Computational Visual Media
https://doi.org/10.1007/s41095-0xx-xxxx-x

Research/Review 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 the 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 rectangling 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 of stitching and rectangling. In the first step, we perform initial stitching by a pre-trained state-of-the-art (SOTA) image stitching model, to produce initially warped stitching results without considering the boundary constraint. Then, we design 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 rarity of the stitching dataset and the difficulty of label generation, we propose to generate a stitching dataset with rectangular stitched images as pseudo Ground Truth (GT) labels, and the performance upper boundary induced from the pseudo GT can be broken by our unsupervised refinement. Qualitative and quantitative results and evaluations demonstrate the advantages of our method over SOTA methods.

Keywords irregular boundaries, learning-based, convolution neural network, regression, stitching, rectangling

1 Introduction

In recent decades, image stitching has been an active topic in computer graphics and vision. The main task of image stitching is to construct a wide field-of-view (FOV) scene from several overlapped images with limited FOV, which 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], etc. 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, low light, and low resolutions. Additionally, preserving geometric structure and visual features necessitates complex optimization and intensive computation, further heightening the difficulty of image stitching.

To overcome the challenges posed by feature matching and structure preservation, learning-based methods have been extensively studied in recent years, stitching 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.

1 Communication University of Zhejiang, Hangzhou 310018, China. E-mail: zhangyun@cuz.edu.cn.

2 Cardiff University, Cardiff CF24 4AG, Wales, UK. E-mail: Yukun.Lai@cs.cardiff.ac.uk.

3 Beijing Jiaotong University, Beijing 100091, China. E-mail: nielang@bjtu.edu.cn.

4 Victoria University of Wellington, Wellington 6012, New Zealand. E-mail: fanglue.zhang@ecs.vuw.ac.nz.

5 University of South Australia, Adelaide 5095, Australia. E-mail: xuyly032@mymail.unisa.edu.au.

Manuscript received: 2024-01-01; accepted: 2024-01-01

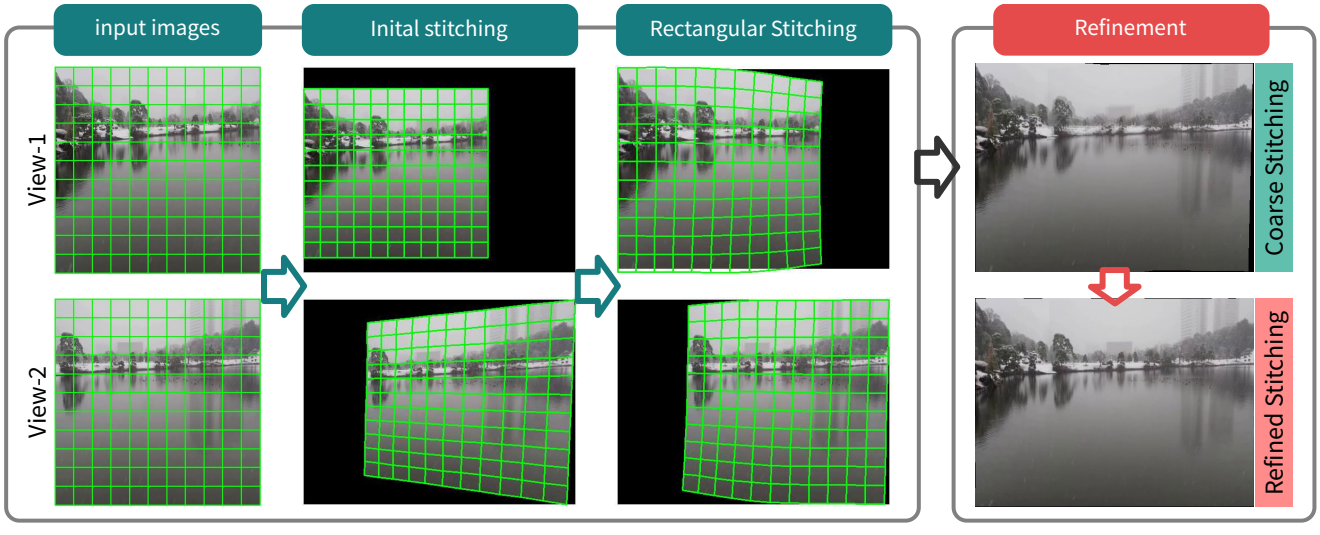


Fig. 1 Pipeline of our method. We take two normal FOV images as input, and then stitch them using pre-trained model. Taking the initial stitching result with irregular boundary as input, we construct the *RectStitchNet*, which can produce coarse stitching results with rectangular boundary. Finally, we further refine the stitching result by an instance-wise unsupervised learning method.

Recently, following previous work on image stitching and rectangling based on conventional optimization frameworks [10, 11], Nie *et al*. [12] proposed the first deep learning solution for image rectangling, which was further extended to image rotation correction [13]. They took the 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 the stitching simultaneously while learning to create rectangular image boundaries. This oversight could potentially lead to the amplification of artifacts in the stitched input after applying their 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 design a three-step progressive stitching approach. Firstly, we conduct an initial stitching process using a state-of-the-art (SOTA) deep stitching technique, acquiring the warped meshes of image pairs. Secondly, we further design a regression network 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 the 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 rectangling are challenging tasks, necessitating a supervised network for effective learning. However, obtaining the ground truth stitching results is difficult due to the absence of a publicly recognized standard. Given that rectangular stitching results can be more standardized and universally recognized, we propose to generate pseudo GT using a state-of-the-art stitching technique with rectangular boundaries [11].

Fig. 1 shows the pipeline of our method. Given two normal FOV images as input, the proposed solution progressively stitches them from an initial stitching image with irregular boundaries to a coarse stitching image with rectangular boundaries to the final stitching result with further refinement on 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 with the previous stitching and rectangling method [11], our method is more robust and efficient due to the 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 fragile and expensive feature matching in traditional methods, so is much more robust and efficient compared with these methods. As

will be demonstrated later using extensive experimental results, our method achieves SOTA performance, both qualitatively and quantitatively, outperforming traditional methods and deep learning baselines.
- To ensure high-quality stitching across a wide range of scenarios, we introduce an unsupervised instance-wise learning strategy to iteratively optimize the stitching results.
- Given the absence of an existing dataset for supervised learning, we have created a new dataset called *RecStitching*, which includes pseudo GT mesh warping results that strictly selected and re-rendered from the traditional stitching results with rectangular boundaries.

2 Related Work

Our deep stitching with rectangular boundaries is closely related to image stitching and image rectangling techniques.

2.1 Traditional Image stitching

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

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

To produce more natural stitching with less perspective distortions, several warping schemes are proposed, which are 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 geometry structure preserving [24].

Recently, Jia *et al*. [25] considered the global collinear structure, which effectively preserves global and local structures while reducing distortions. Zhang *et al*. [26] proposed manifold preserving stitching, and the on-manifold operations help to reduce ghosting and distortion artifacts. To improve the stitching results, seam-cutting methods are applied to removing artifacts in overlapping regions [20, 27].

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

2.2 Deep Image stitching

Different from the methods in Section 2.1, 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 roughly divide recent research into three main types:

Supervised learning. Nie *et al*. [4] and Zhao *et al*. [5] proposed view-free image stitching based on global homography learning, which improves previous learning based stitching method [28] limited by 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 by the pixel-wise warp field, which can well handle the large-parallax problem.

