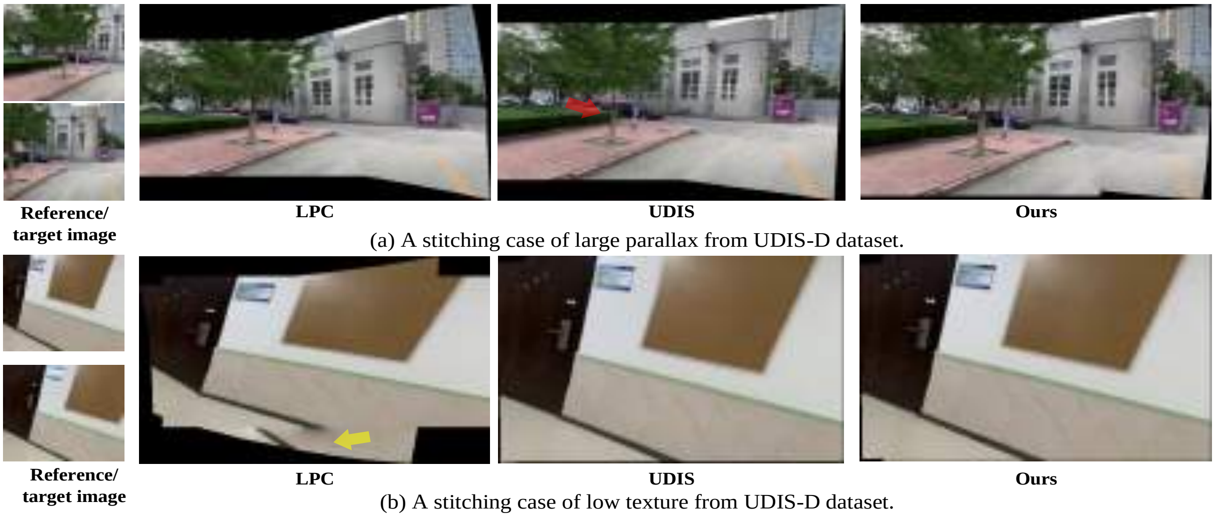
Parallax-Tolerant Unsupervised Deep Image Stitching

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



(a) A stitching case of large parallax from UDIS-D dataset.

(b) A stitching case of low texture from UDIS-D dataset.

Figure 1: Limitations of existing methods. (a) UDIS [41] (learning method) deals with large parallax by blurring parallax regions (highlighted in red). (b) LPC [19] (traditional method) fails in low-texture scenes without sufficient geometric features. Instead, our solution is free from these limitations, achieving promising results in both of the challenging circumstances.

Abstract

Traditional image stitching approaches tend to leverage increasingly complex geometric features (e.g., point, line, edge, etc.) for better performance. However, these hand-crafted features are only suitable for specific natural scenes with adequate geometric structures. In contrast, deep stitching schemes overcome adverse conditions by adaptively learning robust semantic features, but they cannot handle large-parallax cases.

To solve these issues, we propose a parallax-tolerant unsupervised deep image stitching technique (UDIS++). First, we propose a robust and flexible warp to model the image registration from global homography to local thin-plate spline motion. It provides accurate alignment for overlapping regions and shape preservation for non-overlapping regions by joint optimization concerning alignment and distortion. Subsequently, to improve the generalization capability, we design a simple but effective iterative strategy to enhance the warp adaption in cross-dataset and cross-resolution applications. Finally, to further eliminate the parallax artifacts, we propose to composite the stitched image seamlessly by unsupervised learning for seam-driven composition masks. Compared with existing methods, our solution is parallax-tolerant and free from laborious designs of complicated geometric features for specific scenes. Extensive experiments show our superiority over the SoTA methods, both quantitatively and qualitatively. The code is available at https://github.com/nie-lang/UDIS2.

1. Introduction

Image stitching is a practical technology that aims to construct a scene with a wide field-of-view (FoV) from different images with limited FoV. It is useful in a wide range of fields, such as autonomous driving, medical imaging, surveillance videos, virtual reality, etc.

Over the past decades, traditional stitching approaches tend to adopt increasingly complicated geometric features to achieve better content alignment and shape preservation. In the beginning, SIFT [38] is widely used in various image stitching algorithms [4, 13, 50, 5, 34, 25] to extract discriminative key points and calculate adaptive warps. Then, the line segment is proved to be another unique feature to achieve better stitching quality and preserve linear structures [31, 49, 32, 19]. Recently, the large-scale edge is also introduced in [10] to preserve the contour structures. Besides, there is a great variety of other geometric features that are leveraged to improve the stitching quality, such as depth maps [33], semantic planar regions [26], etc.

Having calculated the warps, seam cutting is usually used to remove parallax artifacts. To explore an invisible seam, various energy functions are designed using colors [22], edges [35, 8], salient maps [30], depth [6], etc.

From the broad usage of geometric features, a clear developing trend has been discovered: increasingly sophisticated features are leveraged. We ask: are these complex designs practical in real applications? We attempt to answer this question from two perspectives. 1) These elaborate algorithms with complicated geometric features poorly adapt to scenes without sufficient geometric structures, such as medical images, industrial images, and other natural images with low texture (Fig.9b), low light or low resolution. 2) When there exist abundant geometric structures, the running speed is intolerant (please refer to Table 2,3 for detail). Such a trend seems to violate the "practical" original intent.

Recently, deep stitching technologies using convolutional neural networks (CNNs) have aroused widespread attention in the community. They abandon geometric features and head for high-level semantic features that can be adaptively learned in a data-driven pattern in a supervised [24, 40, 44, 47, 23], weakly-supervised [46], or unsupervised [41] manner. Although they are robust to various natural or unnatural conditions, they cannot handle large parallax and demonstrate unsatisfactory generalization in cross-dataset and cross-resolution conditions. A large-parallax case is shown in Fig.9a, where the tree is in the middle of the car in the reference image while it is on the left in the target image. To deal with parallax, UDIS [41] reconstructs stitched images from feature to pixel. However, the parallax is so large that undesired blurs are produced as a side effect.

In this paper, we propose a parallax-tolerant unsupervised deep image stitching technique, addressing the robustness issue in traditional stitching and the large-parallax issue in deep stitching simultaneously. Actually, the proposed deep learning-based solution is naturally robust to various scenes due to effective semantic feature extraction. Then, it overcomes the large parallax via two stages: warp and composition. In the first stage, we propose a robust and flexible warp to model the image registration. Particularly, we simultaneously parameterize homography transformation and thin-plate spline (TPS) transformation as unified representations in a compact framework. The former offers a global linear transformation, while the latter produces local nonlinear deformation, allowing our warp to align images with parallax. Besides, this warp contributes to both content alignment and shape preservation simultaneously via combined optimization of alignment and distortion. In the second stage, the existing reconstruction-based method [41] treats artifact elimination as a reconstruction process from feature to pixel, leading to inevitable blurs around the parallax regions. To overcome this drawback, we cooperate the motivation of seam-cutting into deep composition and implicitly find a "seam" through unsupervised learning for seam-driven composition masks. To this end, we design boundary and smoothness constraints to restrict the endpoints and route of a "seam", compositing the stitched image seamlessly. In addition to the two stages, we design a simple iterative strategy to enhance the generalization, rapidly improving the registration performance of our warp in different datasets and resolutions.

Furthermore, we conduct extensive experiments about the warp and composition, demonstrating our superiority to other SoTA solutions. The contributions center around:

- We propose a robust and flexible warp by parameterizing the homography and thin-plate spline into unified representations, realizing unsupervised content alignment and shape preservation in various scenes.
- A new composition approach is proposed to generate seamless stitched images via unsupervised learning for composition masks. Compared with the reconstruction [41], our composition eliminates parallax artifacts without introducing undesirable blurs.
- We design a simple iterative strategy to enhance warp adaption in different datasets and resolutions.

2. Related Work

2.1. Traditional Image Stitching

Adaptive warp. AutoStitch [4] leveraged SIFT [38] to extract discriminative keypoints to construct a global homography transformation. After that, SIFT becomes an indispensable feature to calculate various flexible warps, such as DHW [13], SVA [36] APAP [50], ELA [28], TFA [27] for better alignment, SPHP [5], AANAP [34], GSP [7] for better shape preservation. Then, DFW [13] adopted line segments extracted by LSD [48] with keypoints together to enrich structural information in artificial environments. Furthermore, line-guided mesh deformation [49] is designed by optimizing an energy function of various line-preserving terms [32, 19]. To preserve the nonlinear structures, the edge features are used in GES-GSP [10] to achieve a smooth transition between local alignment and structural preservation. In addition to these basic geometric features (point, line, and edge), the depth maps and semantic planes are also used to assist the feature matching using extra depth consistency [33] and planar consensus [26].

Seam cutting. The seam cutting is usually used as a post-processing operation to composite stitched images, which introduces an optimization problem of label assignment along the seam. To obtain a plausible stitched result, an extensive range of energy terms are defined by penalizing photometric differences, such as the Euclidean-metric color difference [22], gradient difference [1, 8], motion- and exposure-aware difference [11], salient difference [30], etc. Then these energy functions are minimized via graph-cut optimization [22]. Besides that, seam cutting is also applied in image alignment to find the best alignment warp with minimal seam-based cost [14, 51, 35, 29].

These complex geometric features are beneficial in natural scenes with adequate geometric structures. However, there are two drawbacks: 1) Without sufficient geometric structures, the strict feature requirements yield inferior stitching quality, even failure. 2) With excessive geometric structures, the computational cost leaps dramatically.

2.2. Deep Image Stitching

In contrast, deep stitching schemes are free from endless designs of geometric features. They learn to capture high-level semantic features from extensive data automatically in a supervised [24, 40, 44, 47, 23], weakly-supervised [46], or unsupervised [41] fashion, making them robust to various challenging scenes. Among them, the unsupervised one [41] is more popular due to the unavailability of real stitched labels. However, it cannot handle large parallax due to the limitation of the homography-based alignment model. The subsequent reconstruction would bring undesirable blurs around parallax regions.

3. Methodology

The overview of our method is shown in Fig.2, where the proposed framework is composed of two stages: warp and composition. In the first stage, our method takes a reference image (I_r) and a target image (I_t) with overlapping regions as input, and regresses a robust and flexible warp. Then the warped images (I_{wr}, I_{wt}) are input to the second stage to predict composition masks (M_{cr}, M_{ct}). The stitched image (S) can be seamlessly composited as follows:

$$S = M_{cr} \times I_{wr} + M_{ct} \times I_{wt}. \tag{1}$$

3.1. Unsupervised Warp Construction

3.1.1 Warp Parameterization

The homography transformation is an invertible mapping from one image to another with 8 degrees of freedom: each two for translation, rotation, scale, and lines at infinity. To guarantee the non-singularity [39] in a regression network, it is commonly parameterized as the motions of four vertices [9], which is solved as a 3×3 matrix using DLT [15].

