Computational Visual Media
https://doi.org/10.1007/s41095-024-0420-6 Vol. 10, No. 4, August 2024, 687–703

Research Article

# RecStitchNet: Learning to stitch images with rectangular boundaries

**Yun Zhang**[1] (✉), **Yu-Kun Lai**[2], **Lang Nie**[3], **Fang-Lue Zhang**[4], **and Lin Xu**[5]



**Abstract** Irregular boundaries in image stitching naturally occur due to freely moving cameras. To deal with this problem, existing methods focus on optimizing mesh warping to make boundaries regular using the traditional explicit solution. However, previous methods always depend on hand-crafted features (e.g., keypoints and line segments). Thus, failures often happen in overlapping regions without distinctive features. In this paper, we address this problem by proposing *RecStitchNet*, a reasonable and effective network for image stitching with rectangular boundaries. Considering that both stitching and imposing rectangularity are non-trivial tasks in the learning-based framework, we propose a three-step progressive learning based strategy, which not only simplifies this task, but gradually achieves a good balance between stitching and imposing rectangularity. In the first step, we perform initial stitching by a pre-trained state-of-the-art image stitching model, to produce initially warped stitching results without considering the boundary constraint. Then, we use a regression network with a comprehensive objective regarding mesh, perception, and shape to further encourage the stitched meshes to have rectangular boundaries with high content fidelity. Finally, we propose an unsupervised instance-wise optimization strategy to refine the stitched meshes iteratively, which can effectively improve the stitching results in terms of feature alignment, as well as boundary and structure preservation. Due to the lack of stitching datasets and the difficulty of label generation, we propose to generate a stitching dataset with rectangular stitched images as pseudo-ground-truth labels, and the performance upper bound induced from the it can be broken by our unsupervised refinement. Qualitative and quantitative results and evaluations demonstrate the advantages of our method over the state-of-the-art.

**Keywords** image stitching; boundaries; convolutional neural network

## 1 Introduction

In recent decades, image stitching has been an active topic in computer graphics and vision. The goal of image stitching is to construct a wide field-of-view (FOV) scene from several overlapping images each having a limited FOV. This has a wide range of applications in virtual reality, autonomous driving, video surveillance, etc. Traditional image stitching methods mainly focus on accurate feature matching, natural warping, shape- and straight line-preservation [1–3]. Despite their great success, most of these methods rely on the performance of hand-crafted feature matching in overlapping regions, and thus have limited generalizability. These methods often struggle to stitch images with unclear textures, or taken in low light, or having low resolution. Additionally, preserving geometric structure and visual features necessitates complex optimization and intensive computation, further heightening the difficulty of image stitching.

1 Key Lab of Film and TV Media Technology of Zhejiang Province, College of Media Engineering, Communication University of Zhejiang, Hangzhou 310018, China. E-mail: zhangyun@cuz.edu.cn (✉).
2 School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK. E-mail: Yukun.Lai@cs.cardiff.ac.uk.
3 Institute of Information Science, Beijing Jiaotong University, Beijing 100091, China. E-mail: nielang@bjtu.edu.cn.
4 School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6012, New Zealand. E-mail: fanglue.zhang@ecs.vuw.ac.nz.
5 STEM, University of South Australia, Adelaide 5095, Australia. E-mail: xuyly032@mymail.unisa.edu.au.
Manuscript received: 2024-01-05; accepted: 2024-02-27

To overcome the challenges posed by feature matching and structure preservation, learning-based methods have been extensively studied in recent years; they stitch images by adaptively learning high-level semantic features from big data. These methods can be roughly divided into three types: supervised [4–6], weakly-supervised [7], and unsupervised [8, 9] methods. They are able to robustly and efficiently stitch images, demonstrating high performance in terms of large parallax tolerance and geometry preservation. However, most of them do not take boundary regularity into consideration. Recently, following previous work on image stitching and imposing rectangularity of results based on conventional optimization frameworks [10, 11], Nie et al. [12] proposed the first deep learning solution for imposing image rectangularity, which was further extended to image rotation correction in Ref. [13]. They took well-stitched images as input and learned to rectify the irregular boundaries while preserving the high-level semantic features. However, their method does not consider optimizing stitching simultaneously with creation of rectangular image boundaries. This oversight could potentially amplify artifacts in the stitched input after applying warping-based rectification.

In this paper, we introduce *RecStitchNet*, a supervised learning network designed for stitching images while ensuring rectangular stitching boundaries. To enable an effective learning process, we have designed a three-step progressive stitching approach. Firstly, we conduct an initial stitching process using a state-of-the-art deep stitching technique, to give warped meshes for the image pair. Secondly, we use a regression network of our own design with a comprehensive objective regarding mesh, perception, and shape to encourage the stitched meshes to have rectangular boundaries with high content fidelity. The output of the network is the predicted mesh motions relative to the initially warped meshes. In this paper, the term "mesh motion" refers to offsets of all vertex positions of the mesh on each image. Finally, to ensure the robustness of our method across various scenarios, we employ an unsupervised instance-wise network to improve the stitching result. This refinement process is guided by an objective function comprising a rectangular boundary term, a feature-matching term, and a shape preservation term, which collectively contribute to the production of high-quality stitching results.

Unlike typical stitching methods that often result in irregular boundaries, our objective is to achieve stitching results with rectangular boundaries. Generally, both stitching and imposing rectangularity are challenging tasks, necessitating a supervised network for effective learning. However, obtaining ground truth stitching results is difficult due to the absence of a publicly recognized standard. So that rectangular stitching results can be more standardized and universally recognized, we propose to generate pseudo-ground-truth using a state-of-the-art stitching technique with rectangular boundaries [11].

Figure 1 shows the pipeline of our method. Given two normal FOV images as input, the proposed

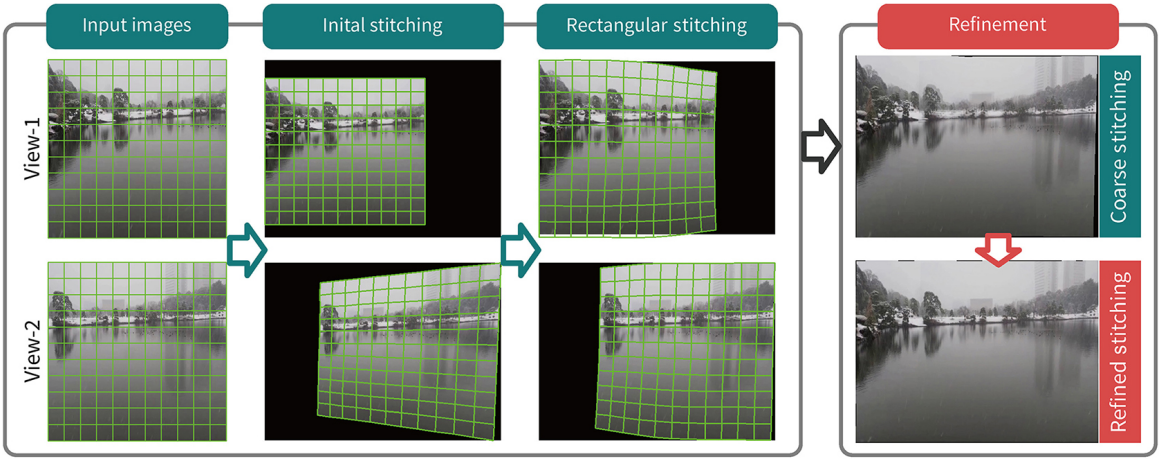



**Fig. 1** Pipeline of our method. We take two normal FOV images as input, and then stitch them using a pre-trained model. Taking the initial stitching result with irregular boundary as input, we use *RecStitchNet* to produce coarse stitching result with a rectangular boundary. We finally refine the coarse stitching result using an instance-wise unsupervised learning method.

solution progressively stitches them from an initial stitched image with irregular boundaries to a coarse stitched image with rectangular boundaries to the final stitching result with further refinement of boundaries and alignment. Extensive experiments and evaluations in this paper show that our approach can effectively stitch images and obtain satisfactory results, with rectangular boundaries.

Compared to a previous stitching method imposing rectangularity [11], our method is more robust and efficient due to its effective high-level feature extraction and matching.

To sum up, our main contributions are as follows:

- We propose a novel deep stitching network called *RecStitchNet*, which does not rely on the fragile and expensive feature matching found in traditional methods, so is much more robust and efficient compared to these methods. As our extensive experimental results demonstrate later, our method achieves state-of-the-art performance, both qualitatively and quantitatively, outperforming traditional methods and deep learning baselines.
- We introduce an unsupervised instance-wise learning strategy to iteratively optimize stitching results, to ensure high-quality stitching in a wide range of scenarios.
- Given the absence of an existing dataset for supervised learning, we have created a new dataset, which includes pseudo-ground-truth mesh warping results, strictly selected and re-rendered from traditional stitching results with rectangular boundaries.

## 2 Related work

### 2.1 Traditional image stitching

Image stitching refers to aligning multiple images with mutual overlaps and producing a new image with a larger FOV. The key problem of image stitching is to keep accurate feature alignment, with unnoticeable distortion. Earlier works based on a single homography [14] and dual-homographies are limited to parallax and perspective variations.

To compensate for the shortcomings of a globally projective model, a number of spatially varying warping models, which can better address local alignment, have been proposed, such as smoothly varying affine stitching [15], as-perspective-as-possible (APAP) stitching [16], piecewise planar region matching [17], and seam-guided warping [18–20].

