of alignment performance on UDIS-D dataset[8]. With the distortion term (ℓ_{inter}+ℓ_{intra}), the alignment performance decrease little.

	Loss	PSNR	SSIM
1	w/o ℓ_{inter}+ℓ_{intra}	25.54	0.841
2	w/o ℓ_{intra}	25.53	0.840
3	w/o ℓ_{inter}	25.48	0.839
4	Our warp	25.43	0.838

Table 2: The superiority of combining TPS with homography. The experiments are conducted on UDIS-D dataset.

	Architecture	PSNR	SSIM
1	Homography + Homography	24.46	0.802
2	TPS + TPS	25.31	0.836
3	Homography + TPS	25.43	0.838

or reconstruction as their post-processing operations. The qualitative results are shown in Fig. 14.

Moreover, we strictly follow the experimental setup in UDIS and conduct user studies to test visual preferences. The participants include 10 volunteers with computer vision backgrounds and 10 outside this community. Specifically, we compare our method with LPC [4] and UDIS [8] one by one. At each time, four images are shown on one screen: the inputs, our stitched result, and the result from LPC/UDIS. The results of ours and the other method are illustrated in random order each time. The user is allowed to zoom in on the images and is required to answer which result is preferred. In the case of "no preference," the user needs to answer whether the two results are "both good" or "both bad". The studies are carried out in the testing set of UDIS-D [8], which means every user has to compare each method with ours in 1,106 images. The results are shown in Fig. 6.

Besides, we demonstrate more results in traditional datasets [5, 7, 9, 2, 10] in Fig. **??**. Our solution can generate natural and seamless results in different scenes with various resolutions and parallax. Also, we promise to release all subjective results, including 1,106 images in UDIS-D and others in traditional datasets.

5.4. Results of Challenging Scenes

We also demonstrate more results in some challenging scenes, such as low texture, low light, etc. As shown in Fig. 10, the traditional scheme fails to stitch these images due to the lack of geometric features. In contrast, our solution succeeds (the reason is discussed in Section 4.1).

5.5. Ablation Studies

As shown in Fig. 7 of the manuscript, the distortion constraints preserve the shape effectively. Also, it produces little negative impact on alignment. The quantitative results

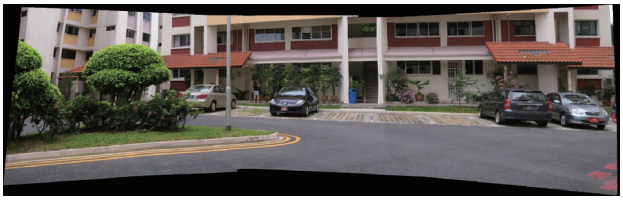
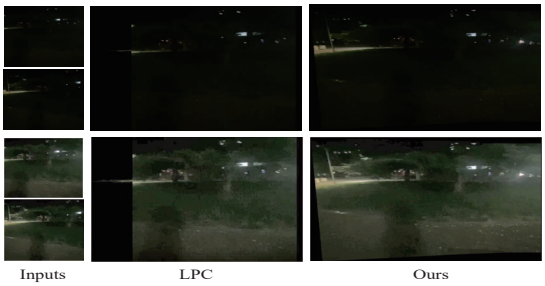


Figure 9: Stitching four images from the traditional dataset[3].

(a) Low-texture cases.



(b) A case in the dark. Top: the original images (inputs and results). Bottom: images after enhancement for better observation.

Figure 10: Results of challenging scenes. Traditional methods fail in these scenes due to the lack of geometric features. All the cases are from UDIS-D dataset [8]

are shown in Table 1, where the SSIM merely decreases 0.03 when we adopt these shape-preserving constraints.

Besides, we replace the TPS prediction with homography prediction in our warp to demonstrate the improvement of TPS deformation. As shown in Table 2, PSNR/SSIM is increased by 0.97/0.036 with TPS deformation, revealing the superiority of TPS over Homography.

5.6. Multi-Image Stitching

Most stitching methods (e.g., LPC[4], UDIS[8]) focus on stitching two images, and so do ours. However, stitching multiple images can be generalized by performing multiple pairwise stitching. Here, we show a case of stitching 4 images in Fig. 9.

