Computers & Graphics (2026)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Revisiting Unsupervised Image Stitching via Efficient Boundary Rectification

ARTICLE INFO

Article history:

Unsupervised image stitching, Boundary rectification, Bidirectional homography, Mesh motion regression, Fine-tuning

ABSTRACT

Unsupervised image stitching aims to align multi-view images with overlaps by learning from unlabeled datasets. Although great progress has been made to improve the robustness and generalization in recent years, most of them focus on the natural warping, large parallax, etc., and neglect the boundary regularity, which may undermine the wide-angle effects. To address the limitations above, we propose *BRecStitch*, which further incorporates the boundary rectification, achieving a good balance between content alignment and boundary regularity. Considering that both stitching and boundary rectification are non-trivial tasks, we propose a two-step unsupervised learning strategy. First, we design a novel unsupervised network that integrates a global bidirectional homography decomposition strategy to encourage balanced warping across views, and a local residual mesh motion regression to ensure precise alignment and boundary regularity. With the output mesh warpings in the first step, we design a fine-tuning approach to further optimize the stitching results by iterative alignment and rectangular boundary convergence. Extensive experiments and evaluations demonstrate the effectiveness of our method and the advantages over state-of-the-art methods.

1. Introduction

Image stitching, a fundamental and critical technique in computer graphics and vision, aims to combine multiple images with overlapping fields of view (FoV) into a single representation with a substantially larger FoV. It has widespread applications in virtual reality, remote sensing, autonomous driving, etc. Traditional image stitching methods [36, 17, 31] heavily rely on the quality of hand-crafted feature matching and often perform poorly in challenging scenes with large parallax, weak textures, or significant illumination variations. Although optimization methods based on global or local distortion alleviate these problems to some extent, they often suffer from high computational complexity and struggle to ensure both alignment accuracy and boundary regularity.

To address the challenges of traditional methods in image stitching, deep learning-based approaches have been widely studied by leveraging high-level semantic features from large datasets. Specifically, these methods are categorized by how the dataset is utilized. Among them, supervised methods are designed to learn the geometric mapping between input images and the labeled ground truth (GT). However, it is difficult to obtain valid stitching labels for supervised learning, and their generalization ability is difficult to guarantee, thus making it difficult to adapt to complex real-world scenarios. To alleviate the reliance on the labeled data and improve generalization ability, unsupervised methods have been proposed to autonomously explore the inherent patterns in large amounts of unlabeled data. A recent representative work is a parallax-tolerant unsupervised deep image stitching technique [23], which can estimate a robust and flexible warp for the target view to align with the reference view. Although effective and robust, single-view stitching methods still suffer from unnatural warping, which may produce extremely irregular stitching boundaries, making it far more difficult to regu-

larize these boundaries into rectangular shapes.

To obtain stitching results with rectangular boundaries, Zhang et al. [35] proposed *RecStitchNet*, which pioneers the combination of stitching and rectangling into a unified network. Although *RecStitchNet* performs well in many challenging scenes, the required large number of pseudo-labels generated by traditional methods introduce artifacts, significantly affecting the data preparation and training processes. In fact, *RecStitchNet* is not designed as an end-to-end network, and the mesh motion regression is based on the pretrained stitching model (Unidirectional Warping), which cannot ensure content alignment and regular boundaries. The model also depends heavily on pseudo-labels, which limits its effectiveness and generalization. In addition, the lack of a differentiable rectangling process that can be integrated into the network architecture hinders the learning from achieving the final fully rectangular stitching results. Evidence from the ablation study confirms that the mask comparison scheme proposed by Nie et al. [20] is ineffective. Although the PolyUnion strategy proposed in *RecStitchNet* [35] works well in unsupervised fine-tuning, the process of PolyUnion is not differentiable, which limits its use in model training.

To address the challenges mentioned above, and improve generalization while ensuring boundary regularity, we revisit unsupervised image stitching via efficient boundary rectification. Specifically, we propose an end-to-end unsupervised regression network, *BRecStitch*, which enhances the approach in [35] by combining bidirectional warping with a stitching boundary rectification strategy. The end-to-end unsupervised network consists of global homography regression and local mesh motion regression. In particular, the global homography aims to obtain initial stitching results by introducing bidirectional warping in overlapping views, while the local regression generates accurate mesh motions to ensure accurate alignment and boundary rectification. For effective boundary rectification, we creatively propose a differentiable boundary rectification loss function, which includes an effective outer boundary extraction and a boundary loss calculation strategy based on a flexible polygon structure. Considering that both stitching and rectangling are non-trivial tasks, we further designed a fine-tuning scheme based on the regressed mesh motion described above, to iteratively refine alignment in the overlapping regions, and encourage the stitching boundaries to converge toward rectangular shapes.

Our method is the first to successfully leverage an end-to-end unsupervised network to solve the stitching and rectangling problems. Extensive experiments and evaluations demonstrate the advantages of our method over state-of-the-art methods [35, 27]. In summary, the main contributions of this paper are as follows:

- We propose *BRecStitch*, the first unsupervised image stitching network, which can ensure good alignment and boundary regularity.
- We design an effective outer boundary extraction and boundary loss calculation strategy for effective rectangular boundary preservation.
- We propose a novel pairwise fine-tuning strategy to refine the feature alignment and boundary regularity efficiently.

2. Related Work

The main goal of image stitching is to combine multiple images with a limited field of view and overlapping areas into a panoramic image with a wide field of view and low distortion. In general, image stitching focuses on two important problems: correct alignment and structural fidelity.

2.1. Traditional image stitching

Early traditional methods centered on hand-crafted geometric features, mainly focused on key points and line segments, and achieved global or local alignment by minimizing projection errors. Lou et al. [17] achieved image alignment through piecewise planar region modeling combined with iterative optimization of energy functions. Brown et al. [3] took invariant features such as SIFT as the core and minimize projection errors to calculate homography matrices for panoramic stitching. Zaragoza et al. [31] dynamically adjusted image transformation parameters and locally minimized projection errors, ensuring alignment smoothness and detail preservation. Methods such as smoothly varying affine stitching [16], global similarity prior [4], Thin-Plate Splines (TPS) [13] and quasi-homography [14] , dual-homography [6] are all solutions to minimize perspective distortion. Although these earlier methods described above did improve alignment accuracy, they mainly relied on the quality of manually crafted features.

For structure and boundary preservation, traditional methods attempt to balance alignment accuracy and boundary regularity through energy functions. For example, Zhang et al. [32] achieved a balance between alignment accuracy and boundary regularity by constructing a multi-term energy function that includes alignment, regularization, scale, etc. Jia et al. [9] integrated global colinear structures into the energy function, combining point-line alignment and distortion terms to balance alignment accuracy and boundary regularity while reducing artifacts.

Although the aforementioned methods have achieved a certain degree of success, their dependence on manually extracted features limits their generalization ability. Consequently, the stitching performance degrades considerably in complex scenarios such as weakly textured or poorly illuminated regions, motivating the development of methods based on deep learning.

2.2. Image stitching based on deep learning

Compared with traditional methods, the advantage of deep learning technology lies in its reliance on data-driven feature learning. This approach not only addresses the challenges of feature extraction and matching but also enables optimization in terms of robustness, alignment accuracy, and the naturalness of results even in complex scenarios. Specifically, the deep learning-based methods can be classified into three categories: supervised, weakly supervised and unsupervised, which complement each other in terms of data dependence, generalization ability and practical value respectively.

2.2.1. Supervised Learning

The edge-guided composition network proposed by Dai et al. [5] adopted supervised learning, which applies deep semantic features to address the limitations of traditional hand-crafted features, thereby enhancing the edge integrity and stitching consistency. Following this, Nie et al. [26] proposed a multi-scale deep homography network that learns alignment relations through multi-scale supervision on a synthetic dataset containing true homography matrices. They also proposed edge-preserving constraints to prevent the distortion of edge structures.

Building upon these supervised approaches, Zhang et al. [35] proposed *RecStitchNet* under the supervised learning framework, which improved the problem of insufficient robustness of the previous method [36] in complex scenarios. This method relies on datasets containing pseudo-real labels and trains the network using a multi-constraint supervised loss to achieve the collaborative optimization of stitching and rectangling, thereby balancing the alignment accuracy and boundary regularity.