However, if a non-planar scene is captured by cameras with different shooting centers, the homography fails to achieve accurate alignment. To solve it, the mesh-based multi-homography scheme [50] is usually used in traditional stitching algorithms. But it cannot be efficiently parallel accelerated, which means it fails to be used in a deep learning framework [43, 42]. *Please refer to Section 2.3 of the supplementary material for specific analysis.* To overcome this issue, we propose to leverage TPS transformation [3, 18] to achieve efficient local deformation.

TPS transformation is a nonlinear, flexible transformation that is usually used to approximate the deformation of non-rigid objects using a thin plate. It is determined by two sets of control points, with a one-to-one correspondence between a flat image and a warped image. Denote N control points on a flat image as $P = [p_1, ..., p_N]^T$ and corresponding points on the warped image as $P' = [p'_1, ..., p'_N]^T$ ($p_i, p'_i \in \mathbb{R}^{2\times1}$). By minimizing an energy function consisting of a data term and a distortion term [20] (*refer to Section 2.1 of the supplementary material for more details*), the TPS transformation can be parameterized as Eq.2:

$$p' = \mathscr{T}(p) = C + Mp + \sum_{i=1}^{N} w_i O(\parallel p - p_i \parallel_2), \tag{2}$$

where p is an arbitrary point on the flat image and p' is the corresponding point on the warped image. $C \in \mathbb{R}^{2\times1}$, $M \in \mathbb{R}^{2\times2}$, and $w_i \in \mathbb{R}^{2\times1}$ are the transformation parameters. $O(r) = r^2 log r^2$ is a radial basis function that indicates the impact of each control point on p. To solve these parameters, we formulate N data constraints using N pairs of control points according to Eq.2, and impose extra dimensional constraints [20] as described in Eq.3:

$$\sum_{i=1}^{N} w_i = 0 \;\; and \;\; \sum_{i=1}^{N} p_i w_i^T = 0. \tag{3}$$

Then, these constraints can be rewritten in the form of matrix calculation and the parameters can be solved as follows:

$$\begin{bmatrix} C \\ M \\ W \end{bmatrix} = \begin{bmatrix} \mathbb{1} & P & K \\ 0 & 0 & \mathbb{1}^T \\ 0 & 0 & P^T \end{bmatrix}^{-1} \begin{bmatrix} P' \\ 0 \\ 0 \end{bmatrix}, \tag{4}$$

where $\mathbb{1}$ is a $N \times 1$ all-one matrix. Each element k_{ij} in $K \in \mathbb{R}^{N\times N}$ is determined by $O(\parallel p_i - p_j \parallel_2)$, and $W = [w_i, ..., w_N]^T$.

Similar to the 4-pt parameterization of the homography, TPS transformation can also be parameterized as the motions of control points. In this work, we define $(U+1) \times (V+1)$ control points being evenly distributed on the target image, and then predict the motions of each control point. To bridge the global homography warp with the local TPS warp, we regress the homography transformation first to provide initial motions of control points. Then we can predict the residual motions for further flexible deformation.

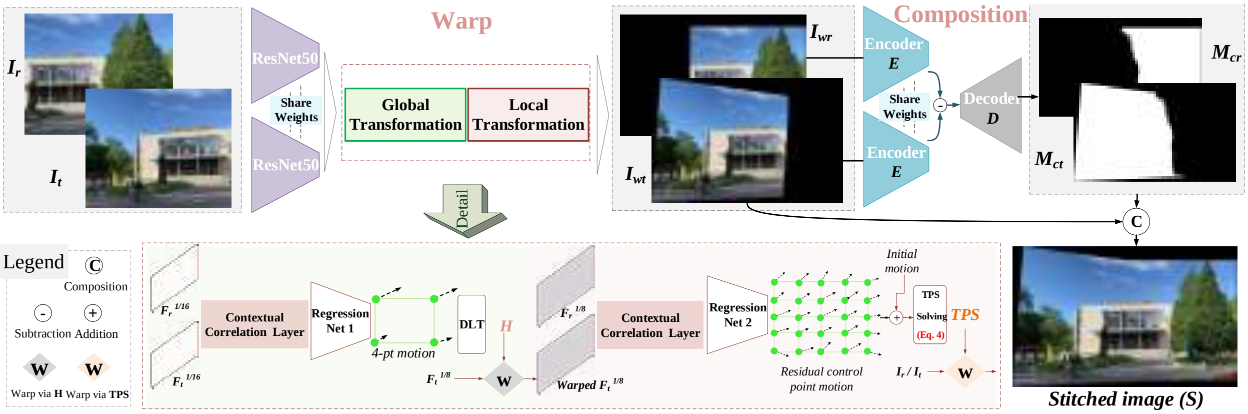


Figure 2: An overview of the proposed parallax-tolerant unsupervised stitching network. Our framework consists of two stages: warp and composition. The first stage predicts a robust and flexible warp to align images with shape preservation. The second stage composites the seamless stitched image by generating composition masks corresponding to warped images.

3.1.2 Pipeline of Warp

As shown in Fig.2, given I_r, I_t, we adopt ResNet50 [17] with pretrained parameters as our backbone to extract semantic features first. It maps a 3-channel image to the high-dimensional semantic features with a resolution scaled to 1/16 of the original. Then the correlation between these feature maps ($F_r^{1/16}$ and $F_t^{1/16}$) can be aggregated into 2-channel feature flows using the contextual correlation layer [43]. Subsequently, a regression network is used to estimate the 4-pt parameterization of the homography warp. This global warp also generates the initial motions of control points.

Next, we warp the feature maps with higher resolution ($F_t^{1/8}$) to embed the homographic prior into the following workflow. After another contextual correlation layer and regression network, the residual motions of control points are predicted, contributing to a robust flexible TPS warp.

3.1.3 Optimization of Warp

To achieve content alignment and shape preservation simultaneously, we design our objective function $\mathcal{L}^w$ concerning two aspects: alignment and distortion.

$$\mathcal{L}^w = \mathcal{L}^w_{alignment} + \omega \mathcal{L}^w_{distortion}. \tag{5}$$

For the alignment, we encourage the overlapping regions to keep consistent at the pixel level. Denoting $\varphi(\cdot,\cdot)$ is the warping operation and $\mathbb{1}$ an all-one matrix with the same resolution as I_r, the alignment loss can be defined as follows:

$$\begin{aligned}\mathcal{L}^w_{alignment} =& \lambda\|I_r \cdot \varphi(\mathbb{1},\mathcal{H}) - \varphi(I_t,\mathcal{H})\|_1 + \\ & \lambda\|I_t \cdot \varphi(\mathbb{1},\mathcal{H}^{-1}) - \varphi(I_r,\mathcal{H}^{-1})\|_1 + \\ & \|I_r \cdot \varphi(\mathbb{1},\mathcal{TPS}) - \varphi(I_t,\mathcal{TPS})\|_1,\end{aligned} \tag{6}$$

where $\mathcal{H}$ and $\mathcal{TPS}$ are warp parameters, and λ is a hyperparameter to balance the impacts of different transformations.

For the distortion, we link adjacent control points in the warped target image to form a mesh and introduce an inter-grid constraint ℓ_{inter} and an intra-grid constraint ℓ_{intra}. The former preserves geometric structures for non-overlapping regions, while the latter reduces projective distortions. In the beginning, we approximate a similar transformation by DLT for every grid in non-overlapping regions and take the 4-pt projective error as the loss. But this constraint that is commonly used in traditional methods [16, 37] does not work in deep learning schemes. Instead, we re-explore the constraints from a more intuitive perspective — the grid edge.

Similar to [42], we penalize the grid edge $\vec{e}$ with the magnitude exceeding a threshold. Denoting $\{\vec{e}_{hor}\}$ and $\{\vec{e}_{ver}\}$ are the collections of horizontal and vertical edges, we describe the intra-grid constraint as follows:

$$\begin{aligned}\ell_{intra} =& \frac{1}{(U{+}1)\times V}\sum_{\{\vec{e}_{hor}\}}\sigma(\langle\vec{e},\vec{i}\rangle - \frac{2W}{V}) + \\ & \frac{1}{U\times(V{+}1)}\sum_{\{\vec{e}_{ver}\}}\sigma(\langle\vec{e},\vec{j}\rangle - \frac{2H}{U}),\end{aligned} \tag{7}$$

where $\vec{i}$ / $\vec{j}$ is the horizontal/vertical unit vector, and $\sigma(\cdot)$ is the $RELU$ function. The projective distortions are reduced by preventing the grid shape from dramatic scaling.

By encouraging the edge pairs (successive edges in horizontal or vertical directions, denoted as $\vec{e}_{s1}, \vec{e}_{s2}$) to be co-linear, we formulate the inter-grid constraint as:

$$\ell_{inter} = \frac{1}{Q}\sum_{\{\vec{e}_{s1},\vec{e}_{s2}\}} S_{s1,s2}\cdot(1 - \frac{\langle\vec{e}_{s1},\vec{e}_{s2}\rangle}{\|\ \vec{e}_{s1}\ \|\cdot\|\ \vec{e}_{s2}\ \|}), \tag{8}$$

where Q is the number of edge pairs and $S_{s1,s2}$ is a 0-1 label that is set to 1 if this edge pair locates on non-overlapping regions. We only preserve the structures in non-overlapping regions, preventing adverse effects on the alignment.

3.2. Unsupervised Seamless Composition

3.2.1 Motivation

UDIS [41] composites a stitched image via unsupervised reconstruction from feature to pixel, but it cannot deal with large parallax. Traditional seam cutting eliminates artifacts by finding a seamless cutting path using dynamic programming [2] or graph-cut optimization [22], but it shows over-reliance on photometric differences.

An intuitive idea is to cooperate the motivation of seam cutting into a learning framework. Nevertheless, how to make our unsupervised deep stitching approach work with seam cutting and be effective is a major difficulty. For example, dynamic programming is not differential; graph-cut optimization assigns absolute integers to the labels, which truncates gradients in the backpropagation. In this stage, we propose to relax the hard label to a *soft mask* with float numbers, innovatively supervising the generation of seam-inspired masks via the balancing effect of two constraints with special designs.

3.2.2 Pipeline of Composition

At first, we concatenate warped images as input and exploit the UNet-like network [45] as our composition network. But this pattern coarsely mixes the features from different images. It is challenging for such a network to perceive the semantic difference between warped images.