To produce more natural stitching with less perspective distortion, several warping schemes have proposed, characterized by shape-preserving half-projection (SPHP) [21], adaptive as-natural-as-possible (AANAP) [3], global similarity prior [1], quasi-homography [22], single perspective [23], and geometric structure preserving [24].

Recently, Jia et al. [25] considered global collinear structures, effectively preserving global and local structures while reducing distortions. Zhang and Huang [26] proposed manifold preserving stitching: Using on-manifold operations helps to reduce ghosting and distortion artifacts. To improve stitching results, seam-cutting methods have been applied to removing artifacts in overlapping regions [20, 27].

The most relevant work to our paper comes from Zhang et al. [11], in which boundary regularity constraints are incorporated into the stitching framework, helping to solve the irregular boundary problem in image stitching. Although successful in many examples including some challenging cases, the method in Ref. [11] may fail in situations such as those with unclear textures, low lighting, and low resolution. In addition, the two-step energy optimization process is also time-consuming.

### 2.2 Deep image stitching

Unlike the above methods, deep stitching learns to stitch images by extracting high-level features from large datasets, which avoids the difficulties in feature matching, global and local structure preservation, etc. We may roughly divide recent research into three main types.

#### *2.2.1 Supervised learning*

Nie et al. [4] and Zhao et al. [5] proposed view-free image stitching based on global homography learning, which improves upon the previous learning based stitching method [28] which is limited to relatively fixed views. To tolerate parallax in stitching, they generate a synthetic dataset from an existing real image dataset. Instead of homography based learning, Kweon et al. [6] recently proposed a novel deep stitching framework using a pixel-wise warp field, which can handle the large-parallax problem well.

#### *2.2.2 Weakly supervised learning*

To overcome the difficulties in dataset and ground truth generation, Song et al. [7] proposed a weakly-

supervised learning method to train the stitching model without using real ground truth images. They have further extended their method to stitching multiple images and creating 360-degree panoramas.

*2.2.3 Unsupervised learning*

Considering the difficulties in data label generation, some works focus on unsupervised learning methods, which train stitching models without labels. Nie et al. [8] proposed an unsupervised image stitching method, which consists of unsupervised coarse image stitching and image reconstruction. Very recently, Nie et al. [9] further proposed a parallax-tolerant unsupervised image stitching method which is characterized by combining homography and thin-plate splines (TPS) into a unified framework.

### 2.3 Enforcing image rectangularity

Enforcing image rectangularity aims to regulate the irregular boundaries caused by image stitching, rotation, etc. The pioneering works in imposing image rectangularity are Refs. [10, 29], in which content-aware warping methods based on mesh optimization are proposed. Wu et al. [30] further extended imposing rectangularity to videos, incorporating temporal coherence into the warping-based optimization. Nie et al. [12] proposed a one-stage learning baseline of deep rectangularity imposition for image stitching. Compared to the two-stage methods in Refs. [10, 29], the method in Ref. [12] is more efficient and robust, and can well preserve non-linear structures thanks to high-level feature extraction in the learning framework. Liao et al. [31] proposed a rectangularity imposing rectification network, which applies the TPS module to perform non-linear and non-rigid transformations for imposing rectangularity on wide-angle rectified images. Very recently, Zhou et al. [32] combined stitching and imposing rectangularity into a unified end-to-end framework using a synthetic dataset. Although effective in producing stitching results with rectangular boundaries, it still suffers from content loss and ghosting effects in the overlapping regions.

## 3 Method

### 3.1 Overview

Like recent work on stitching and imposing rectangularity [1, 9, 11, 12, 23], we also stitch images by content-aware mesh warping. Mesh warping is widely used in image manipulation due to its simplicity and efficiency. Traditional methods [1, 11, 23] are based on energy optimization with constraints on all grid vertices of the mesh. Let $V = \{V^i, i = 1, \ldots, N\}$ be the sets of all vertices of the input images, where $N$ is the number of images. We aim to obtain warped mesh vertices $\hat{V} = \{\hat{V}^i\}$ by minimizing the energy function $E(\hat{V})$, which includes several content-aware constraints, such as feature alignment, shape preservation, and straight line preservation. Such methods usually focus on designing energy terms that are effective in stitching and easy to optimize. Unlike traditional methods, deep learning based methods [9, 12] focus on dataset preparation, network construction, and mesh regression. For effective mesh regression, we have to focus on designing the objective function to effectively guide the training process, and the total loss, to ensure satisfactory convergence. To calculate the feature loss after mesh warping, an effective and efficient warping operation is required, which must also be differentiable for effective gradient propagation.

Inspired by the methods above, we propose a novel method to achieve stitching and imposing boundary rectangularity simultaneously in a learning-based framework. Figure 2 shows an overview of our method for deep stitching with rectangular boundaries. We take two images of the same size with partial overlap as input; the output is a rectangular stitching result with no loss of content. We first perform initial stitching, which aims to warp the high-level features extracted from input images. The warping is guided by the initial meshes generated by a state-of-the-art deep stitching model [9]. We further learn to regulate the boundary of the stitching result by designing a regression network, which generates suitable mesh motion to be applied to the initial meshes. Finally, with the combination of initial meshes and these mesh motions, the final stitching result can easily be produced by warping and average blending.

### 3.2 Initial warping

In this stage, the input images are initially stitched using a leading deep stitching model [9]. As illustrated in Fig. 3, given two input images $\{I^i, i = 1, 2\}$ to be stitched, a uniform quad mesh $v^i$ is placed on each image $I^i$. The initial warping produces warped meshes $\{\xi^i\}$ after the deep stitching process. To

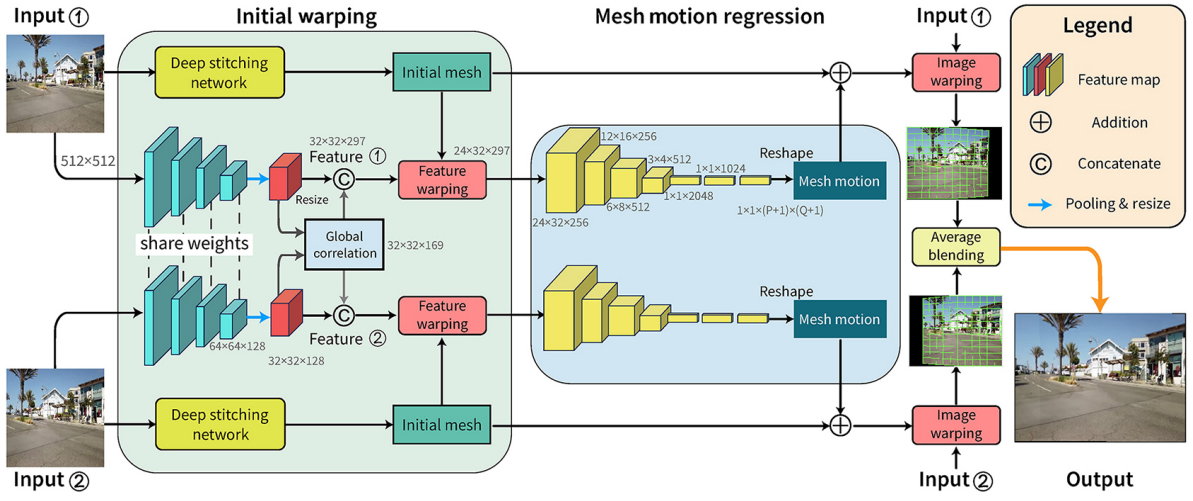



**Fig. 2** Overview of our network for deep stitching with rectangular boundaries. Our supervised learning network consists of initial warping and mesh motion regression stages. The first stage warps the high-level features extracted from the input images; the warping is guided by meshes generated by the deep stitching model by Nie et al. (2023). Using the warped features as input, we further obtain the mesh motions (vertex offsets to apply to the initial meshes) through mesh motion regression. Final stitching results are obtained by averaging the images warped by the meshes produced by combining the initial meshes and mesh motions.

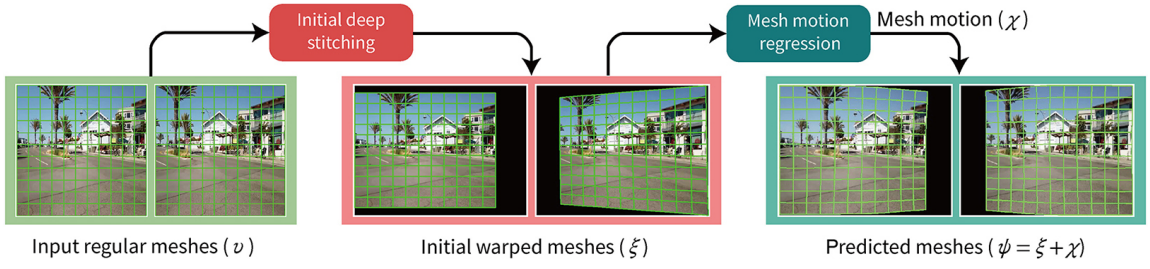



**Fig. 3** Mesh manipulations in initial stitching and mesh motion regression.

facilitate the deep stitching task, the first image is consistently kept unchanged, while the other image is warped into alignment with it. Next, high-level semantic features are extracted from each input image (without warping) through a series of convolution and pooling blocks (blue solid blocks in Fig. 2); each block comprises two convolution layers. After the first, second, and third blocks, a max-pooling layer is applied. We set the number of channels to 64 and 128 for the convolution layers in the first two and the last two blocks, respectively. Subsequently, following the last blocks, an adaptive pooling layer is employed to standardize the resolution of the features.