Although, straightforward and effective, supervised learning still has an inherent limitation, namely its high dependence on large-scale high-quality annotated datasets. Moreover, ground-truth annotation for image stitching tasks is very difficult, and there is currently no widely recognized method for generating labels.

2.2.2. Weakly Supervised and Unsupervised Learning

To overcome the challenges in dataset construction, label generation, and data annotation required in supervised learning, numerous weakly supervised and unsupervised learning approaches have been proposed.

Weakly supervised methods aim to reduce the reliance on fully annotated data while still using limited supervision to guide network learning. Song et al. [28] proposed a weakly supervised stitching network that avoids dependence on ground-truth wide field-of-view images, which effectively mitigates the limitations of supervised learning arising from its strong dependence on high-quality annotated data and exhibits greater advantages in adaptability to real-world scenarios.

Unsupervised learning, on the other hand, completely removes the need for manual data labeling, significantly reducing the cost and workload of data preparation. Jiang et al. [10] constrained unsupervised stitching with a global-aware loss, and upgraded to the global-aware quadrature pyramid regression architecture by [11] which does not require manual annotation of spectral alignment labels. Similarly, Nie et al. [23] proposed a parallax-tolerant unsupervised image stitching method trained under an unsupervised paradigm and leveraging the collaborative optimization of homography matrices and TPS. This method enables robust stitching in large-parallax scenarios even without any ground-truth supervision.

2.3. Image Rectangling and Rectification

Image rectangling and rectification, the key technology for addressing irregular boundary in stitching results, aims to correct irregular boundary into regular rectangle or approximate rectangle while retaining the effective content. In this field, He et al. [7] proposed a warping method based on mesh optimization to rectify panoramic images into rectangular ones, laying an important foundation for traditional image rectangling and rectification techniques.

Benefiting from the ability of the learning framework to extract high-level semantic features, deep image stitching methods exhibit better robustness and higher efficiency. Mei et al. [18] proposed a solution that achieves image stitching by estimating the homography matrix. Nie et al. [20] proposed a one-stage deep learning baseline model for rectangling constraints, which optimizes mesh accuracy using a residual progressive regression strategy and achieves relatively higher robustness in image stitching. They further address the rectangling of wide-angle images by Thin-Plate Spline model and DoF-based Curriculum learning [15]. Inspired by [20], Zhou et al. [37] further proposed RecDiffusion, which firstly applies the diffusion-based learning framework to rectangling, and Zhang et al.[35] proposed a unified supervised learning framework that combines image stitching and rectangling. Compared with the mesh regression-based rectangling [20], it achieves superior performance and establishes a new state-of-the-art (SOTA). Although effective and convincing, the diffusion-based method is computationally intensive, and the generalization can not be ensured due to the reliance on labels.

For smarter and more robust rectification, Nie et al. [22] proposed a neural network to predict optical flows that can help warp the tilted images without angle priors. More recently, Nie et al. [24] proposed a Semi-Supervised Coupled TPS Model for Rotation Correction, rectangling, etc. To achieve a more flexible and powerful transformation, they proposed an iterative search scheme to predict new control points according to the current latent condition.

3. BRecStitch

As shown in Fig. 1, *BRecStitch* is designed as an end-to-end regression network. It takes a reference image and a target image as input, and the output corresponds to

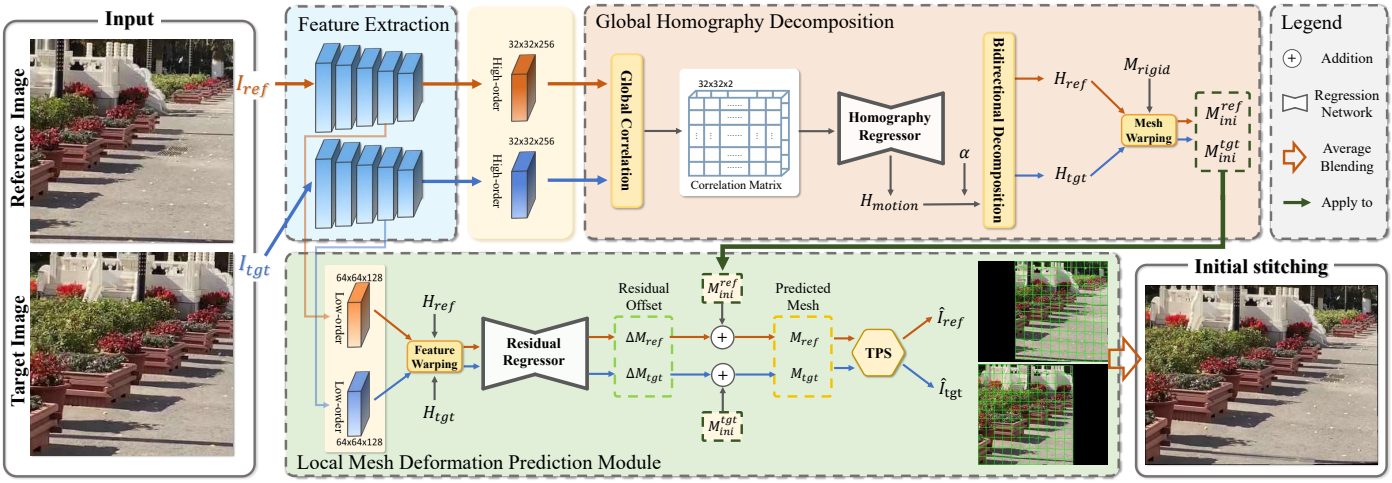

Fig. 1: The overall architecture of our network, which takes a reference image and a target image as inputs. The output consists of the predicted meshes of the input, which are further used to guide the warping of the input images.

the predicted meshes for the input images, which are further used to warp the input image for stitching with the target image using the TPS transformation. Specifically, we adopt a bidirectional decomposition strategy and design a novel loss function to encourage content alignment in the overlapping regions and ensure well-shaped rectangular boundaries in the stitching result. To ensure the effectiveness and robustness of the approach, we further develop a fine-tuning scheme to iteratively refine the stitching result. We detail our algorithm in this section.

3.1. Initial Stitching Stage

The unidirectional warping employed in the stitching process may induce excessive distortion in certain regions of the input image. This issue should be addressed during the unsupervised initial stitching stage to ensure boundary regularity and accurate alignment of overlapping areas. Drawing on typical distortion patterns from global to local in UDIS++ [23] and the bidirectional decomposition in StabStitch++ [25], the global homography decomposition is applied in this stage for overall alignment. Meanwhile, the local mesh deformation is employed to tackle parallax issues. Additionally, a dedicated boundary constraint term is incorporated into the loss function to produce more regular stitching boundaries.

As shown in Fig. 1, we take a pair of partially overlapping images as input (I_{ref}, I_{tgt}), and we obtain the warped meshes for both views as output after the Feature Extraction, Global Homography Decomposition, and Local Mesh Deformation Prediction steps.

Feature Extraction. To extract discriminative and generalizable semantic features that are critical for unsupervised alignment, we adopt a Siamese feature extractor based on the pretrained ResNet-18 [8] , a backbone network with four stages that gradually extracts features evolving from structural to semantic representations. Specifically, we employ the intermediate layers (Stages 2 and 3) of ResNet-18 [8] for feature extraction, where Stage 3 is used to extract semantic features for global homography estimation, and Stage 2 is used to extract structural features for local alignment.

Global Homography Decomposition. Inspired by [25], we employ a bidirectional decomposition module for more flexible warping in image stitching. We first calculate the contextual correlation [21] between two high-order feature maps to quantify the pixel-level feature similarity. With this correlation matrix (Corr) as input, our homography regression network GLNet($\cdot$), which is composed of a convolutional feature encoder and a fully connected parameter predictor, regresses the displacements of the four corner points for each input image H_{motion} = GLNet(Corr). After that, global homography H can be obtained by combining the four corners of input image S and their displacements H_{motion} through the Direct Linear Transformation (DLT) [2], using $H = DLT(S, S + H_{motion})$.

We further construct a virtual intermediate plane using bidirectional decomposition to balance the warping between the two views. Similar to [25], we obtain the displacement of I_{tgt} towards the intermediate plane $H_{motion}^{\text{tgt}} = \alpha \cdot H_{motion}$ through a decomposition coefficient $\alpha \in [0, 1]$. Then, the homography matrix H_{tgt} can be obtained through the same DLT, and $H_{\text{ref}} = H^{-1} \cdot H_{\text{tgt}}$.