To overcome it, we use the encoder of the network to extract semantic features from I_{wr} and I_{wt} separately with shared weights. For skip connections, we replace them by subtracting the features of I_{wt} from that of I_{wr} and delivering the residuals at each resolution to the decoder. We set the filter number and activation function of the last layer to 1 and *sigmoid* to predict M_{cr} for the warped reference image. The other mask M_{ct} for the warped target image can be easily obtained by simple post-processing.

3.2.3 Optimization of Composition

The optimization goal of our unsupervised composition includes a boundary term and a smoothness term as follows:

$$\mathcal{L}^c = \alpha \mathcal{L}^c_{boundary} + \beta \mathcal{L}^c_{smoothness}. \tag{9}$$

The former indicates the start point and end point of the "seam" while the latter constrains the route.

We expect the endpoints to be the intersections of the boundaries of warped images. To achieve it, we leverage 0-1 boundary masks M_{br}, M_{bt} to indicate the boundary positions of overlapping regions on both sides of the "seam". *More details are available in Section 3.1 of the supplementary material.* Then, we formulate the boundary loss as follows:

$$\mathcal{L}^c_{boundary} = \| (S - I_{wr}) \cdot M_{br} \|_1 + \| (S - I_{wt}) \cdot M_{bt} \|_1 . \tag{10}$$

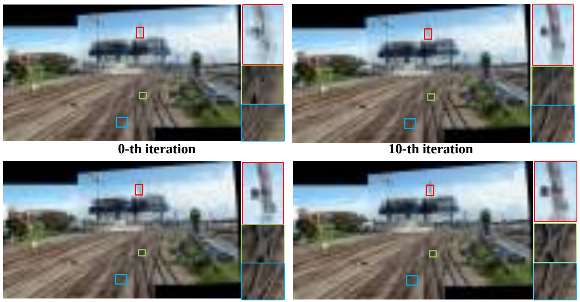


Figure 3: We demonstrate the process of iterative warp adaption on "railtrack" dataset [50] (cross-dataset and cross-resolution).

This loss constrains boundary pixels of overlapping regions in S from either I_{wr} or I_{wt}. However, M_{br} and M_{bt} share common intersections, which produces ambiguity for the belongs of intersections. But it is the ambiguity that fixes the endpoints of a "seam" to the intersections.

To measure the smoothness of a seam, traditional seam-cutting approaches define various energy functions with different photometric differences. In this work, we adopt the simplest photometric difference as $D = (I_{wr} - I_{wt})^2$ to demonstrate our effectiveness. Then we define the smoothness on the difference map as follows:

$$\begin{aligned} \ell_D = & \sum_{i,j} |M^{i,j}_{cr} - M^{i+1,j}_{cr}|(D^{i,j} + D^{i+1,j}) + \\ & \sum_{i,j} |M^{i,j}_{cr} - M^{i,j+1}_{cr}|(D^{i,j} + D^{i,j+1}), \end{aligned} \tag{11}$$

where i, j are the Cartesian coordinates. To produce a smooth transition between both sides of the "seam", we also define the smoothness of the stitched image as follows:

$$\begin{aligned} \ell_S = & \sum_{i,j} |M^{i,j}_{cr} - M^{i+1,j}_{cr}| \cdot |S^{i,j} - S^{i+1,j}| + \\ & \sum_{i,j} |M^{i,j}_{cr} - M^{i,j+1}_{cr}| \cdot |S^{i,j} - S^{i,j+1}|. \end{aligned} \tag{12}$$

By adding ℓ_D and ℓ_S, we formulate the complete smoothness term $\mathcal{L}^c_{smoothness}$. Note that, our network is trained to facilitate the capability to extract semantic differences. In the inference process, the proposed method no longer relies on photometric differences.

3.3. Iterative Warp Adaption

To transfer a pretrained model to other datasets (cross-scene and cross-resolution), the most common way is to fine-tune on the new dataset. However, it usually requires labels to assist the adaption process. In this work, we address this limitation by setting an unsupervised optimization goal as follows:

$$\mathcal{L}^w_{adaption} = \| I_r \cdot \varphi(\mathbb{1}, \mathcal{TPS}) - \varphi(I_t, \mathcal{TPS}) \|_1 . \tag{13}$$

Compared with Eq. 5, we remove the homography alignment loss and distortion loss. Because these constraints have been well learned by the pretrained model, what we do is adjust the local alignment on different data.

Furthermore, we consider a special case that the new dataset only contains one sample. Experiments exhibit that our model can also be optimized stably for adapting to only sample in an iterative fashion. In particular, we set a threshold τ and a maximum iteration number T. The adaption process stops when the iteration number reaches T or consecutive optimization errors (Eq. 13) are lower than τ.

We show an iterative adaption example in Fig. 3, where the artifacts are significantly reduced with the increase of iteration number. It takes about 0.1s to finish an iteration.

4. Experiments

4.1. Dataset and Implement Details

Dataset: To make an intuitive and fair comparison with deep stitching methods, we also train our model on UDIS-D [41] dataset. The evaluation is conducted on UDIS-D dataset and other traditional datasets [50, 13, 34, 28, 35].

Details: We train our warp and composition networks for 100 and 50 epochs using Adam [21] with an exponentially decaying learning rate with an initial value of 10^{-4}. For the warp stage, ω and λ are set to 10 and 3, and we adopt $(12 + 1) \times (12 + 1)$ control points to provide the flexible TPS transformation. For the second stage, we set α and β to 10,000 and 1,000. As for the warp adaption, τ and T are assigned as 10^{-4} and 50. All implementations are based on PyTorch using a single GPU with NVIDIA RTX 3090 Ti.

4.2. Comparative Experiments

To demonstrate our effectiveness comprehensively, we conduct extensive experiments on warp, composition, and the complete stitching framework, respectively.

4.2.1 Comparisons of Warp

We compare our warp with SIFT [38]+RANSAC [12] (the pipeline of AutoStitch [4]), APAP [50], ELA [28], SPW [32], LPC [19], and UDIS's warp [41]. We implement SIFT+RANSAC by ourselves and adopt the official codes for other methods with default parameters such as mesh resolutions. All the methods, including ours, use the average fusion as the post-processing operation. Because this simple fusion is fast and can better highlight the misalignments.

Quantitative comparison: We first carry on quantitative comparisons with the same metrics as UDIS [41] on UDIS-D dataset [41] that has 1,106 samples for the evaluation. The results are shown in Table 1, where $I_{3\times3}$ takes the identity matrix as a "no-warping" transformation for reference. The results are divided into three parts according to the performance as [41, 43]. The programs of traditional methods might crash in some challenging samples due to the lack of geometric features. When that happens, we use $I_{3\times3}$ as an alternative transformation for the evaluation.

Qualitative comparison: Qualitative results are shown in Fig. 4, where we zoom in on two regions at different depth surfaces to highlight parallax artifacts. From this figure, our warp outperforms the other solutions by a large margin on UDIS-D dataset [41].

Cross-dataset comparison: We use the pretrained model to evaluate our performance on other datasets, as illustrated in Fig. 5. The iterative adaption strategy is used to further improve the alignment performance.

Speed comparison: To evaluate the speed objectively, we test it on three traditional public datasets [50, 34, 13] with three different resolutions. As reported in Table 2, our warp has a speed far exceeding the others with GPU acceleration, while traditional warps cannot be accelerated by GPU. For traditional mesh-based warps, the runtime does not vary linearly with the resolution, and in scenes with rich geometric features (e.g., "railTrack"), the speed becomes a disaster.

4.2.2 Comparisons of Composition

We compare our composition with the perception-based seam-cutting approach [30] and reconstruction-based method [41]. To show the parallax artifacts more intuitively, we warp the images by SIFT+RANSAC and give the results of average fusion for reference.

Qualitative comparison: Traditional seam-cutting methods find the seam by dynamic programming [2] or graph-cut optimization [22]. The values in traditional masks are integers while that in ours are float. Therefore, we cannot evaluate our composition quantitatively with traditional indicators. Instead, we show qualitative results in Fig. 6. Besides, we promise to release all subjective results, including 1,106 images in UDIS-D and others in traditional datasets.

Speed comparison: Here, we warp the inputs with the proposed warp first. Then these warped images are used for speed evaluation on different composition methods. As illustrated in Table. 3, our composition shows significant speed superiority over the others with GPU acceleration.

4.2.3 More Comparisons

Here, we evaluate the performance of our complete stitching framework with other SoTA methods. The results are illustrated in Fig. 9, where LPC [19] and UDIS [41] adopt the perception-based seam cutting [30] and reconstruction [41] for the post-processing operations. *For clarity, more experimental results including qualitative comparisons, user studies, challenging cases, and cross-dataset evaluations are depicted in the supplementary material.*

4.3. Ablation studies

We first conduct ablation studies on different warp constraints. As shown in Fig. 7(top), the inter-gird constraint preserves the structures whiles the intra-grid one reduces

Table 1: Quantitative comparison of warp on UDIS-D dataset [41]. The best is marked in red and the second best is in blue.

	PSNR ↑				SSIM ↑			
	Easy	Moderate	Hard	Average	Easy	Moderate	Hard	Average
$I_{3\times3}$	15.87	12.76	10.68	12.86	0.530	0.286	0.146	0.303
SIFT[38]+RANSAC[12]	28.75	24.08	18.55	23.27	0.916	0.833	0.636	0.779
APAP[50]	27.96	24.39	**20.21**	23.79	0.901	0.837	0.682	0.794
ELA[28]	**29.36**	**25.10**	19.19	**24.01**	**0.917**	**0.855**	**0.691**	**0.808**
SPW[32]	26.98	22.67	16.77	21.60	0.880	0.758	0.490	0.687
LPC[19]	26.94	22.63	19.31	22.59	0.878	0.764	0.610	0.736
UDIS's warp[41]	25.16	20.96	18.36	21.17	0.834	0.669	0.495	0.648
Our warp	**30.19**	**25.84**	**21.57**	**25.43**	**0.933**	**0.875**	**0.739**	**0.838**