Weakly Supervised Learning. To overcome the difficulties in dataset and Ground Truth (GT) generation, Song *et al*. [7] proposed a weakly-supervised learning method to train the stitching model without the real GT images. They further extended their method to stitching multiple images and creating 360-degree panoramas.

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, which consists of unsupervised coarse image stitching and image reconstruction. Very recently, Nie *et al*. [9] further proposed a parallax-tolerant unsupervised image stitching which is characterized by combining homography and thin-plate splines (TPS) into a unified framework.

2.3 Image rectangling

Image rectangling aims to regulate the irregular boundaries caused by image stitching, rotation, etc. The pioneering work of image rectangling is [29], in which a content-aware warping method was proposed through a two-stage warping based optimization on meshes. Wu *et al*. [30] further extended the image rectangling to videos, which incorporates temporal coherence into the warping based optimization. Nie *et al*. [12] proposed a one-stage learning baseline of deep rectangling for image stitching. Compared with the two-stage method in [29], the method in [12] is more efficient and robust, and can well

preserve non-linear structures thanks to the high-level feature extraction in the learning framework. Liao *et al*. [31] proposed a rectangling rectification network, which applies the TPS module to perform non-linear and non-rigid transformations for wide-angle rectified image rectangling. Very recently, Zhou *et al*. [32] combined stitching and rectangling into a unified end-to-end framework using a synthesized dataset. Although effective in producing stitching results with rectangular boundaries, it still suffers from the content loss and ghost effects in the overlapping regions.

3 Our Method

Similar to recent progress in stitching and rectangling [1, 9, 11, 12, 23], we also stitch images by the 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 the energy optimization with constraints on all grid vertices of the mesh. Let $V = \{V^i, i = 1, 2, ..., N\}$ be the sets containing all vertices of 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 contains several content-aware constrains, such as feature alignment, shape preserving, straight line preserving etc., and these methods usually focus on designing energy terms that are effective in stitching and easy to optimize. Different from traditional methods, the deep learning based methods [9, 12] focus on the dataset preparation, network construction, and mesh regression. For effective mesh regression, we have to focus on designing the objective function, which can effectively guide the training process, and the total loss can achieve satisfactory convergence. To calculate the feature loss after the mesh warping, an effective and efficient warping operation is required, which is also differentiable for effective gradient propagation.

Inspired by the methods above, we propose a novel method to achieve stitching and boundary rectangling simultaneously in a learning-based framework. Fig. 2 shows an overview of our deep stitching with rectangular boundaries. We take two images of the same size with partial overlap as input, and 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, and the warping is guided by the initial meshes generated by the SOTA 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 on top of the initial meshes. Finally, with the combination of initial meshes and their mesh motions, the final stitching result can be easily produced by warping and average blending.

3.1 Initial Warping

In this stage, the input images are initially stitched using the state-of-the-art deep stitching model [9]. As illustrated in Fig. 3, given two input images $\{I^i, i = 1, 2\}$ to be stitched, a rigid quad mesh υ^i is placed on each image I^i, the initial warping produces the warped meshes $\{\xi^i\}$ after the deep stitching process. To facilitate the deep stitching task, the first image is consistently kept unchanged, while the other image is warped to align with it. Next, high-level semantic features are extracted from each input image (without warping) through a series of convolution and pooling blocks. These blocks, represented by the blue solid blocks in Fig. 2, and each block comprises two convolution layers. After the first, second, and third blocks, a max-pooling layer is applied. We set the channel numbers 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 two images in the overlapping regions, we concatenate the global correlation [4] with the extracted features of each image. Given the extracted features (F^1, F^2) of the input images, their global correlation refers to the feature-wise similarities, and is defined as

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

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. Consequently, the warped features $\{\kappa^i\}$ of input images, representing the features of the initial stitching results, are generated and serve as the input for the mesh motion regression process.

3.2 Mesh Motion Regression

In this stage, our goal is to obtain the mesh motion, which denotes the offsets of the initially warped mesh vertices, and helps regulate the shape of the final stitching boundaries. As depicted in the middle section of Fig. 2, the input for this stage consists of the extracted high-level features that have been warped by the meshes from the initial stitching. We design 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 acquire the predicted meshes $\{\psi^i\}$, by combining the mesh motion $\{\chi^i\}$ with the initial warped

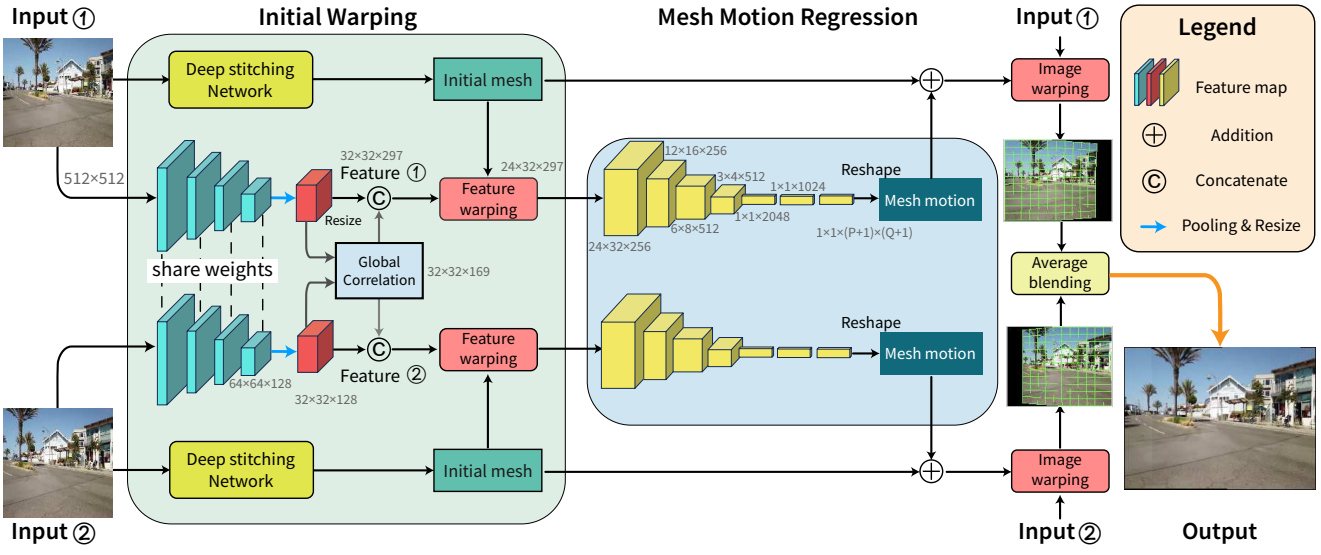


Fig. 2 Overview of the deep stitching with rectangular boundaries network. Our supervised learning network consists of Initial Warping and Mesh Motion Regression steps. In the first step, we warp the high-level features extracted from input images, and the warping is guided by the meshes generated by a SOTA deep stitching model [9]. With the warped features as input, we further obtain the mesh motion (vertex offsets) on the initial meshes through mesh motion regression. Finally, the stitching results are obtained by averaging the images warped by the mesh combined by the initial mesh and mesh motion.