Local Mesh Deformation Prediction. Based on the global homography, we further introduce a local mesh deformation prediction sub-network, which aims to handle large parallax and achieve precise local alignment while ensuring global geometric consistency.

First, we place a rigid grid M_{rigid} on each input image, which is used to control the warping of input images. Then, we warp the rigid grid onto the virtual intermediate plane using the bidirectional homography matrices H_{ref} and H_{tgt} to generate the initial meshes $M_{\text{ini}}^{\text{ref}}$ and $M_{\text{ini}}^{\text{tgt}}$.

To ensure correct alignment and structure preservation, we further design a local mesh deformation network to estimate the residual mesh offsets relative to the initial

Fig. 2: Illustration of the boundary loss. We take the predicted meshes of the reference image and the target image as input, and then compute the boundary constraint by calculating the sum of the minimum distances from the outer boundary points of the stitched image to the minimum bounding rectangle (marked in green). As the legend shows, the red and blue meshes represent the predicted meshes of the reference and target images, respectively, and the red and blue points are the extracted vertices from those predicted meshes.

mesh. To obtain the input of the network, we first warp the low-order feature maps ($F_{\text{ref}}^{(128)}$, $F_{\text{tgt}}^{(128)}$) according to the global homography matrices H_{ref} and H_{tgt}. The final warped meshes are then obtained by adding the residual offsets to the corresponding initial meshes, as defined below.

$$\begin{cases} M_{\text{ref}} = M_{\text{ini}}^{\text{ref}} + \text{LocNet}(\mathcal{H}(F_{ref}^{(128)}, H_{ref})) \\ M_{\text{tgt}} = M_{\text{ini}}^{\text{tgt}} + \text{LocNet}(\mathcal{H}(F_{tgt}^{(128)}, H_{tgt})), \end{cases} \tag{1}$$

where LocNet$(\cdot)$ refers to the output of the local mesh regression network, and $\mathcal{H}(\cdot,\cdot)$ represents the warping of the feature map based on the homography transformation matrix.

At last, based on the final meshes M_{ref} and M_{tgt}, we obtain the warped results for both views using the TPS transformation and blend these warped results to produce the final stitching result, denoted as Γ. The definitions are given below.

$$\Gamma = \mathcal{B}(\mathcal{W}(I_{\text{ref}}, M_{\text{ref}}), \mathcal{W}(I_{\text{tgt}}, M_{\text{tgt}})), \tag{2}$$

where $\mathcal{W}(\cdot,\cdot)$ refers to image warping based on the TPS transformation, and $\mathcal{B}(\cdot,\cdot)$ is used to combine the results of both warped views. We prefer to use the *"Average Blending"* scheme, which achieves a balance between efficiency and visual quality.

3.2. Boundary Rectification

We propose a novel boundary rectification approach. However, it is non-trivial to define an effective and efficient boundary rectification constraint, due to the complexity of mesh overlap across different views.

In [20], the boundary constraint is simply defined as the difference between the stitched mask and the all-one mask as in Eq. 3. In our experiments, we find that incorporating this constraint into our network has no significant effect on the boundaries while incurring a substantially higher computational cost.

$$\mathcal{L}'_{\text{bdy}} = \|E - \mathcal{W}(m, M_{ref}) \cup \mathcal{W}(m, M_{tgt})\|_1, \tag{3}$$

Here, E denotes an all-one matrix with the same dimension as the stitched mask, while m represents the initial mask of the corresponding input image. However, the ablation study presented in [20] and Table 1 demonstrates its deficiency in preserving rectangular boundaries.

Furthermore, Zhang et al. [35] proposed a novel boundary constraint term during the fine-tuning step. The main idea of their method is to extract the outer boundary points of the two overlapping meshes using the polygon Boolean union operation [1]. With the extracted outer boundary, the loss term can be directly defined as the sum of the differences between all vertices and their target positions, where two attributes (constraint direction and target value) are assigned to each vertex. Although [35] effectively rectified irregular boundaries, the Boolean Union operation used for outer boundary extraction is computationally expensive and non-differentiable, thereby limiting its applicability during model training. Additionally, this method is less efficient than ours, as shown in the runtime comparison in Table 4.

To address the aforementioned efficiency and training issues, we propose a novel solution for more effective and efficient boundary rectification, characterized by efficient and differentiable outer boundary extraction. As illustrated in Fig. 2, taking the predicted meshes M_{ref} and M_{tgt} as in-

put, we first normalize their coordinates, and then extract the boundary points of the two meshes. As shown in the 2^{nd} row of Fig. 2, the red and blue points denote the extracted boundary points. The next step is to identify the outer boundary points of the stitched images. The main idea is to collect the boundary points of each mesh that are not enclosed by the polygons of the other mesh, and these points are defined as outer boundary points. Let $\mathcal{O}_{\text{ref}}$ and $\mathcal{O}_{\text{tgt}}$ denote the outer boundary points of each mesh. The outer boundary of the stitched image is defined as $\mathcal{O} = \mathcal{O}_{\text{ref}} \cup \mathcal{O}_{\text{tgt}}$. The definitions are provided below.

$$\begin{cases} \mathcal{O}_{\text{ref}} = \{u \mid u \in \partial M_{\text{ref}} \ \cap \ u \notin \text{int}(M_{\text{tgt}})\} \\ \mathcal{O}_{\text{tgt}} = \{v \mid v \in \partial M_{\text{tgt}} \ \cap \ v \notin \text{int}(M_{\text{ref}})\}, \end{cases} \tag{4}$$

Here, ∂M refers to the boundary points of mesh M, while $\text{int}(M)$ denotes the region enclosed by the boundary. The vertices u and v are located on the boundaries of M_{ref} and M_{tgt}, respectively.

We describe the outer boundary extraction process in Alg. 1, which includes extraction from the reference and target images, and we take the reference image part as an example. The double loop in Alg. 1 represents the Cartesian product of "point × edge". For each point $u \in \partial M_{\text{ref}}$, we check whether it intersects each edge formed by vertices $v \in \partial M_{\text{tgt}}$, as illustrated in Fig. 3b. This can be achieved by casting a horizontal ray to the right from each point u, which essentially identifies intersections between the line $y = u^y$ and polygon edges lying to the right of u. Next, we count the number (n_{inter}) of intersections and use the odd-even rule to verify whether u is inside the polygon, and retain the points where n_{inter} is even to obtain $\mathcal{O}_{\text{ref}}$. Finally, the outer boundary can be represented by $\mathcal{O} = \mathcal{O}_{\text{ref}} \cup \mathcal{O}_{\text{tgt}}$. As illustrated in Fig. 3a, we assume that there are two meshes, Mesh A and Mesh B, with points P_{i} and P_{o} both belonging to Mesh A. From P_{i} and P_{o}, we cast horizontal rays to the right and count the number of intersections (n_{inter}) between these rays and the boundary of Mesh B. If n_{inter} is even, it indicates that the point is outside the polygon (as shown in P_{o} in Fig. 3a).

To clarify the algorithm, we further provide a detailed analysis of the outer boundary determination. We first discuss the polygonal approximation of our mesh. In our warping-based learning framework, we apply rigid meshes to the reference and target images, and only update vertex coordinates through mesh motion regression. Thus, the mesh boundary can be approximated as a polygon instead of a curved contour, which facilitates the aforementioned ray-intersection-based outer boundary extraction. As shown in Alg. 1, a boundary vertex of a mesh is identified as a point on the global outer boundary by verifying the number of intersections between a horizontal ray cast from the vertex and the polygon boundary of the other mesh. For the special cases in Fig. 3c, when the horizontal ray intersects a vertex of the other mesh, the number of intersections can also be correctly determined using the original logic of Alg. 1. In addition, when the ray cast from a vertex is collinear with an edge of the other mesh, we directly skip this vertex and do not count it as an intersection.

We further analyze the differentiability of Alg. 1. It is evident that most operations involved are differentiable. In particular, although the "modulo" operation appears non-differentiable during boundary extraction, it does not affect the differentiability of the overall algorithm. In our opinion, the modulo operation is equivalent to the "**if**" conditional statement. In the forward propagation step, only the operations in the selected branch appear in the computational graph. In the backpropagation step, the network only computes derivatives for the operations along the actual path traversed during forward propagation, and propagates the gradients back along this path. Indeed, operations such as the O_{ref} update support gradient propagation.