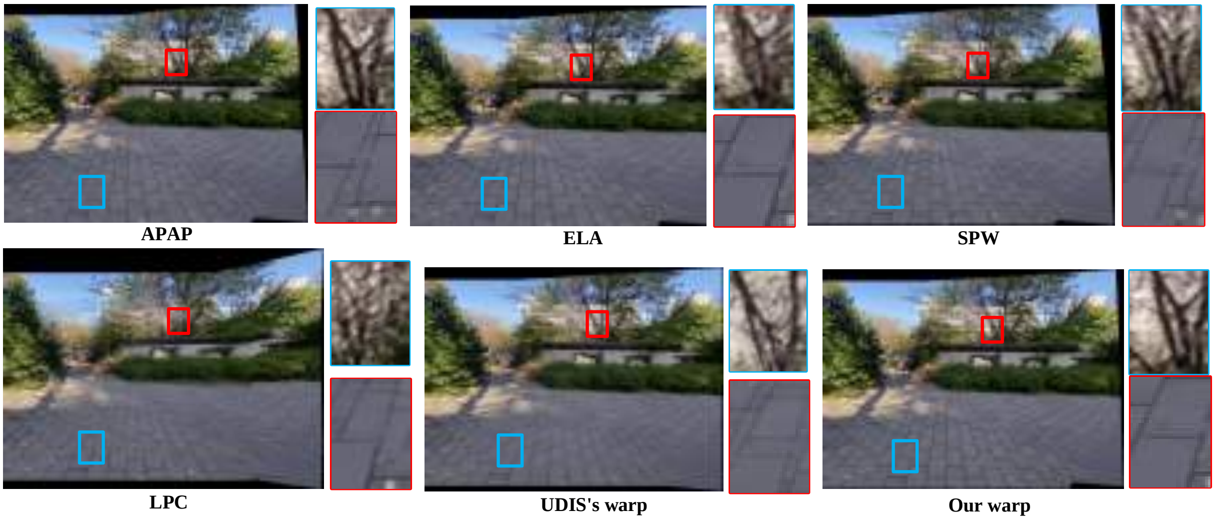

APAP ELA SPW LPC UDIS's warp Our warp

Figure 4: Qualitative comparison of warp on UDIS-D dataset [41]. We zoom in on a near region and a far region to show the alignment performance. *For clarity, we show the inputs and more comparative results in the supplementary material.*

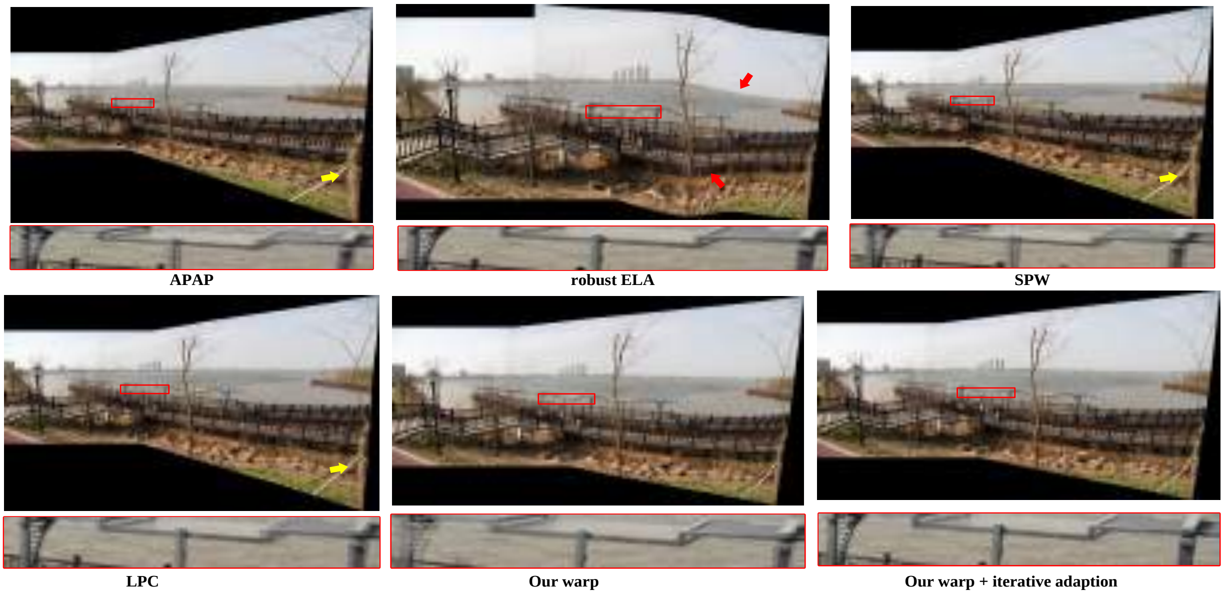

APAP robust ELA SPW LPC Our warp Our warp + iterative adaption

Figure 5: Qualitative comparison of warp on "boardingBridge" dataset [28] with a resolution of 1440 × 2160 for inputs. The yellow and red arrows highlight projective and structural distortions.

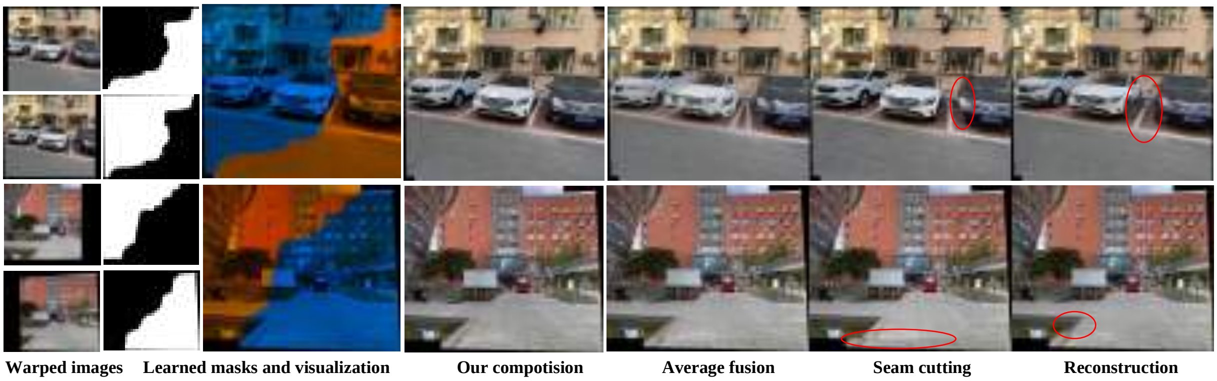


Figure 6: The comparison of composition. *For clarity, more results are reported in the supplementary material.*

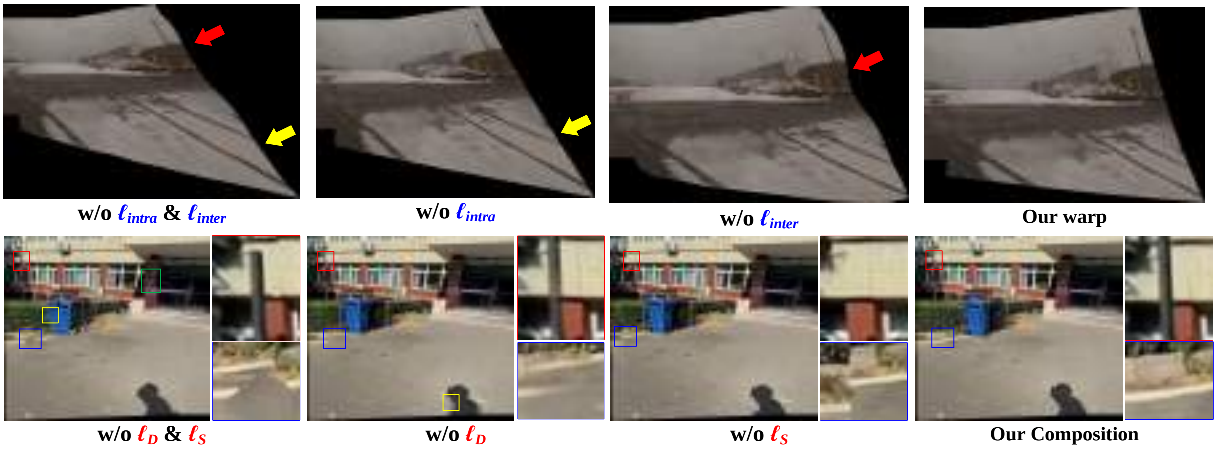


Figure 7: Ablation studies on our warp and composition. Top: the red and yellow arrows highlight the structural and projective distortions, respectively. Bottom: the rectangles indicate the discontinuous regions.

Table 2: Comparison of warp on elapsed time (s). 1: tested with Intel i7-9750H 2.60GHz CPU; 2: tested with NVIDIA RTX 3090Ti GPU.

Dataset	Railtrack [50]	Fence [34]	Carpark [13]
Resolution	1500 × 2000	1088 × 816	490 × 653
APAP [50]^1	20.921	4.427	2.005
ELA [28]^1	18.982	4.739	2.179
SPW [32]^1	227.762	4.787	6.583
LPC [19]^1	2805.3	9.115	40.443
Our warp^1	12.073	5.025	3.486
Our warp^2	**0.731**	**0.210**	**0.117**

Table 3: Comparison of composition on elapsed time (s). 1: tested with Intel i7-9750H 2.60GHz CPU; 2: tested with NVIDIA RTX 3090Ti GPU.

Dataset	Railtrack [50]	Fence [34]	Carpark [13]
Resolution (after warping)	1831 × 3193	1298 × 1320	718 × 1186
Seam cutting [30]^1	46.657	4.058	0.873
Reconstruction [41]^1	304.963	80.837	10.734
Our composition^1	22.778	6.666	3.286
Our composition^2	**0.532**	**0.143**	**0.071**

projective distortions. Moreover, these constraints bring little adverse impact on alignment. *Quantitative results are reported in the supplementary material.*

Then we study the impacts of smoothness term in our composition. The results are shown in Fig. 7(bottom), where we highlight the discontinuous regions by rectangles. With the smoothness constraints on the difference map and stitched image, the discontinuity is significantly improved.

5. Conclusion

In this paper, we propose a parallax-tolerant unsupervised deep stitching solution. First, a robust flexible warp is adaptively learned for both content alignment and shape preservation. We also present the seam-inspired composition to further reduce artifacts. Besides, a simple iterative warp adaption strategy is designed to effectively enhance the generalization in cross-dataset and cross-resolution cases. Compared with existing solutions, our method can address both challenging scenes and large-parallax cases. With increasingly popular GPUs, our solution exhibits incredible efficiency.