References

[1] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 11(6):567–585,

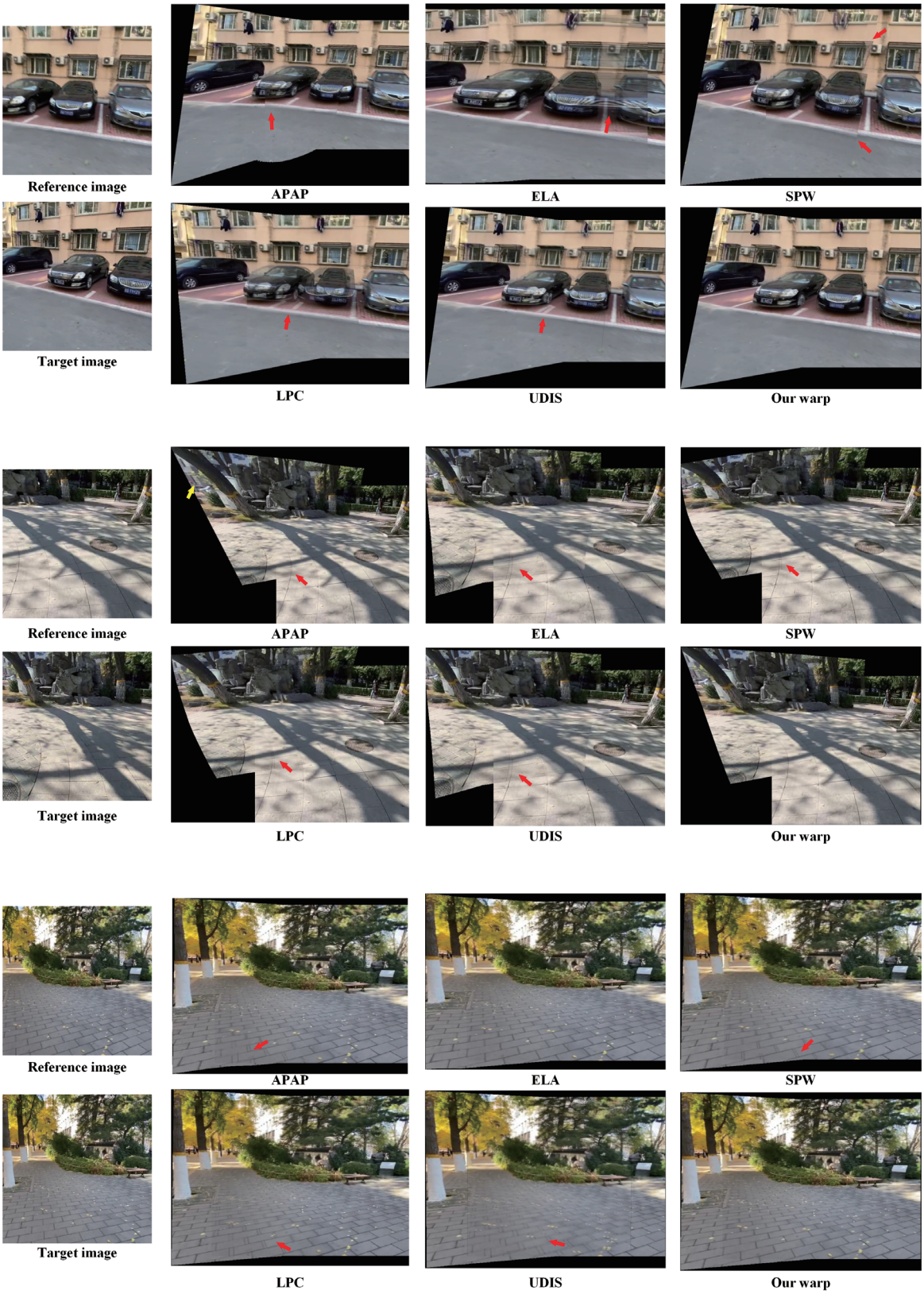


Figure 11: Comparative results of warp on UDIS-D dataset[8]. The red arrows highlight the artifacts.