Given the outer boundary vertices of warped meshes after stitching, we compute their maximum and minimum coordinates to obtain the minimal bounding rectangle. We then constrain each vertex on the outer boundary to be close to the nearest side of the smallest bounding rectangle.

Algorithm 1 Outer boundary extraction algorithm

Input: ∂M_{ref}, ∂M_{tgt} ▷ The boundary points of the two meshes
Output: $\mathcal{O}_{\text{ref}}$ ▷ The extracted outer boundary from ∂M_{ref}
$\mathcal{O}_{\text{ref}} \leftarrow \emptyset$
$\mathcal{R}[\,] \leftarrow \partial M_{\text{ref}}$, $\mathcal{T}[\,] \leftarrow \partial M_{\text{tgt}}$
for each u in $\mathcal{R}$ do
  $n_{\text{inter}} \leftarrow 0$ ▷ Initialize the number of intersection points
  for k from 1 to n do
    $v \leftarrow \mathcal{T}[k]$ ▷ Start point of the edge
    $v_{next} \leftarrow \mathcal{T}[(k+1) \bmod n]$ ▷ Next point of the edge
    if $v^y = v^y_{next}$ then
      continue
    end if
    if $u^y > \min(v^y, v^y_{next})$ and $u^y \leq \max(v^y, v^y_{next})$ then
      Let $\delta = \overline{vv_{\text{next}}} \cap (y = u^y)$
      if $u^x \leq \max(v^x, v^x_{next})$ and $u^x \leq \delta^x$ then
        $n_{\text{inter}} \leftarrow n_{\text{inter}} + 1$ ▷ Intersections to the right of u
      end if
    end if
  end for
  if $(n_{\text{inter}} \bmod 2) == 0$ then
    $\mathcal{O}_{\text{ref}} \leftarrow \mathcal{O}_{\text{ref}} \cup \{u\}$
  end if
end for
return $\mathcal{O}_{\text{ref}}$

3.3. Loss Function

The loss function of the local deformation network contains three constraint terms: alignment loss, shape preservation loss, and rectangular boundary loss. We define the total loss as Eq. (5):

$$\mathcal{L}_{\text{total}} = \varphi_a \mathcal{L}_{\text{align}} + \varphi_s \mathcal{L}_{\text{shape}} + \varphi_b \mathcal{L}_{\text{bdy}}, \tag{5}$$

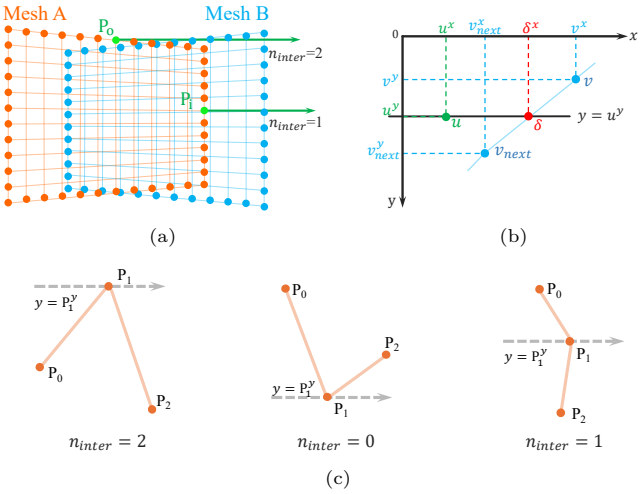

Fig. 3: Outer boundary extraction. (a) Shows how to determine the internal points of the polygon; (b) Provides details for judging the intersection point in Alg. 1; (c) Describes the special cases of the outer boundary extraction.

where φ_a, φ_s, and φ_b denote the corresponding weights.

Alignment Loss. To ensure accurate feature alignment in the overlapping regions, we propose a coarse-to-fine scheme to define the alignment loss. Specifically, this constraint is defined as a combination of the alignment losses from global and local warping, as defined below.

$$\mathcal{L}_{\text{align}} = \varphi_h \mathcal{L}_{\text{h}} + \varphi_t \mathcal{L}_{\text{t}}, \tag{6}$$

where φ_h and φ_t are corresponding weights.

We define the global alignment loss based on the pixel-level differences between overlapping regions of warped images guided by the global bidirectional homography [23], as defined below.

$$\mathcal{L}_{\text{h}} = \|\mathcal{H}(I_{ref}, H_{ref}) \cdot \mathcal{P}_{\text{h}} - \mathcal{H}(I_{tgt}, H_{tgt}) \cdot \mathcal{P}_{\text{h}}\|_1, \tag{7}$$

where $\mathcal{P}_{\text{h}} = \mathcal{H}(m, H_{ref}) \cap \mathcal{H}(m, H_{tgt})$ is used to extract the overlapping regions from the warped images.

Similar to the global loss, we define the local alignment loss based on the pixel-level differences between the overlapping regions of warped images guided by the predicted local deformation mesh, as detailed below.

$$\mathcal{L}_{\text{t}} = \|\mathcal{W}(I_{ref}, M_{ref}) \cdot \mathcal{P}_{\text{t}} - \mathcal{W}(I_{tgt}, M_{tgt}) \cdot \mathcal{P}_{\text{t}}\|_1, \tag{8}$$

where $\mathcal{P}_{\text{t}} = \mathcal{W}(m, M_{ref}) \cap \mathcal{W}(m, M_{tgt})$ denotes the mask corresponding to the overlapping area.

Shape Preservation Loss. To prevent excessive deformation in non-overlapping areas, the shape preservation loss imposes intra-grid and inter-grid constraints on the final mesh, as defined below.

$$\mathcal{L}_{\text{shape}} = \mathcal{L}_{\text{intra}} + \mathcal{L}_{\text{inter}}. \tag{9}$$

Similar to [20], the intra-grid constraint prevents excessive stretching by limiting the maximum size of each mesh cell, and enforces that the edge length does not exceed twice the original length of the rigid grid edge. The inter-grid constraint is formulated to preserve the local smoothness of the grid structure. This is achieved by calculating the cosine of the angles formed by horizontally and vertically adjacent edges within the grid. The angle error derived from these calculations serves as the inconsistency metric. Please refer to [20] for details.

Rectangular Boundary Loss. With the outer boundary $\mathcal{O}$ and the minimal boundary rectangle defined in Section 3.2, we define the rectangular boundary loss as follows. For each outer boundary point $(x_k, y_k) \in \mathcal{O}$, we compute the minimum distance from the point to the four edges of the rectangle. The sum of these minimum distances is taken as the boundary loss, as defined below.

$$\begin{aligned}\mathcal{L}_{\text{bdy}} = \sum_{k=1}^{\mathcal{N}} \min\big(&|x_k - \mathcal{O}^x_{\min}|,\ |x_k - \mathcal{O}^x_{\max}|,\\ &|y_k - \mathcal{O}^y_{\min}|,\ |y_k - \mathcal{O}^y_{\max}|\big),\end{aligned} \tag{10}$$

where $\mathcal{O}^x_{\max}$, $\mathcal{O}^x_{\min}$, $\mathcal{O}^y_{\max}$ and $\mathcal{O}^y_{\min}$ refer to the four boundary coordinates of the bounding rectangle, and $\mathcal{N}$ is the total number of vertices on the outer boundary.

3.4. Fine-tuning

To further improve the alignment accuracy and boundary regularity of the overlapping regions, we conduct a pairwise fine-tuning process. The results are presented in Fig. 4. In the fine-tuning stage, we design a weighted alignment loss to ensure that the features of the two images are well aligned within the overlapping area. Specifically, it is formulated by assigning weight ω to Eq. 8, where ω denotes the overlap proportion weight, as given below.

$$\omega = \lambda \cdot \max\left(0.1, \frac{|\Phi|}{H \times W}\right). \tag{11}$$

Here, $|\Phi|$ represents the total number of pixels in the overlapping region, and λ is a scaling factor that amplifies the influence of the overlap weight.

The significance of the weight design is that when the overlapping region is very small, a minimum weight of 0.1 is assigned to avoid training instability caused by an excessively small weight. When the overlapping region is relatively large, the weight increases linearly with the overlap ratio. This approach reinforces the alignment constraint and facilitates the adaptation to image pairs with varying degrees of overlap.

In this stage, the total loss function is a linear combination of weighted alignment loss and rectangular boundary (see Section 3.3), as defined below.

$$\mathcal{L}_{\text{ft}} = \varphi_o \omega \mathcal{L}_{\text{t}} + \varphi_b \mathcal{L}_{\text{bdy}}, \tag{12}$$

where φ_o and φ_b are the corresponding weights that control their relative importance.