To establish the relationship between the two images in the overlapping regions, we concatenate the global correlation [4] with the features extracted for each image. Given the extracted features $(F^1, F^2)$ of the input images, their global correlation refers to their feature-wise similarities, defined by

$$\mathrm{Cor}(x_1, x_2) = \frac{< F^1(x_1), F^2(x_2) >}{|F^1(x_1)||F^2(x_2)|} \qquad (1)$$

where $x_1, x_2$ represent the locations of the feature vector in each feature map. We limit the range of feature similarity comparison for fast calculation of global correlation. These features are then warped using the meshes $\{\xi^i\}$ obtained through the initial stitching process. These warped features $\{\kappa^i\}$ of the input images, representing the features of the initial stitching results, serve as input to the mesh motion regression process.

### 3.3 Mesh motion regression

In this stage, our goal is to obtain the mesh motion, as offsets from the initially warped mesh vertices; this helps regulate the shape of the final stitching boundaries. As indicated in the middle section of Fig. 2, the input to this stage consists of the extracted high-level features that have been warped by the meshes from the initial stitching. We provide a simple yet effective fully convolutional network to predict both the vertical and horizontal motion, denoted as $\{\chi^i\}$, for all vertices relative to those in the initially

warped meshes $\{\xi^i\}$. The output of the regression is of dimension $(P+1)\times(Q+1)$, where $P$ and $Q$ denote the resolution of the meshes. Subsequently, we determine the predicted meshes $\{\psi^i\}$ by combining the mesh motion $\{\chi^i\}$ with the initial warped meshes $\{\xi^i\}$, as shown in Fig. 3. With the incorporation of mesh motion, the outer boundary of the combined meshes more closely approximates a rectangle.

### 3.4 Loss functions

#### *3.4.1 Loss*

Our regression network learns the motion of the mesh vertices that can ensure both feature alignment and boundary regularity. We use three loss terms and define the total loss as Eq. (2):

$$L_{\text{train}} = \varphi_{\text{m}} l_{\text{m}} + \varphi_{\text{p}} l_{\text{p}} + \varphi_{\text{s}} l_{\text{s}} \tag{2}$$

where $l_{\text{m}}$, $l_{\text{p}}$, $l_{\text{s}}$ refer to the mesh, perception, and shape preserving loss terms respectively, and $\varphi_{\text{m}}$, $\varphi_{\text{p}}$, $\varphi_{\text{s}}$ are corresponding weights.

For our supervised training framework, we have prepared a large dataset (refer to Section 3.6), which contains input image pairs, pseudo-ground-truth mesh labels representing warped mesh vertices, and pseudo-ground-truth stitching result labels. The pseudo-ground-truth data are produced by a leading traditional stitching method which produces rectangular boundaries [11].

#### *3.4.2 Mesh loss*

Given the predicted meshes $\{\psi^i\}$, we simply constrain them to be close to the ground truth labels of meshes $\Psi^i$, using:

$$l_{\text{m}} = \sum_{i=1}^{2} \sum_{j=1}^{(P+1)*(Q+1)} \|\psi_j^i - \Psi_j^i\|_1 \tag{3}$$

where $\psi_j^i$ and $\Psi_j^i$ refer to the vertex positions of the predicted mesh and the pseudo-ground-truth mesh respectively.

#### *3.4.3 Perceptual loss*

We further constrain the result to be visually appealing, and to preserve the structure in the input image, such as linear or salient structures. We define the corresponding loss as Eq. (4):

$$l_{\text{p}} = \sum_{i=1}^{2} \|\Gamma(\Omega_{\text{tps}}(\psi^i, I^i)) - \Gamma(\Omega_{\text{tps}}(\Psi^i, I^i))\|_1 \tag{4}$$

where $\Omega_{\text{tps}}(\cdot)$ refers to the TPS transformation [33], which is used to warp the input images $\{I^i\}$ guided by the warped mesh, and $\Gamma(\cdot)$ refers to the VGG-19 [34] feature extractor.

#### *3.4.4 Shape preserving loss*

Following Ref. [12], we also preserve the shape of the mesh using intra-grid and inter-grid shape similarity constraints, using:

$$l_{\text{s}} = l_{\text{s}}^{\text{intra}} + l_{\text{s}}^{\text{inter}} \tag{5}$$

The intra-grid constraint is employed to enforce both the scale and direction of the grid edges, and is defined as Eq. (6):

$$\begin{aligned} l_{\text{s}}^{\text{intra}} = &\sum_{i=1}^{2} \frac{\sum_{\vec{e}_j \in \vec{h}_i} \vartheta(\Delta_x(\vec{e}_j) + \sigma W/Q)}{(P+1)Q} \\ &+ \sum_{i=1}^{2} \frac{\sum_{\vec{e}_k \in \vec{v}_i} \vartheta(\Delta_y(\vec{e}_k) + \sigma H/P)}{P(Q+1)} \end{aligned} \tag{6}$$

where $\vec{e}_j$ and $\vec{e}_k$ refer to all horizontal and vertical edges of a mesh respectively, and $\Delta_x(\vec{e}_j)$ and $\Delta_y(\vec{e}_k)$ refer to projections of the edge vectors onto $x$ and $y$ directions respectively. $\vartheta$ is the ReLU function, which is used to cause the direction of the horizontal and vertical edges to be right and bottom, and enforce their scale to be more than $\sigma W/Q$ and $\sigma H/P$; we set $\sigma = 0.8$ in this paper.

The inter-grid constraint aims to cause pairs of successive horizontal and vertical grid edges $\{\vec{e}_{t1}, \vec{e}_{t2}\}$ to undergo linear changes (i.e., encouraging their angle to be close to zero). It is defined as Eq. (7):

$$l_{\text{s}}^{\text{inter}} = \sum_{i=1}^{2} \frac{1}{|\Lambda^i|} \sum_{\{\vec{e}_{t1}, \vec{e}_{t2}\} \in \Lambda^i} (1 - \cos(\vec{e}_{t1}, \vec{e}_{t2})) \tag{7}$$

where $\cos(\vec{e}_{t1}, \vec{e}_{t2})$ calculates the cosine of the angle between $\vec{e}_{t1}$ and $\vec{e}_{t2}$, $\Lambda^i$ refers to the set of all successive grid edges in the mesh of the $i$th image, and $|\Lambda^i|$ is the total number of successive grid edges.

### 3.5 Unsupervised instance-wise stitching refinement

#### *3.5.1 Approach*

In the mesh motion regression step, our loss functions are designed to cause the predicted stitching result to be close to both the pseudo-ground-truth mesh and the stitched image while preserving mesh shape. However, our experiments showed that some predicted results may not exhibit perfect boundary regularity and feature matching (see Fig. 4). Simply incorporating feature matching and rectangular boundary constraints into the network training process does not yield satisfactory results. This is because the refinement objective (unsupervised learning) is slightly contradictory to the original optimization goal of *RecStitchNet* (supervised

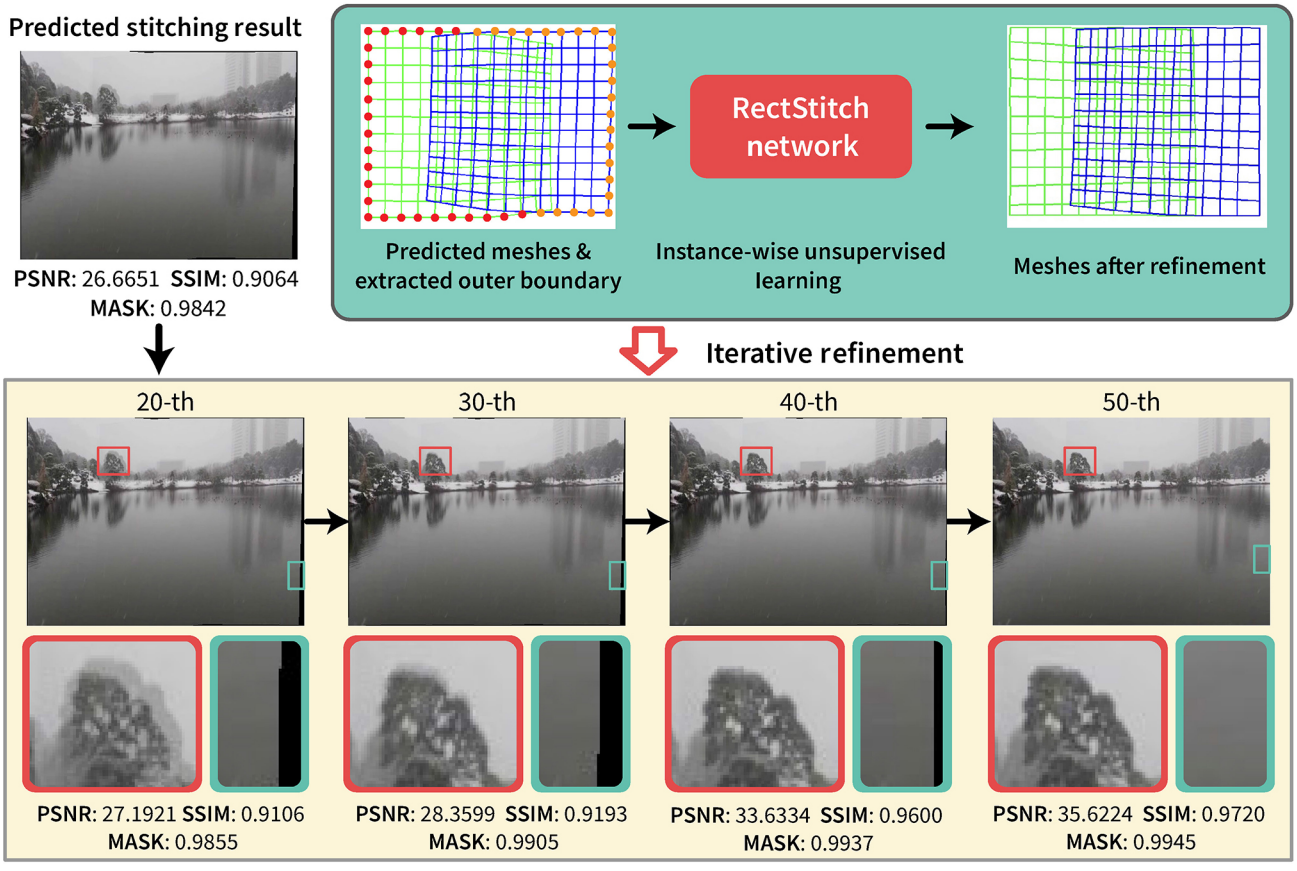



**Fig. 4** Refinement of stitching results. The input to this step comprises the predicted meshes and the corresponding input image pair. We first extract the outer boundary of the predicted meshes using a polygon Boolean operation, and then predict the refined meshes using the instance-wise unsupervised learning framework in an iterative manner. Finally, we obtain an optimized stitching result by warping and blending.

