Content-Preserving Image Stitching with Piecewise Rectangular Boundary Constraints

Yun Zhang, Yu-Kun Lai, *Member, IEEE,* and Fang-Lue Zhang, *Member, IEEE*

***Abstract*—This paper proposes an approach to content-preserving image stitching with regular boundary constraints, which aims to stitch multiple images to generate a panoramic image with a piecewise rectangular boundary. Existing methods treat image stitching and rectangling as two separate steps, which may result in suboptimal results as the stitching process is not aware of the further warping needs for rectangling. We address these limitations by formulating image stitching with regular boundaries in a unified optimization. Starting from the initial stitching results produced by the traditional warping-based optimization, we obtain the irregular boundary from the warped meshes by polygon Boolean operations which robustly handle arbitrary mesh compositions. By analyzing the irregular boundary, we construct a piecewise rectangular boundary. Based on this, we further incorporate line and regular boundary preservation constraints into the image stitching framework, and conduct iterative optimization to obtain an optimal piecewise rectangular boundary. Thus we can make the boundary of the stitching results as close as possible to a rectangle, while reducing unwanted distortions. We further extend our method to video stitching, by integrating the temporal coherence into the optimization. Experiments show that our method efficiently produces visually pleasing panoramas with regular boundaries and unnoticeable distortions.**

***Index Terms*—content-preserving image stitching, panoramic image, rectangling, polygon Boolean operations, piecewise rectangular boundary.**

I. Introduction

The rapid recent advances in digital visual media mean that the public can now capture and produce high-quality images and videos, which has promoted computer graphics applications that utilize visual data captured by ordinary users. Image/video panorama is one of these successful applications. With the integrated panorama module in their smart phones and portable cameras, people can easily take panoramic photos by simply moving their cameras. It is also the most accessible way to get virtual reality content for immersive visual experience. However, unlike well calibrated images captured by professional devices with camera arrays, the intrinsic and extrinsic parameters of the images captured by consumer-level devices are difficult to estimate. Thus, robust image stitching methods which directly stitch visual content are highly important for such applications designed for ordinary users.

Recently, much progress has been made in image stitching. However, due to the casual motion of hand-held cameras, existing stitching methods usually produce panoramic images with irregular boundaries after the local feature alignment. But the common application scenario for stitched images is to display full panoramas on normal screens, or generate free-viewpoint photos from part of the whole scene recorded as an image collection, which means that we can only show them in rectangular windows. To achieve this, a simple and direct method is cropping, but it usually causes loss of important content in the stitched panorama, and reduces the impression of wide-angle photography. In order to produce panoramic images with rectangular boundaries, image completion techniques [2] [3] are used to synthesize missing regions in the bounding box of panoramic images. However, these methods are not stable, and may fail when synthesizing regions with rich structures and semantically meaningful objects.

He et al. [1] proposed a *rectangling* method to produce visually pleasing panoramic images with desired rectangular boundaries by warping the initial stitched panoramas. Although effective in many examples, their method suffers from the following problems: (1) In their method, stitching and rectangling are two *separate* processes, so the latter rectangling step may distort the optimized stitching result, making it hard to get an optimal rectangular panorama. Moreover, making arbitrary boundaries rectangular may also introduce excessive distortions unacceptable for target applications. (2) Their method relies on placing a grid mesh on the stitched irregular panorama for rectangling, where the mesh may contain pixels outside the stitched image due to the irregularity of the boundary, leading to the resulting rectangular image containing small "holes" near the boundary; see the zoom-in view of Fig. 1(g). (3) The warping-based method may cause large distortions and destroy feature alignment when turning an incompletely captured scene to a rectangle. In summary, when the gap between the target rectangular boundary and irregular panorama boundary is large, or there are "holes" which are difficult to fill in by inpainting or warping, a better approach is required to create panoramic images with regular boundaries while avoiding large distortions.

In this paper, we propose a novel approach for content-preserving image stitching, which aims to regularize the boundary of the stitched panorama, and preserve as much content as possible in a rectangular cropping window. Our method is based on the following observations: (1) Rectangling and stitching are tightly related, and optimizing the two

Yun Zhang is with the College of Media Engineering, Communication University of Zhejiang, Hangzhou 31008, China.
E-mail: zhangyun@cuz.edu.cn
Yu-Kun Lai is with the School of Computer Science and Informatics, Cardiff University, Cardiff, CF24 3AA, UK.
E-mail: LaiY4@cardiff.ac.uk
Fang-Lue Zhang is with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6012, New Zealand.
E-mail: fanglue.zhang@ecs.vuw.ac.nz

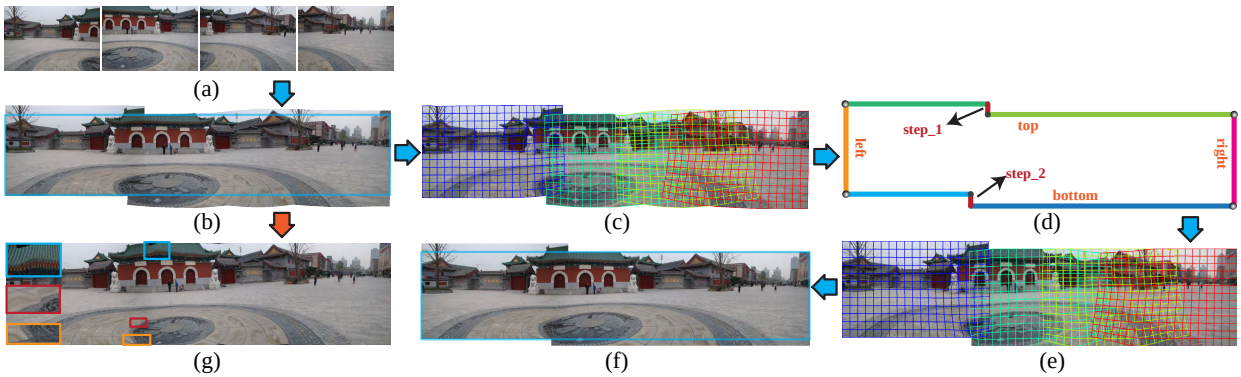


Figure 1. Pipeline of our method with piecewise rectangular boundaries. (a) input images, (b) initial stitching with an irregular boundary, (c) meshes of initial stitching, (d) piecewise rectangular boundary, (e) warped meshes for piecewise rectangling, (f) our result, (g) rectangular panorama result of [1].

processes *simultaneously* can help produce better rectangular panoramas in a content-aware manner. (2) The aim of panorama rectangling is to preserve as much image content as possible in a rectangular window while avoiding unexpected distortions. To achieve this, an irregular boundaries should not be simply optimized to be a single rectangle as evidenced by Fig. 1(g). We propose to instead use a more flexible *piecewise rectangular* (a.k.a. *rectilinear*) boundary to ensure the regularity of the boundary while avoiding excessive distortions; see Fig. 1(d) for an example. Specifically, a piecewise rectangle is defined as the union of one or more rectangles, and a piecewise rectangular boundary refers to the outer boundary of such a shape. We provide a user study to evaluate the advantage of stitching with piecewise rectangular boundaries. Using piecewise rectangular boundaries also has the advantage that traditional rectangular boundaries are treated as a special case, and will provide rectangular results when appropriate. Our method works well even for challenging cases and can produce visually pleasing results without user interactions; see Fig. 12 for some examples. After stitching the input images using the traditional method with the global similarity prior [4], we extract the outer boundary of the stitching result and analyze the boundary constraints, and finally perform a global optimization taking all the constraints into account to obtain the stitching results with piecewise rectangular boundary. Our method can robustly stitch a large number of images. To achieve this, we treat each image in the initial stitching result as a warped mesh, and utilize polygon Boolean operations to extract irregular boundaries and suitable boundary constraints for piecewise rectangling. In the global optimization stage, we take into account regular boundary, shape preservation, line preservation and global similarity constraints in a unified optimization framework. To obtain panoramic images with optimal piecewise rectangular boundaries, we firstly automatically extract the piecewise rectangular boundary, and then iteratively combine the boundary segments connected by steps to simplify the shape of the panorama boundary while avoiding large distortions. Finally, after minimizing the energy function, we get the stitching result by warping and blending. When the target boundary is simply a rectangle, our method performs stitching and rectangling simultaneously, and can produce panoramas with a rectangular boundary; see some examples in Fig. 13. Our method can help users easily crop panoramas while preserving as much content as possible in a rectangular cropping window, and avoiding unwanted distortions, thus can enhance the panorama viewing experiences. Furthermore, our method can be extended to video stitching with regular boundaries.

The main contributions of this paper are summarized as:

- We propose a global optimization approach to producing panoramic images by simultaneously stitching images and optimizing the boundary regularity in a unified framework. By doing so, our method reduces undesired distortions compared with traditional approaches where stitching and rectangling are treated as two separate steps.
- We propose to use piecewise rectangular boundaries to achieve regular boundaries while preserving content from input images as much as possible and avoiding excessive distortions compared with the traditional rectangling. To effectively balance distortion and boundary simplicity, we further develop a fully automatic algorithm to produce optimized piecewise rectangular boundaries.

II. RELATED WORK

We briefly review the techniques most related to our work.

Image stitching. Image stitching aims to create seamless and natural photo-mosaics. A comprehensive survey of image stitching algorithms is given in [5]. Brown et al. [6] proposed a method for fully automatic panoramic image stitching, which aligns multiple images by a single homography. Their method is effective under the following assumptions: (1) the camera only rotates around its optical center; (2) the images are shot from the same viewpoint; (3) the scenes are nearly planar. However, for images shot by hand-held cameras, they always contain parallax, which limits the application of their method. Given the limitation of a single homography, Gao et al. [7] proposed to use two homographies to perform nonlinear alignment, where the scene is modeled by dominant distant and ground planes. However, their method is only effective when there are no local perspective variations.

For better performance in image alignment, Zaragoza et al. [8] proposed the as-projective-as-possible (APAP) warping based on the Moving Direct Linear Transformation (DLT), and can seamlessly align images with different projective

models. Their method can handle global perspectives, while allowing local non-projective deviations, thus can deal with some challenging cases. This technique has been widely applied in image alignment due to its excellent performance and in this paper we also use APAP for our initial stitching before optimization. Based on APAP, researchers attempted to obtain more natural panoramas. Lin et al. [9] combined local homography and global similarity transformation to achieve more continuous and smooth stitching results, which have less visible parallax and perspective distortions. Li et al. [10] proposed a dual-feature warping-based model by combining keypoints and line segment features. However, the 2D model proposed in [10] cannot handle large parallax and depth variations, and it is difficult to determine the line correspondences in images with large parallax. Chang et al. [11] proposed the parametric warping which combines projective and similarity transformation. By combining this with APAP [8], their method can significantly reduce distortions in the stitching results. Chen et al. [4] further proposed the natural image stitching with a global similarity prior. They designed a selection scheme to automatically determine the proper global scale and rotation for each image.