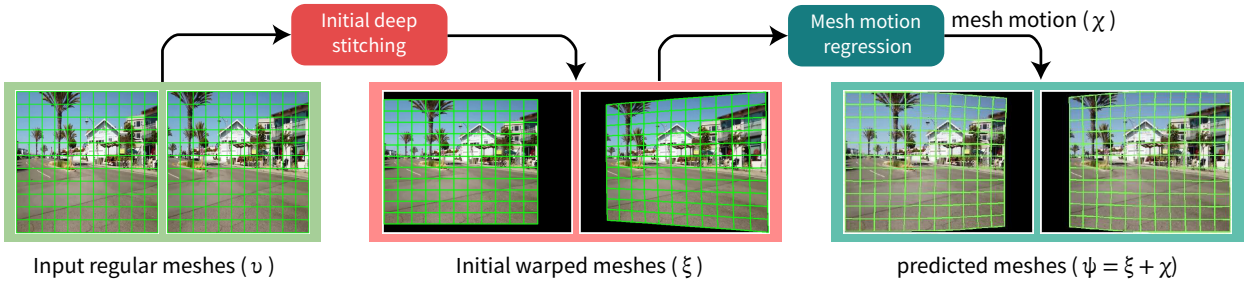


Fig. 3 Mesh manipulations in initial stitching and mesh motion regression.

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.

3.3 Loss Functions

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 follows:

$$L_{train} = \varphi_m l_m + \varphi_p l_p + \varphi_s l_s, \tag{2}$$

where l_m, l_p, l_s refer to the mesh, perception, shape preserving loss terms, and φ_m, φ_p, φ_s are the corresponding weights.

In our supervised training framework, we have prepared a large dataset (refer to Section 3.5), which contains the input image pairs, pseudo GT mesh labels representing warped mesh vertices, pseudo GT stitching result labels, where the pseudo GT data is produced by a SOTA traditional stitching method with rectangular boundaries [11].

3.3.1 *Mesh Loss*

Given the predicted meshes $\{\psi^i\}$, we simply constrain them to be close to the ground truth (GT) label of meshes Ψ^i, which is defined as:

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

where ψ_j^i and Ψ_j^i refer to the vertex positions of the predicted mesh and the pseudo GT mesh.

3.3.2 *Perceptual Loss*

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

$$l_p = \sum_{i=1}^{2} \|\Gamma(\Omega_{tps}(\psi^i, I^i)) - \Gamma(\Omega_{tps}(\Psi^i, I^i))\|_1, \quad (4)$$

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

3.3.3 *Shape Preserving Loss*

Similar to [12], we also preserve the shape of the mesh using the intra-grid and inter-grid shape similarity constraints, which are defined as:

$$l_s = l_s^{intra} + l_s^{inter}. \quad (5)$$

Intra-grid constraint is employed to enforce both the scale and direction of the grid edges, and is defined as follows:

$$l_s^{intra} = \sum_{i=1}^{2} \frac{\sum_{\vec{e}_j \in \vec{h}_i} Re(\Delta_x(\vec{e}_j) + \sigma \frac{W}{Q})}{(P+1) \times Q} + \sum_{i=1}^{2} \frac{\sum_{\vec{e}_k \in \vec{v}_i} Re(\Delta_y(\vec{e}_k) + \sigma \frac{H}{P})}{P \times (Q+1)}, \quad (6)$$

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

Inter-grid constraint is to enforce two 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), and it is defined as:

$$l_s^{inter} = \sum_{i=1}^{2} \frac{1}{N} \sum_{\{\vec{e}_{t1}, \vec{e}_{t2}\} \in \Lambda^i} (1 - \cos(\vec{e}_{t1}, \vec{e}_{t2})), \quad (7)$$

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

3.4 Unsupervised Instance-wise Stitching Refinement

In the mesh motion regression step, our loss functions are designed to enforce that the predicted stitching result is close to both the pseudo GT mesh and the stitched image while preserving mesh shape. However, experiments showed that some predicted results may not guarantee perfect boundary regularity and feature matching (refer to 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 learning using pseudo mesh labels), preventing the network parameters from being optimized to the best optima. 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 the 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 ghost in image stitching.

As illustrated in Fig. 4, 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 Alg. 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 Φ. To iteratively optimize the stitching boundary, we first need to obtain the boundary vertices of the stitching result. Drawing inspiration from [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 follows

$$\hat{P} = \hat{P}^1 \bigcup \hat{P}^2. \quad (8)$$

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 iteration terminates 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 Φ through blending the warped images.

In the refinement step, we use a different set of loss functions for the instance-wise unsupervised learning, and the definitions are described below.

Algorithm 1 Refinement of 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 Φ;
Let $\hat{P}^i$ be the boundary vertices of ψ^i;
Let $\hat{P}$ be the outer boundary vertices of the two meshes $\{\psi^i\}$, $i = 1, 2$, and $\hat{P}$ is calculated using Equ. 8;
foreach $j \in [1, 200]$ **do**
  foreach $i \in [1, 2]$ **do**
    χ_i = *RecStitchNet*(ψ^i, I^i);
    ψ^i = ψ^i + χ^i;
    $\Theta^i = \psi^i$;
  end foreach
  if $j == 1$ **then**
    $Loss_{pre} = Loss(RecStitchNet)$;
  end if
  else
    $Loss_{now} = Loss(RecStitchNet)$;
    if $|Loss_{now} - Loss_{pre}| < e^{-5}$ **then**
      break;
    end if
    $Loss_{pre} = Loss_{now}$
  end if
end foreach
foreach $i \in [1, 2]$ **do**
  $R^i = \Omega_{tps}(\Theta^i, I^i)$
end foreach
$\Phi = Blending(R^1, R^2)$;

3.4.1 *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.

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

where $\Omega_{tps}(\cdot)$ 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 $\{\psi^i\}$.

3.4.2 *Rectangular Boundary Loss*

In [12], the rectangular boundary loss is simply defined as the difference between the $\{0, 1\}$ mask of the result and the all '1' mask. However, we found in the experiment 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 stitching boundary, we first extract the outer boundary $\hat{P}$ of the warped and overlaid meshes $\{\psi^i\}$. For each vertex ν_k in $\hat{P}$, we assign several attributes to it, including their constraint direction $\rho(\nu_k) \in \{[1,0], [0,1]\}$ ($[1, 0]$) and $[0, 1]$ refer to the constraint on x and y directions); 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 the target locations as follows:

$$l_b = \left\| \sum_{\nu_k \in \hat{P}} (\nu_k \cdot \rho(\nu_k)) - \tau(\nu_k) \right\|_1. \quad (10)$$

In this stage, the total loss function is a linear combination of feature matching, rectangular boundary, and shape preserving constraints (see details in Sec. 3.3.3) as follows

$$L_{refine} = \varphi_f l_f + \varphi_b l_b + \varphi_s l_s, \quad (11)$$

where φ_f, φ_b, φ_s are the weights to control the importance of the three loss constraints.

3.5 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 called the *RecStitching* dataset, which comprises input images, mesh labels, and image labels (refer to Fig. 5). The details of our data preparation are as follows:

- **Stitching**: The input image pairs are sourced from the dataset of [8]. We perform stitching using the traditional warping-based method described in [11]. This method is capable of generating stitching results with rectangular boundaries, along with the 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 amount of missing content. Furthermore, our stitching method does not have to account for piecewise rectangular boundaries as in [11].