learning using pseudo mesh labels), preventing the network parameters from being optimized. To enhance stitching performance and enable the transfer of the pretrained model to other datasets, we propose an instance-wise unsupervised learning method constrained by feature matching, rectangular boundary, and shape preservation constraints, which are designed to further optimize the mesh grid, so as to refine the imperfect rectangular boundaries and the ghosting in stitched images.

As Fig. 4 shows, while the predicted stitching result appears quite satisfactory, it still exhibits irregular boundaries and misalignment in the overlapping regions. To further refine the stitching result, we introduce an instance-wise unsupervised learning scheme to iteratively optimize the stitching (see Algorithm 1). The input consists of the predicted meshes $\{\psi^i\}$ generated by our pre-trained regression network, along with the corresponding input image pair $\{I^i\}$. The output comprises the optimized meshes $\{\Theta^i\}$, for $i = 1, 2$, and the stitching result $\Phi$. To iteratively optimize the stitching boundary, we first need to obtain the boundary vertices of the stitching result. Drawing inspiration from Ref. [11], we treat the outer boundaries of the meshes $\{\psi^i\}$ as polygons $\{\hat{P}^i\}$. Subsequently, the outer boundary $\hat{P}$ of the two meshes is calculated using a polygon Boolean union operation [35], as Eq. (8):

$$\hat{P} = \hat{P}^1 \cup \hat{P}^2 \tag{8}$$

With the outer boundary vertices of the stitching results, we are able to construct an effective constraint for the rectangular boundary. In each iteration, we first predict the mesh motions $\{\chi_i\}$ relative to the current meshes $\{\psi^i\}$ using an unsupervised learning network with the same architecture as *RecStitchNet*. $\{\psi^i\}$ is then updated for the next iteration. We then compare the current loss value with the value from the previous iteration. The iterations terminate when the difference in loss is sufficiently small. Finally, we warp the input images using the final optimized meshes $\{\Theta^i\}$, and obtain the final stitching result $\Phi$ through average blending of the warped images.

**Algorithm 1** Refine the stitching results

**Input**: Predicted meshes $\{\psi^i\}$ produced by our pre-trained regression network, and input image pair $\{I^i\}$, $i = 1, 2$;
**Output**: Optimized meshes $\{\Theta^i\}$, $i = 1, 2$ and stitching results $\Phi$;
Let $\hat{P}^i$ be the boundary vertices of $\psi^i$;
Let $\hat{P}$ be the outer boundary vertices of the two meshes $\{\psi^i\}$, $i = 1, 2$, calculated by Eq. (8);
**foreach** $j \in [1, 200]$ **do**
  **foreach** $i \in [1, 2]$ **do**
    $\chi_i = \text{RecStitchNet}(\psi^i, I^i)$;
    $\psi^i = \psi^i + \chi^i$;
    $\Theta^i = \psi^i$;
  **end foreach**
  **if** $j == 1$ **then**
    $\text{Loss}_{\text{pre}} = \text{Loss}(\text{RecStitchNet})$;
  **end if**
  **else**
    $\text{Loss}_{\text{now}} = \text{Loss}(\text{RecStitchNet})$;
    **if** $|\text{Loss}_{\text{now}} - \text{Loss}_{\text{pre}}| < \text{e}^{-5}$ **then**
      break;
    **end if**
    $\text{Loss}_{\text{pre}} = \text{Loss}_{\text{now}}$
  **end if**
**end foreach**
**foreach** $i \in [1, 2]$ **do**
  $R^i = \Omega_{\text{tps}}(\Theta^i, I^i)$
**end foreach**
$\Phi = \text{AverageBlend}(R^1, R^2)$;

In the refinement step, we use a different set of loss functions for the instance-wise unsupervised learning, defined as below.

*3.5.2 Feature matching loss*

The feature matching constraint is designed to ensure that the features of the two images in the overlapping regions are well-aligned. It is defined as the difference between the warped image features in the overlapping regions, as Eq. (9):

$$l_{\text{f}} = \left\| \sum_{i=1}^{2} (\Gamma(\Omega_{\text{tps}}(\psi^i, I^i) * M * (-1)^{i-1})) \right\|_1 \tag{9}$$

where $\Omega_{\text{tps}}(\cdot)$ refers to the TPS transformation, and $\Gamma(\cdot)$ refers to the VGG-19 [34] feature extractor. $M$ is the intersection of the warped masks guided by the predicted meshes $\{\psi^i\}$.

*3.5.3 Rectangular boundary loss*

In Ref. [12], the rectangular boundary loss is simply defined as the difference between the $\{0, 1\}$ mask of the result and the all-one mask. However, we found in experiments that incorporating this form of loss has almost no effect on shaping the rectangular boundary probably due to the difficulty of gradient propagation. To effectively optimize the stitched boundary, we first extract the outer boundary $\hat{P}$ of the warped and overlaid meshes $\{\psi^i\}$. We assign several attributes to each vertex $\nu_k$ in $\hat{P}$, including their constraint directions $\rho(\nu_k) \in \{[1,0], [0,1]\}$ (in $x$ and $y$ directions), and their target values $\tau(\nu_k)$ (the values in the top/bottom/left/right directions). Finally, the loss is defined as the sum of the differences between all vertices and their target locations, as Eq. (10):

$$l_{\text{b}} = \left\| \sum_{\nu_k \in \hat{P}} (\nu_k \cdot \rho(\nu_k)) - \tau(\nu_k) \right\|_1 \tag{10}$$

In this stage, the total loss function is a linear combination of feature matching, rectangular boundary, and shape preserving constraints (see details in Section 3.4.4), as Eq. (11):

$$L_{\text{refine}} = \varphi_{\text{f}} l_{\text{f}} + \varphi_{\text{b}} l_{\text{b}} + \varphi_{\text{s}} l_{\text{s}} \tag{11}$$

where $\varphi_{\text{f}}$, $\varphi_{\text{b}}$, $\varphi_{\text{s}}$ are corresponding weights to control their relative importance.

### 3.6 Data preparation

Recently, there have been very few datasets available for image stitching, and defining their labels (i.e., ground-truth results) is quite challenging. To the best of our knowledge, no dataset suitable for our method exists yet. Unlike traditional stitching methods, which often yield results with irregular boundaries, our objective is to achieve stitching results with rectangular boundaries. This makes it considerably easier to define labels for the stitching process.

To train a deep learning network for image stitching with rectangular boundaries, we have established a new dataset, which comprises input images, mesh labels, and image labels (refer to Fig. 5). Data preparation was carried out as follows.

- *Stitching*: Input image pairs were sourced from the *training* dataset of UDIS-D proposed in Ref. [8]. We performed stitching using the traditional warping-based method described in Ref. [11]. This method is capable of generating stitching results with rectangular boundaries, along with corresponding meshes for each input image. Given our focus on stitching with rectangular boundaries, we prefer to omit data where the stitching result contains an excessive

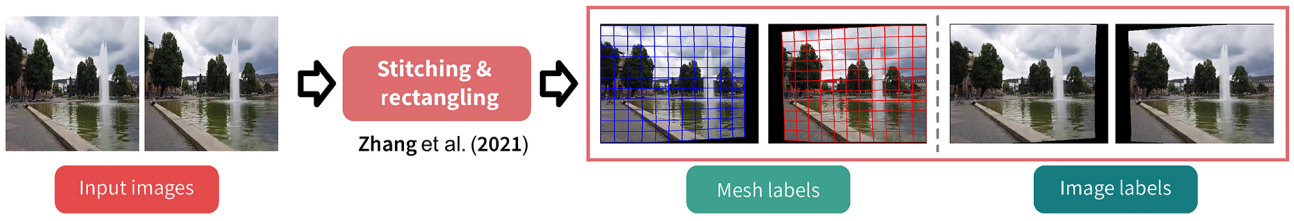



**Fig. 5** Dataset preparation. Give a pair of input images, we first stitch them using the method from Zhang et al. (2021), and then output the resulting mesh labels and the corresponding warped image labels.

amount of missing content. Furthermore, our stitching method does not have to account for piecewise rectangular boundaries as in Ref. [11].