There are also methods focusing on local alignment adjustment for eliminating stitching artifacts. To stitch images with large parallax, Zhang et al. [12] proposed a local stitching method, which is based on the observation that overlapping regions do not need to be perfectly aligned. Lin et al. [13] proposed the seam-guided local alignment where optimal local alignment is guided by the seam estimation. In their method, salient curves and line structures are preserved by local and non-local similarity constraints. Very recently, Li et al. [14] proposed the robust elastic warping for parallax-tolerant image stitching. To ensure robust alignment, they applied the Bayesian model to remove incorrect local matches.

However, none of these methods above consider how to achieve better results in the display window. He et al. [1] proposed a content-aware warping method to produce rectangular images from the stitched panoramas. Their method is effective to rectify irregular boundaries caused by projections and casual camera movements. However, their two-step warping strategy separates the stitching and rectangling processes, and therefore cannot ensure optimal solutions. Moreover, their method cannot cope well with scenes that are not completely captured. Unlike [1], we incorporate stitching and rectangling into a unified framework, and construct a global optimization to obtain piecewise rectangular panoramic images.

Video stitching. Compared with image stitching, video stitching is more difficult, due to the camera motion, dynamic foreground and large parallax. For static camera settings, such as multi-camera surveillance [15] [16], videos from different cameras are aligned only once, and the main challenge is to avoid ghosting and artifacts caused by moving objects. For moving cameras with relatively fixed positions, such as camera arrays fixed poles [17], cameras can be pre-calibrated for global stitching of videos, and spatio-temporally coherent warping and minimizing distortion are the main challenges due to the motion and parallax. Google Street View [18] also utilized moving camera arrays for street view capture and panorama generation. To generate high-quality panoramic videos for those captured by fixed camera arrays, Zhu et al. [19] proposed a method for real-time panoramic video blending. Meng et al. [20] proposed a multi-UAV (unmanned aerial vehicle) surveillance system that supports real-time video stitching. Recently, many researchers focused on stitching algorithms for videos shot by multiple hand-held cameras. El-Saban et al. [21] improved the optimal seam selection blending for fast video stitching; however, they do not consider video stabilization. Lin et al. [22] were the first to propose a robust framework to stitch videos from moving hand-held cameras, which incorporates stabilization and stitching into a unified framework. Guo et al. [23] and Nei et al. [24] further improved the performance of a joint video stabilization and stitching framework. Their main contributions include: estimation of inter-motions between cameras and intra-motions in a video, and common background identification for multiple input videos. In this paper, we further extend our content-preserving image stitching to videos that are captured from unstructured camera arrays [17].

III. Overview

Fig. 1 gives the pipeline of our content-preserving stitching method. The input to our approach is a number of images with partial overlaps, and the goal is to obtain a panoramic image with regular boundaries. Similar to previous warping-based stitching, we place a separate quad mesh on each image, and construct energy functions with constraints on all image meshes. The core of our approach is a unified optimization framework that combines image stitching and piecewise rectangling, which contains the following key steps:

Preprocessing. We first calculate the image match graph using the method proposed in [6]. The images that are connected in the match graph are aligned in the stitching process. This automatic matching process allows stitching with complex image overlaps; see examples in Fig. 9. For line and global feature preservation, we detect straight lines in all images using the fast line segment detector [25].

Initial image stitching. The goal of this step is to initialize our content-preserving stitching, which also provides the basis for analyzing regular boundary constraints. The stitching strategy in this step is also incorporated into the optimization of our piecewise rectangling stitching. We apply APAP [8] for accurate feature alignment. Inspired by [4], we also add a global similarity term for more natural stitching with less distortions.

Piecewise rectangular stitching. After the initial image stitching, we extract the contour of each warped mesh, and obtain the irregular boundary of the stitching result by polygon Boolean union operations. Then, we analyze the vertices and intersections on the irregular boundary to get regular boundary constraints for our energy optimization. To balance boundary simplicity and content distortion, we further iteratively optimize the piecewise rectangular boundary by combining the boundary segments connected by steps on the regular boundary. Finally, we minimize the energy function, and get the stitching result by warping and blending.

IV. Initial Image Stitching

Like previous image stitching methods [4] [9], each input image is represented using a regular quad mesh placed on it. Let $V = \{V^i\}$ and $E = \{E^i\}$ be the sets containing all the vertices and edges of input images, where $i = 1, 2, \ldots, N$, and N is the number of images to be stitched. The j^{th} vertex of V^i is then represented as V^i_j. We aim to obtain the deformed vertices $\hat{V}$ by minimizing the energy function $\Phi(\hat{V})$, which contains the following three terms.

Feature alignment. Given its good performance in piecewise alignment, we apply APAP [8] for feature alignment, and define the term as

$$\phi_a(\hat{V}) = \sum_{(i,j)\in G} \sum_{m^{ij}_k \in M^{ij}} \|\tilde{v}(m^{ij}_k(i)) - \tilde{v}(m^{ij}_k(j))\|^2, \quad (1)$$

where G refers to the image matching graph which contains all the matched image pairs (i, j), and m^{ij}_k represents one of the feature matchings for image pair (i, j). The position of the deformed matched feature point is denoted by $\tilde{v}(m^{ij}_k(i))$, which is represented by interpolating the vertex positions of the mesh grid on image i that contains $m^{ij}_k(i)$. $\tilde{v}(m^{ij}_k(j))$ is similarly defined.

Shape preservation. We use the shape preservation term defined in [26], which splits each grid cell into two triangles and applies ARAP warping [27]. The term is defined as

$$\phi_s(\hat{V}) = \sum_{i=1}^{N} \sum_{\hat{V}^i_j \in \hat{V}^i} \|\hat{V}^i_j - \hat{V}^i_{j_1} - \xi \mathbf{R}(\hat{V}^i_{j_0} - \hat{V}^i_{j_1})\|^2 \quad (2)$$

where $\mathbf{R} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ and $\xi = \|V^i_j - V^i_{j_1}\| / \|V^i_{j_0} - V^i_{j_1}\|$ are the 90° rotation matrix and scaling parameters. To achieve shape-preserving warping, the deformed vertices $\hat{V}^i_j$, $\hat{V}^i_{j_0}$, $\hat{V}^i_{j_1}$ should satisfy the similarity transform.

Global similarity. We use the global similarity term proposed in [4] to preserve the naturalness of panoramic images. We first set image I_1 as a reference, and specify its desired rotation angle θ_1 to 0° with its scaling s_1 set to 1. For any other image I_i ($2 \le i \le N$), the desired scaling s_i and rotation angle θ_i w.r.t. I_1 are calculated according to [4]. The global similarity term is defined as

$$\phi_g(\hat{V}) = \sum_{i=2}^{N} \sum_{\hat{e}^i_j \in \hat{E}^i} \beta(\hat{e}^i_j)[\|c_x(\hat{e}^i_j) - s_i \cos\theta_i\|^2 + \|c_y(\hat{e}^i_j) - s_i \sin\theta_i\|^2], \quad (3)$$

where $c_x(\hat{e}^i_j)$ and $c_y(\hat{e}^i_j)$ refer to the coefficients of the warped grid edges $\hat{e}^i_j$ for similarity transforms in the x and y directions; see details in [28]. $\beta(\hat{e}^i_j)$ is used to assign more importance to the edges in overlapping regions; see detailed definition in [4].

With the energy terms above, we define the overall energy for image stitching as

$$\Phi_{stitch}(\hat{V}) = \gamma_a \phi_a(\hat{V}) + \gamma_s \phi_s(\hat{V}) + \gamma_g \phi_g(\hat{V}), \quad (4)$$

where γ_a, γ_s, γ_g are used to control the importance of the three energy terms. In our experiments, we fix $\gamma_a = 1$ and set $\gamma_s = 6.5$, $\gamma_g = 0.5$ by default. We give more importance to preserve the shape of image meshes for less distortion in the warping based optimization. The warped meshes of the stitched images with irregular boundaries will serve as the initial state for the optimization to produce the piecewise rectangular stitching result.

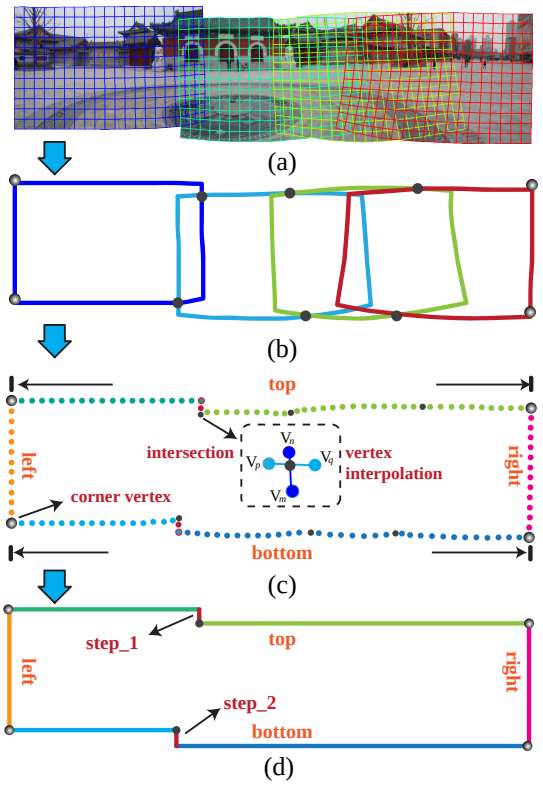


Figure 2. Irregular boundary extraction and piecewise rectangular boundary construction. (a) meshes of initial stitching, (b) boundary Boolean operations, (c) irregular boundary extraction, (d) piecewise rectangular boundary.