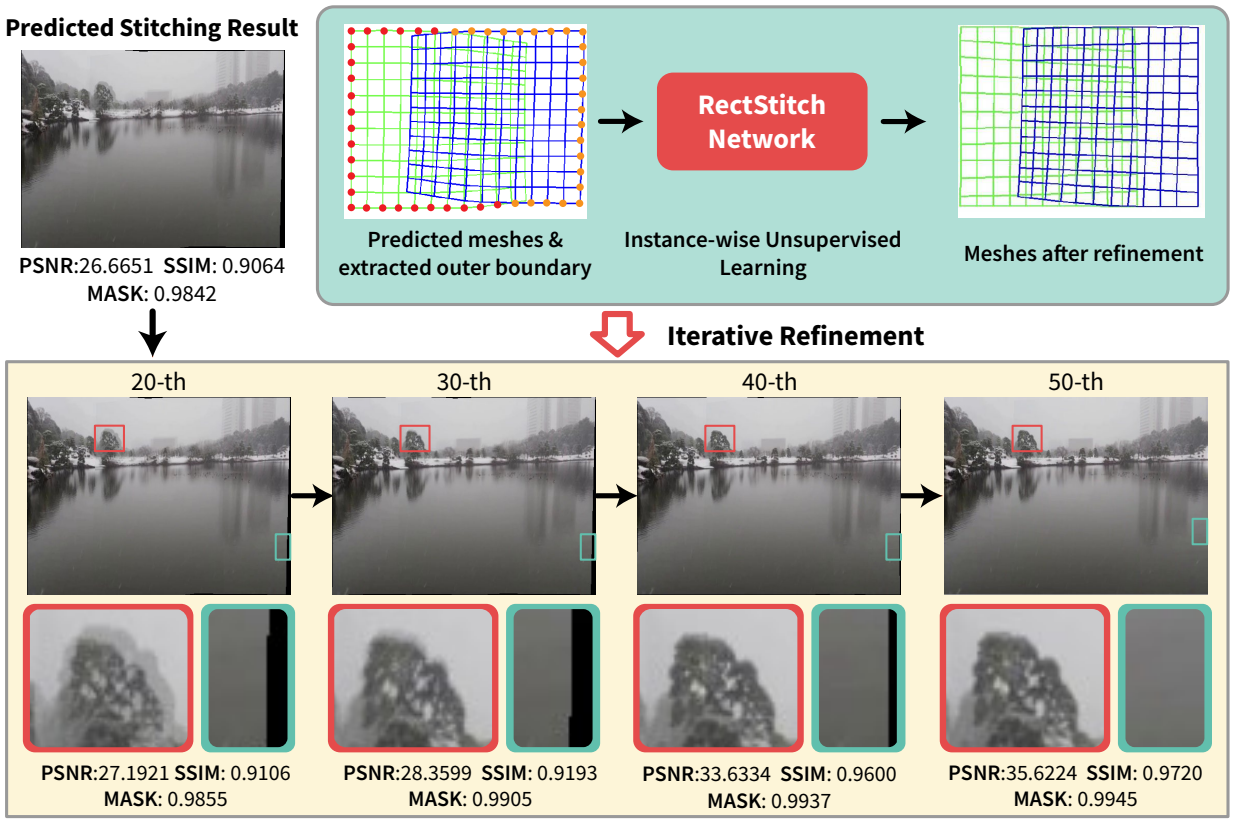


Fig. 4 Refinement of the stitching result. The input of this step is the predicted meshes and the corresponding input image pair. We first extract the outer boundary of the predicted meshes using the polygon Boolean operations [35], then we predict the refined meshes using the instance-wise unsupervised learning framework in an iterative manner. Finally, we obtain an optimized stitching result by warping and blending.

- **Normalization**: For effective training, the 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 $W_s \times H_s$, and for each vertex of the mesh with coordinates (x, y), we convert them to $(x/w_t \times W_s, y/h_t \times H_s)$, where w_t and h_t represent the outer boundary size of the stitching result.
- **Rendering**: We further render the stitching results by warping the input images guided by the normalized meshes. To achieve smoother stitching results, our rendering is performed using the TPS transformation [33], which provides more natural transitions and smoother interpolation compared to mesh-based warping. At this stage, the resolution of each rendered image is also set to $W_s \times H_s$.

Actually, the stitching labels produced by [11] cannot be considered as ideal labels due to limitations in [11]. This may impact the performance of the training model. In this paper, we utilize these labels, considered as *pseudo* Ground Truth(GT), for supervised training. To break the bottleneck of the pseudo GT, we further refine the stitching results using our unsupervised instance-wise learning. Experimental results and evaluations in Section 4 demonstrate that our refined results outperform the training labels produced by [11].

4 Experiments

4.1 Implementation details

In the data preparation and training stages, we set the mesh resolution of each image to 11×11, and resolution of each input image is normalized to 512×512. In the feature extraction and regression stages, we set $kernal\_size = 3$, $stride = 2$ for all convolution blocks and $kernal\_size = 2$, $stride = 2$ for all max pooling layers; we set the search range to 6 to efficiently calculate the correlation of two feature maps (32×32), and the size of the correlation is $(4 \times 32 \times 32 \times 169)$. In the training stage, we use the linear combination of $conv4\_2$, $cov3\_2$ and $conv2\_2$ of the VGG-19 features as the high-level feature of images. The weights of loss terms are set to $\varphi_m = 1$, $\varphi_p = 0.000006$, $\varphi_s = 0.8$. Similar to many CNN-based networks [12], we use the Adam optimizer with a learning rate initialized to $1e-4$ for 100k iterations, and

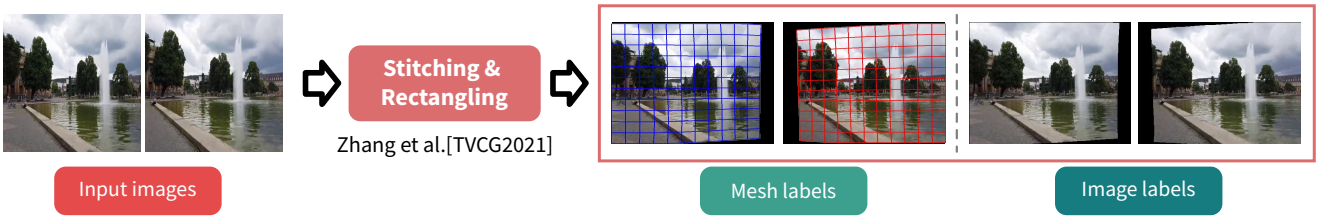


Fig. 5 Dataset preparation. Give a pair of input images, we first stitch them using the method in [11], and then output the resulting mesh labels and the corresponding warped image labels.

the decay rate is set to 0.90. We set $batch\_size = 4$ and use 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 is initialized to 0.002 for fast refinement. All implementation is based on Tensorflow using a single GPU with NVIDIA RTX 4090. To better compare the performance of different stitching methods, we simply use the average blending to composite the overlapping regions.

4.2 Evaluations

To demonstrate the effectiveness of our method, we conduct both qualitative and quantitative evaluations. We compare our method with the SOTA methods that have released their source code to the public.