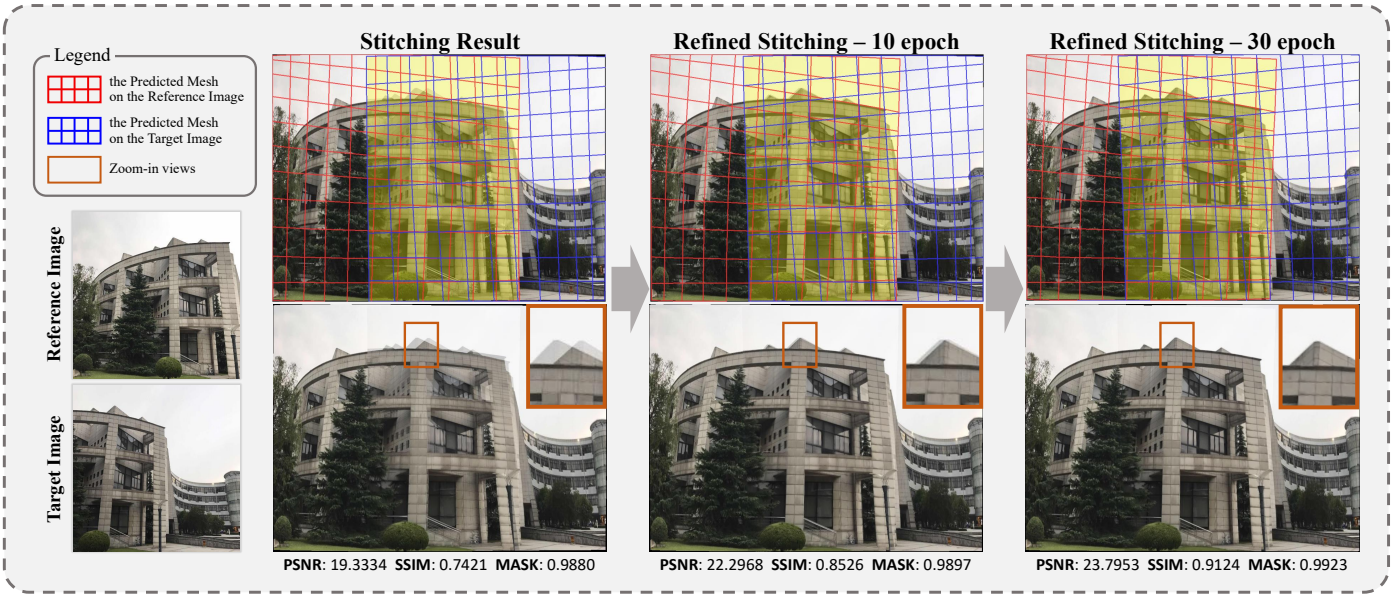

Fig. 4: Pairwise fine-tuning process. Given a pair of images as input, the image stitching parameters are iteratively optimized via a pretrained neural network using weighted alignment loss and rectangular boundary loss, yielding the refined stitching result. Columns 2 to 4 present the stitching result of *BRecStitch*, as well as the results after 10 and 30 epochs, respectively. The overlapping regions are highlighted in yellow for clearer and more intuitive illustration.

4. Experiments

4.1. Implementation Detail

Our implementation was based on PyTorch using a single NVIDIA A40 GPU for training and inference. In the data preparation and training stages, we uniformly set the resolutions of the input image and the mesh to 512×512 and 11×11, respectively. Our network was trained for 260 k iterations using the Adam optimizer [12] with an exponentially decaying learning rate initialized to 10^{-4}. We set the batch size to 4 and used ReLU as the activation function.

For the decomposition coefficient, we conducted comparative experiments using the values (0.3, 0.4, and 0.5) respectively. Finally, we set $\alpha = 0.4$ to achieve improved boundary regularity. In the training stage, we set $\varphi_a = 1$, $\varphi_s = 1$, $\varphi_b = 0.0001$, $\varphi_h = 3$, and $\varphi_t = 1$. In the fine-tuning step, we set $\lambda = 10$, $\varphi_o = 3$, and $\varphi_b = 0.001$ to achieve effective boundary rectification. We set the maximum number of iterations to 50 for each example, and most examples converged rapidly, with an average of 30 iterations.

We further analyze the setting of the loss weights in detail. In our experiments, the alignment loss, shape preservation loss, and boundary loss at the beginning of training are 0.8022, 0.0001, and 80.3288, respectively. Notably, the shape preservation loss is initially very small because the mesh undergoes negligible deformation at the beginning of training, and it gradually increases throughout training. Since alignment and shape preservation constitute the core objectives of high-quality stitching, we set their weights to 1 to ensure both constraints are sufficiently enforced. In contrast, the boundary loss is orders of magnitude larger, so a comparable weight would dominate the total loss and severely degrade the alignment and shape preservation. Thus, we set the weight of boundary loss to 0.0001 so that it serves as a weak regularizer, balancing the three terms and stabilizing training. At the fine-tuning stage, the total loss is formulated as a linear combination of the local alignment loss and the boundary loss. We introduce a scaling factor to further enhance alignment within overlapping regions. Accordingly, we increase the weight of the boundary loss to 0.001 for better regularization of the global stitching boundary.

We use two datasets to evaluate our method: the UDIS-D dataset [19] and an out-of-domain dataset. The former contains 10,440 training image pairs and 1,106 test image pairs, whereas the latter comprises 143 image pairs in total, including 119 captured by us and 24 selected from traditional datasets. This dataset is designed to evaluate the generalization capability of our model across diverse scenes, including challenging cases such as low-texture (9.09%), low-contrast (4.20%), low-light (7.69%), and low-overlap (9.79%).

4.2. Evaluation

To evaluate the effectiveness of our method, we conducted extensive quantitative and qualitative evaluations, as well as a user study.

4.2.1. Quantitative and Qualitative Comparison

We first compare our method with RecStitchNet[35], a state-of-the-art learning-based method for image stitching and rectangling, which is highly relevant to our work. For a more comprehensive evaluation, we combine state-of-the-art learning-based stitching methods[25] with rectangling methods [20] to generate stitching results with rectangular