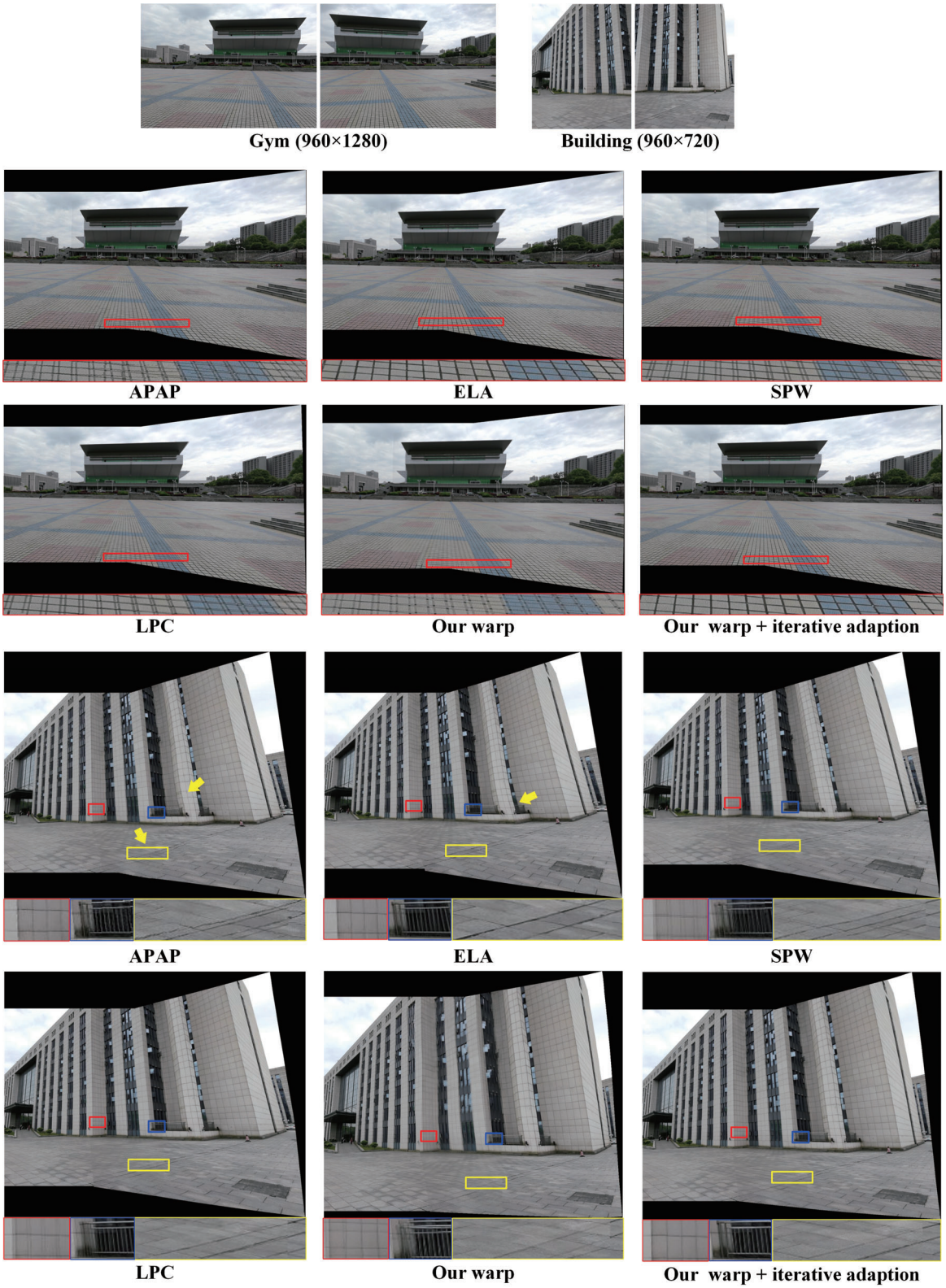
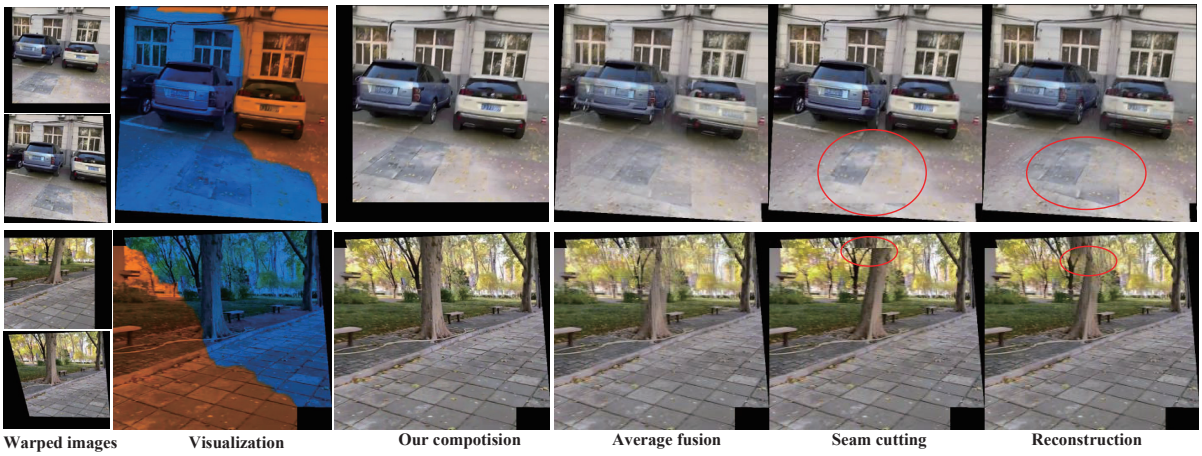