- *Normalization*: For effective training, mesh labels should be constrained within a certain range. However, the scale of stitching results tends to vary greatly. Consequently, we set the resolution of the stitching result to be $W_s \times H_s$; for each vertex of the mesh with coordinates $(x, y)$, we converted them to $(xW_s/w_t, yH_s/h_t)$, where $w_t$ and $h_t$ represented the outer boundary size of the stitching result.
- *Rendering*: We further rendered the stitching results by warping the input images guided by the normalized meshes. To achieve smoother stitching results, our rendering was performed using the TPS transformation [33], which provides more natural transitions and smoother interpolation than mesh-based warping. At this stage, the resolution of each rendered image was also set to $W_s \times H_s$.

Actually, the stitching labels produced by Ref. [11] cannot be considered as ideal labels due to limitations in their approach. This may impact the performance of the training model. In this paper, we utilize these labels, considered as pseudo-ground-truth, for supervised training. To break the bottleneck of the pseudo-ground-truth, we further refined the stitching results using our unsupervised instance-wise learning. Experimental results and evaluations in Section 4 demonstrate that our refined results improve upon the training labels produced by Ref. [11].

## 4 Experiments

### 4.1 Implementation details

In the data preparation and training stages, we set the mesh resolution of each image to $11 \times 11$, and resolution of each input image was normalized to $512 \times 512$. In the feature extraction and regression stages, we set kernel-size = 3, stride = 2 for all convolution blocks and kernel-size = 2, stride = 2 for all max pooling layers; we set the search range to 6 to efficiently calculate the correlation of two feature maps ($32 \times 32$); the size of the correlation is ($4 \times 32 \times 32 \times 169$). In the training stage, we used a linear combination of conv4_2, conv3_2, and conv2_2 layers of the VGG-19 features as the high-level feature of an image. The weights of loss terms were set to $\varphi_m = 1$, $\varphi_p = 0.000006$, $\varphi_s = 0.8$. As for many CNN-based networks [12], we used the Adam optimizer with a learning rate initialized to $10^{-4}$ for $10^5$ iterations, and a decay rate of 0.9. We set batch-size = 4 and used ReLU as the activation function. In the stitching refinement stage, we set $\varphi_f = 0.0001$, $\varphi_b = 1$, $\varphi_s = 1$, and the decaying learning rate was initialized to 0.002 for fast refinement. All implementation was based on TensorFlow using a single GPU with an Nvidia RTX 4090. To better compare the performance of different stitching methods, we simply used average blending to composite the overlapping regions.

### 4.2 Evaluation

To assess the effectiveness of our method, we conducted both qualitative and quantitative evaluations. We compared our method to state-of-the-art methods that have public source code.

Figure 6 displays several stitching results from examples sourced from the *testing* dataset of UDIS-D, which were unseen during training. Given a pair of input images, we shows results of performing stitching using the methods from Refs. [11, 12] and our method separately. For the method in Ref. [12], we initiate the process with an initial stitching using the deep stitching method from Ref. [9]. By subsequently utilizing the stitching result and corresponding mask, the final stitching result is obtained through the deep rectangularity imposition method from Ref. [12]. Zhang et al.'s method [11], carries out stitching through a global optimization process. To obtain

our result, we first stitch images using the proposed *RecStitchNet*, and then refine the stitching to produce improved results. In comparison to Refs. [11, 12], our method excels at shaping the rectangular boundary and ensuring precise alignment in the overlapping regions. The marked red and green boxes (close-up views), along with the PSNR (peak signal-to-noise ratio) and SSIM (structural similarity) metrics, highlight the advantages of our method in terms of shape preservation, boundary regularity, and feature alignment. To quantify the performance of boundary regularity, we define the Mask metric by calculating the proportion of white pixels inside the warped stitching mask, which serves as a demonstration of our proficiency in preserving rectangular boundaries.

To validate the effectiveness of our method, we conducted further tests on data unseen in UDIS-D, and some of which was previously utilized in certain traditional stitching methods [1, 11]. Figure 7 showcases stitching results and makes comparisons to the methods presented in Refs. [11, 12]. The close-up views illustrate that our method excels in aligning salient structures, such as lines and characters, and better maintains rectangular boundaries. Figure 8 provides more results and comparisons. The red rectangles highlight the shortcomings of Refs. [11, 12] in terms of structure preservation, feature alignment, and boundary regularity. Furthermore, we offer a quantitative evaluation in Table 1, which vividly compares the performance of different methods.