V. Piecewise Rectangular Stitching

For a given image collection, directly warping them to align with a single rectangle may not be preferable when large regions are missing. For example, as shown in Fig. 1(g), the warping-based rectangling [1] may introduce unwanted distortions when the gap between the irregular boundary of the initial stitching result and the target rectangular boundary is too large. To avoid such undesirable artifacts, we propose to generate *piecewise rectangular* boundaries which can make the target boundary as rectangular as possible, while avoiding excessive distortions if there are large missing regions in the whole scene. We also consider the content-preserving constraints simultaneously when optimizing the warped meshes. Compared with [1], using the piecewise rectangular boundary, the stitching result can be easily cropped into a rectangular photo to display more contents in a screen; see examples in Fig. 12. We first extract and analyze the irregular boundaries from the initial stitching results in Section IV, and then design an optimization objective for stitching that considers the piecewise rectangular boundary constraints.

A. Irregular Boundary Extraction

The irregular boundary extraction is an important step for panorama rectangling. Unlike [1], which places only one mesh over the stitched panorama with an irregular boundary, our method places a separate mesh for each image to be stitched. As a result, the irregular boundary consists of vertices from different image meshes, and around overlapping regions, edges from different meshes intersect with each other, as shown in Fig. 2(a). Although using a single mesh as in [1] makes the representation simpler, it has an unavoidable limitation that due to the boundary irregularity, the mesh grid may contain regions outside the stitched images, leading to small "holes" in the rectangling results; see Fig. 1(g). In our method, to cope with multiple meshes and arbitrary overlapping situations, we notice that the overall irregular boundary is formed by the boundaries of the warped meshes. More specifically, the boundary of each warped mesh is a polygon, and the union of all such polygons forms a compound polygon corresponding to the stitched image. The irregular boundary of the stitching result can be simply obtained as the boundary of the compound polygon, as illustrated in Fig. 2(b).

Therefore, we propose a simple and effective algorithm for irregular boundary extraction, based on polygon Boolean union operations [29]; see Alg. 1. The input includes the mesh vertices of all warped images, and the goal is to obtain the vertices on the irregular boundary. As shown in Fig. 2(c), to simplify discussion we assume that the irregular boundary is split into four sides, denoted as B^k ($k = 1, 2, 3, 4$) corresponding respectively to the *top*, *right*, *bottom* and *left* sides (in the clockwise order). Let $\hat{P}^i$ be the polygon of the i^{th} warped image. We use the algorithm in [29] to efficiently calculate the compound polygon $\hat{P}$ as the union of all these image polygons

$$\hat{P} = \bigcup_{i=1}^{N} \hat{P}^i. \tag{5}$$

Denote by $\hat{P}_j$ the j^{th} vertex of $\hat{P}$ in the clockwise order. We similarly use it to represent the position of the vertex when there is no ambiguity. $\hat{P}_j$ can either be a boundary vertex from a warped mesh, or the intersection of two warped mesh edges. We introduce an indicator function $\zeta(\hat{P}_j)$, which is 1 if it is a vertex from a warped mesh, and 0 otherwise. For the former case, we use $\hat{V}_{k_j}$ to indicate the warped vertex. In the latter case, the position of the intersection point is obtained using a linear interpolation of the 4 vertices from the two intersecting grid edges. Denote by $\boldsymbol{\kappa}_j = [\hat{V}_{m_j}, \hat{V}_{n_j}, \hat{V}_{p_j}, \hat{V}_{q_j}]$ the vector containing 4 vertices, and $\boldsymbol{\eta}_j = [c_{m_j}, c_{n_j}, c_{p_j}, c_{q_j}]$ their contributing weights. The position of the intersection point $\hat{P}_j = \boldsymbol{\kappa}_j \cdot \boldsymbol{\eta}_j$.

To work out the irregular boundary sides B^k ($k = 1, 2, 3, 4$), we first obtain the axis-aligned bounding rectangle $\hat{R}$ of $\hat{P}$. Denote by $\hat{C}_k$ the 4 corners of the rectangle $\hat{R}$. The 4 corner vertices V_{C_k} on the warped meshes are then defined as the vertices on the warped meshes closest to $\hat{C}_k$, i.e.

$$V_{C_k} = \underset{\hat{V}_j \in \hat{V}}{\arg\min} \|\hat{V}_j - \hat{C}_k\|. \tag{6}$$

Algorithm 1: Irregular boundary extraction

Input: Mesh vertices $\hat{V}^i$ of each warped image I_i, $i = 1, 2, \ldots, N$;
Output: Indexes of boundary vertices B^k, $k = 1, 2, 3, 4$ corresponding to *top*, *right*, *bottom* and *left* sides of the boundary;

```
Let P̂^i be the polygon of I_i;
Calculate P̂ using polygon union operators in Equ. 5;
foreach P̂_j ∈ P̂ do
    Use ζ(P̂_j) to indicate if it is a vertex (1) or an
     intersection point (0);
    if ζ(P̂_j) == 1 then
        Record the vertex of the warped mesh V̂_{k_j};
    end
    else
        Record the relevant vertices and their weights:
          κ_j = [V̂_{m_j}, V̂_{n_j}, V̂_{p_j}, V̂_{q_j}];
          η_j = [c_{m_j}, c_{n_j}, c_{p_j}, c_{q_j}];
    end
end
Determine the bounding rectangle R̂ of P̂;
Find the 4 corners Ĉ_k of R̂;
Calculate the 4 corner vertices V_{C_k} using Equ. 6;
Split P̂ into B^k using V_{C_k} (k = 1, 2, 3, 4);
```

The 4 corner vertices V_{C_k} can easily split the compound polygon $\hat{P}$ into 4 sides, which are denoted as B^k.

As shown in Fig. 2, the initial image stitching result has an irregular boundary formed by the overlapping of 4 image meshes. Fig. 2(b) shows contours of all the meshes, where each contour is shown in a different color, and the black circles are the intersections of these contours. As shown in Fig. 2(c), after the polygon Boolean union operations, the irregular boundaries are correctly extracted and classified into 4 sides.

B. Piecewise Rectangular Boundary Constraints

Given vertices from each irregular boundary side B^k ($k = 1, 2, 3, 4$), the aim of this step is to group them to form boundary sections, where each section S_j^k represents a sequence of boundary vertices that are in the same direction and should be aligned horizontally or vertically in the target piecewise rectangular shape. As illustrated in Fig. 2(d), each section is shown in a different color. We first sequentially collect all the corner vertices and intersection points $\{\Psi_j^k\}$ from B^k, and initialize the boundary section $\{S_j^k\}$ with the vertices between adjacent corners and intersections (including the two endpoints). We then repeatedly merge two adjacent boundary sections $S_{j_1}^k$ and $S_{j_2}^k$ if they are in the same direction, i.e. $dir(S_{j_1}^k) = dir(S_{j_2}^k)$, where $dir(\cdot)$ works out the dominant direction as either horizontal (0) or vertical (1). When no further merging is possible under this rule, we further merge the boundary sections before and after the very short sections with less than 2 vertices (referred to as small steps), to avoid overly complicated boundary structure. After analyzing the irregular boundary, we calculate the target boundary value of

Algorithm 2: Piecewise rectangular boundary analysis

Input: Irregular boundary sides B^k from Alg. 1;
Output: Boundary sections $S^k = \{S_j^k\}$ corresponding to the boundary side B^k;
$S^k = \emptyset$;
Collect corners and intersections $\Psi^k = \{\Psi_j^k\}$ from B^k;
foreach $\Psi_j^k, \Psi_{j+1}^k \in B^k$ **do**
  $S_j^k = \{\Psi_j^k ...$ intermediate vertices $... \Psi_{j+1}^k\}$;
  Add S_j^k to S^k;
end
repeat
  foreach *adjacent boundary section* $(S_{j_1}^k, S_{j_2}^k)$ $S_{j_1}^k \bigcap S_{j_2}^k \neq \emptyset$ **do**
    if $dir(S_{j_1}^k) == dir(S_{j_2}^k)$ **then**
      Merge $S_{j_2}^k$ to $S_{j_1}^k$;
    end
  end
  foreach *boundary section* S_j^k, $len(S_j^k) \leq 2$ **do**
    Merge boundary sections before and after S_j^k;
  end
until *no further merging*;
foreach *boundary section* S_j^k **do**
  if $dir(S_j^k) == 0$ *(horizontal)* **then**
    $val(S_j^k) = Avg(B_t^k.y)(\forall B_t^k \in S_j^k)$;
  end
  else
    $val(S_j^k) = Avg(B_t^k.x)(\forall B_t^k \in S_j^k)$;
  end
end

each section $val(S_j^k)$ by averaging their coordinates in the corresponding direction. The algorithm is summarized in Alg. 2. As shown in Fig. 2(d), the top and bottom boundary sides contain 3 segments each, and steps orthogonal to the sides are essential to reduce distortions in panorama rectangling.

C. Piecewise Rectangular Stitching

We design a global optimization which simultaneously finds the optimal image stitching and piecewise rectangling results. Our energy function contains terms relating to feature alignment, shape preservation and global similarity constraints that are used for stitching. Besides, we also consider regular boundary and line preservation constraints to avoid unexpected distortions when rectangling irregular boundaries. The energy terms for stitching have been defined in Section IV, and we now define energy terms for irregular boundary rectangling as follows.

Regular boundary preservation. With the piecewise rectangular boundary constraints, we define the regular boundary preservation energy as

$$\phi_r(\hat{V}) = \sum_{k=1}^{4} \sum_{S_j^k \in B^k} \sum_{\hat{V}_t \in S_j^k} \|\mathbf{\Lambda}_{dir(S_j^k)}[\zeta(\hat{V}_t)\hat{V}_t + (1-\zeta(\hat{V}_t))(\boldsymbol{\kappa}_t \cdot \boldsymbol{\eta}_t)] - val(S_j^k)\|^2, \tag{7}$$

where B^k, $k = 1, 2, 3, 4$ refers to the boundary sides in the *top*, *right*, *bottom* and *left* directions; S_j^k represents the j^{th} boundary section in the k^{th} side, and $val(\cdot)$ refers to the value of the target boundary section. As defined before, $\zeta(\hat{V}_t)$ indicates the type of the boundary point, either as a mesh vertex (1) or the intersection of two edges (0). $\mathbf{\Lambda}_0 = [0 \quad 1]$ and $\mathbf{\Lambda}_1 = [1 \quad 0]$ are 1×2 matrices, used to extract the y and x components of the coordinates respectively, to constrain the position of the boundary point to be close to the desired values.