References

[1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *SIGGRAPH*, pages 294–302. 2004. 3

[2] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *TOG*, 26(3):10–es, 2007. 5, 6

[3] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 11(6):567–585, 1989. 3, 11

[4] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 74(1):59–73, 2007. 1, 2, 6

[5] Che-Han Chang, Yoichi Sato, and Yung-Yu Chuang. Shape-preserving half-projective warps for image stitching. In *CVPR*, pages 3254–3261, 2014. 1, 2

[6] Xin Chen, Mei Yu, and Yang Song. Optimized seam-driven image stitching method based on scene depth information. *Electronics*, 11(12):1876, 2022. 2

[7] Yu-Sheng Chen and Yung-Yu Chuang. Natural image stitching with the global similarity prior. In *ECCV*, pages 186–201, 2016. 2

[8] Qinyan Dai, Faming Fang, Juncheng Li, Guixu Zhang, and Aimin Zhou. Edge-guided composition network for image stitching. *PR*, 118:108019, 2021. 2, 3

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 3

[10] Peng Du, Jifeng Ning, Jiguang Cui, Shaoli Huang, Xinchao Wang, and Jiaxin Wang. Geometric structure preserving warp for natural image stitching. In *CVPR*, pages 3688–3696, 2022. 2

[11] Ashley Eden, Matthew Uyttendaele, and Richard Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *CVPR*, pages 2498–2505, 2006. 3

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 6, 7

[13] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *CVPR*, pages 49–56, 2011. 1, 2, 6, 8, 14, 19

[14] Junhong Gao, Yu Li, Tat-Jun Chin, and Michael S Brown. Seam-driven image stitching. In *Eurographics*, pages 45–48, 2013. 3, 15

[15] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3

[16] Kaiming He, Huiwen Chang, and Jian Sun. Rectangling panoramic images via warping. *TOG*, 32(4):1–10, 2013. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *NIPS*, 28, 2015. 3

[19] 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. 1, 2, 6, 7, 8, 14, 15, 18

[20] JT Kent and KV Mardia. The link between kriging and thin-plate splines. *Probability, Statistics and Optimization*, pages 326–339, 1994. 3

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[22] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *TOG*, 22(3):277–286, 2003. 2, 3, 5, 6

[23] Hyeokjun Kweon, Hyeonseong Kim, Yoonsu Kang, Youngho Yoon, Wooseong Jeong, and Kuk-Jin Yoon. Pixel-wise deep image stitching. *arXiv preprint arXiv:2112.06171*, 2021. 2, 3

[24] Wei-Sheng Lai, Orazio Gallo, Jinwei Gu, Deqing Sun, Ming-Hsuan Yang, and Jan Kautz. Video stitching for linear camera arrays. *arXiv preprint arXiv:1907.13622*, 2019. 2, 3

[25] Kyu-Yul Lee and Jae-Young Sim. Warping residual based image stitching for large parallax. In *CVPR*, pages 8198–8206, 2020. 1

[26] Aocheng Li, Jie Guo, and Yanwen Guo. Image stitching based on semantic planar region consensus. *TIP*, 30:5545–5558, 2021. 2

[27] Jing Li, Baosong Deng, Rongfu Tang, Zhengming Wang, and Ye Yan. Local-adaptive image alignment based on triangular facet approximation. *TIP*, 29:2356–2369, 2019. 2

[28] 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, 6, 7, 8, 12, 14, 17, 19

[29] Jiaxue Li and Yicong Zhou. Automatic color image stitching using quaternion rank-1 alignment. In *CVPR*, pages 19720–19729, 2022. 3

[30] Nan Li, Tianli Liao, and Chao Wang. Perception-based seam cutting for image stitching. *Signal, Image and Video Processing*, 12(5):967–974, 2018. 2, 3, 6, 8, 12, 13, 14, 18

[31] Shiwei Li, Lu Yuan, Jian Sun, and Long Quan. Dual-feature warping-based motion model estimation. In *ICCV*, pages 4283–4291, 2015. 2

[32] Tianli Liao and Nan Li. Single-perspective warps in natural image stitching. *TIP*, 29:724–735, 2019. 2, 6, 7, 8

[33] Tianli Liao and Nan Li. Natural image stitching using depth maps. *arXiv preprint arXiv:2202.06276*, 2022. 2

[34] Chung-Ching Lin, Sharathchandra U Pankanti, Karthikeyan Natesan Ramamurthy, and Aleksandr Y Aravkin. Adaptive as-natural-as-possible image stitching. In *CVPR*, pages 1155–1163, 2015. 1, 2, 6, 8

[35] 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. 2, 3, 6, 14, 18, 19, 20

[36] Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. Smoothly varying affine stitching. In *CVPR*, pages 345–352, 2011. 2

[37] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3d video stabilization. *TOG*, 28(3):1–9, 2009. 4

[38] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2, 6, 7

[39] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018. 3

[40] Lang Nie, Chunyu Lin, Kang Liao, Meiqin Liu, and Yao Zhao. A view-free image stitching network based on global homography. *Journal of Visual Communication and Image Representation*, 73:102950, 2020. 2, 3

[41] 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. 1, 2, 3, 5, 6, 7, 8, 13, 14, 15, 16, 18, 20

[42] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Deep rectangling for image stitching: A learning baseline. In *CVPR*, pages 5740–5748, 2022. 3, 4

[43] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Depth-aware multi-grid deep homography estimation with contextual correlation. *CSVT*, 32(7):4460–4472, 2022. 3, 4, 6

[44] Lang Nie, Chunyu Lin, Kang Liao, and Yao Zhao. Learning edge-preserved image stitching from multi-scale deep homography. *Neurocomputing*, 491:533–543, 2022. 2, 3

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[46] Dae-Young Song, Geonsoo Lee, HeeKyung Lee, Gi-Mun Um, and Donghyeon Cho. Weakly-supervised stitching network for real-world panoramic image generation. *arXiv preprint arXiv:2209.05968*, 2022. 2, 3

[47] Dae-Young Song, Gi-Mun Um, Hee Kyung Lee, and Donghyeon Cho. End-to-end image stitching network via multi-homography estimation. *SPL*, 28:763–767, 2021. 2, 3

[48] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *TPAMI*, 32(4):722–732, 2008. 2

[49] Tian-Zhu Xiang, Gui-Song Xia, Xiang Bai, and Liangpei Zhang. Image stitching by line-guided local warping with global similarity constraint. *PR*, 83:481–497, 2018. 2

[50] 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. 1, 2, 3, 5, 6, 7, 8, 11, 13, 14, 19

[51] Fan Zhang and Feng Liu. Parallax-tolerant image stitching. In *CVPR*, pages 3262–3269, 2014. 3, 14, 19

A. Supplemental Material

In this document, we provide the following supplementary contents:

- Details of warp (Section B).
- Details of composition (Section C).
- Analysis on robustness and distortion (Section D).
- More results (Section E).

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.

B. More Details of Warp

B.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 [3]:

$$\varepsilon = \varepsilon_{alignment} + \lambda \varepsilon_{distortion}, \tag{14}$$

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

$$\begin{aligned} \varepsilon_{alignment} &= \sum_{i=1}^{N} \parallel p' - \mathscr{T}(p) \parallel^2, \\ \varepsilon_{distortion} &= \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 \mathscr{T}}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 \mathscr{T}}{\partial x \partial y} \right)^2 \right. \\ &\quad \left. + \left(\frac{\partial^2 \mathscr{T}}{\partial y^2} \right)^2 \right) dxdy, \end{aligned} \tag{15}$$

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. 14, we are able to determine the warp function, which is derived as follows (see Eq. 2 in the manuscript):

$$p' = \mathscr{T}(p) = C + Mp + \sum_{i=1}^{N} w_i O(\parallel p - p_i \parallel_2). \tag{16}$$

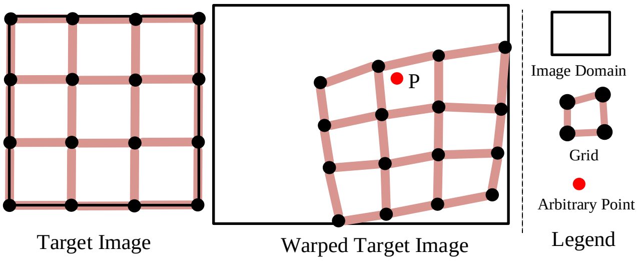


Figure 8: Backward interpolation.

B.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 avoid increasing λ in Eq. 14 and instead assume that control points are evenly distributed in the target image, and their motion is smooth. We form a mesh by connecting control points and introduce an intra-grid constraint (Eq. 7 of the manuscript) and an inter-grid constraint (Eq. 8 of the manuscript) 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.

B.3. Multiple Homography vs. TPS

The TPS warp is more appropriate than the traditional mesh-based multi-homography warp [50] 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. 8. 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. 8, 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. 16), 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.

B.4. Difference to Stitching Methods using TPS

The existing stitching methods using TPS are all traditional feature-based solutions. For example, ELA [28] calculates TPS transformations using matched keypoints such as SIFT. This transformation is then processed to reduce computational cost and distortions.

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).

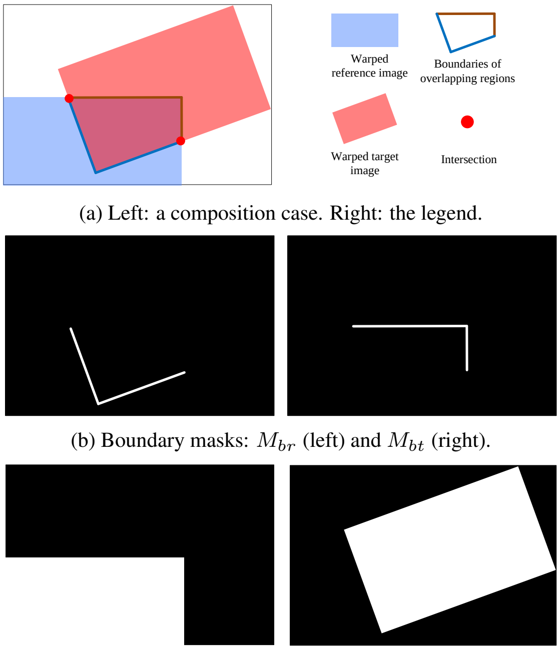
C. More Details of Composition

C.1. Boundary Term

Considering a composite case (Fig. 9a), we aim to fix the endpoints of a seam on the intersections. To achieve this, we define two boundary masks, as shown in Fig. 9b: 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:

$$\mathcal{L}^{c}_{boundary} = \| (S - I_{wr}) \cdot M_{br} \|_1 + \| (S - I_{wt}) \cdot M_{bt} \|_1 . \tag{17}$$

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.9a. These common pixels inevitably yield ambiguity for the belongs of intersections, and the ambiguity helps to determine the seam endpoints.



(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 9: Details of the boundary term for the composition.

Next, we describe how to get the boundary masks. Given the warped masks M_r, M_t (as shown in Fig. 9c), we obtain boundary masks by the following formulation:

$$\begin{aligned} M_{br} &= M_r \cdot \mathscr{E}(M_t), \\ M_{bt} &= M_t \cdot \mathscr{E}(M_r), \end{aligned} \tag{18}$$

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

C.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. 10 shows the masks from seam cutting [30] and ours, where our "seam" is significantly wider. That is why we cannot quantitatively evaluate our composition in traditional metrics.