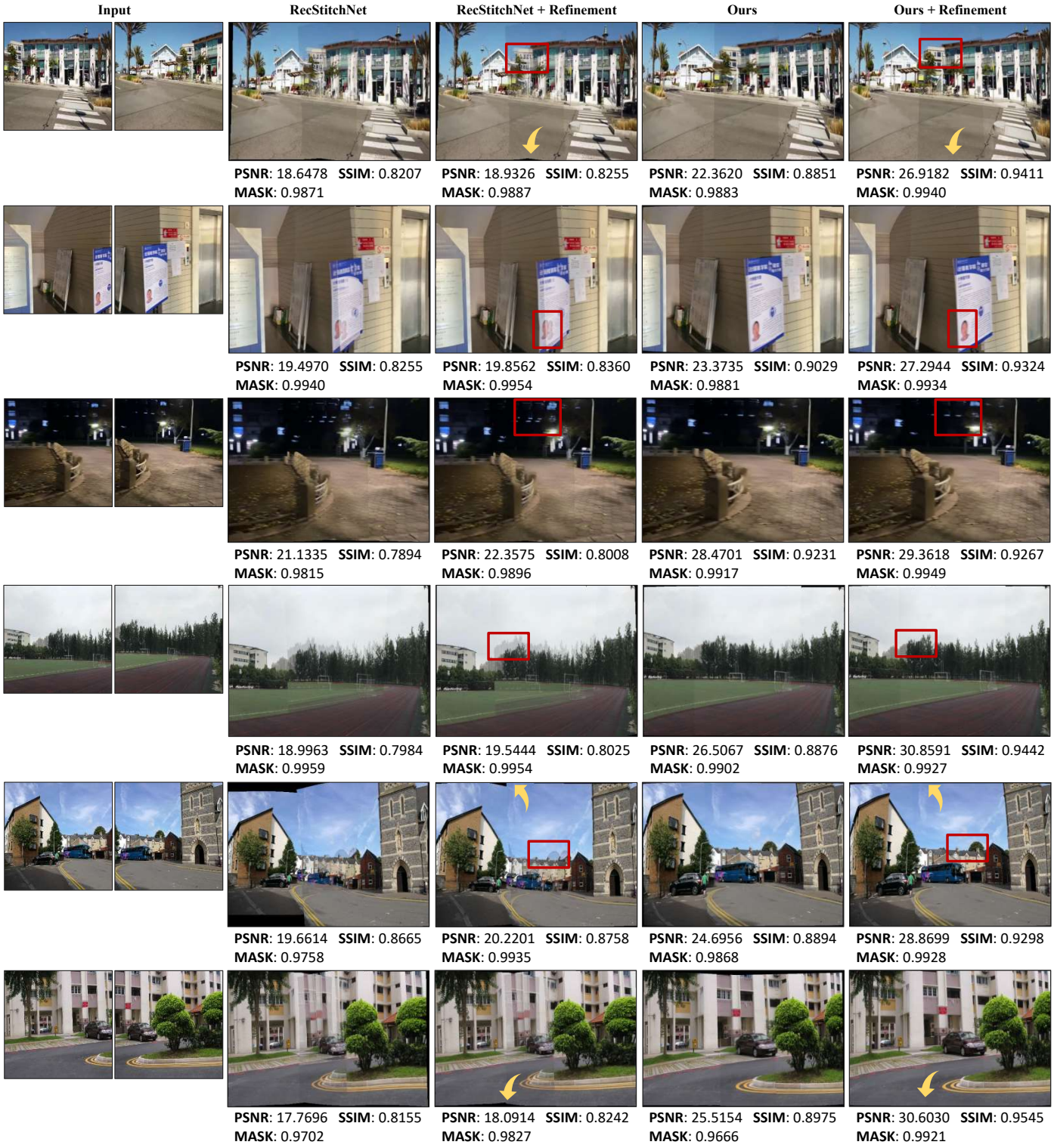

Fig. 5: Qualitative and quantitative comparison with RecStitchNet [35]. The inputs of the first three lines are from the UDIS-D dataset [19] and the remaining inputs are from our collected dataset. The first column presents the input images to be stitched. The 2^{nd} and 3^{rd} columns provide the stitching results of RecStitchNet [35] and its results after fine-tuning. The last two columns show the results of our method and its fine-tuning results.

Fig. 6: Comparison with StabStitch++ [25], Rectangling [20] and traditional Rectangular Stitching [36]. The first column shows the input images for stitching. The 2^{nd} and 3^{rd} columns present the results of StabStitch++ [25] and their results after rectangling [20], respectively. The 4^{th} column shows the results of the traditional rectangular stitching method [36]. The last two columns present the results of our method and our refined results. The red boxes and yellow arrows highlight artifacts produced by other methods.

Table 1: Ablation study and quantitative evaluation on the UDIS-D *testing* dataset [19]. The metrics used in the ablation study include PSNR, SSIM, and MASK, where "Mask" denotes the ratio of valid pixels within the bounding box. Columns 2–5 report the metrics of the stitching results without alignment, shape preservation, boundary constraints, and bidirectional decomposition, respectively; Columns 6–7 present the metrics of our method before and after fine-tuning; Columns 8 and 9 further report the metrics of [35] before and after fine-tuning; The last column presents the metrics of the stitching results when the boundary loss is replaced by the one proposed in [20].

	w/o Align.	w/o Shape	w/o Bdy.	w/o BiDir.	Ours	Ours+FT	[35]	[35]+FT	$\mathcal{L}'_{\text{bdy}}$
PSNR	13.0821	23.9570	27.1685	24.4752	25.0359	25.9178	21.3263	24.3630	26.6716
SSIM	0.3221	0.8130	0.8844	0.8205	0.8347	0.8603	0.7015	0.8094	0.8745
MASK	0.9940	0.9881	0.9349	0.9871	0.9911	0.9935	0.9858	0.9874	0.9382

boundaries. Specifically, we first employ the bidirectional decomposition-based stitching method [25] to obtain the initial stitching result and its corresponding mask, and then apply the rectangling method in [20] to rectify the irregular boundary. To evaluate the stitching quality in overlapping regions, we adopt PSNR and SSIM as evaluation metrics. To evaluate boundary regularity, we design the "MASK" metric to measure the closeness of the warped mask to the all-one matrix, where the warped mask denotes the corresponding 0-1 mask of the stitching result. We also present stitching results after fine-tuning and compare them with the fine-tuning results of RecStitchNet [35] over 30 iterations.

Fig. 5 presents extensive quantitative and qualitative results of our method, together with comparisons with RecStitchNet [35]. The first three inputs are from the UDIS-D dataset [19], and the last three examples take inputs from a self-built dataset. For a fair comparison, we compare our results with those of RecStitchNet [35] before and after fine-tuning. Both quantitative metrics and qualitative results confirm the advantages of our method in terms of alignment and boundary regularity before fine-tuning. We further refine the stitching results through a pairwise-image fine-tuning process. Experimental results show that although fine-tuning improves stitching quality for both our method and RecStitchNet [35], a comparison between their fine-tuned outputs still clearly demonstrates the superiority of our approach.

As shown in Fig. 6, we further compare with the SOTA works [25, 20] and the traditional rectangular stitching method [36]. By applying the bidirectional warping technique, StabStitch++[25] achieves stitching results with more balanced warping across both views, and the stitching boundary is rectified through the rectangling operation described in [20]. Although well-aligned thanks to the frozen initial stitching result, the boundary is not correctly

Table 2: Comprehensive quantitative evaluation on the UDIS-D dataset [19], as well as comparison with the state-of-the-art method [35]. To evaluate alignment in the overlapping regions, we use FSIM [33] to quantify alignment accuracy, RMSE to compute the root mean square of pixel differences, and LPIPS [34] to measure perceptual visual similarity. We further employ LBE to evaluate geometric fidelity based on LSD [29], and use GMSD [30] to measure the consistency of edge structures and fusion smoothness.

	FSIM↑	RMSE↓	LBE↓	LPIPS↓	GMSD↓
Ours	0.9175	19.7144	0.8796	0.1499	0.1259
Zhang[35]	0.8652	30.9369	0.8804	0.1550	0.1697

and robustly rectified because of the limitation of the single mesh regression scheme. The 4^{th} column shows results of the traditional rectangular stitching method [36]. Although [36] achieves comparable performance on common evaluation metrics, including PSNR, SSIM, and MASK, its performance in structure preservation and alignment is less stable, as shown in the regions marked with red boxes. In addition, different from deep learning frameworks, this method usually requires solving a computationally expensive optimization problem. Furthermore, it may fail to obtain correct stitching results under conditions such as "Low Texture" and "Low Light" scenes, as illustrated in Fig. 4.2.1. Both the quantitative and qualitative comparisons indicate that our method achieves a favorable balance between alignment and boundary regularity. Results in the last two columns demonstrate that our results are visually pleasing and acceptable even without a fine-tuning scheme.

To evaluate our method more comprehensively, we adopt a richer set of evaluation metrics, including FSIM [33], RMSE, and LPIPS [34], to assess alignment in overlapping regions across the frequency, spatial, and visual perception domains, as well as LBE and GMSD [30] to quantify geometric fidelity and structural consistency. The results in Table 2 clearly indicate that our method outperforms the most closely related state-of-the-art approach [35].

To validate the effectiveness and generalizability of our method, we conducted more experiments on challenging examples characterized by low texture, low contrast, low light, and low overlap. As shown in Fig. 4.2.1, our method can robustly stitch images and produce results with improved alignment and rectangular boundaries. Quantitative metrics and highlighted regions clearly demonstrate that the refinement step significantly improves stitching quality.

4.2.2. Evaluation of Boundary Rectification

Our method prioritizes boundary rectification to enhance the wide-angle effect of stitching results in an unsupervised framework. Therefore, we conduct separate evaluations of boundary rectification by comparing with SOTA methods [20, 35].

To compare with [20], we redesigned the training scheme by replacing our boundary loss $\mathcal{L}_{bdy}$ with $\mathcal{L}'_{bdy}$, as defined in [20] (see details in Section 3.3). Experimental results indicate that this solution has two main drawbacks. First, owing to the requirement for stitching mask generation, the computational overhead increases, and the actual training time is more than twice as long as ours. Second, the results generated by the retrained model are similar to those without the boundary loss, as shown in the last column of Table 1, indicating that the boundary constraint defined in [20] has no significant effect on boundary rectification. In comparison, our boundary loss function, which constrains the outer boundary of the stitching meshes, is more intuitive and effective, enabling efficient training and boundary rectangularization.