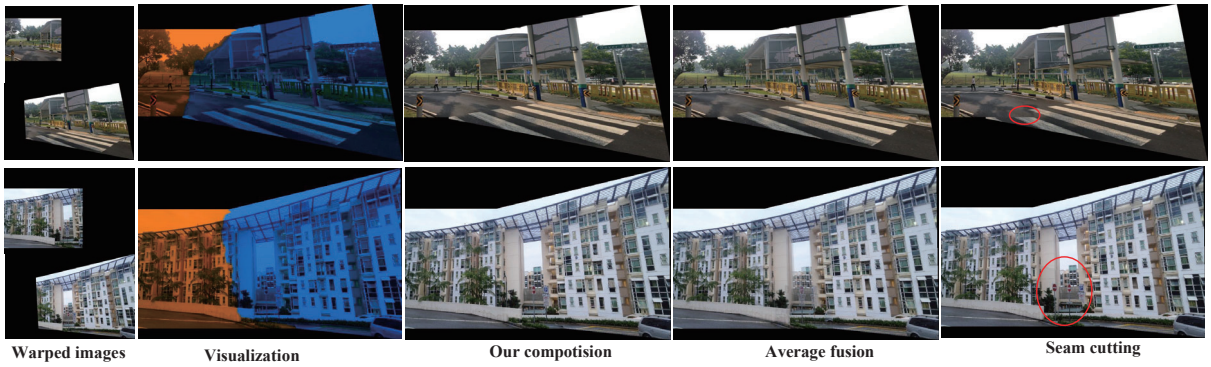