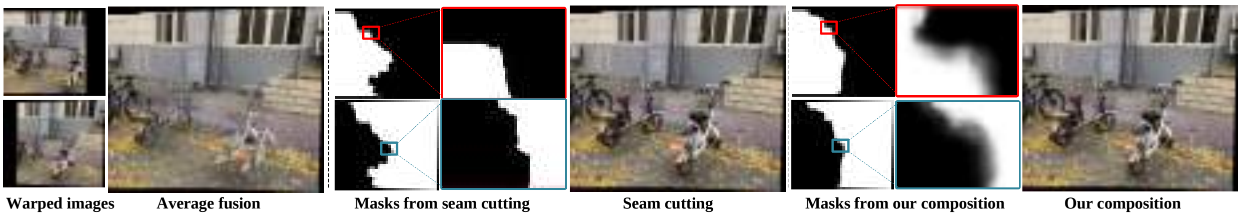
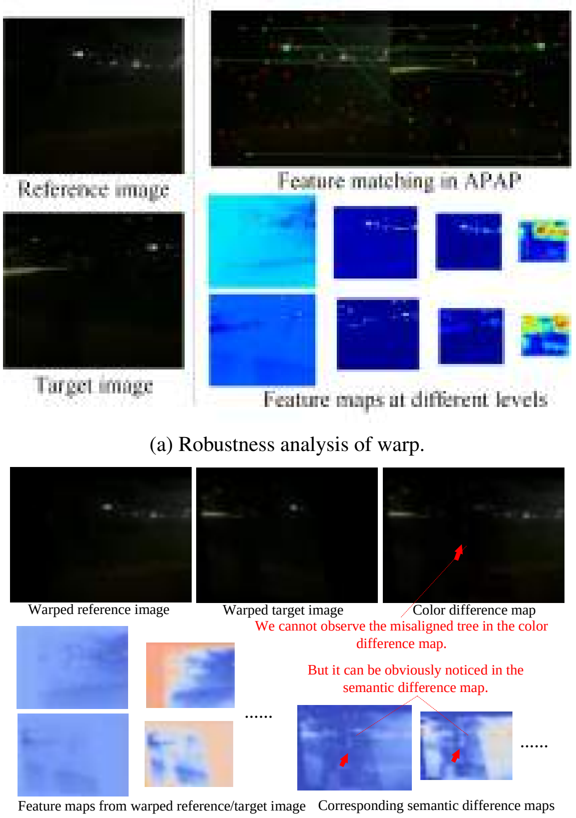


Figure 10: The difference between masks from seam cutting [30] and our composition.


Reference image
Feature matching in APAP
Target image
Feature maps at different levels
Warped reference image
Warped target image
Color difference map
We cannot observe the misaligned tree in the color difference map.
But it can be obviously noticed in the semantic difference map.
......
......
Feature maps from warped reference/target image
Corresponding semantic difference maps

(a) Robustness analysis of warp.

(b) Robustness analysis of composition.

Figure 11: Robustness analysis.

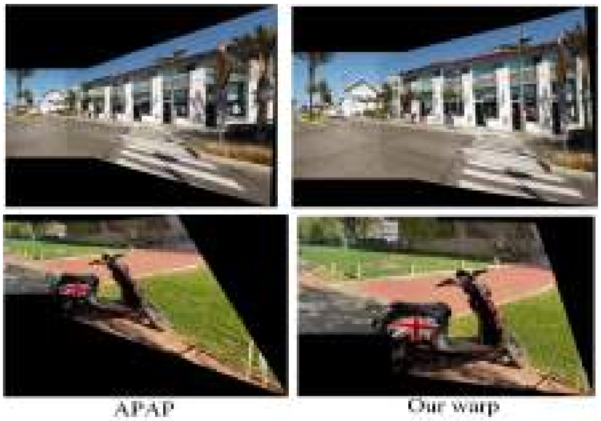


Figure 12: Projective distortion: APAP vs. ours. These instances are from UDIS-D dataset[41].

D. Analysis

D.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 [50], which represents traditional solutions. In Fig. 11a, 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. 11a, 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.

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. 11b displays an example where the tree (highlighted by red arrows) is missing in the color difference map. The proposed deep composition method overcomes 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. 11b, where the tree can be obviously noticed in semantic difference maps.

D.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).

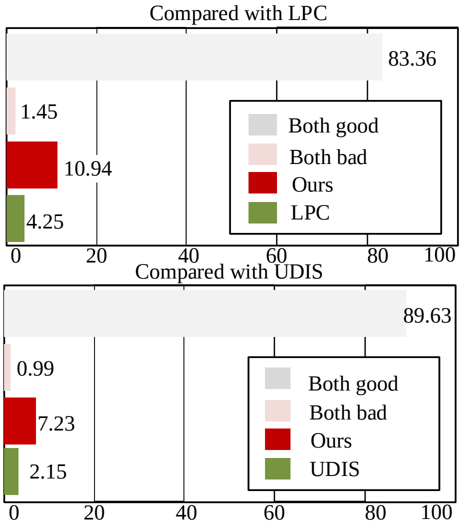


Figure 13: User study of visual preferences with existing SoTA solutions. The results are presented in percentage and averaged on 20 participants.

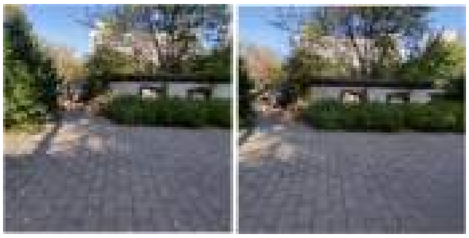


Figure 14: The input images of Fig. 4 in the manuscript.

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.

E. More Results

E.1. Results of Warp

The Fig. 14,15 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. 18 and Fig. 19.

E.2. Results of Composition

Here, we illustrate more comparative results of large-parallax composition in Fig. 20. 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 methods (perception-based seam cutting [30] and reconstruction [41]) in UDIS-D [41] and other large-parallax datasets [35].

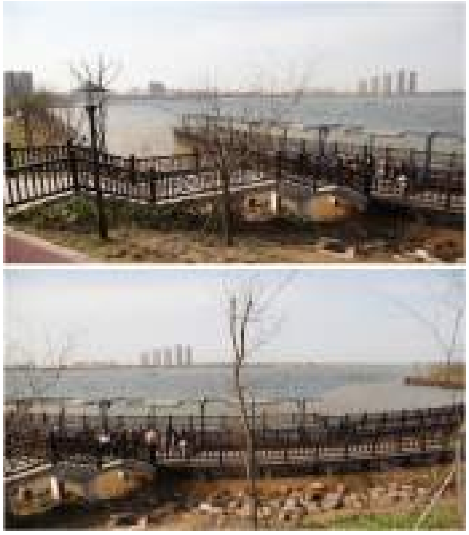


Figure 15: The input images of Fig. 5 in the manuscript.

E.3. Results of Complete Solutions

Then, we compare our complete framework with other SoTA solutions (LPC [19] and UDIS [41]) with seam cutting or reconstruction as their post-processing operations. The qualitative results are shown in Fig. 21.

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 [19] and UDIS [41] 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 [41], which means every user has to compare each method with ours in 1,106 images. The results are shown in Fig. 13.

Besides, we demonstrate more results in traditional datasets [28, 35, 50, 13, 51] in Fig. 22. 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.

E.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. 17, 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 D.1).

Table 4: Ablation studies of alignment performance on UDIS-D dataset[41]. 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 5: 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

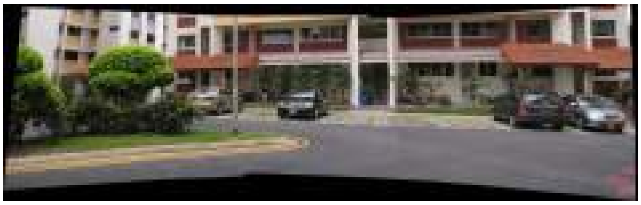


Figure 16: Stitching four images from the traditional dataset[14].

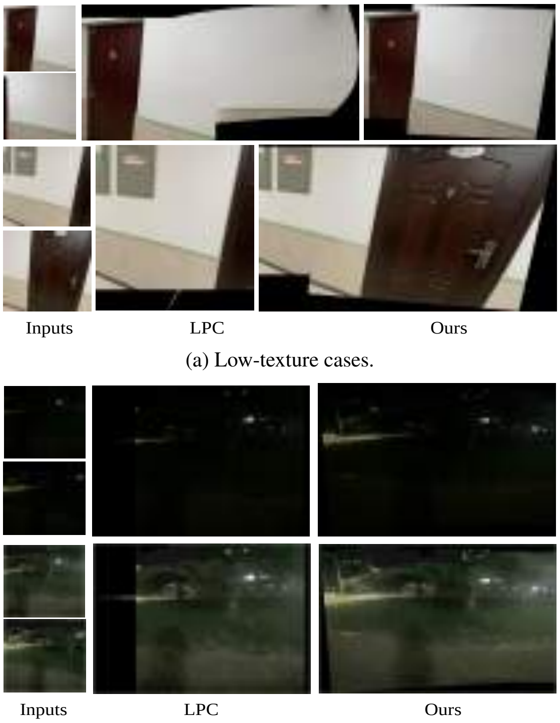
E.5. Ablation Studies

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 are shown in Table 4, where the SSIM merely decreases 0.03 when we adopt these shape-preserving constraints.

Besides, we demonstrate the superiority of combining TPS with homography in Table 5. Compared with only homography, the combination can significantly improve the alignment performance. Compared with only TPS, the combination reaches slightly better performance with less computational cost.

E.6. Multi-Image Stitching

Most stitching methods (e.g., LPC[19], UDIS[41]) 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. 16.