We further compare our method with RecStitchNet [35], a state-of-the-art learning-based method for image stitching with boundary rectification. In [35], the boundary rectification constraint is employed in the fine-tuning scheme, and irregular boundaries can be effectively rectified within 30–40 iterations. However, the Boolean union operation [1] used to extract the outer boundary of the stitched mesh is computationally expensive. As shown in Table 4, our method achieves a significantly convergence speed during the fine-tuning than RecStitchNet [35]. Furthermore, the Boolean union operation is non-differentiable, and thus cannot be incorporated into the training pipeline..

4.2.3. Ablation Study

To evaluate the effectiveness of each constraint, we retrained the model by removing one constraint at a time and then conducted an ablation study on the UDIS-D dataset [19]. Fig. 8 shows the stitching results without alignment, shape preservation, boundary constraints, and bidirectional decomposition. It is evident that, without the alignment constraint, the stitching result exhibits severe ghosting, and important structures are distorted without the shape constraint. Moreover, we observe that without the boundary constraint, our method degenerates to StabStitch++ [25]. Additionally, the stitching results without bidirectional decomposition fail to achieve a balance between alignment quality and boundary rectification.

Table 1 presents further quantitative results for our ablation study. The metrics include PSNR, SSIM and MASK, where MASK denotes the ratio of valid pixels within the bounding box. Columns 2–5 report the metrics of the stitching results obtained without alignment, shape preservation, boundary constraints, and bidirectional decomposition, respectively. Columns 6–7 and 8–9 show the results obtained by our method and [35] before and after fine-tuning, respectively. Both the visual comparison and the evaluation metrics demonstrate that each constraint is essential for producing satisfactory stitching results.

4.2.4. User Study

To explore user preference for image stitching, we conducted a user study with college students with basic computer skills, including students majoring in digital media, liberal arts, and design. Similar to Fig. 5, we present the

Fig. 7: Challenging examples characterized by low-texture, low-contrast, low-light, and low-overlap. The 1^{st} to 3^{rd} rows respectively show the inputs and the results of our method before and after fine-tuning, with red boxes highlighting the performance improvements from fine-tuning.

Fig. 8: The stitching results of the ablation study on the UDIS-D dataset [19]. Columns 2 to 5 show the stitching results obtained without alignment, shape preservation, boundary constraints, and bidirectional decomposition, respectively. The last column presents our results with all constraints incorporated, while red boxes indicate artifacts in the results obtained without these key constraints.

Table 3: A user study on the UDIS-D dataset [19] based on the preferences of 30 participants for the two methods before and after refinement.

	Content Alignment	Shape Preservation
RecStitchNet	$\mu(3.67)\sigma(0.66)$	$\mu(3.57)\sigma(0.57)$
Ours	$\mu(4.03)\sigma(0.61)$	$\mu(4.00)\sigma(0.53)$
p-value	0.014	0.002
RecStitchNet+FT	$\mu(4.07)\sigma(0.45)$	$\mu(4.13)\sigma(0.35)$
Ours+FT	$\mu(4.43)\sigma(0.57)$	$\mu(4.46)\sigma(0.51)$
p-value	0.005	0.010

Table 4: Average runtime of different methods. The 1^{st} column presents the average runtime of the combination of bidirectional stitching [25] and rectangling; the 2^{nd} and 3^{rd} columns show the average runtime of stitching by RecStitchNet [35] and BRecStitch, and the fine-tuning time in each iteration. All the experiments are carried out on the UDIS-D dataset [19].

Stitch&Rect.		RecStitchNet		BRecStitch	
Stitch	Rect.	w/o FT	Iter.	w/o FT	Iter.
$0.092s$	$0.094s$	$0.151s$	$0.656s$	$0.095s$	$0.075s$

stitching results produced by RectStitchNet [35] and our *BRecStitch*, along with the corresponding results after fine-tuning. In the user study, each participant evaluated 30 groups of stitching results, divided into two categories: the results before fine-tuning and the results after fine-tuning. To ensure effective and fair user ratings, each group of results was displayed on a separate page, with the presentation order randomized for each participant. Additionally, users can zoom in/out on the stitching results to facilitate their evaluations. Each example was rated on a 5-point scale ("1 = poor, 5 = excellent") by participants according to two core metrics: *Content Alignment* and *Shape Preservation*. To validate the statistical reliability of user preferences, we conducted a two-tailed paired t-test on the collected rating data. As shown in Table 3, our method achieves higher mean scores (μ) across all dimensions. Since the p-value of each comparison group is less than 0.05, these differences are statistically significant. This confirms that users prefer our method and demonstrates the effectiveness and robustness of our framework.

4.3. Performance

To test the effectiveness of our method, we report the performance of our *BRecStitch* and compare it with [25] and RecStitchNet [35] on the UDIS-D dataset [19]. As shown in Table 4, the first column refers to the runtime of separate stitching [25] and rectangling [20]. The 2^{nd} and 3^{rd} columns show the time cost of RecStitchNet [35] and our BRecStitch, as well as their fine-tuning stage which contains 30 iterations. Specifically, "w/o FT" and "Iter." denote the average stitching time without fine-tuning and the average duration of each iteration during fine-tuning, respectively. It is clear that our method (w/o FT) is much faster than [35] and the combination of stitching [25] and rectangling [20]. In addition, our fine-tuning scheme is more efficient than that of RecStitchNet [35]. We believe that the high performance lays an important foundation for the subsequent research on video stitching.

4.4. Discussion

In this paper, we focus on the boundary rectification problem in image stitching and propose a novel boundary rectification constraint, which is successfully integrated into an unsupervised image stitching framework. Instead of relying on the Boolean union operation [1], we design a novel and efficient solution to extract the outer stitching boundaries via a parallelized point-in-polygon check. Our boundary term is formulated by constraining the vertices of the outer boundary to remain close to their minimum bounding rectangle. Both experimental results and ablation studies clearly demonstrate the effectiveness of our boundary constraint within the unsupervised learning framework, which helps achieve a good balance between alignment and boundary regularity, and produce stitching results with satisfactory boundaries and more immersive effects.

The effectiveness of our unsupervised stitching also depends on the feature selection from the backbone network. To validate our selection of intermediate features from Stages 2 and 3 of ResNet-18 [8], we conduct additional comparative experiments on the UDIS-D dataset [19]. We compare the results from our selected intermediate layers with those from shallow layers (Stages 1 and 2) and deep layers (Stages 3 and 4) of ResNet-18. As shown in Fig. 9, the zoom-in view and the quantitative metrics indicate the superiority of our method in terms of feature alignment and structural preservation. The performance reported in Table 5 further demonstrates that our strategy achieves superior stitching performance while requiring lower computational cost.

Our method has several limitations. As shown in Fig. 10, the top of the building is severely distorted after the stitching with boundary rectification. The reason for this is the failure to preserve salient structures, such as straight lines. In fact, the local distortion and misalignment are more likely to manifest when a significant portion of content is missing due to large parallax, occlusion, etc. In such cases, the rectangular boundary constraint requires more aggressive mesh deformation to converge to a rectangle, which may stretch local structures or slightly compromise the pixel-level alignment in non-critical overlapping subregions. As a long-standing and challenging issue in image stitching, the trade-off between local structural fidelity and global boundary regularity still calls for more effective and practical solutions.

5. Conclusion

This paper proposes *BRecStitch*, an end-to-end network that is the first to successfully incorporate the boundary rectification constraint into the unsupervised stitching

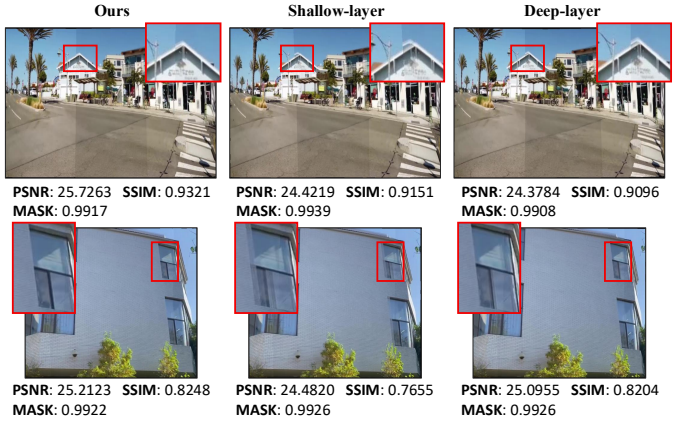

Fig. 9: Comparison of results obtained using features from the selected intermediate, shallow, and deep layers of ResNet-18. The zoom-in view and the metrics indicate the superiority of our method in terms of feature alignment and structural preservation.