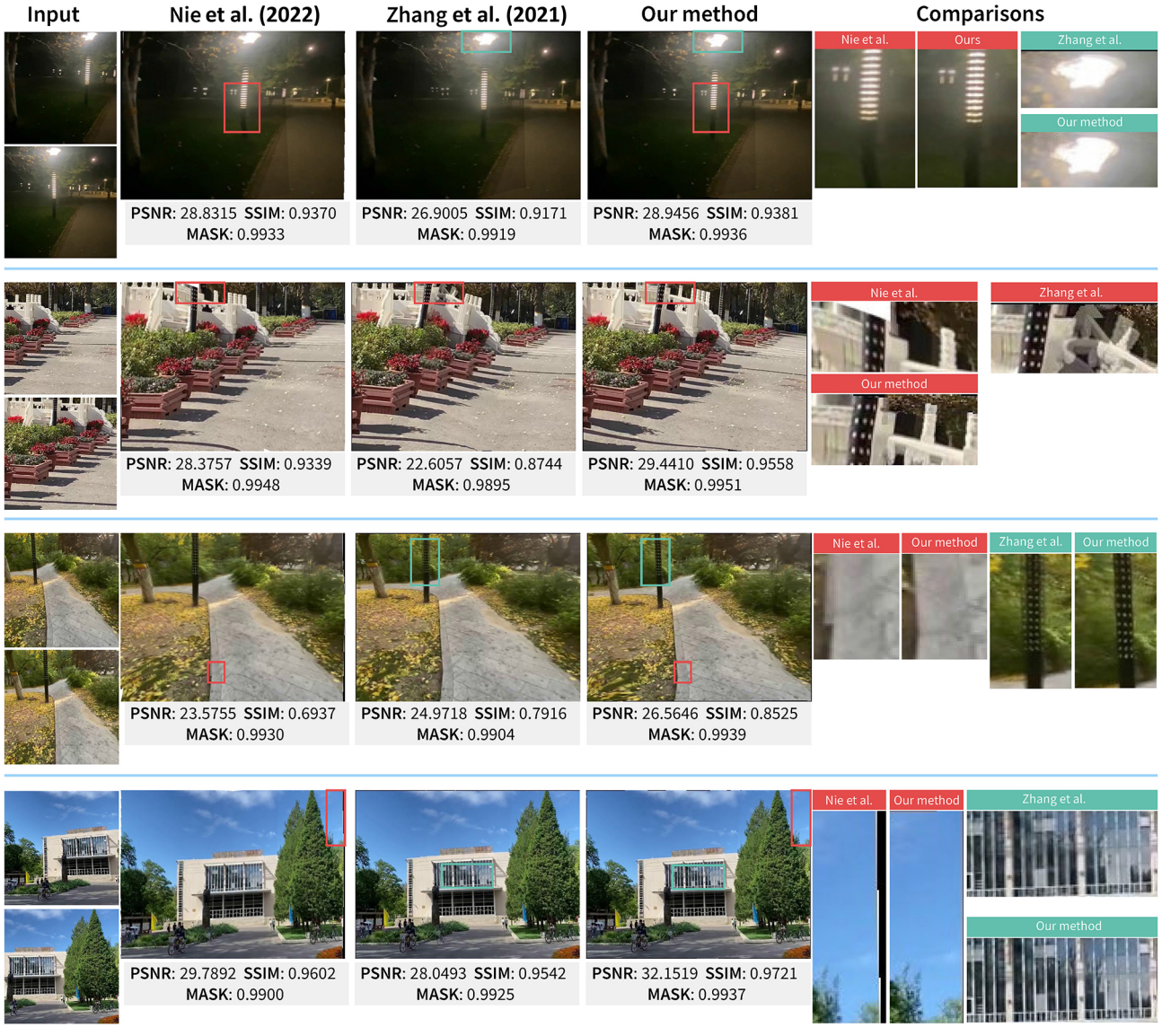



**Fig. 6** Stitching results and comparisons on the *testing* dataset of UDIS-D.

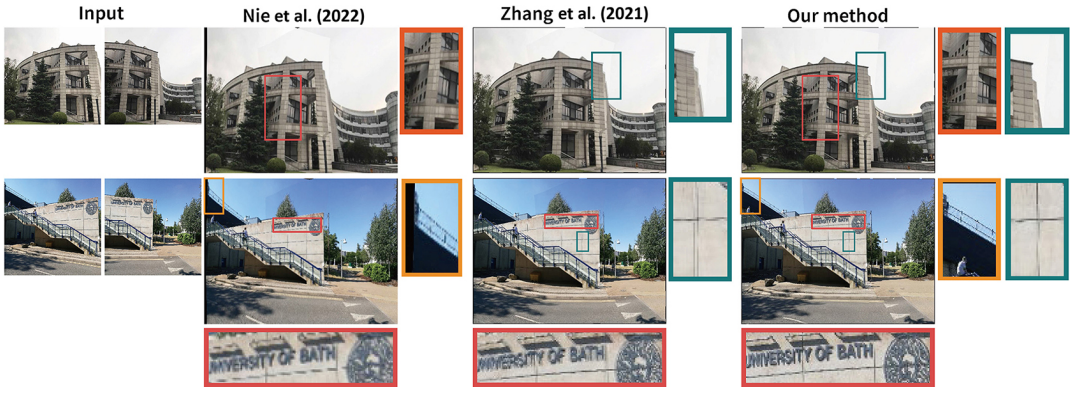



**Fig. 7** Stitching results and comparisons on data unseen in UDIS-D.

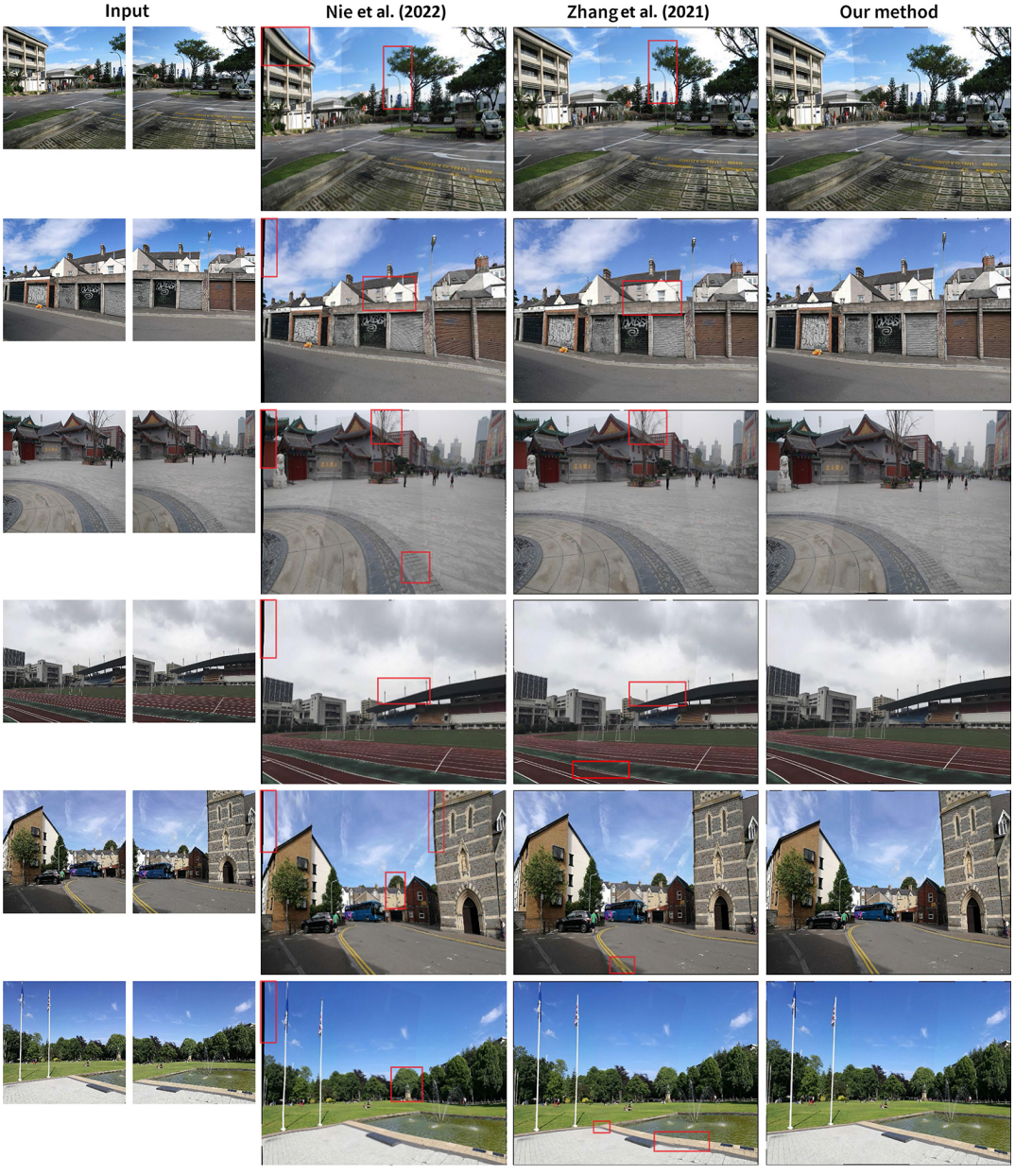



**Fig. 8** Further results and comparisons. The red rectangles point out artifacts in feature alignment, structure preservation, and rectangular boundary preservation.

**Table 1** Metrics for examples in Figs. 7 and 8

| Metrics | Fig. 7(1) | Fig. 7(2) | Fig. 8(1) | Fig. 8(2) | Fig. 8(3) | Fig. 8(4) | Fig. 8(5) | Fig. 8(6) |
|---|---|---|---|---|---|---|---|---|
| Nie et al. (2022) | | | | | | | | |
| PSNR | **25.7282** | **26.8942** | 28.3332 | 27.8733 | 28.3954 | 30.1785 | 30.6975 | 29.9269 |
| SSIM | 0.9442 | **0.9390** | 0.9522 | **0.9430** | 0.9278 | 0.9490 | **0.9668** | **0.9499** |
| Mask | 0.9905 | 0.9873 | 0.9895 | 0.9873 | 0.9906 | 0.9925 | 0.9944 | 0.9927 |
| Zhang et al. (2021) | | | | | | | | |
| PSNR | 23.3887 | 23.1461 | 25.9588 | 24.9477 | 26.6624 | 27.2616 | 27.8547 | 28.2692 |
| SSIM | 0.8947 | 0.8597 | 0.9161 | 0.8844 | 0.9102 | 0.8901 | 0.9335 | 0.9171 |
| Mask | 0.9903 | 0.9908 | 0.9920 | 0.9924 | 0.9921 | 0.9920 | 0.9919 | 0.9920 |
| Ours | | | | | | | | |
| PSNR | 25.4717 | 26.0382 | **29.0105** | **30.0713** | **28.8899** | **33.6945** | **32.9202** | **32.0031** |
| SSIM | **0.9618** | 0.9098 | **0.9619** | 0.9394 | **0.9454** | **0.9647** | 0.9660 | 0.9486 |
| Mask | **0.9951** | **0.9935** | **0.9936** | **0.9950** | **0.9940** | **0.9932** | **0.9947** | **0.9934** |

Results obtained from Ref. [12] excel in feature alignment due to the complete separation of stitching and imposing rectangularity. However, they cannot guarantee rectangular boundaries and preservation of salient structures. In Ref. [11], where stitching and imposing rectangularity are accomplished through global optimization, the rectangular boundaries are well preserved, but artifacts tend to appear in terms of feature alignment.

We performed further extensive quantitative evaluations on the *testing* dataset of UDIS-D, as shown in Table 2. Pseudo-ground-truth in the column 2 refers to the metrics of the results from Ref. [11], which are used as training labels. The last two columns present the metrics of our stitching results before and after refinement. The metrics include PSNR, SSIM, and Mask, which are used to measure feature alignment in the overlapping regions as well as the boundary regularity. In experiments, we find that limitations of Ref. [11] may cause it to fail to generate stitching results when images exhibit characteristics such as low light, low texture, low contrast, low overlap, etc. Out of the 1106 examples in the *testing* dataset of UDIS-D, 1068 examples were successfully stitched by Ref. [11], and the remaining 38 examples could not be stitched correctly. For a fair comparison, the quantitative evaluation was performed on the selected 1068 examples and the remaining 38 examples, separately. The results in Table 2 vividly show the advantages of our method.

Additionally, we select some stitching results produced from the remaining 38 examples of the *testing* dataset of UDIS-D, which exhibit characteristics such as low light, low texture, low contrast, and low overlap. Both the visual results and metrics in Fig. 9 illustrate that our method is effective and robust in challenging scenarios. Both quantitative and qualitative results affirm the effectiveness of our method in terms of feature alignment, regular boundary preservation, and structure preservation.

### 4.3 Ablation study

As in the case of the quantitative evaluation in Table 2, we also selected the *testing* dataset of UDIS-D for an ablation study.

**Table 2** Quantitative evaluation on the *testing* dataset of UDIS-D. The upper part and the lower part give quantitative evaluations on the selected 1068 examples and the remaining 38 examples of the *testing* dataset of UDIS-D

| Metrics | Pseudo-ground-truth | Nie et al. (2022) | Ours | Ours+refinement |
|---|---|---|---|---|
| Selected 1068 testing examples of UDIS-D | | | | |
| PSNR | 25.0656 | 25.9212 | 21.3544 | **27.7812** |
| SSIM | 0.8454 | 0.8581 | 0.7020 | **0.8958** |
| Mask | 0.9903 | 0.9913 | 0.9889 | **0.9941** |
| Boundary | 0.0002 | 0.00017 | 0.0016 | **0.00014** |
| Remaining 38 testing examples of UDIS-D | | | | |
| PSNR | — | 24.7549 | 20.9781 | **26.7898** |
| SSIM | — | 0.8292 | 0.7281 | **0.8772** |
| Mask | — | 0.9909 | 0.9742 | **0.9920** |