Line preservation. To avoid unexpected distortions after warping, we also need to preserve straight lines in panoramas. In the initial stitching step, we are only concerned with obtaining the irregular boundary, thus the line preservation term is not necessary in that step. We use the line preservation term from [22], and the line segments are detected using [25]. Let L^i be the set of all detected line segments in image I_i. For a given line segment $l \in L^i$, assume that it contains p subsegments, with sample points $l_0, l_1, \ldots, l_p$. Each sample point l_j is represented by interpolating vertices of the mesh grid that contains l_j. Specifically, $l_j = \hat{\mathbf{V}}_{l_j}^i \cdot \mathbf{\Omega}_{l_j}^i$, where $\hat{\mathbf{V}}_{l_j}^i$ refers to the warped grid vertices, and $\mathbf{\Omega}_{l_j}^i$ are the corresponding weights before warping. The line preservation term is defined such that the position of a sample point l_j should be close to the position obtained by a linear interpolation of two endpoints l_0 and l_p with weights $(1 - j/p)$ and j/p. We define the energy term as

$$\phi_l(\hat{V}) = \sum_{i=1}^{N} \sum_{l \in L_i} \sum_{j=1}^{p-1} \|(1 - \frac{j}{p})\hat{\mathbf{V}}_{l_0}^i \cdot \mathbf{\Omega}_{l_0}^i + \frac{j}{p}\hat{\mathbf{V}}_{l_p}^i \cdot \mathbf{\Omega}_{l_p}^i - \hat{\mathbf{V}}_{l_j}^i \cdot \mathbf{\Omega}_{l_j}^i\|^2. \tag{8}$$

Total energy. With the piecewise rectangular boundary and line preservation constraints, the total energy function for our content-preserving image stitching is defined as

$$\Phi(\hat{V}) = \Phi_{stitch}(\hat{V}) + \gamma_r \phi_r(\hat{V}) + \gamma_l \phi_l(\hat{V}), \tag{9}$$

where Φ_{stitch} is the stitching energy function defined in Equ. 4, γ_r and γ_l are weights to control the importance of energy terms. We set $\gamma_r = 10^3$ to ensure the regularity of the boundary. In our experiments, we find that line preservation is more important than local shape preservation, thus γ_l is set higher to 15 to avoid too much distortion in straight lines.

D. Refinement of Piecewise Rectangular Boundaries

As shown in Fig. 3(f), our piecewise rectangular boundary may contain some unnecessary steps, which are defined as *short* boundary sections orthogonal to the direction of the side, which may degrade the rectangling effects. For optimal stitching with a regular boundary, we further propose to iteratively refine the piecewise rectangular boundary. After minimizing the total energy defined in Equ. 9, we calculate the energy value $\Phi(\hat{V})$ using the optimized vertices, denoted as E_0. Then, we repeat the following steps at each iteration until no further improvement can be made.

In the t^{th} iteration ($t = 1, 2, \ldots$), we first analyze the feature points and line detection results near the boundary sections connected by steps. If such feature points and lines

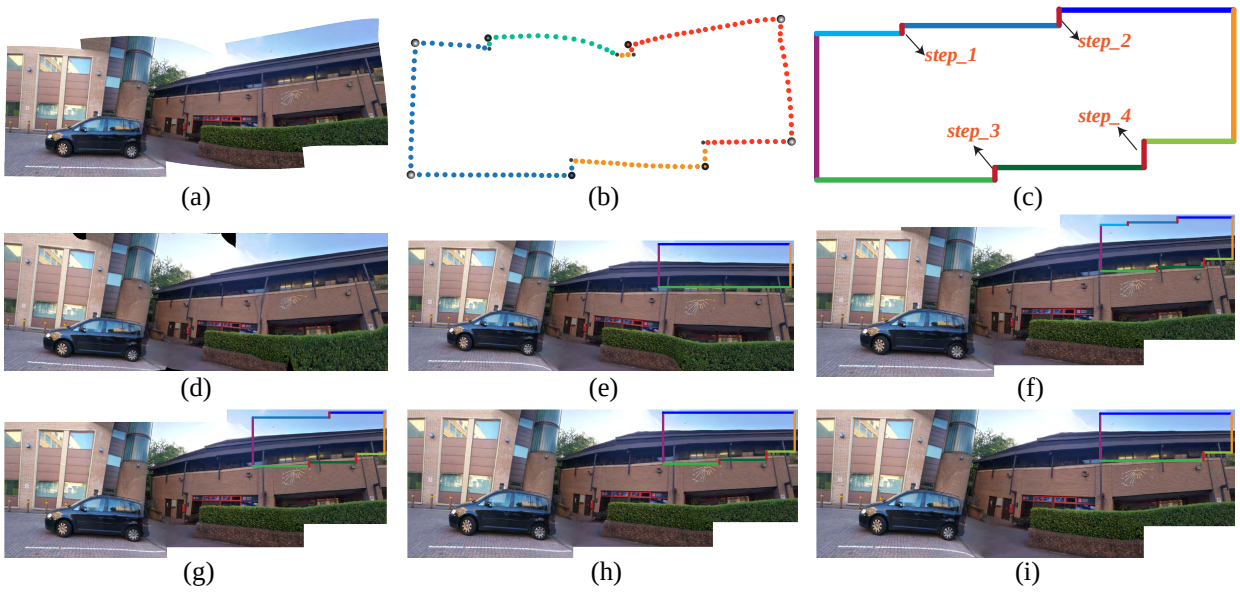


Figure 3. Piecewise rectangling in image stitching. (a) initial stitching result with an irregular boundary, (b) irregular boundary extraction, (c) target boundary estimation, (d) rectangular stitching result of [1], (e) our rectangular stitching result, (f-i) stitching results obtained by piecewise rectangling with iterative refinement, (i) our final stitching result.

exist, the corresponding step cannot be removed; see the steps in Fig. 1(d). When there are few features and lines, e.g. the local image contains featureless grass and sky, we further analyze such steps as follows: For each step, we attempt to remove it and merge the boundary sections before and after it, which leads to a simplified boundary, and then we apply the same image stitching by minimizing the total energy defined in Equ. 9. The minimum energy obtained by removing a step is denoted as E_t. We further compare E_t with E_{t-1} from the last iteration. If $E_t - E_{t-1} < \sigma$, which means that the distortion in this iteration is acceptable, we accept the new result and proceed to the next iteration. Otherwise, the new result is rejected, and we return the result from the last iteration as the final result. In this paper, we set the threshold σ to $\xi \cdot E_{t-1}$, and $\xi = 0.05$ by default, which works well in most examples. Our method is general in that the panorama rectangling proposed by He et al. [1] can be classified as a special case of our piecewise rectangling, when there are no steps in the target boundary.

Fig. 3(e) is the rectangling result of our method when all steps are removed, and there exists considerable distortion in the bottom-right corner. Compared with the result of [1] in Fig. 3(d) which contains "holes" and distortions, our rectangling result is more reasonable. Figs. 3(f-i) show results of our piecewise rectangling in each iteration, and the top-right corner of each result shows the shape of the target piecewise rectangular boundary. These results demonstrate that each iteration makes the boundary of panorama closer to a rectangle. Finally we get the panorama with optimal piecewise rectangular boundary and unnoticeable distortions; see Fig. 3(i).

E. Optimization and Result Generation

For initial image stitching, we first minimize $\Phi_{stitch}(\hat{V})$ defined in Equ. 4, which is global translation invariant. To ensure a unique solution, we fix the first vertex of the first mesh. Note that each energy term is quadratic and the variables are mesh vertices of each image, and therefore the energy function can be efficiently minimized by solving a sparse linear system. Since this stitching step is only used to get the target rectangle and irregular boundary, we do not need to render the stitching result by warping and blending.

After the irregular boundary extraction, we minimize the total energy defined in Equ. 9 which incorporates the regular boundary and line preservation constraints into the stitching framework, thus can simultaneously optimize both stitching and boundary regularity. Since both the added terms are quadratic, Equ. 9 can also be efficiently minimized.

With the optimized vertices of each mesh, we further warp each image by texture mapping and suppress seams between different meshes by multi-band blending [19]. For efficiency, we can also simply apply linear blending, which works well in most cases. Fig. 1(f) is the stitching result constrained by our piecewise rectangular boundary. Compared with the traditional stitching in Fig. 1(b) and existing rectangling method [1] in Fig. 1(g), our method makes a better balance between distortion and boundary regularity, and preserves the panorama content in the rectangular window as much as possible.

F. Discussions

We first discuss the effectiveness of all energy terms in Equ. 9. All the five energy terms are necessary to make satisfactory stitchings, and we analyze their effects by removing each of them at a time in the following ablation study. Fig. 4(a) shows the 3 input images with partial overlaps for stitching, and Fig. 4(b) is the initial stitching result which

Figure 4. Ablation study of our energy terms for rectangular stitching. (a) input images, (b) initial stitching result, (c, d, f, g) stitching results obtained by removing ϕ_a, ϕ_s, ϕ_r and ϕ_l respectively, (e) initial stitching result with a very small weight for ϕ_g, as no result is produced without this term, (h) our stitching result using all energy terms.

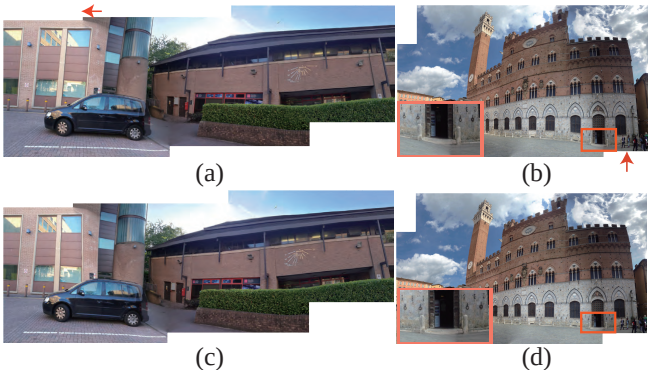


Figure 5. Optimizing piecewise rectangular boundaries by sliding vertices on borders. (a) and (c) are stitching results before and after sliding vertices of a step on the top border in the horizontal direction, (b) and (d) are stitching results before and after sliding in the vertical direction.