Fig. 6 displays several stitching results from examples sourced from the *testing* dataset in [8] unseen during training. Given a pair of input images, we perform stitching using the methods from [11, 12] and our method separately. Concerning the method in [12], we initiate the process with an initial stitching using the deep stitching method from [9]. By subsequently utilizing the stitching result and corresponding mask, the final stitching result is obtained through the deep rectangling method from [12]. For Zhang *et al.*'s method [11], the stitching is carried out 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 with [11, 12], our method excels at shaping the rectangular boundary and ensuring precise alignment in the overlapping regions. The marked red and green boxes (zoomed-in 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 in the warped stitching mask, which serves as a demonstration of our proficiency in preserving rectangular boundaries.

To validate the effectiveness of our method, we conduct further tests on other images that have been previously utilized in some traditional stitching methods [1, 11]. Fig. 7 showcases the stitching results and a comparison with the methods presented in [11, 12]. The zoomed-in views illustrate that our method excels in aligning salient structures, such as lines and characters, and better maintains rectangular boundaries. Fig. 9 provides more results and comparisons. The red rectangles highlight the shortcomings of [11, 12] in terms of structure preservation, feature alignment, and boundary regularity. Furthermore, we offer quantitative evaluations in Table 1, which vividly demonstrate the performance comparison of different methods. Results obtained from [12] excel in feature alignment due to the complete separation of stitching and rectangling. However, they cannot guarantee rectangular boundaries and the preservation of salient structures. In [11], where stitching and rectangling are accomplished through global optimization, the rectangular boundaries are well preserved, but artifacts tend to appear in terms of feature alignment.

We further perform extensive quantitative evaluations on the testing dataset from [8], as shown in Table 2. The *Pseudo GT* in the 2^{nd} column refers to the metrics of the results from [11] used as training labels. The last two columns present the metrics of our stitching results before and after refinement. Our metrics include PSNR, SSIM, Mask, which are used to measure feature alignment in the overlapping regions as well as the boundary regularity. In experiments, we find that due to the limitation of [11], it may fail to generate stitching results when images exhibit characteristics such as low contrast, low light, large parallax, and texturelessness. Out of the 1106 examples in the testing dataset from [8], 1068 examples can be successfully stitched by [11], and the others cannot be stitched correctly. For a fair comparison, the quantitative evaluation are performed on the selected 1068 examples and the other 38 examples, respectively, and the comparison results vividly show the advantages of our method, see details in Table 2.

Additionally, we select some stitching results produced

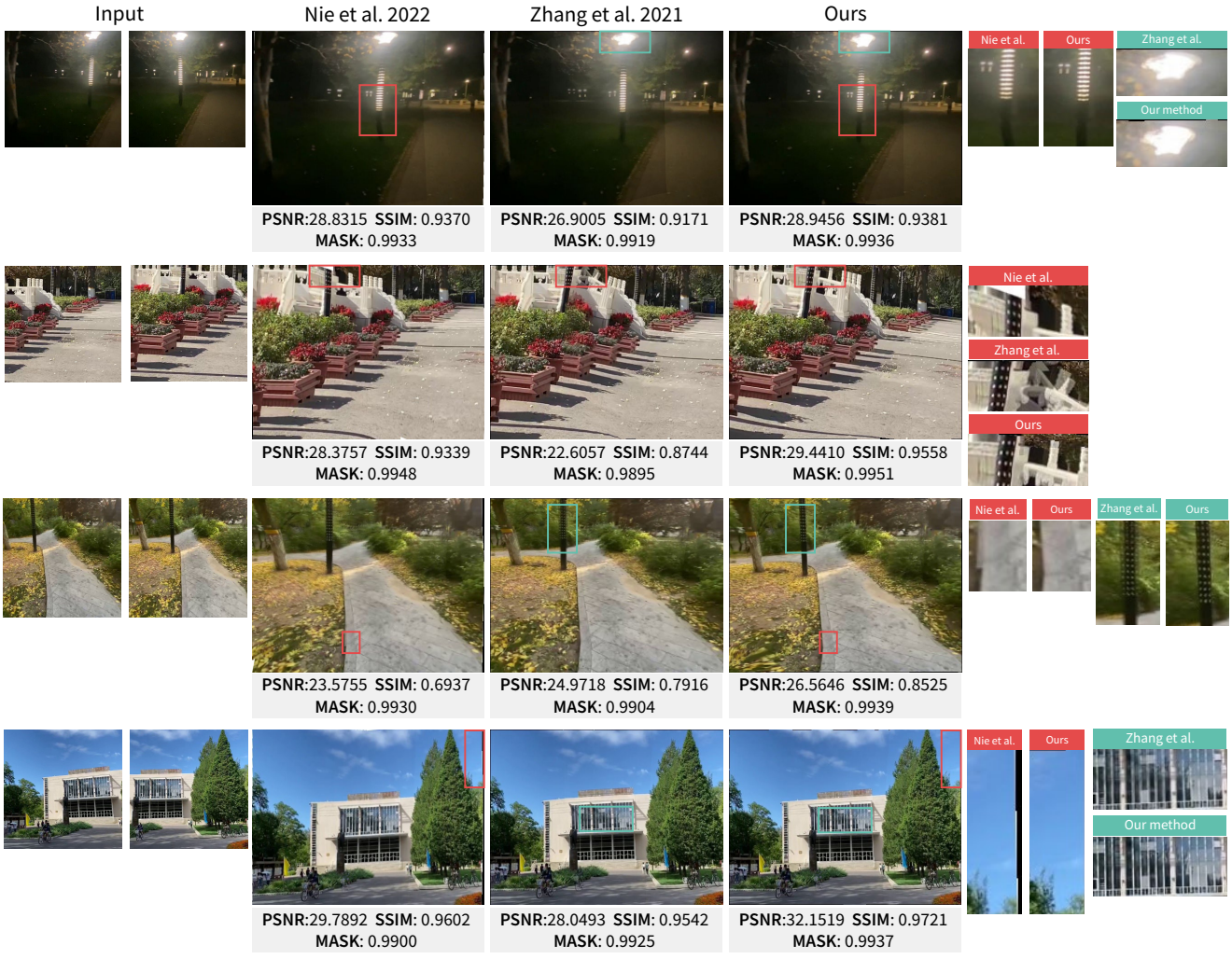


Fig. 6 Stitching results and comparisons of testing dataset in [8].

from the testing dataset from the selected 38 examples [8], which exhibit characteristics such as low light, textureless-ness, low contrast and low overlap. Both the visual results and metrics in Fig. 10 illustrate that our method 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 Abalation Study

Similar to the quantitative evaluations in Table 2, we also select the testing dataset from [8] for abalation study.

We first perform visual comparison of the abalation study, as shown in Fig. 8. From the zoomed-in views and quantitative metrics, it is easy to find that without the mesh and shape constraints, the results are completely unacceptable, and 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 conduct quantitative evaluations for the abalation study to assess the role of each constraint term and the global correlations, and our metrics also include 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 (see details in Section 3.4.2). As shown in Table 3, all the constraint terms and the global correlation take an important role in improving the performance of stitching.

4.4 Performance

In terms of running times, the experiments show that the

Fig. 7 Stitching results and comparisons of other data.

Table 1 Performance comparison of the examples from Figs. 7 and 9.

Metrics	Fig. 7(1)	Fig. 7(2)	Fig. 9(1)	Fig. 9(2)	Fig. 9(3)	Fig. 9(4)	Fig. 9(5)	Fig. 9(6)
Nie et al. [12]								
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. [11]								
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**

average time for image stitching in the testing dataset is 51 ms, which is significantly faster than the traditional method [11]. As for the stitching refinement, each iteration requires 35 ms, with an average of 50 iterations. We provide a comparison of the average running time for testing data from [8] in Table 4. For Nie *et al.*'s [12] method, the running time includes both the time spent in initial stitching using the learning-based method [9] and their rectangling process. In comparison, our learning-based method is proved to be more efficient.