(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 17: 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 [41]

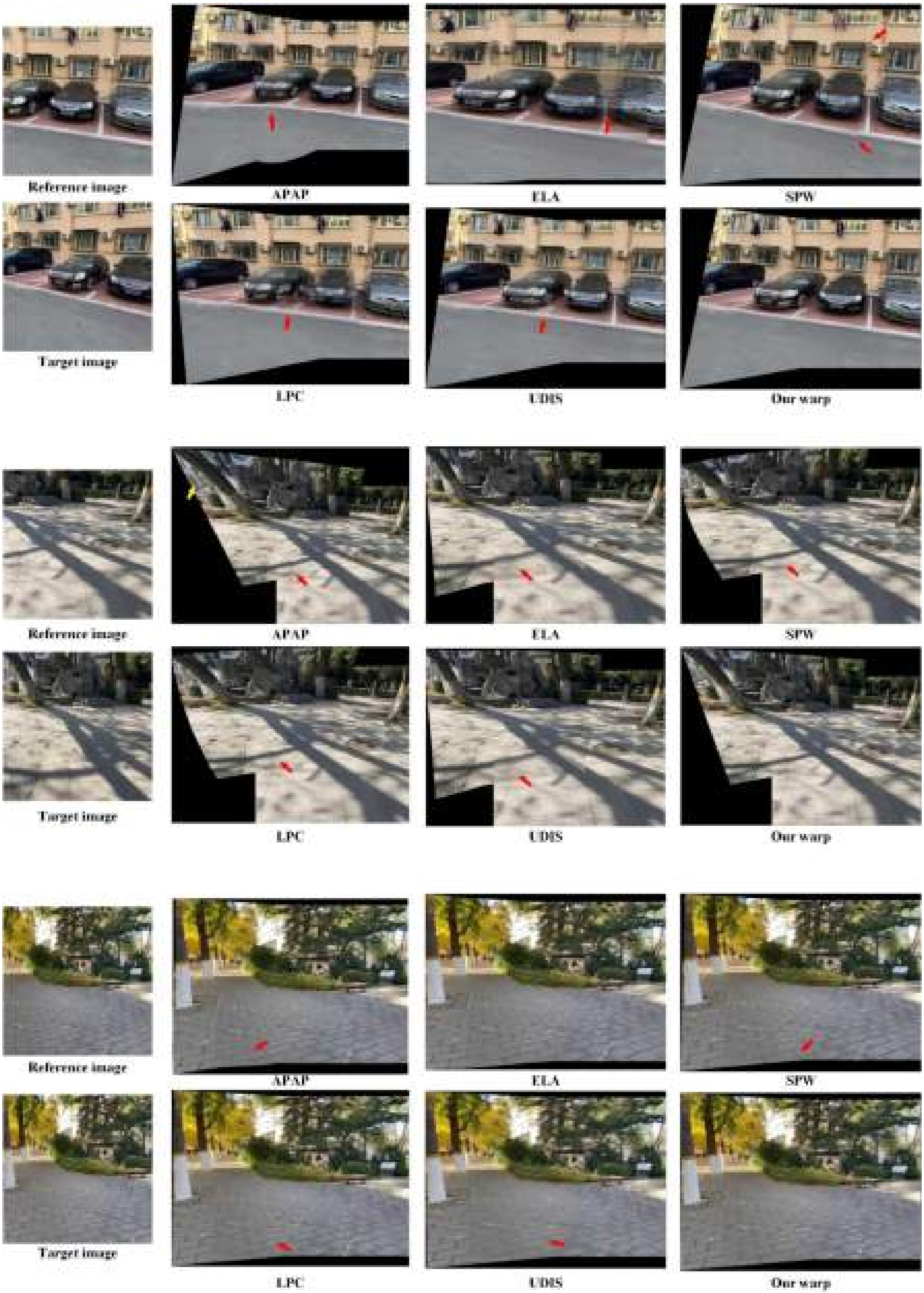


Figure 18: Comparative results of warp on UDIS-D dataset[41]. The red arrows highlight the artifacts.

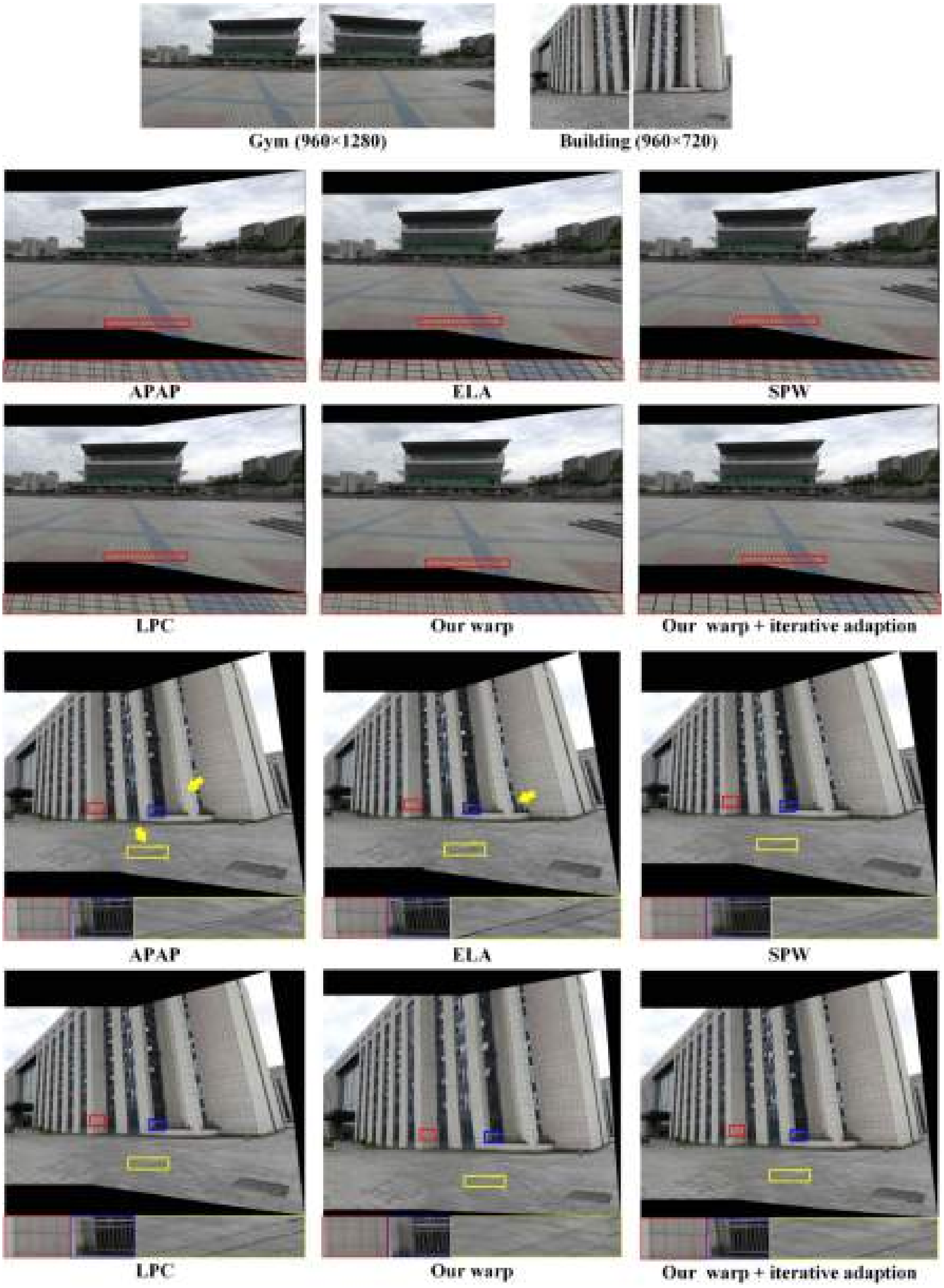
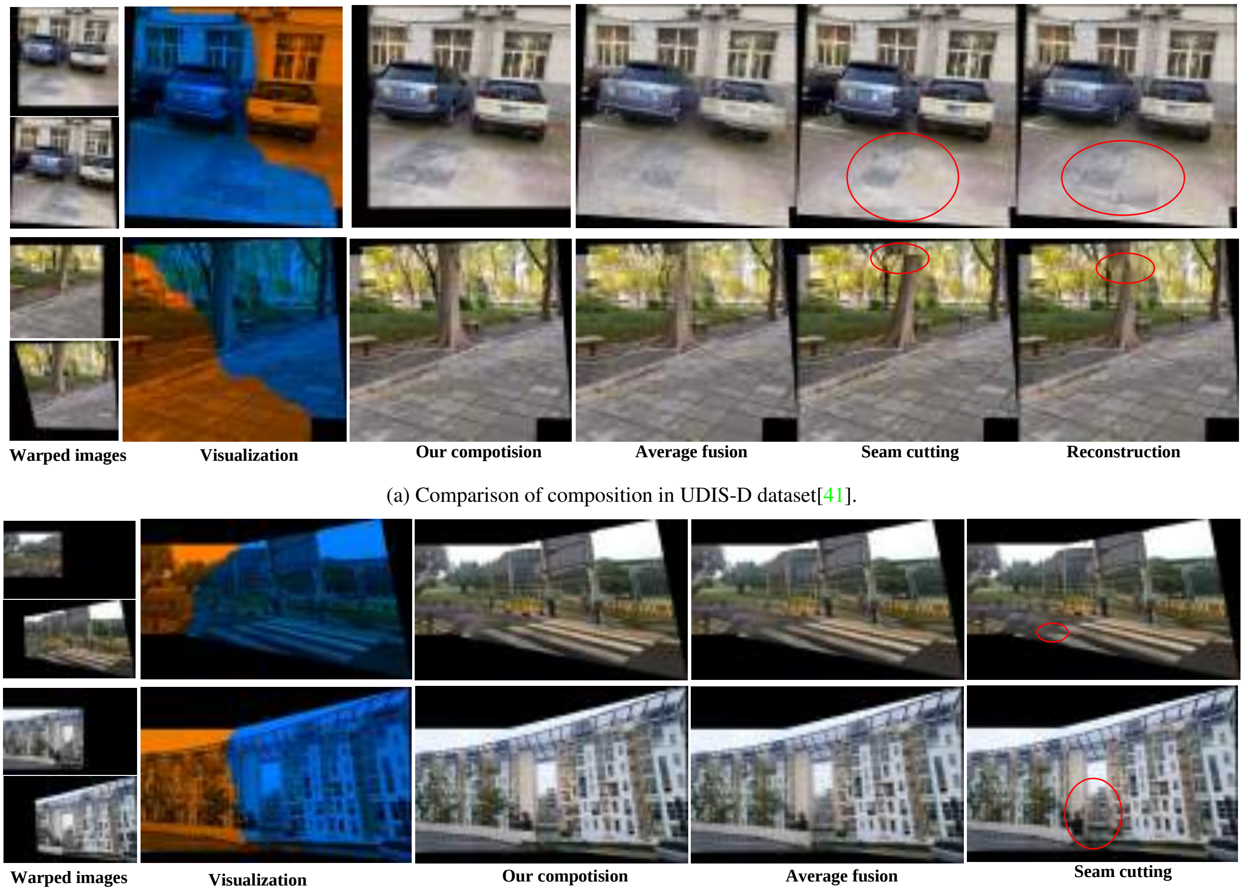


Figure 19: Comparative results of warp in cross-dataset cases[28].The arrows highlight the distortions.



(a) Comparison of composition in UDIS-D dataset[41].

(b) Comparison of composition in traditional large-parallax dataset[35].

Figure 20: 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.