Figure 12: Comparative results of warp in cross-dataset cases[5].The arrows highlight the distortions.



(a) Comparison of composition in UDIS-D dataset[8].



(b) Comparison of composition in traditional large-parallax dataset[7].

Figure 13: Comparative results of composition. We warp large-parallax cases using SIFT+RANSAC and all the composition methods take the warped images as input. The red circles highlight the seam discontinuity or blur.

Figure 14: Comparative results of complete stitching frameworks. LPC[4] and UDIS[8] leverage perception-based seam cutting[6] and reconstruction[8]as the composition methods.

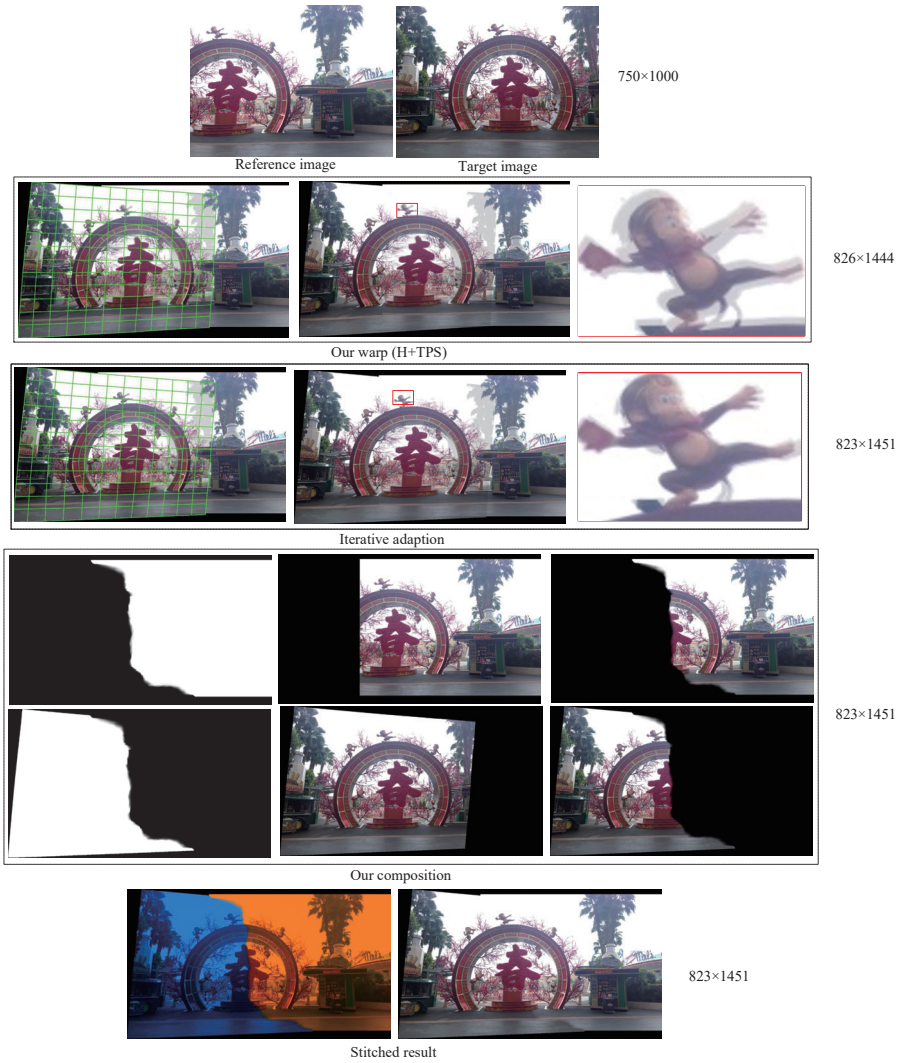


Figure 15: The complete pipeline of the proposed stitching framework. We show our intermediate result in a large-parallax cross-dataset case[7]. We link the predicted control points to form a mesh for clear visualization. Note that for the images from UDIS-D dataset[8], we do not conduct warp adaption iterations.

1989. 1

[2] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *CVPR*, pages 49–56, 2011. 5

[3] Junhong Gao, Yu Li, Tat-Jun Chin, and Michael S Brown. Seam-driven image stitching. In *Eurographics*, pages 45–48, 2013. 5

[4] 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 *CVPR*, pages 12186–12195, 2021. 4, 5, 8

[5] Jing Li, Zhengming Wang, Shiming Lai, Yongping Zhai, and Maojun Zhang. Parallax-tolerant image stitching based on robust elastic warping. *TMM*, 20(7):1672–1687, 2017. 2, 5, 7

[6] Nan Li, Tianli Liao, and Chao Wang. Perception-based seam cutting for image stitching. *Signal, Image and Video Processing*, 12(5):967–974, 2018. 3, 4, 8

[7] Kaimo Lin, Nianjuan Jiang, Loong-Fah Cheong, Minh Do, and Jiangbo Lu. Seagull: Seam-guided local alignment for parallax-tolerant image stitching. In *ECCV*, pages 370–385, 2016. 4, 5, 8, 9

[8] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Unsupervised deep image stitching: Reconstructing stitched features to images. *TIP*, 30:6184–6197, 2021. 4, 5, 6, 8, 9

[9] Julio Zaragoza, Tat-Jun Chin, Michael S Brown, and David Suter. As-projective-as-possible image stitching with moving dlt. In *CVPR*, pages 2339–2346, 2013. 2, 3, 5

[10] Fan Zhang and Feng Liu. Parallax-tolerant image stitching. In *CVPR*, pages 3262–3269, 2014. 5