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

4.5 Discussions

In this paper, we propose *RecStitchNet*, which combines rectangling and stitching in a unified learning based framework. It is natural to compare our method with the two-network cascade approach for stitching and rectangling. Acturally, in this paper, the results of Nie *et al.* [12] are produced by the cascaded stitching and rectangling. By comparison, our method is superior to the cascaded one, and the advantages are as follows: 1) the artifacts, such as feature misalignment, in the first stitching step cannot be solved in the following rectangling step, and might be amplified due to the latter warping; In addition, the stitching performance of different methods may also affect the rectangling effects. 2) the cascaded solution cannot ensure globally optimized results in terms of shape preserving, rectangular boundary preserving and feature alignment, while our method takes two normal images as input, and learns to perform stitching and rectangling in a unified framework. With the reasonable and effective network and the unsupervised refinement, our method can stably produce high-quality stitching results.

As a supervised learning framework, we use pseudo GT as labels for the training. The reason is that so far there is no recognized GT for learning based stitching with rectangular boundary, and it is true that the pseudo GT is theoretically the upper boundary of the learned model in this step. However, we

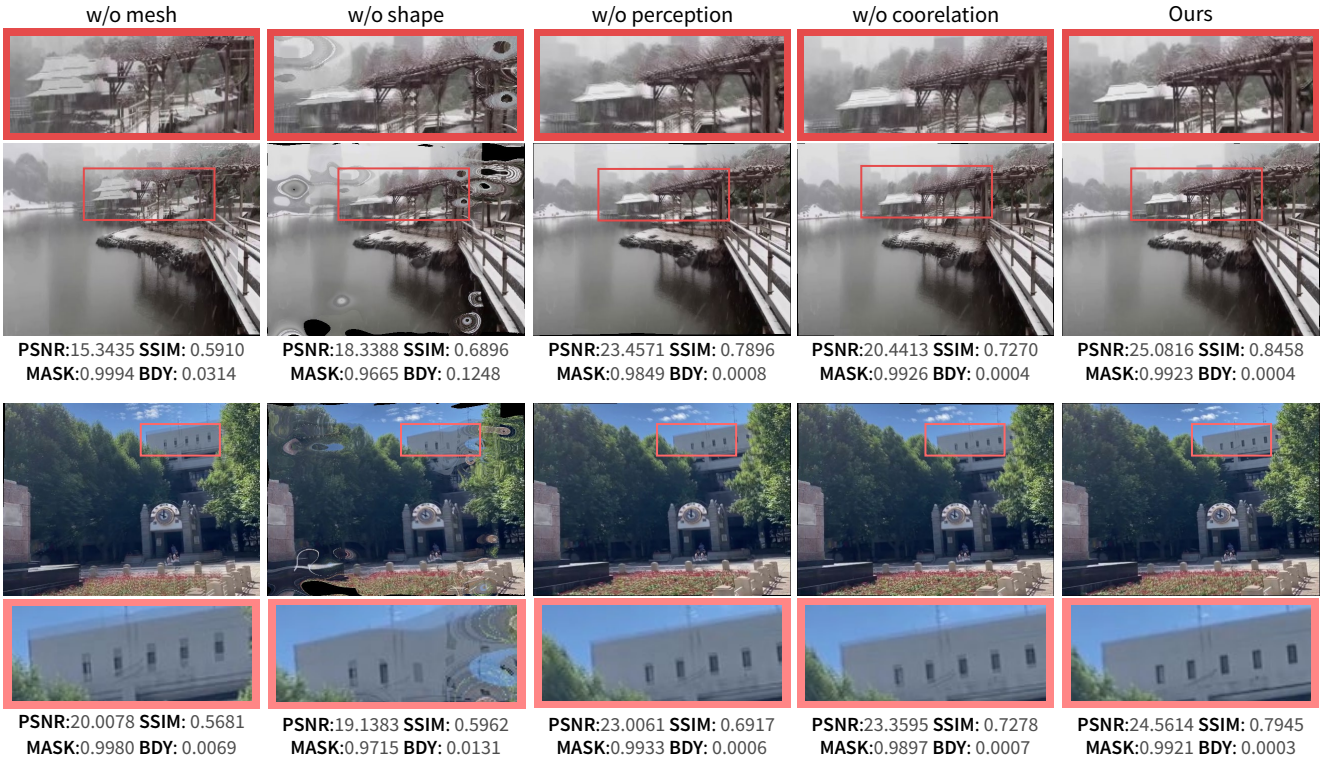


Fig. 8 Visual comparison of ablation study.

Table 2 Quantitative evaluations on the testing dataset in [8]. The upper part and the lower part give quantitative evaluations on dataset that [11] can successfully stitch and fail to stitch, respectively.

Metrics	Pseudo GT[11]	Nie[12]	Ours	Ours+Refinement
Evaluations on dataset that [11] *can successfully stitch*				
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**
Evaluations on dataset that [11] *fails to stitch*				
PSNR	-	24.7549	20.9781	**26.7898**
SSIM	-	0.8292	0.7281	**0.8772**
Mask	-	0.9909	0.9742	**0.9920**

have to point out that it would be very difficult for training our *RectStitchNet* without labels, and after training using pseudo GTs, we can obtain acceptable stitching results at a very small cost, with only a few artifacts regarding feature alignment and rectangular boundary preserving, which also exist in the pseudo-labels. To break the bottleneck of pseudo labels and further improve the 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 GT, as shown in the quantitative and qualitative evaluations in the paper.

5 Conclusions

This paper presents *RecStitchNet*, a novel learning-base framework for image stitching with regular boundaries. Compared with 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 preserve salient structures near the stitching boundaries (e.g. straight lines) 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 SOTA methods.