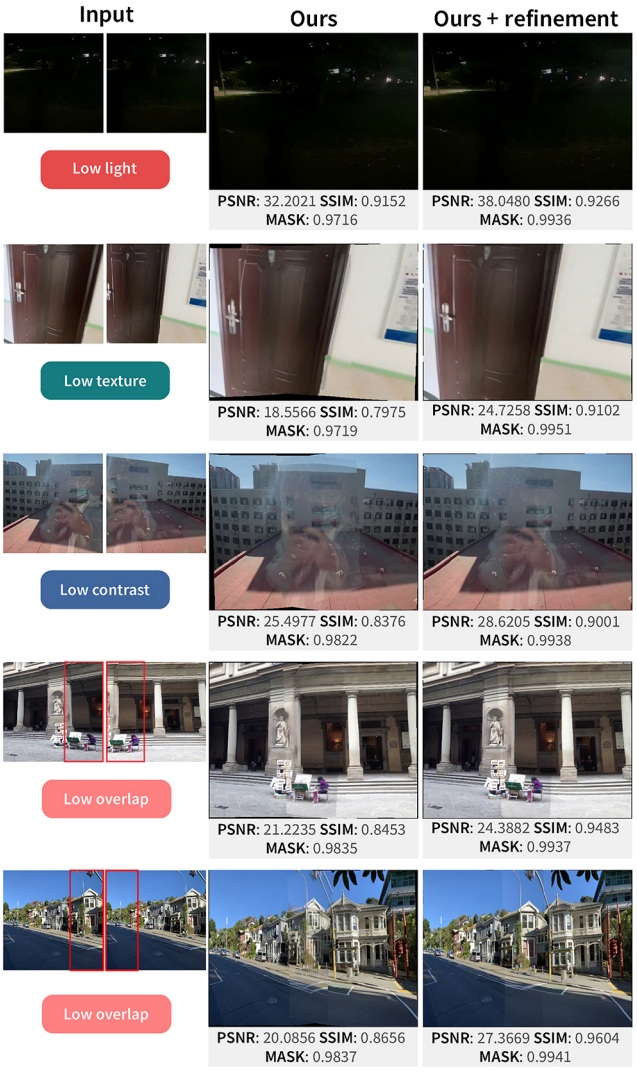



**Fig. 9** Challenging examples of the *testing* dataset of UDIS-D, including low light, low texture, low contrast, and low overlap.

We first provide visual results from the ablation study in Fig. 10. From the close-up views and quantitative metrics, it is easy to see that without the mesh and shape constraints, the results are completely unacceptable: there are significant artifacts in feature alignment and shape distortions. Without the perception and correlation constraints, the results are much better, but still have noticeable ghosting and irregular boundary artifacts.

We further conducted a quantitative evaluation for the ablation study test to assess the role of each constraint term and the global correlations, using as metrics PSNR, SSIM, and Mask. In the ablation study, we observed that the "Mask" metric may not accurately represent the regularity of boundaries, as the mesh vertices often exceed the target rectangular boundary, especially when there is no mesh label loss. Therefore, we additionally employed a Boundary metric, which measures the distance between the vertices on the outer boundary and their target positions (as detailed in Section 3.5.3). Table 3 shows that all the constraint terms and the global correlation play an important role in improving the performance of stitching.

### 4.4 Speed

In terms of running time, our experiments show that the average time for image stitching for the *testing* dataset is 51 ms, which is significantly faster than the traditional method in Ref. [11]. Each iteration of stitching refinement requires 35 ms, with an average of 50 iterations. We provide a speed comparison on the *testing* dataset of UDIS-D in Table 4. For Nie et al.'s method [12], the running time includes both the time spent in initial stitching using the learning-based method [9] and their rectangularity imposition process. In comparison, our learning-based method is shown to be more efficient.

Following refinement, our speed is comparable to that of the traditional method [11]. However, our results surpass it in terms of feature alignment and boundary regularity. Furthermore, our method demonstrates greater robustness in many challenging cases.

### 4.5 Discussion

In this paper, we propose *RecStitchNet*, which combines imposing rectangularity and stitching in a unified learning based framework. It is natural to compare our method with the two-network cascade approach for stitching and imposing rectangularity. Actually, in this paper, the results of Nie et al.

**Table 3** Ablation study metrics

| Metric | w/o shape | w/o perception | w/o mesh | w/o correlation | Our method |
|---|---|---|---|---|---|
| PSNR | 18.1769 | 21.0704 | 17.8817 | 20.7431 | **21.3544** |
| SSIM | 0.6147 | 0.6690 | 0.5753 | 0.6758 | **0.7020** |
| Mask | 0.9617 | 0.9828 | 0.9937 | 0.9814 | **0.9889** |
| Boundary | 0.0382 | 0.0017 | 0.0756 | 0.0020 | **0.0016** |

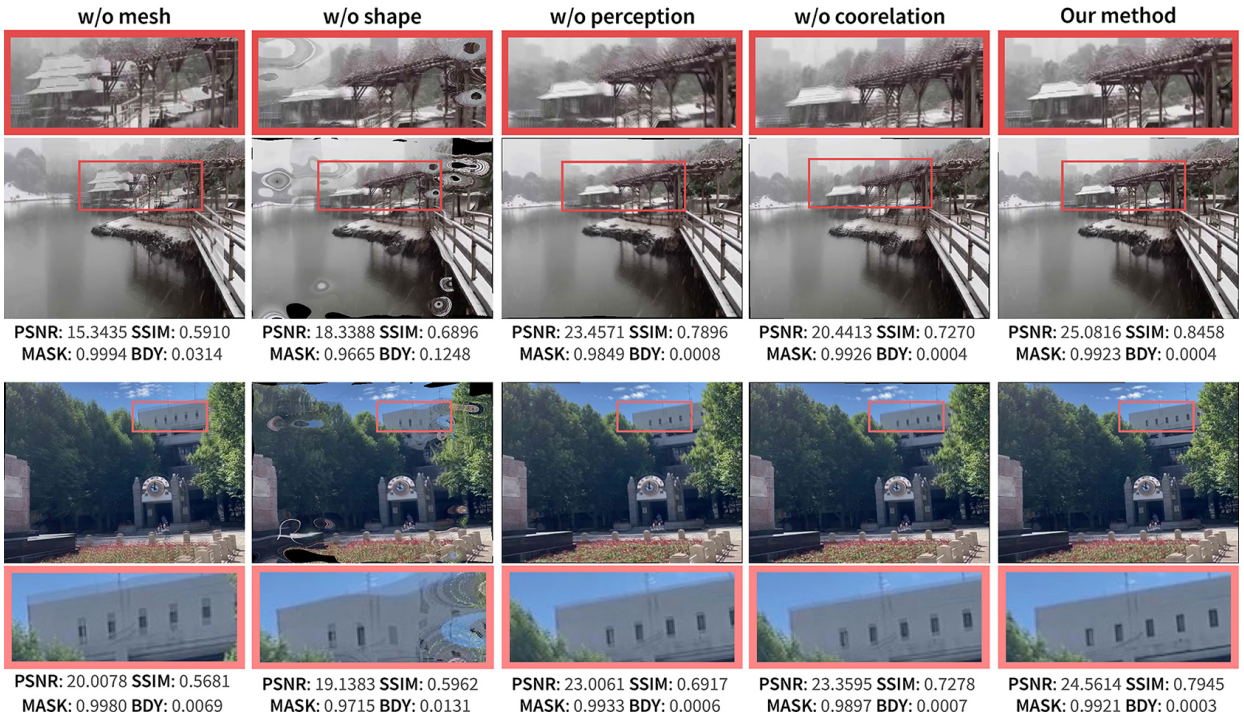



**Fig. 10** Output images in ablation study using the *testing* dataset of UDIS-D.

**Table 4** Average running time (Unit: s)

| Nie et al. | Zhang et al. | Ours | Ours+refinement |
|---|---|---|---|
| 0.256 | 1.413 | **0.211** | 1.921 |

[12] are produced by the cascaded stitching and imposing rectangularity. Our method is superior to the cascaded one, and its advantages are as follows. Firstly, artifacts, such as feature misalignment, in the first stitching step cannot be fixed in the following rectangularity imposition step, and can be amplified by warping. In addition, the stitching performance of different methods may also affect the rectangularity imposition effects. Secondly, a cascaded solution cannot ensure globally optimal results in terms of shape preservation, rectangular boundary imposition, and feature alignment, while our method takes two normal images as input, and learns to perform stitching and impose rectangularity in a unified framework. With a reasonable and effective network and the unsupervised refinement, our method can stably produce high-quality stitching results.

Having a supervised learning framework, we use pseudo-ground-truth as labels for training. The reason is that so far there is no recognized ground-truth for learning based stitching with rectangular boundary, and it is true that the pseudo-ground-truth is theoretically the upper limit to the boundary of the learned model in this step. However, we have to point out that it would be very difficult to train our *RecStitchNet* without labels, and after training using the pseudo-ground-truth, we can obtain acceptable stitching results at a very small cost, with only a few artifacts regarding feature alignment and rectangular boundary preservation, which also exist in the pseudo labels. To break the bottleneck of pseudo-labels and further improve stitching performance, we further refine the stitching results using an unsupervised learning method, which can produce high quality stitching results with better performance than the pseudo-ground-truth, as shown in our quantitative and qualitative evaluations.

## 5 Conclusions

This paper has presented *RecStitchNet*, a novel learning-base framework for image stitching with regular boundaries. Compared to traditional stitching and recent learning-based methods, our method can effectively ensure feature alignment, boundary regularity, and salient structure preservation. Our stitching refinement stage enables our model to better adapt to various scenarios and datasets. Although simple yet effective, our method still has some limitations. See Fig. 11: our method may fail to


**Fig. 11** Our method may fail to preserve salient structures (e.g., straight lines) near stitching boundaries.

preserve salient structures (e.g., straight lines) near stitching boundaries when there is large content loss. In addition, our method may fail to stitch correctly when there is very little overlap in the images, and this is also challenging for other methods.

In future, we hope to produce more image-stitching datasets with diverse scenarios and high-quality labels, and further explore more effective networks and constraints for better stitching. In addition, we also would like to extend our learning based framework to video stitching [36, 37], in which stabilization [38] and feature tracking [39] across frames should be considered.