Table 5: Metrics of the stitching results and average runtime obtained using our selected intermediate layers, as well as the shallow and deep layers of ResNet-18, on the UDIS-D dataset [19].

	Ours	Shallow-layer	Deep-layer
PSNR	25.0359	24.9542	24.4412
SSIM	0.8347	0.8344	0.8186
MASK	0.9911	0.9911	0.9918
AvgTime	0.091s	0.093s	0.099s

framework. To achieve more balanced distortions between two views, we first incorporate the bidirectional warping strategy from [25] into our global homography. Based on the global warping, a local regression network is further designed to generate local mesh motions for more accurate feature alignment and more regular stitching boundaries. Specifically, the highlight of local regression lies in the newly proposed rectangular boundary loss, which is featured by efficient outer stitching boundary extraction and effective boundary constraint in the training step. Experimental results demonstrate that our end-to-end network can stitch images effectively and efficiently, achieving a favorable balance between feature alignment and boundary regularity. To improve stitching results, we further design a fine-tuning scheme that can efficiently enhance alignment and boundary regularity within 10–20 iterations. In the future, we will investigate effective solutions for preserving salient structures (e.g., straight lines) and explore the feasibility of unsupervised video stitching with boundary rectification.

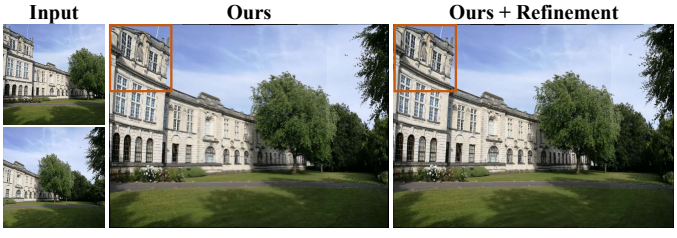

Fig. 10: Our method may fail to preserve salient structures such as straight lines when seeking to achieve both good alignment and rectangular boundaries.

Acknowledgements The authors would like to thank all anonymous reviewers for their valuable comments and Dr. Tong Li for her professional proofreading. This work was supported by "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No.2025C02014), the National Natural Science Foundation of China (No. 62502057).

References

[1] A new algorithm for computing boolean operations on polygons. *Computers & Geosciences*, 35(6):1177–1185, 2009.

[2] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, 1989.

[3] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.*, 74(1):59–73, 2007.

[4] Yu-Sheng Chen and Yung-Yu Chuang. Natural image stitching with the global similarity prior. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference*, volume 9909 of *Lecture Notes in Computer Science*, pages 186–201. Springer, 2016.

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

[6] Junhong Gao, Seon Joo Kim, and Michael S. Brown. Constructing image panoramas using dual-homography warping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 49–56. IEEE, 2011.

[7] Kaiming He, Huiwen Chang, and Jian Sun. Rectangling panoramic images via warping. *ACM Trans. Graph.*, 32(4):79:1–79:10, 2013.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778. IEEE, 2016.

[9] Qi Jia, Zhengjun Li, Xin Fan, Haotian Zhao, Shiyu Teng, Xinchen Ye, and Longin Jan Latecki. Leveraging line-point consistence to preserve structures for wide parallax image stitching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 12186–12195. IEEE, 2021.

[10] Zhiying Jiang, Zengxi Zhang, Xin Fan, and Risheng Liu. Towards all weather and unobstructed multi-spectral image stitching: Algorithm and benchmark. In *The 30th ACM International Conference on Multimedia, MM*, pages 3783–3791. ACM, 2022.

[11] Zhiying Jiang, Zengxi Zhang, Jinyuan Liu, Xin Fan, and Risheng Liu. Multispectral image stitching via global-aware quadrature pyramid regression. *IEEE Trans. Image Process.*, 33:4288–4302, 2024.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015.

[13] Jing Li, Zhengming Wang, Shiming Lai, Yongping Zhai, and Maojun Zhang. Parallax-tolerant image stitching based on robust elastic warping. *IEEE Trans. Multim.*, 20(7):1672–1687, 2018.

[14] Nan Li, Yifang Xu, and Chao Wang. Quasi-homography warps in image stitching. *IEEE Trans. Multim.*, 20(6):1365–1375, 2018.

[15] Kang Liao, Lang Nie, Chunyu Lin, Zishuo Zheng, and Yao Zhao. Recrecnet: Rectangling rectified wide-angle images by thin-plate spline model and dof-based curriculum learning. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 10766–10775. IEEE, 2023.

[16] Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong Fah Cheong. Smoothly varying affine stitching.

In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 345–352. IEEE, 2011.

[17] Zhongyu Lou and Theo Gevers. Image alignment by piecewise planar region matching. *IEEE Trans. Multim.*, 16(7):2052–2061, 2014.

[18] Yuan Mei, Lichun Yang, Mengsi Wang, Tianxiu Yu, and Kaijun Wu. Dunhuangstitch: Unsupervised deep image stitching of dunhuang murals. *IEEE Trans. Vis. Comput. Graph.*, 31(8):4226–4240, 2025.

[19] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Unsupervised deep image stitching: Reconstructing stitched features to images. *IEEE Trans. Image Process.*, 30:6184–6197, 2021.

[20] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Deep rectangling for image stitching: A learning baseline. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5730–5738. IEEE, 2022.

[21] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Depth-aware multi-grid deep homography estimation with contextual correlation. *IEEE Trans. Circuits Syst. Video Technol.*, 32(7):4460–4472, 2022.

[22] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Deep rotation correction without angle prior. *IEEE Trans. Image Process.*, 32:2879–2888, 2023.

[23] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Parallax-tolerant unsupervised deep image stitching. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 7365–7374. IEEE, 2023.

[24] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Semi-supervised coupled thin-plate spline model for rotation correction and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):9192–9204, 2024.

[25] Lang Nie, Chunyu Lin, Kang Liao, Yun Zhang, Shuaicheng Liu, and Yao Zhao. Stabstitch++: Unsupervised online video stitching with spatiotemporal bidirectional warps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(9):7443–7456, 2025.

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

[27] Lang Nie, Yuan Mei, Kang Liao, Xunqiu Xu, Chunyu Lin, and Bin Xiao. Robust image stitching with optimal plane. *IEEE Trans. Vis. Comput. Graph.*, pages 1–11, 2026.

[28] Dae-Young Song, Geonsoo Lee, Heekyung Lee, Gi-Mun Um, and Donghyeon Cho. Weakly-supervised stitching network for real-world panoramic image generation. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference*, volume 13676 of *Lecture Notes in Computer Science*, pages 54–71. Springer, 2022.

[29] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):722–732, 2010.

[30] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C. Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Transactions on Image Processing*, 23(2):684–695, 2014.

[31] Julio Zaragoza, Tat-Jun Chin, Quoc-Huy Tran, Michael S. Brown, and David Suter. As-projective-as-possible image stitching with moving DLT. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1285–1298, 2014.

[32] Guofeng Zhang, Yi He, Weifeng Chen, Jiaya Jia, and Hujun Bao. Multi-viewpoint panorama construction with wide-baseline images. *IEEE Trans. Image Process.*, 25(7):3099–3111, 2016.

[33] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. FSIM: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.*, 20(8):2378–2386, 2011.

[34] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 586–595. IEEE, 2018.

[35] Yun Zhang, Yu-Kun Lai, Lang Nie, Fang-Lue Zhang, and Lin Xu. Recstitchnet: Learning to stitch images with rectangular boundaries. *Comput. Vis. Media*, 10(4):687–703, 2024.

[36] Yun Zhang, Yu-Kun Lai, and Fang-Lue Zhang. Content-preserving image stitching with piecewise rectangular boundary constraints. *IEEE Trans. Vis. Comput. Graph.*, 27(7):3198–3212, 2021.

[37] Tianhao Zhou, Haipeng Li, Ziyi Wang, Ao Luo, Chen-Lin Zhang, Jiajun Li, Bing Zeng, and Shuaicheng Liu. Recdiffusion: Rectangling for image stitching with diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2692–2701. IEEE, 2024.