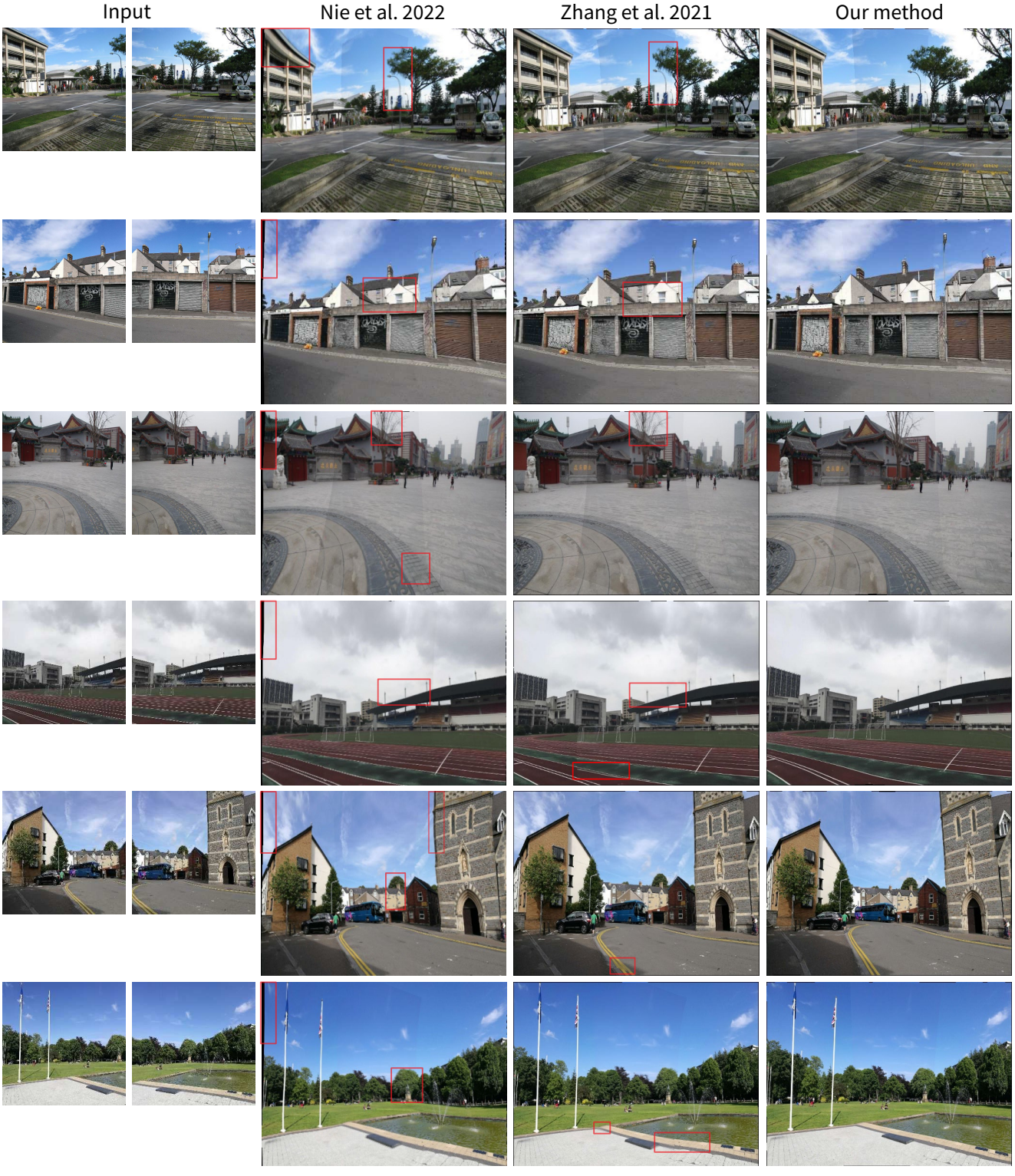


Fig. 9 More results and comparisons. The 'red' rectangles show the artifacts in feature alignment, structure preservation, and rectangular boundary preserving.

Table 3 Ablation Study on the testing dataset in [8].

Metrics	w/o shape	w/o perception	w/o mesh	w/o corr.	Ours
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	0.002	**0.0016**

Table 4 Comparison of average running time (Sec.).

Nie [12]	Zhang [11]	Ours	Ours+Refinement
0.256s	1.413s	**0.211s**	1.921s

In the future, we will 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), 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.

References

[1] Chen Y, Chuang Y. Natural Image Stitching with the Global Similarity Prior. In B Leibe, J Matas, N Sebe, M Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, Springer2016, 186–201.

[2] Gao J, Kim SJ, Brown MS. Constructing image panoramas using dual-homography warping. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, IEEE Computer Society2011, 49–56.

[3] Lin C, Pankanti S, Ramamurthy KN, Aravkin AY. Adaptive as-natural-as-possible image stitching. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, IEEE Computer Society2015, 1155–1163.

[4] Nie L, Lin C, Liao K, Liu M, Zhao Y. A view-free image stitching network based on global homography. *J. Vis. Commun. Image Represent.*, 2020, 73: 102950.

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

[6] Kweon H, Kim H, Kang Y, Yoon Y, Jeong W, Yoon KJ. Pixel-Wise Warping for Deep Image Stitching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, 37(1): 1196–1204, doi:10.1609/aaai.v37i1.25202.

[7] Song D, Lee G, Lee H, Um G, Cho D. Weakly-Supervised Stitching Network for Real-World Panoramic Image Generation. In S Avidan, GJ Brostow, M Cissé, GM Farinella, T Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XVI*, volume 13676 of *Lecture Notes in Computer Science*, Springer2022, 54–71.

[8] Nie L, Lin C, Liao K, Liu S, Zhao Y. Unsupervised Deep Image Stitching: Reconstructing Stitched Features to Images. *IEEE Trans. Image Process.*, 2021, 30: 6184–6197.

[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 (ICCV)*, 2023, 7399–7408.

[10] He K, Chang H, Sun J. Rectangling panoramic images via warping. *ACM Transactions on Graphics (TOG)*, 2013, 32(4): 1–10.

[11] Zhang Y, Lai Y, Zhang F. Content-Preserving Image Stitching With Piecewise gular Boundary Constraints. *IEEE Trans. Vis. Comput. Graph.*, 2021, 27(7): 3198–3212.

[12] Nie L, Lin C, Liao K, Liu S, Zhao Y. Deep Rectangling for Image Stitching: A Learning Baseline. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE2022, 5730–5738.

[13] Nie L, Lin C, Liao K, Liu S, Zhao Y. Deep Rotation Correction Without Angle Prior. *IEEE Trans. Image Process.*, 2023, 32: 2879–2888.

[14] Brown M, Lowe DG. Automatic Panoramic Image Stitching using Invariant Features. *Int. J. Comput. Vis.*, 2007, 74(1): 59–73.

[15] Lin W, Liu S, Matsushita Y, Ng T, Cheong LF. Smoothly varying affine stitching. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, IEEE Computer

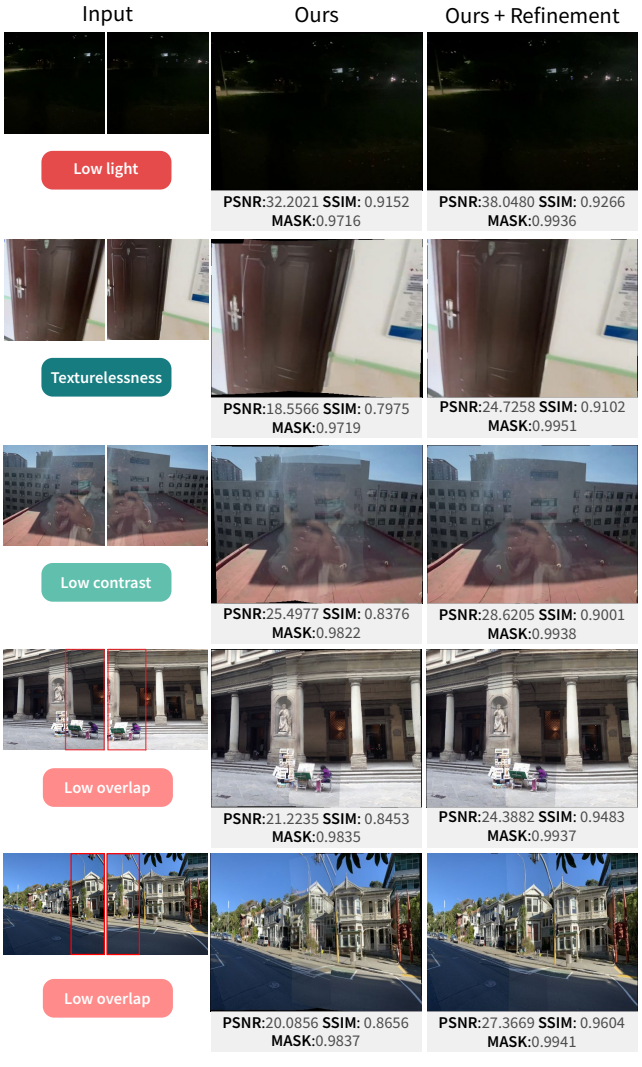


Fig. 10 Some challenging examples of the testing dataset in [8], such as low light, texturelessness, low contrast and low overlap.