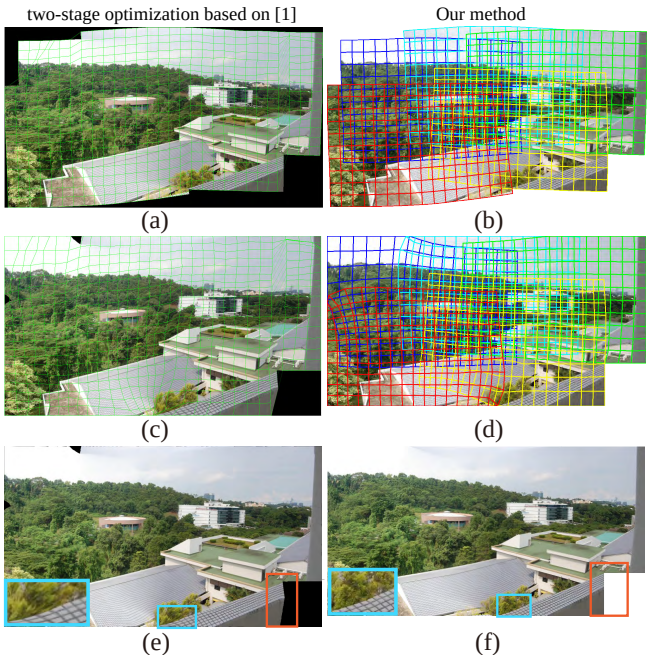


Figure 6. A comparison with a two-stage optimization based on [1]. (a) and (c) are the initial and the second stage stitching results obtained by the two-stage optimization based on [1], (b) and (d) are the initial and global stitching results of our method, (e) and (f) show the comparison of the final stitching results without overlaid grids.

has an irregular boundary and does not preserve straight lines. Figs. 4(c) and (d) are the results obtained by removing the feature alignment term ϕ_a and local shape preservation term ϕ_s, and the stitching results have ghosting and severe distortion. In our experiment, when removing the global similarity term ϕ_g, our method fails to produce valid initial stitching results, because ϕ_g controls the scale and shape of each stitched image. Without this constraint, the images cannot be properly scaled. To show the effects of ϕ_g, we give a very small weight to this term, and find that the scale is largely reduced and the shape is severely distorted; see Fig. 4(e). By removing the boundary constraint term ϕ_r, the stitching result shown in Fig. 4(f) has an irregular boundary. Finally, we remove the line preservation term ϕ_l, and the salient straight lines are not well preserved, as shown in Fig. 4(g). Our stitching result with all the energy terms in Equ. 9 is shown in Fig. 4(h), which is visual pleasing with a rectangular boundary and unnoticeable distortions. The zoom-in views in Fig. 4 further show the effects of each energy term.

We discuss an approach to further optimizing piecewise rectangular boundaries. In this paper, the piecewise rectangular boundary is obtained by averaging the vertex coordinates in each section; see details in Alg. 2. This simple heuristic provides generally plausible piecewise rectangular boundaries. In fact, if we allow the boundaries to be adjusted during the second-stage optimization of our method, the distortions could be further reduced by sliding vertices on the top and bottom borders in the horizontal direction, or sliding vertices on the left and right borders in the vertical direction. For corner vertices, they are treated as on two borders and therefore can move in both horizontal and vertical directions. To obtain an optimal sliding, we try sliding each vertex in all plausible directions. For each direction, we search for the least energy value defined in Equ. 9 with the boundary constraints generated by the slid vertex positions, using the gradient descent method. Finally, we apply the sliding that leads to the least energy value to obtain our optimal piecewise rectangular boundary. As shown in Figs. 5(a) and (c), by sliding the vertices of a step on the top border in the horizontal direction, the slanted building wall above the car can be rectified. Figs. 5(b) and (d) show another example, which rectifies the threshold of a door by sliding in the vertical direction; see the zoom-in views. Although effective, searching an optimal

sliding for each vertex during the energy optimization is very time-consuming, so this optimization is not included in our default pipeline. We plan to develop a more efficient boundary optimization solution in our future work.

Finally, we compare our method with an alternative approach using a two-stage optimization based on [1] to get a piecewise rectangular result. As shown in Fig. 6(a), in the first stage, we perform the initial stitching, and resample a mesh on the stitching result using the backward warping in [1]. In the second stage, we construct an optimization which takes constraints of line and regular boundary preservation from our method, to globally warp the resampled mesh. Similar to [1], due to the irregular boundary of the initial stitching result, the resampled mesh usually contains invalid pixels, which may result in "holes" along borders. In addition, the resampled mesh could not adapt well to the structure of the initial stitched image, which may introduce large distortions near the piecewise rectangular boundaries; see Fig. 6(c). Our initial and final stitching results with overlaid meshes on them are shown in Figs. 6(b) and (d). Compared with the two-stage optimization based on [1], our result has less distortions and more regular boundaries, see Figs. 6(e) and (f), as highlighted in the blue and orange rectangles.

Note that our approach is fundamentally different from previous method [1]. The initial stitching in our method is only used to get regular boundary constraints. Then, the optimal warping is obtained through a global optimization taking all the constraints into account simultaneously, with meshes representing individual input images deforming separately. In comparison the method in [1] takes the pre-stitched image as input, and applies further warping for rectangling. To fit the stitched image with a single mesh while reducing distortions in rectangling, [1] adopts seam carving for the initial mesh generation. However, their single-mesh approach is likely to produce "holes" near boundaries, and the follow-up warping for rectangling introduces further distortions. In contrast, our method optimizes stitching and rectangling in a unified framework, and our iterative refinement process further identifies suitable piecewise rectangular boundaries to spread out distortions, reaching a balance between boundary regularity and distortion.

VI. RESULTS AND EVALUATIONS

In this section, we show a variety of panoramic images generated by our image stitching with regular boundary constraints, comparisons with state-of-the-art methods, and an application to video stitching. Then, we report the runtime performance, quantitative evaluation and user studies to evaluate the effectiveness of our method. Finally, we discuss our limitations. In this paper, we use the datasets provided by Chen et al. [4] for image stitching, and Perazzi et al. [17] for video stitching, along with our own captured images for providing diverse and challenging input. For clearer presentation, we only provide input for examples shot by ourselves.

A. Results and Comparisons

Fig. 7 shows a comparison of our method with [1] for producing rectangular panoramas. Fig. 7(a) is the initial stitching result in the first step of our method. For fair comparison, we also take it as the input to He et al.'s method [1]. In [1], a single mesh is placed on the initial stitched panorama with an irregular boundary, and it is common that the mesh contains regions out of the stitched panorama; see Fig. 7(b). As given in Figs. 7(c) and (d), after the global warping, the rectangling result of [1] may contain "holes", which degrades the quality of the final rectangular panorama. In addition, the method in [1] treats stitching and rectangling as two individual processes, thus cannot well preserve the local and global structures of the scene. Compared with [1], we utilize the meshes (see Fig. 7(e)) from the initial stitching step, thus can avoid the "hole" problem entirely. With these meshes, a global optimization combining stitching and rectangling constraints is constructed, and the final result can not only obtain a regular boundary, but also well preserve the local and global structures; see Figs. 7(f) and (g).

Fig. 8 gives comparisons with state-of-the-art methods in terms of line preservation. For fair comparison, we improve the method in [4] by imposing the line preservation constraint, and the stitching result is shown in Fig. 8(a). Using the stitched panorama from Fig. 8(a), the method in [1] fails to preserve salient straight lines; see the orange rectangles in Fig. 8(b). The reason is that [1] treats stitching and rectangling as two separate steps, thus cannot ensure an optimal solution. Figs. 8(c) and (d) are our initial piecewise rectangular stitching results without and with line preservation, which demonstrate the effectiveness of the line preservation energy term, and the benefit of our simultaneous stitching and rectangling. Fig. 8(e) is our final stitching result after several iterations, which not only preserves straight lines, but also provides rectangular boundaries. Although the straight lines are not strictly preserved, our method makes a good balance between line preservation and distortion.

Fig. 9 presents the results and comparisons of stitching for scenes with large missing contents. We provide two examples to show the effectiveness of our method for such challenging cases. For each example, (a) gives the initial stitching result, which is also used as the input to He et al.'s method [1]. (b) and (c) show the meshes after initial stitching and the extracted irregular boundaries respectively. (d) and (e) are the rectangular stitching results of [1] and our method. Although both of them have distortions, our result is more reasonable and visually pleasing. In addition, due to the drawback of the mesh representation in [1], the warped panoramas may contain "holes". With the optimized piecewise rectangular boundary, the distortion in our result in (f) is unnoticeable, while the content in the rectangular window is preserved as much as possible.

An alternative approach to generating rectangular panoramas is image completion. Fig. 10 compares panorama completion results, using the results of traditional stitching and our method as input. Fig. 10(a) is the stitching result of [4], which has an irregular boundary and a large amount of missing content. By completing the "holes" in Fig. 10(a) using Huang et al.'s method [30], we get the rectangular panorama shown in Fig. 10(c). The close-up windows show that their method [30] is poor in synthesizing semantic content. Fig. 10(b) is the

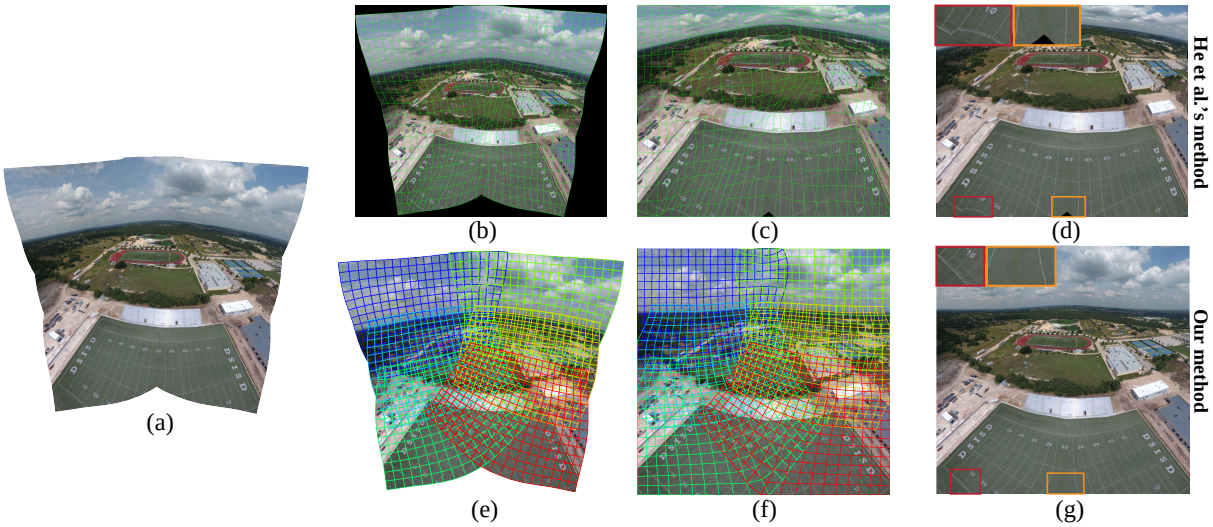


Figure 7. Comparison with He et al.'s [1] method. (a) the initial stitching result, which is also used as the input to the method in [1]. Results of [1]: (b) mesh of the initial stitching, (c) mesh after global warping, (d) rectangular panorama by [1]. Results of our method: (e) meshes of initial stitching, (f) meshes after the global warping, (g) our rectangular panorama.