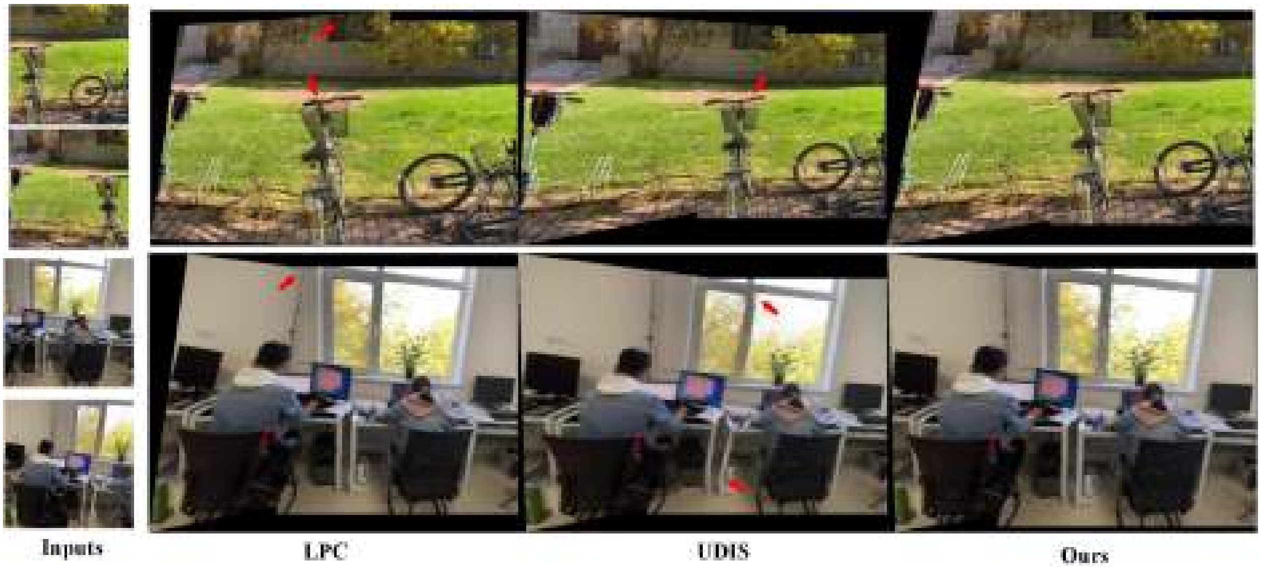


Figure 21: Comparative results of complete stitching frameworks. LPC[19] and UDIS[41] leverage perception-based seam cutting[30] and reconstruction[41]as the composition methods.

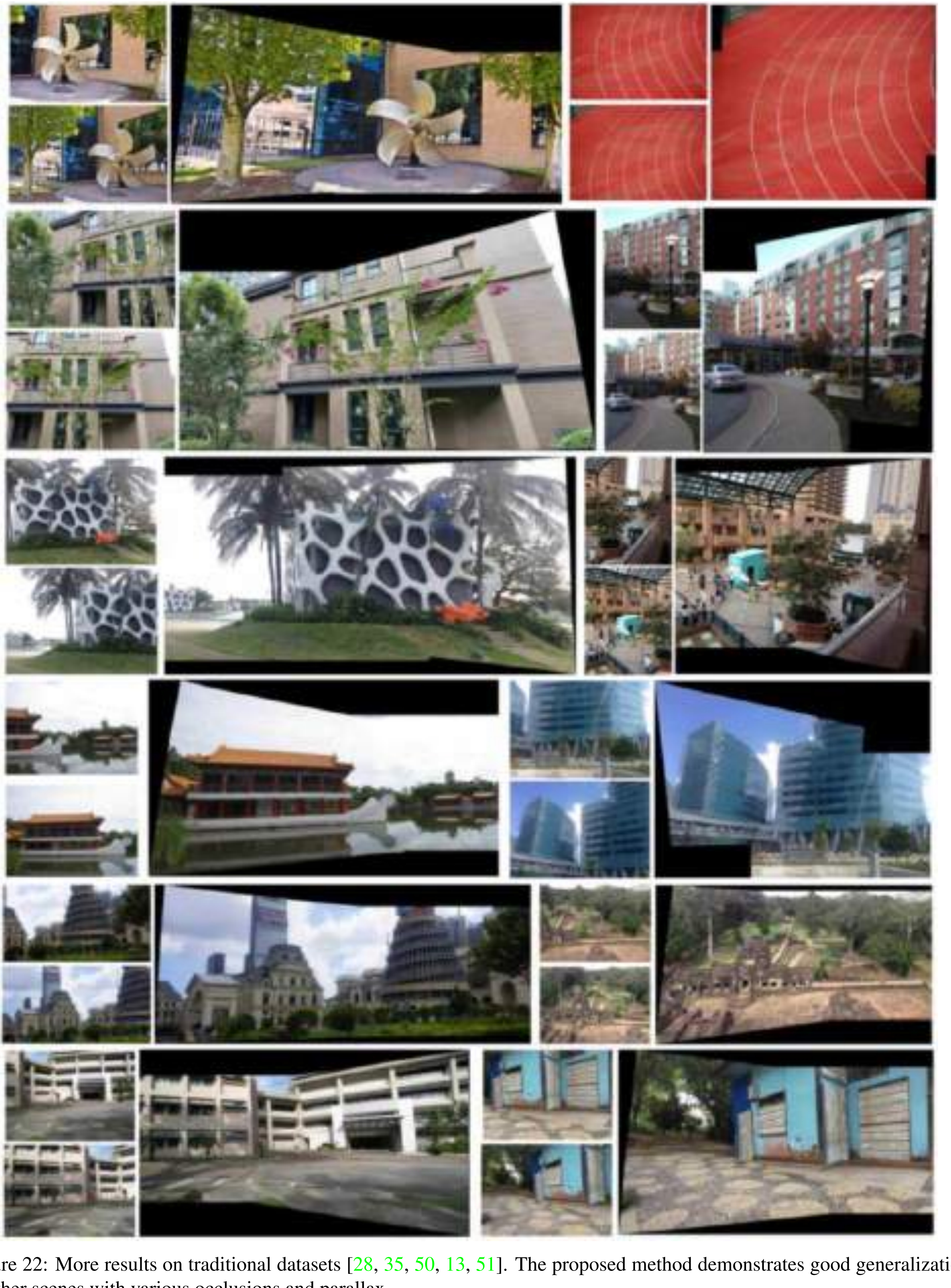


Figure 22: More results on traditional datasets [28, 35, 50, 13, 51]. The proposed method demonstrates good generalization in other scenes with various occlusions and parallax.

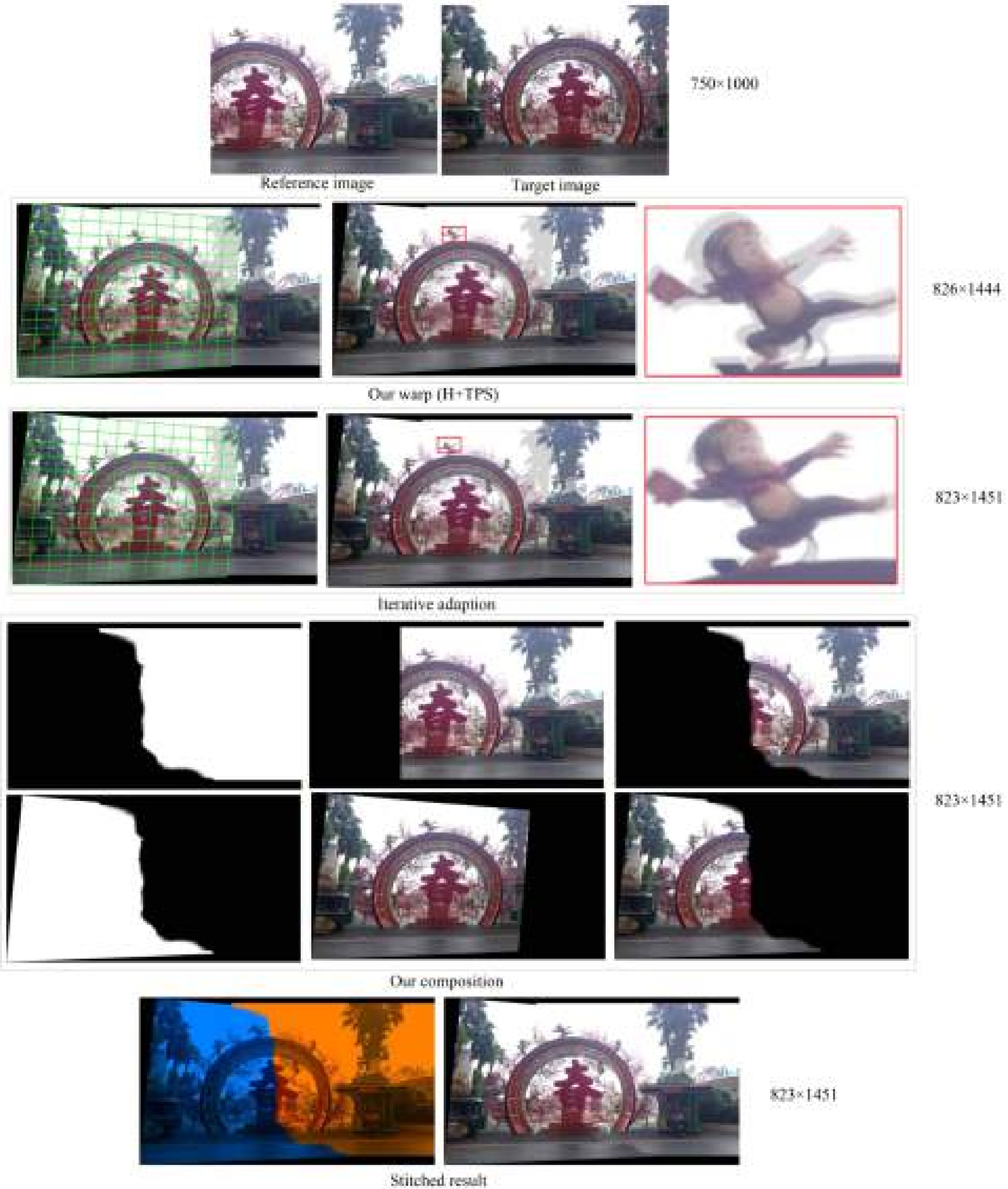


Figure 23: The complete pipeline of the proposed stitching framework. We show our intermediate result in a large-parallax cross-dataset case[35]. We link the predicted control points to form a mesh for clear visualization. Note that for the images from UDIS-D dataset[41], we do not conduct warp adaption iterations.