Fig. 11 Limitations of our method. Our method may fail to preserve salient structures (e.g. straight lines) near the stitching boundaries.

Society2011, 345–352.

[16] Zaragoza J, Chin T, Tran Q, Brown MS, Suter D. As-Projective-As-Possible Image Stitching with Moving DLT. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, 36(7): 1285–1298.

[17] Lou Z, Gevers T. Image Alignment by Piecewise Planar Region Matching. *IEEE Trans. Multim.*, 2014, 16(7): 2052–2061.

[18] Lin K, Jiang N, Cheong L, Do MN, Lu J. SEAGULL: Seam-Guided Local Alignment for Parallax-Tolerant Image Stitching. In B Leibe, J Matas, N Sebe, M Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, Springer2016, 370–385.

[19] Zhang F, Liu F. Parallax-Tolerant Image Stitching. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, IEEE Computer Society2014, 3262–3269.

[20] Gao J, Li Y, Chin T, Brown MS. Seam-Driven Image Stitching. In MA Otaduy, O Sorkine, editors, *34th Annual Conference of the European Association for Computer Graphics, Eurographics 2013 - Short Papers, Girona, Spain, May 6-10, 2013*, Eurographics Association2013, 45–48.

[21] Chang C, Sato Y, Chuang Y. Shape-Preserving Half-Projective Warps for Image Stitching. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, IEEE Computer Society2014, 3254–3261.

[22] Li N, Xu Y, Wang C. Quasi-Homography Warps in Image Stitching. *IEEE Trans. Multim.*, 2018, 20(6): 1365–1375.

[23] Liao T, Li N. Single-Perspective Warps in Natural Image Stitching. *IEEE Trans. Image Process.*, 2020, 29: 724–735.

[24] Du P, Ning J, Cui J, Huang S, Wang X, Wang J. Geometric Structure Preserving Warp for Natural Image Stitching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE2022, 3678–3686.

[25] Jia Q, Li Z, Fan X, Zhao H, Teng S, Ye X, Latecki LJ. Leveraging Line-Point Consistence To Preserve Structures for Wide Parallax Image Stitching. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE2021, 12186–12195.

[26] Zhang L, Huang H. Image Stitching With Manifold Optimization. *IEEE Trans. Multim.*, 2023, 25: 3469–3482.

[27] Li N, Liao T, Wang C. Perception-based seam cutting for image stitching. *Signal Image Video Process.*, 2018, 12(5): 967–974.

[28] Lai W, Gallo O, Gu J, Sun D, Yang M, Kautz J. Video Stitching for Linear Camera Arrays. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, BMVA Press2019, 130.

[29] He K, Chang H, Sun J. gling panoramic images via warping. *ACM Trans. Graph.*, 2013, 32(4): 79:1–79:10.

[30] Wu J, Shi J, Zhang L. Rectangling irregular videos by optimal spatio-temporal warping. *Comput. Vis. Media*, 2022, 8(1): 93–103.

[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. *CoRR*, 2023, abs/2301.01661.

[32] Zhou H, Zhu Y, Lv X, Liu Q, Zhang S. Rectangular-Output Image Stitching. In *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, 2800–2804.

[33] Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K. Spatial Transformer Networks. In C Cortes, ND Lawrence, DD Lee, M Sugiyama, R Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, 2017–2025.

[34] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Y Bengio, Y LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

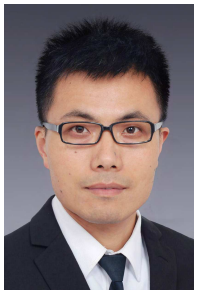
[35] Martínez F, Rueda AJ, Feito FR. A new algorithm for computing Boolean operations on polygons. *Computers & Geosciences*, 2009, 35(6): 1177–1185.

[36] Wang M, Shamir A, Yang G, Lin J, Lu S, Hu S. BiggerSelfie: Selfie Video Expansion With Hand-Held Camera. *IEEE Trans. Image Process.*, 2018, 27(12): 5854–5865.

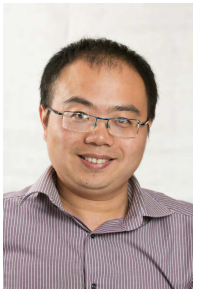
[37] Nie Y, Su T, Zhang Z, Sun H, Li G. Dynamic Video Stitching via Shakiness Removing. *IEEE Trans. Image Process.*, 2018, 27(1): 164–178.

[38] Wang M, Yang G, Lin J, Zhang S, Shamir A, Lu S, Hu S. Deep Online Video Stabilization With Multi-Grid Warping Transformation Learning. *IEEE Trans. Image Process.*, 2019, 28(5): 2283–2292.

[39] Rong J, Zhang L, Huang H, Zhang F. IMU-Assisted Online Video Background Identification. *IEEE Trans. Image Process.*, 2022, 31: 4336–4351.

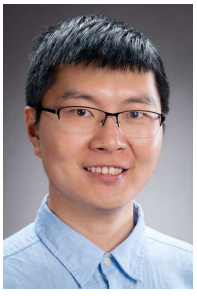


Yun Zhang is currently a Professor at the Communication University of Zhejiang in China. He received his doctoral degree from Zhejiang University in 2013. Before that, he received Bachelor and Master degrees from Hangzhou Dianzi University in 2006 and 2009, respectively. He visited the Visual Computing Group of Cardiff University in 2018 and 2023, and the Computational Media Innovation Centre of Victoria University of Wellington in 2019. His research interests include Computer Graphics, Image and Video Editing, Virtue Reality. He is a Senior Member of CCF.



Yu-Kun Lai received his bachelor's degree and PhD degree 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 the B.S degree in computer science and technology from Beijing Jiaotong University, Beijing, China, in 2019, and is currently pursuing the Ph.D. degree in signal and information processing from the Institute of Information Science, Beijing Jiaotong University, Beijing, China. His current research interests include image and video processing, 3-D vision, and multi-view geometry.



Fang-Lue Zhang received the Ph.D. degree from Tsinghua University, in 2015. He is currently a Senior Lecturer in Computer Graphics at the Victoria University of Wellington, Wellington, New Zealand. His research interests include image and video editing, computer vision, and computer graphics. He received the Victoria Early-Career Research Excellence Award, in 2019, and the Fast-Start Marsden Grant from the New Zealand Royal Society, in 2020. He is on the editorial board of Computer & Graphics. He is a member of ACM and a committee member of IEEE Central New Zealand Sector.

Lin Xu is currently pursuing her Ph.D. at the University of South Australia, with an anticipated graduation date in 2024. She received her bachelor's degree and master's degree, 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 Universite libre de Bruxelles in 2017. Her research interests include computer vision, multi-media system, and intelligence computing.