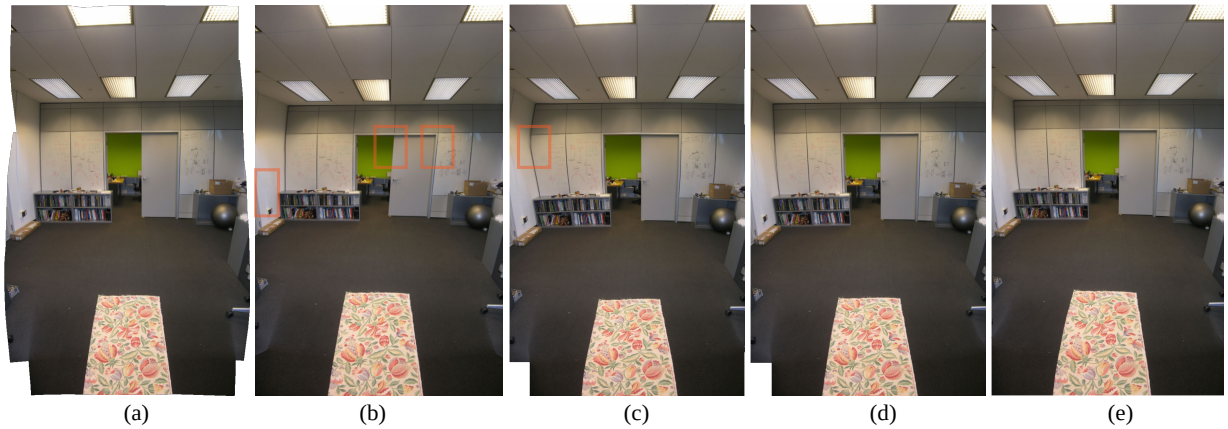


Figure 8. Comparisons with state-of-the-art methods in terms of line preservation. (a) improved stitching result obtained by [4] with the line preservation constraint, (b) rectangling result of [1], (c) and (d) are our piecewise rectangling results in the 1^{st} iteration without and with the line preservation constraint, (e) our final stitching result.

piecewise rectangling result of our method, which preserves regular boundaries while preventing undesirable distortions. Based on our result, it is much easier for image completion to synthesize regular "holes" in the top left corner. The result in Fig. 10(d) shows that the combination of our method and image completion is successful.

Fig. 11 gives the results of selfie panoramas using our method. We first take photos of the panorama view using the back camera on a mobile phone, and then shoot the selfie portrait using the front camera facing the background of the panorama. Fig. 11(a) shows all the input images, including photos for the panorama background and the portrait photo. We can see that the result of [4] in Fig. 11(b) contains irregular boundaries. Fig. 11(c) is the result of our piecewise rectangular stitching method without considering face features, where the portrait is distorted too much. To avoid the distortion on faces, we first detect the face region from the portrait photo, and modify Equ. 2 (shape preservation term) as

$$\phi_s(\hat{V}) = \sum_{i=1}^{N} \sum_{\hat{V}_j^i \in \hat{V}^i} \alpha_j^i \|\hat{V}_j^i - \hat{V}_{j_1}^i - \xi \mathbf{R}(\hat{V}_{j_0}^i - \hat{V}_{j_1}^i)\|^2 \quad (10)$$

where α_j^i refers to the saliency value of vertex $\hat{V}_j^i$. A larger value ($\alpha_j^i = 20$) is specified for the vertices in the face region, and 1 otherwise. As shown in Fig. 11(d), the proposed method with constraints on face shape preservation generates a better selfie panorama.

Fig. 12 gives the results of challenging cases, which contain a large amount of missing content, thus previous panorama rectangling method [1] cannot produce plausible results. Row (a) shows the stitching results of [4], which have irregular boundaries, and Row (b) shows our piecewise rectangling panoramas. According to the possible cropping windows (red and yellow) in Row (a) and (b), Row (c) further gives the final cropped panoramas based on the stitched panoramas obtained by [4] (cyan border) and our method (green border). It is obvi-

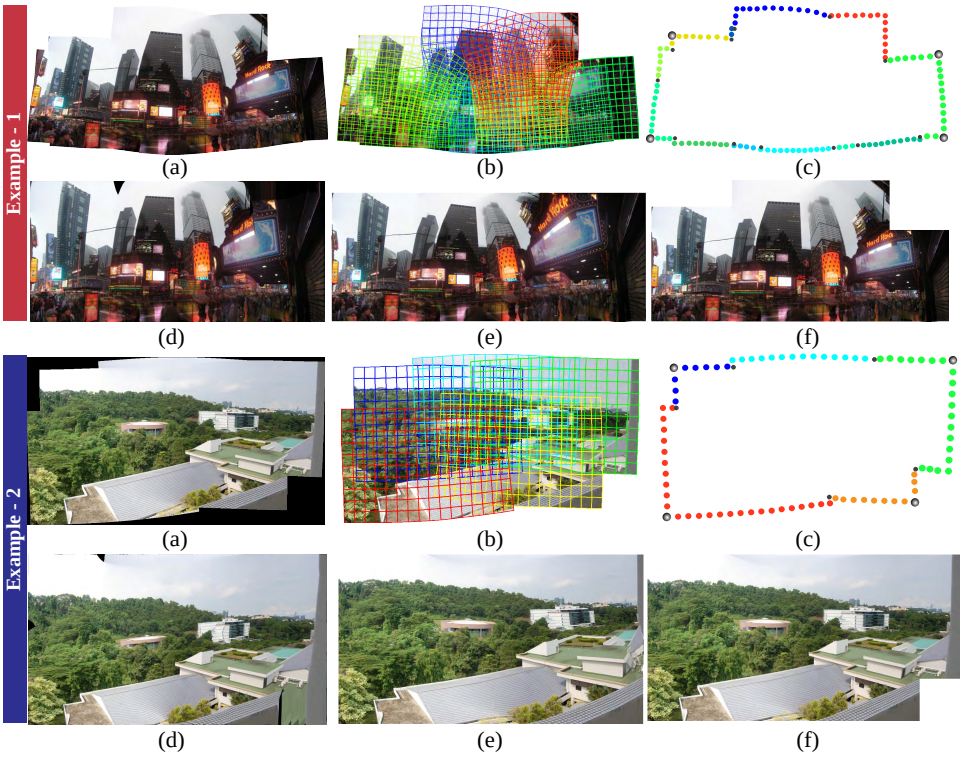


Figure 9. Results and comparisons of stitching with large missing contents. Two examples are presented as follows: (a) initial stitching result with an irregular boundary, (b) warped meshes of initial stitching, (c) irregular boundary extraction, (d) and (e) rectangular stitching results of [1] and our method respectively, (f) our piecewise rectangular stitching result.

Figure 10. Comparison with image completion. (a) stitching result of [4], (b) our piecewise rectangling panorama, (c, d) image completion results obtained by applying [30] to (a) and our stitching result in (b) respectively.

Figure 11. Selfie expansion. (a) input photos, (b) initial stitched image with an irregular boundary, (c) result of our method with a regular boundary, which distorts the human face, (d) result of our method with a regular boundary and face preservation, which can avoid the unwanted face distortion.

ous that, with the generated piecewise rectangular boundaries using our method, the panoramic images can be easily cropped or even completed, and we can obtain panoramic images with more content and unnoticeable distortions by choosing a rectangular window in our results. Thus, compared with traditional stitching with irregular boundaries, our method is effective to improve the visual effects and wide-angle viewing experience of panoramic images.

Fig. 13 presents more results using our method. Compared with the results of the *initial stitching*, our *final results* have regular boundaries and unnoticeable distortions, which can provide better wide-angle viewing experience, and preserve more image content in a rectangular window.

B. Application to Rectangling Video Panoramas

We further apply our method to video stitching. In fact, it is difficult to stitch videos from individual hand-held cameras, and rectangling them is even more challenging. The reason is that the regular boundary in each frame would be different and the temporal coherence is difficult to maintain due to the shaking in each video. Inspired by [17], we aim to produce rectangular panoramic videos from unstructured camera arrays with *fixed* camera configurations. This is more manageable,

Figure 12. Results of challenging cases. (a) Initial stitching results with irregular boundaries, (b) results of our piecewise rectangling stitching, (c) cropped images based on the stitched panoramas of [4] (cyan border) and our results (green border).

as the warping parameters for stitching individual frames are nearly invariant. For temporal coherence, we propose a simple and effective scheme as follows: We first divide a video into several blocks (35 frames per block in our experiments with neighboring blocks having overlaps of 15 frames). For each block, we compute the stitched panorama for the first frame, and the warping parameters are used for the other frames in the block. For the overlapping frames, the warping parameters are a linear combination of neighboring blocks, gradually transitioning from the first set of parameters to the second. Fig. 14 shows two sets of results, and each set shows video panoramas of different frames obtained by [17] and our method. The comparisons show that our method is effective in rectangling video panoramas shot by fixed camera arrays. Please refer to the supplementary video for results of our panoramic videos, and comparisons with [17].

C. Performance

We report the performance of our method on an Intel Core i7 8550U 1.8GHz laptop with 16G RAM. Take Fig. 1 as an example. The input contains 5 images (800×600), and the total processing time is about 3.5 seconds. Specifically, the initial stitching costs 0.81 seconds, which includes feature matching, line detection, energy construction and optimization. Then, the stitching with rectangular boundary constraints costs 0.49 seconds, which includes the irregular boundary extraction, boundary constraint construction and iterative optimization. Finally, with the warped vertices, texture mapping and blending are performed, and the time cost is 2.17 seconds. In the optimization, we only construct the same energy terms once. In addition, since all energy terms are quadratic, the optimization can be solved efficiently. For our iterative optimization in the piecewise rectangling, results in each iteration are similar thus we take the result of the last iteration as initialization, and apply the conjugate gradient method to make the optimization more efficient. For high resolution images, we first downsample each image to a fixed size (0.5 Mega-pixel), and the initial stitching and warping are performed on these downsampled images. Then we upsample the warped vertices through bilinear interpolation, and the final results are obtained by efficient texture mapping and blending of the original high resolution images.