## Acknowledgements

This research was supported by the Zhejiang Province Basic Public Welfare Research Program (No. LGG22F020009), Key Lab of Film and TV Media Technology of Zhejiang Province (No. 2020E10015), and Marsden Fund Council managed by the Royal Society of New Zealand (No. MFP-20-VUW-180). We also would like to express our special gratitude to Dr. Yaqi Wang for her great work in diagram enhancement.

## Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Fang-Lue Zhang is the Guest Editor of the CVM 2024 Special Issue.

## References

[1] Chen, Y. S.; Chuang, Y. Y. Natural image stitching with the global similarity prior. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9909*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 186–201, 2016.

[2] Gao, J.; Kim, S. J.; Brown, M. S. Constructing image panoramas using dual-homography warping. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, 49–56, 2011.

[3] Lin, C. C.; Pankanti, S. U.; Ramamurthy, K. N.; Aravkin, A. Y. Adaptive as-natural-as-possible image stitching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1155–1163, 2015.

[4] Nie, L.; Lin, C.; Liao, K.; Liu, M.; Zhao, Y. A view-free image stitching network based on global homography. *Journal of Visual Communication and Image Representation* Vol. 73, Article No. 102950, 2020.

[5] Zhao, Q.; Ma, Y.; Zhu, C.; Yao, C.; Feng, B.; Dai, F. Image stitching via deep homography estimation. *Neurocomputing* Vol. 450, 219–229, 2021.

[6] Kweon, H.; Kim, H.; Kang, Y.; Yoon, Y.; Jeong, W.; Yoon, K. J. Pixel-wise warping for deep image stitching. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 37, No. 1, 1196–1204, 2023.

[7] Song, D. Y.; Lee, G.; Lee, H.; Um, G. M.; Cho, D. Weakly-supervised stitching network for real-world panoramic image generation. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13676*. Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 54–71, 2022.

[8] Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Unsupervised deep image stitching: Reconstructing stitched features to images. *IEEE Transactions on Image Processing* Vol. 30, 6184–6197, 2021.

[9] Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Parallax-tolerant unsupervised deep image stitching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 7365–7374, 2023.

[10] He, K.; Chang, H.; Sun, J. Rectangling panoramic images via warping. *ACM Transactions on Graphics* Vol. 32, No. 4, Article No. 79, 2013.

[11] Zhang, Y.; Lai, Y. K.; Zhang, F. L. Content-preserving image stitching with piecewise rectangular boundary constraints. *IEEE Transactions on Visualization and Computer Graphics* Vol. 27, No. 7, 3198–3212, 2021.

[12] Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Deep rectangling for image stitching: A learning baseline. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5730–5738, 2022.

[13] Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Deep rotation correction without angle prior. *IEEE Transactions on Image Processing* Vol. 32, 2879–2888, 2023.

[14] Brown, M.; Lowe, D. G. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision* Vol. 74, No. 1, 59–73, 2007.

[15] Lin, W. Y.; Liu, S.; Matsushita, Y.; Ng, T. T.; Cheong, L. F. Smoothly varying affine stitching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 345–352, 2011.

[16] Zaragoza, J.; Chin, T. J.; Tran, Q. H.; Brown, M. S.; Suter, D. As-projective-as-possible image stitching with moving DLT. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 7, 1285–1298, 2014.

[17] Lou, Z.; Gevers, T. Image alignment by piecewise planar region matching. *IEEE Transactions on Multimedia* Vol. 16, No. 7, 2052–2061, 2014.

[18] Lin, K.; Jiang, N.; Cheong, L. F.; Do, M.; Lu, J. SEAGULL: seam-guided local alignment for parallax-tolerant image stitching. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9907*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 370–385, 2016.

[19] Zhang, F.; Liu, F. Parallax-tolerant image stitching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3262–3269, 2014.

[20] Gao, J.; Li, Y.; Chin, T.; Brown, M. S. Seam-driven image stitching. In: Proceedings of the 34th Annual Conference of the European Association for Computer Graphics, 2013.

[21] Chang, C. H.; Sato, Y.; Chuang, Y. Y. Shape-preserving half-projective warps for image stitching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3254–3261, 2014.

[22] Li, N.; Xu, Y.; Wang, C. Quasi-homography warps in image stitching. *IEEE Transactions on Multimedia* Vol. 20, No. 6, 1365–1375, 2018.

[23] Liao, T.; Li, N. Single-perspective warps in natural image stitching. *IEEE Transactions on Image Processing* Vol. 29, 724–735, 2020.

[24] Du, P.; Ning, J.; Cui, J.; Huang, S.; Wang, X.; Wang, J. Geometric structure preserving warp for natural image stitching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3678–3686, 2022.

[25] Jia, Q.; Li, Z.; Fan, X.; Zhao, H.; Teng, S.; Ye, X.; Latecki, L. J. 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, 12181–12190, 2021.

[26] Zhang, L.; Huang, H. Image stitching with manifold optimization. *IEEE Transactions on Multimedia* Vol. 25, 3469–3482, 2023.

[27] Li, N.; Liao, T.; Wang, C. Perception-based seam cutting for image stitching. *Signal, Image and Video Processing* Vol. 12, No. 5, 967–974, 2018.

[28] Lai, W. S.; Gallo, O.; Gu, J.; Sun, D.; Yang, M. H.; Kautz, J. Video stitching for linear camera arrays. In: Proceedings of the 30th British Machine Vision Conference, 2019.

[29] He, K.; Chang, H.; Sun, J. Content-aware rotation. In: Proceedings of the IEEE International Conference on Computer Vision, 553–560, 2013.

[30] Wu, J. L.; Shi, J. J.; Zhang, L. Rectangling irregular videos by optimal spatio-temporal warping. *Computational Visual Media* Vol. 8, No. 1, 93–103, 2022.

[31] Liao, K.; Nie, L.; Lin, C.; Zheng, Z.; Zhao, Y. RecRecNet: Rectangling rectified wide-angle images by thin-plate spline model and DoF-based curriculum learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 10766–10775, 2023.

[32] Zhou, H.; Zhu, Y.; Lv, X.; Liu, Q.; Zhang, S. Rectangular-output image stitching. In: Proceedings of the IEEE International Conference on Image Processing, 2800–2804, 2023.

[33] Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 2, 2017–2025, 2015.

[34] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the 3rd International Conference on Learning Representations, 2015.

[35] Martínez, F.; Rueda, A. J.; Feito, F. R. A new algorithm for computing Boolean operations on polygons. *Computers & Geosciences* Vol. 35, No. 6, 1177–1185, 2009.

[36] Wang, M.; Shamir, A.; Yang, G. Y.; Lin, J. K.; Yang, G. W.; Lu, S. P.; Hu, S. M. BiggerSelfie: Selfie video expansion with hand-held camera. *IEEE Transactions on Image Processing* Vol. 27, No. 12, 5854–5865, 2018.

[37] Nie, Y.; Su, T.; Zhang, Z.; Sun, H.; Li, G. Dynamic video stitching via shakiness removing. *IEEE Transactions on Image Processing* Vol. 27, No. 1, 164–178, 2018.

[38] Wang, M.; Yang, G. Y.; Lin, J. K.; Zhang, S. H.; Shamir, A.; Lu, S. P.; Hu, S. M. Deep online video stabilization with multi-grid warping transformation

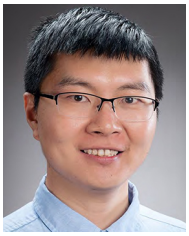
learning. *IEEE Transactions on Image Processing* Vol. 28, No. 5, 2283–2292, 2019.

[39] Rong, J. X.; Zhang, L.; Huang, H.; Zhang, F. L. IMU-assisted online video background identification. *IEEE Transactions on Image Processing* Vol. 31, 4336–4351, 2022.

**Yun Zhang** is currently a professor in the College of Media Engineering, Communications University of Zhejiang, China. He received his doctoral degree from Zhejiang University in 2013. Before that, he received his bachelor and master degrees from Hangzhou Dianzi University in 2006 and 2009, respectively. He visited the Visual Computing Group of Cardiff University in 2018 and 2023. His research interests include computer graphics, image and video editing, and virtual reality. He is a senior member of the CCF.

**Yu-Kun Lai** received his bachelor and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a professor in the School of Computer Science & Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing, and computer vision. He is on the editorial boards of *IEEE Transactions on Visualization and Computer Graphics* and *The Visual Computer*.

**Lang Nie** received his B.S degree in computer science and technology from Beijing Jiaotong University, China, in 2019, and is currently pur-suing a Ph.D. degree in signal and information processing from the Institute of Information Science, Beijing Jiaotong University. His current research interests include image and video processing, 3D vision, and multi-view geometry.



**Fang-Lue Zhang** received his Ph.D. degree from Tsinghua University, in 2015. He is currently a senior lecturer in computer graphics at the Victoria University of Wellington, New Zealand. His research interests include image and video editing, computer vision, and computer graphics. He received a Victoria Early-Career Research Excellence Award, in 2019, and a Fast-Start Marsden Grant from the New Zealand Royal Society, in 2020. He is on the editorial board of *Computers & Graphics*.

**Lin Xu** is currently pursuing her Ph.D. degree at the University of South Australia, with an anticipated graduation date in 2024. She received her bachelor and master degrees, as well as her first Ph.D. degree from Fujian Normal University of China in 2006, 2010, and 2014, respectively. Lin furthered her academic pursuits with visits to the College of Creativity and Technology at Fo Guang University in 2015 and the Faculty of Sciences at Université Libre de Bruxelles in 2017. Her research interests include computer vision, multi-media systems, and intelligence computing.