D. Quantitative Evaluation

We first give quantitative evaluation, as shown in Table I. For the 6 examples in Fig. 13, we compare the average feature alignment error and cropping ratio using the initial stitching and our piecewise rectangling method. The average feature alignment error is obtained by averaging the distances of matched features in the overlapped regions, and the cropping ratio is the ratio of cropped image content by a rectangle to the whole stitched panorama. As shown in Table I, the initial stitching results have slightly lower feature alignment errors than our results which is understandable, as we need to make a balance between feature alignment and boundary regularity in the global optimization. The average per-pixel feature alignment errors and the differences are rather small, which implies that our method is comparable to the initial stitching with regard to feature alignment. In terms of cropping ratios, our method has an obvious advantage over the initial stitching, thanks to the piecewise regular boundary constraint. For examples #1, #2, #4, our cropping ratios are 100%, because the target boundary is rectangular. For examples #3, #5, #6, our cropping ratios are close to 100% due to the piecewise rectangular boundary, which helps to retain as much content as possible in the cropping window.

E. User Study

To evaluate the visual quality of our stitching results, we invited 20 participants with ages ranging from 19 to 23, and split them evenly into two groups to take part in two user studies respectively. After watching these panoramas in the supplemental material for 3∼5 minutes, they were asked to give scores for each example. In the 1^{st} user study, the first group of participants were asked to score two indicators (visual quality and wide-angle effects) for panoramas obtained by initial stitching and our rectangling, where all examples in the supplemental material are used. In the 2^{nd} user study, the second group of participants were asked to score another two indicators (wide-angle effects and free of distortion) for panoramas obtained by our direct rectangling and piecewise rectangling. Each indicator was graded by an integer ranging from 1 to 5 (from worst to best). We report the mean value and standard deviation of users' scores for each example and the aggregate data in Table II.

In user study I, the scores of our method are better than those of initial stitching for each example, and the aggregate data further shows that our stitching ($\mu = 4.60, 4.80$) is more satisfactory than the initial stitching ($\mu = 4.20, 4.13$) in terms of visual quality and wide-angle effects, showing that users tend to prefer regular boundaries. The two sample t-test shows that the differences between the scores of "visual quality" and "wide-angle effects" are both significant (p-values: 0.049 and 0.002). Thus, it is obvious that panoramas

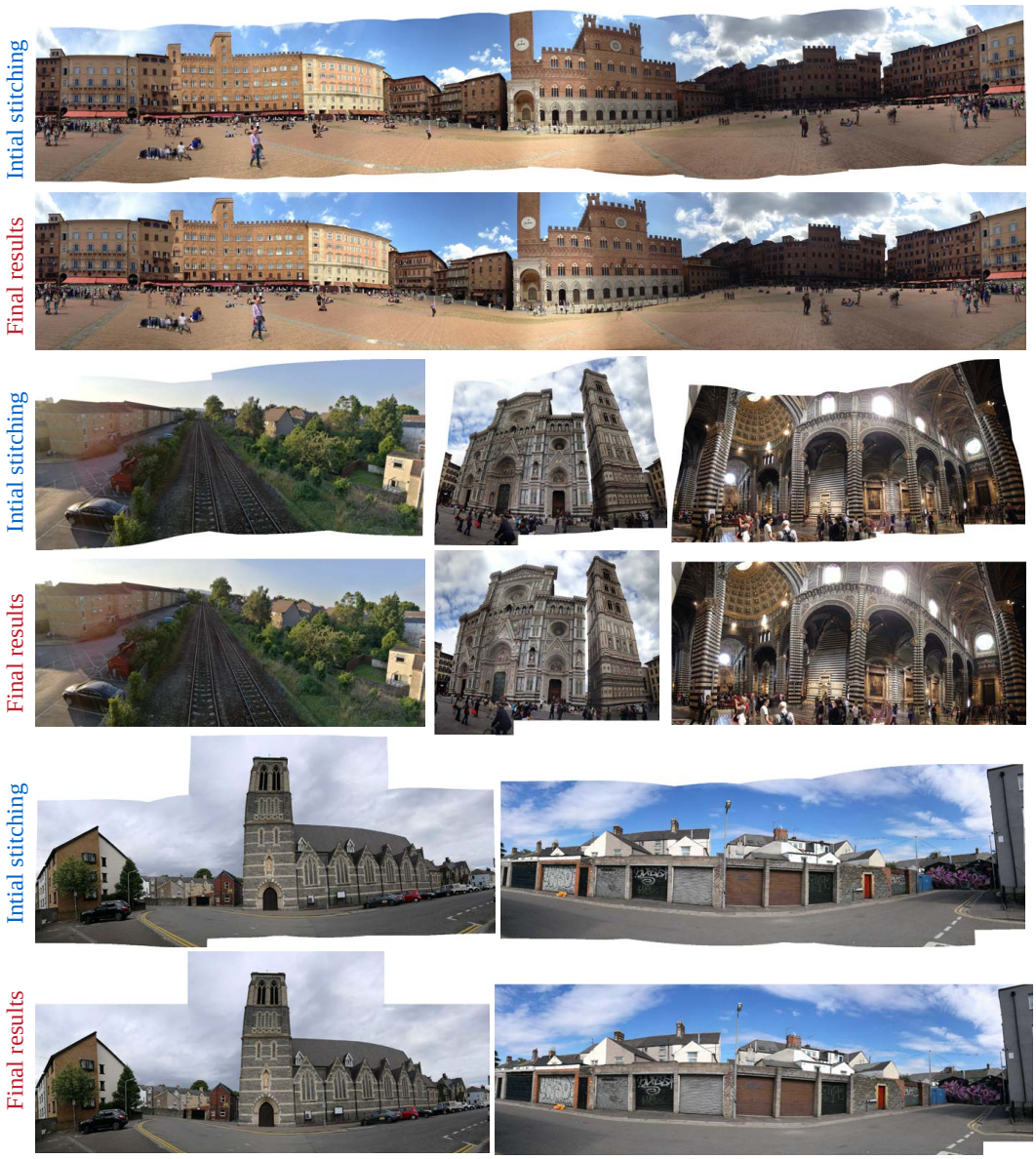


Figure 13. More results. The *initial stitching* results are generated without the regular boundary constraint, and *final results* are obtained by our piecewise rectangular stitching.

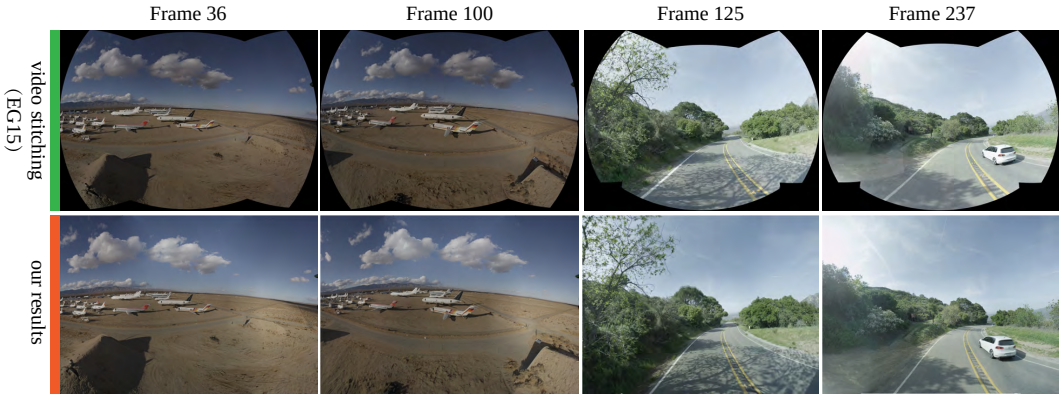


Figure 14. Application to rectangling video panoramas. We give two examples, and for each example show stitched panoramas of 2 different frames obtained by [17] and our method respectively.

Table I
QUANTITATIVE EVALUATIONS OF AVERAGE FEATURE ALIGNMENT ERRORS IN PIXELS AND CROPPING RATIOS

	#1	#2	#3	#4	#5	#6
Inital Stitching	0.17\82.8%	0.21\86.3%	0.16\79.2%	0.26\78.7%	0.28\75.0%	0.18\83.9%
Our method	0.32\100%	0.21\100%	0.19\96.7%	0.29\100%	0.31\85.5%	0.21\92.7%

Table II
RESULTS OF TWO USER STUDIES

	User study I				**User study II**			
	visual quality		wide-angle effects		wide-angle effects		free of distortion	
	Initial stitch	Our result	Initial stitch	Our result	direct rect	piecewise	direct rect	piecewise
#1	4.2 / 0.42	4.4 / 0.52	4.4 / 0.70	4.5 / 0.71	4.7 / 0.48	4.4 / 0.52	3.5 / 0.53	4.3 / 0.67
#2	4.0 / 0.47	4.2 / 0.42	3.9 / 0.57	4.7 / 0.48	4.6 / 0.70	4.6 / 0.52	3.4 / 0.52	4.5 / 0.53
#3	3.9 / 0.57	4.6 / 0.52	3.6 / 0.52	4.6 / 0.84	4.6 / 0.52	4.1 / 0.32	3.5 / 0.71	4.1 / 0.32
#4	4.0 / 0.47	4.5 / 0.53	4.3 / 0.48	4.5 / 0.71	4.7 / 0.48	4.7 / 0.48	3.6 / 0.52	4.4 / 0.52
#5	4.5 / 0.53	4.7 / 0.67	4.2 / 0.42	4.8 / 0.42	4.6 / 0.52	4.3 / 0.48	3.8 / 0.42	4.6 / 0.52
#6	4.0 / 0.47	4.8 / 0.42	3.6 / 0.52	4.7 / 0.67	4.5 / 0.53	4.1 / 0.32	3.8 / 0.63	4.5 / 0.53
#7	4.1 / 0.57	4.4 / 0.70	4.0 / 0.67	4.9 / 0.32	4.6 / 0.52	4.4 / 0.52	4.1 / 0.57	4.4 / 0.52
#8	4.1 / 0.74	4.5 / 0.71	4.3 / 0.48	4.8 / 0.63	4.7 / 0.48	4.5 / 0.53	3.7 / 0.67	4.6 / 0.52
#9	3.8 / 0.42	4.6 / 0.52	3.8 / 0.63	4.8 / 0.42	4.5 / 0.53	4.3 / 0.48	3.3 / 0.67	4.5 / 0.53
#10	3.8 / 0.63	4.3 / 0.48	3.7 / 0.48	4.8 / 0.42	4.5 / 0.71	4.1 / 0.32	3.0 / 0.47	4.3 / 0.48
#11	4.5 / 0.71	4.5 / 0.53	4.4 / 0.52	4.9 / 0.32	4.9 / 0.32	4.5 / 0.53	4.2 / 0.63	4.7 / 0.67
#12	4.6 / 0.52	4.8 / 0.42	4.6 / 0.70	5.0 / 0.00				
#13	4.5 / 0.53	4.9 / 0.32	4.3 / 0.48	5.0 / 0.00				
#14	4.4 / 0.52	4.8 / 0.42	4.3 / 0.67	5.0 / 0.00				
#15	4.7 / 0.48	4.9 / 0.32	4.6 / 0.52	5.0 / 0.00				
aggregate	**4.20 / 0.56**	**4.60 / 0.50**	**4.13 / 0.63**	**4.80 / 0.41**	**4.63 / 0.50**	**4.36 / 0.48**	**3.81 / 0.60**	**4.45 / 0.52**

with piecewise rectangular boundaries are preferred by most users, and provide a better wide-angle effects.

In user study II, the direct and piecewise rectangling refer to our stitching method with rectangular and piecewise rectangular boundaries respectively. Since not all stitching results have piecewise rectangular boundaries, we only used the subset with piecewise rectangular boundaries, as shown in the supplemental material (Figures 1∼11). The mean values of each example and aggregate data show that the direct rectangling performs slightly better in terms of "wide-angle effects", and the piecewise rectangling performs much better in terms of "free of distortion". After performing the two sample t-tests over the comparison of "wide-angle effects" and "free of distortion", we find that the p-values $= 0.29 > 0.05$ and $0.015 < 0.05$, which show that the advantage of the direct rectangling in terms of wide-angle effects is not statistically significant, while the piecewise rectangling method significantly reduces undesirable distortions. Combining the comparisons of the two indicators, the piecewise rectangling makes a good balance, and can obtain panoramas with unnoticeable distortions and acceptable wide-angle effects.

F. Limitations

Due to the free movement of hand-held cameras, panoramic images inevitably have irregular boundaries and missing content. Our piecewise rectangling stitching can effectively rectify these problems by warping-based optimizations with regular boundary constraints. However, there are still some limitations: (1) Similar to most warping-based methods, our method cannot preserve all lines well when there are many lines in local regions. (2) Our method may fail when there are salient structures near the intersection of neighboring meshes. See Fig. 15 for an example, where the zoom-in views show that our piecewise rectangling scheme may introduce unwanted distortion in order to preserve the piecewise rectangular boundary.

(a) (b)

Figure 15. Failure case: when there is strong structure in the intersection of meshes, our method may fail to preserve the structure. (a) initial stitching result with an irregular boundary; (b) our image stitching result with a piecewise rectangular boundary.

VII. CONCLUSION

In this paper, we have proposed an efficient approach for content-preserving stitching with regular boundary constraints, which can generate panoramic images with piecewise rectangular boundaries. Our main contribution is a global optimization which incorporates the regular boundary constraint in the framework of image stitching. Based on the traditional stitching with irregular boundaries, we analyze the warped meshes and extract the outer irregular boundary, and then setup the piecewise rectangular boundary constraint for the optimization to get the final content-preserving stitching result. Experiments, comparisons and an application show that our method is effective and outperforms state-of-the-art methods. Specifically, for panoramic scenes with missing contents, our piecewise rectangling not only regularizes the stitching boundary as much as possible, but also avoids unwanted distortions.

We also conduct quantitative evaluation and two user studies to further demonstrate the advantages of our method.

In the future, we will consider more features to improve the performance of panorama rectangling, such as visual saliency, scene structure, etc. For video stabilization and stitching, the warping-based method may also introduce irregular boundaries. Regularizing the boundary of warped videos can preserve more content in a cropping window and improve the viewing experience. However, for videos shot by freely moving hand-held cameras, it is difficult to define the regular boundary constraints, and maintain the spatial-temporal coherence. We leave these problems as our future work.

Acknowledgements The authors would like to thank all anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (No. 61602402), Zhejiang Province Public Welfare Technology Application Research (No. LGG19F020001), Victoria Early Career Research Excellence Award (No. 224525) and the Royal Society (No. IES\R1\180126).

REFERENCES

[1] K. He, H. Chang, and J. Sun, "Rectangling panoramic images via warping," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 79:1–79:10, 2013.

[2] S. Yen, H. Yeh, and H. Chang, "Progressive completion of a panoramic image," *Multimedia Tools Appl.*, vol. 76, no. 9, pp. 11 603–11 620, 2017.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24:1–24:11, 2009.

[4] Y. Chen and Y. Chuang, "Natural image stitching with the global similarity prior," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, Proceedings, Part V*, 2016, pp. 186–201.

[5] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 1, 2006.

[6] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[7] J. Gao, S. J. Kim, and M. S. Brown, "Constructing image panoramas using dual-homography warping," in *2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, June 20-25*, 2011, pp. 49–56.

[8] J. Zaragoza, T. Chin, Q. Tran, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving DLT," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1285–1298, 2014.

[9] C. Lin, S. Pankanti, K. N. Ramamurthy, and A. Y. Aravkin, "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, pp. 1155–1163.

[10] S. Li, L. Yuan, J. Sun, and L. Quan, "Dual-feature warping-based motion model estimation," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13*, 2015, pp. 4283–4291.

[11] C. Chang, Y. Sato, and Y. Chuang, "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, pp. 3254–3261.

[12] F. Zhang and F. Liu, "Parallax-tolerant image stitching," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28*, 2014, pp. 3262–3269.

[13] K. Lin, N. Jiang, L. Cheong, M. N. Do, and J. Lu, "SEAGULL: seam-guided local alignment for parallax-tolerant image stitching," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, Proceedings, Part III*, 2016, pp. 370–385.

[14] J. Li, Z. Wang, S. Lai, Y. Zhai, and M. Zhang, "Parallax-tolerant image stitching based on robust elastic warping," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1672–1687, 2018.

[15] B. He and S. Yu, "Parallax-robust surveillance video stitching," *Sensors*, vol. 16, no. 1, p. 7, 2016.

[16] X. Yin, W. Li, B. Wang, Y. Liu, and M. Zhang, "A novel video stitching method for multi-camera surveillance systems," *KSII Trans. Internet and Information Systems*, vol. 8, no. 10, pp. 3538–3556, 2014.

[17] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. H. Gross, "Panoramic video from unstructured camera arrays," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 57–68, 2015.

[18] D. Anguelov, C. Dulong, D. Filip, C. Früh, S. Lafon, R. Lyon, A. S. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *IEEE Computer*, vol. 43, no. 6, pp. 32–38, 2010.

[19] Z. Zhu, J. Lu, M. Wang, S. Zhang, R. R. Martin, H. Liu, and S. Hu, "A comparative study of algorithms for realtime panoramic video blending," *IEEE Trans. Image Processing*, vol. 27, no. 6, pp. 2952–2965, 2018.

[20] X. Meng, W. Wang, and B. Leong, "Skystitch: A cooperative multi-uav-based real-time video surveillance system with stitching," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26 - 30*, 2015, pp. 261–270.

[21] M. A. El-Saban, M. Ezz, A. Kaheel, and M. Refaat, "Improved optimal seam selection blending for fast video stitching of videos captured from freely moving devices," in *18th IEEE International Conference on Image Processing, ICIP 2011, Brussels, Belgium, September 11-14*, 2011, pp. 1481–1484.

[22] K. Lin, S. Liu, L. Cheong, and B. Zeng, "Seamless video stitching from hand-held camera inputs," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 479–487, 2016.

[23] H. Guo, S. Liu, T. He, S. Zhu, B. Zeng, and M. Gabbouj, "Joint video stitching and stabilization from moving cameras," *IEEE Trans. Image Processing*, vol. 25, no. 11, pp. 5491–5503, 2016.

[24] Y. Nie, T. Su, Z. Zhang, H. Sun, and G. Li, "Dynamic video stitching via shakiness removing," *IEEE Trans. Image Processing*, vol. 27, no. 1, pp. 164–178, 2018.

[25] R. G. von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, 2010.

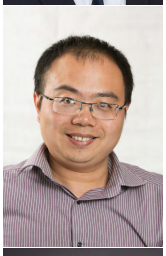
[26] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 78:1–78:10, 2013.

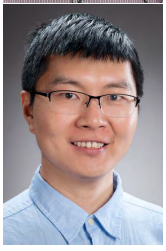
[27] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1134–1141, 2005.

[28] T. Igarashi and Y. Igarashi, "Implementing as-rigid-as-possible shape manipulation and surface flattening," *J. Graphics, GPU, & Game Tools*, vol. 14, no. 1, pp. 17–30, 2009.

[29] F. Martínez, A. J. Rueda, and F. R. Feito, "A new algorithm for computing boolean operations on polygons," *Computers & Geosciences*, vol. 35, no. 6, pp. 1177–1185, 2009.

[30] J. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 129:1–129:10, 2014.





Yun Zhang is an associate professor at Communication University of Zhejiang. 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. From Feb. to Aug. 2018, he was a visiting scholar at Cardiff University. His research interests include Computer Graphics, Image and Video Editing, Computer Vision and Virtual Reality.

Yu-Kun Lai received his bachelor's and Ph.D. degrees in computer science from Tsinghua University, China, in 2003 and 2008, respectively. He is currently a Reader at the School of Computer Science & Informatics, Cardiff University. His research interests include Computer Graphics, Geometry Processing, Computer Vision and Image Processing. He is on the editorial boards of Computer Graphics Forum and The Visual Computer.

Fang-Lue Zhang is a lecturer at Victoria University of Wellington. He received his doctoral degree from Tsinghua University in 2015 and bachelor degree from Zhejiang University in 2009. He joined Victoria Graphics Group in 2017. His research interests include image and video editing, computer graphics and vision. He is a member of ACM and IEEE. He received Victoria Early Career Research Excellence Award in 2019.