Multi-Viewpoint Panorama Construction with Wide-Baseline Images

Guofeng Zhang, *Member, IEEE,* Yi He, Weifeng Chen, Jiaya Jia, *Senior Member, IEEE,* and Hujun Bao, *Member, IEEE*

***Abstract*—We present a novel image stitching approach, which can produce visually plausible panoramic images with input taken from different viewpoints. Unlike previous methods, our approach allows a wide baseline between images and non-planar scene structures. Instead of 3D reconstruction, we design a mesh-based framework to optimize alignment and regularity in 2D. By solving a global objective function consisting of alignment and a set of other prior constraints, we can construct a visually plausible panorama which is locally as-perspective-as-possible but nearly orthogonal on global aspect. We also improve solutions for seamless composition and achieve decent performance on misaligned area. Experimental results on challenging data demonstrate the effectiveness of the proposed method.**

***Index Terms*—image stitching, multi-view point panorama, image alignment, wide-baseline, mesh optimization**

I. Introduction

With the prevalence of digital cameras and smart phones, sharing captured and collected images has become a popular way to entertain people and brand ourselves. Since a normal camera has a limited field of view, many applications have been developed to provide a panoramic shooting mode, where the users can capture images under guidance to generate a panorama.

Although panoramic stitching from a single viewpoint has been maturely studied, it is still difficult to produce visually pleasing results from a set of images under wide baselines, which is however an inevitable component in many applications. For example, to produce a large field-of-view image for an object close to the camera by circling the camera around it is much more difficult than handling camera pan with a fixed center. Images captured from multiple cameras put in different locations raise similar challenges. All these applications require panorama techniques considering non-ignorable baselines among different cameras.

Many previous image stitching methods require that the camera rotates with a fixed center [1], [2], [3], or the scene is planar [4]. Violation of these assumptions may lead to severe problems. Recent methods [5], [6], [7] relaxed these constraints by dual-homography [5], or smoothly varying affine/homography [6], [7]. They work well for images with moderate parallax, but are still problematic in the wide-baseline condition, as demonstrated in Figure 11.

In this paper, we propose a stitching approach for wide-baseline images. Our main contribution is a mesh-based framework combining kinds of terms to optimize the image alignment. Especially, a novel scale preserving term is introduced which can make the alignment nearly orthogonal on global aspect but still allows local perspective correction. In addition, a new seam-cut model is proposed to reduce visual artifacts caused by misalignment that is difficult to be handled by traditional graph cuts [8]. Figure 1 shows a challenging urban scene example where 14 images are captured in different viewpoints with large baselines. Our generated panorama is visually compelling.

G. Zhang, Y. He, W. Chen and H. Bao are with the State Key Lab of CAD&CG, Zijingang Campus, Zhejiang University, Hangzhou, 310058, P.R. China. E-mail: {zhangguofeng, bao}@cad.zju.edu.cn; heyi@zjucadcg.cn; wfchen@umich.edu. G. Zhang is also affiliated with Innovation Joint Research Center for Cyber-Physical-Society System.

Jiaya Jia is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. E-mail: leojia@cse.cuhk.edu.hk.

II. Related Work

A. 3D Reconstruction

Given the dense depth maps of a scene, the panoramic view can be generated by 3D modeling with texture mapping. However, multi-view stereo techniques [9], [10], [11], [12], [13] are constrained by a series of conditions including camera motion and Lambertian surface assumption. It is difficult to produce perfect 3D models in many cases, especially when there are only a few images. In the application of video stabilization where the baseline between source and target images is small, the reconstructed sparse 3D points may be enough for content-preserving warp [14], [15]. But this does not work that well for wide-baseline images with complex structure.

Agarwala [4] constructed multi-viewpoint panoramas for approximately planar scenes. Structure-from-motion was used to recover camera poses and sparse 3D points. Then a dominant plane was selected manually so that the input images can be projected for stitching. In contrast, our method is an automatic approach without requiring the recovery of camera motion and 3D structures.

B. Mesh Optimization

Mesh optimization and manipulation perform well on image retargeting [16], [17], resizing [18], [19], [20] rectangling [21], and video stabilization [14], [15]. These methods use different global energy functions depending on their targets, and solve for the optimal mesh configuration. A similarity constraint was usually used for regularization, which however is not

(a) (b)

Fig. 1: Automatically constructed urban panorama with 14 wide-baseline images. (a) Input images. (b) The reconstructed panorama.

appropriate for perspective projection. For instance, parallel lines remain parallel under affine transformation but not perspective transformation. In contrast, our proposed straightness constraint does not involve the parallelism constraint and is more appropriate for perspective transformation.

C. Panoramic Mosaics

Panoramic image techniques [1], [3] work best for camera that undergoes only rotation. For other types of camera motion, misalignment artifacts could be introduced. Although seam optimization [8] and gradient domain blending techniques [22] can be used, they do not solve the problem by nature. Recently, Gao [5] proposed a dual-homography model to align overlapping images, where the warping can be modeled by a linear interpolation of two homographies. It is still insufficient for complex scenes.

Lin [6] employed a smoothly varying affine model to stitch images with parallax. Zaragoza [7] extended this method to more general scenes with smoothly varying homographies. They use feature correspondences with adaptive weights to estimate locally coherent homographies. Both these methods assumed that a global affine/homography can approximately represent image transformation and the local deviation is minor. This assumption is violated on wide-baseline images.

Different from the above methods, Zhang [23] focused on improving seam optimization. This method randomly picks some subsets of the correspondences and align only local parts of images. This process is repeated for optimizing seams to generate multiple candidate panoramas. The best panorama can be measured and chosen. Chang [24] proposed combining the projective and similarity transformation to reduce distortions. This technique can be combined with that of [7]. But it still faces challenges in handling wide-baseline images. All above methods select one image as the reference view, and warp other images to it, which may cause large perspective distortion for photographing a long scene.

D. Seamless Composition

Graph cuts [8] stitches images by optimizing a Markov Random Field (MRF), which has been widely used. Agarwala [4] proposed incorporating 3D information to the data term, while several other methods used a binary function depending on the visibility of pixels. The smoothness term penalizes the color differences on the seams. In our wide-baseline cases, misaligned pixels might coincidentally have similar colors, which make it difficult to detect bad seams via color. In order to address this problem, we propose to combine alignment errors and colors in a new way.

III. Feature Matching with Outliers Rejection

Like most previous approaches [1], [3], [7], we also use SIFT [25] to find correspondences. For most challenging data, we can use ASIFT [26] to obtain more feature matches.

Estimating epipolar geometry with RANSAC [27] can reject mismatched correspondences. We find that a few outliers along the epipolar line are usually difficult to be eliminated, which may influence stitching. If the scene is planar, global homography estimation can also reject outliers. The method of [7] extends this method by enlarging the error threshold to accept feature correspondences from different planes. It works well only for small-baseline images. Our method is different from them – we use local homographies to more robustly remove outliers, which works even in wide-baseline images.

For each feature point, we assume there is a plane in its local area, so that all of its neighbors are approximately on the same plane. For two arbitrary feature points, we consider they are neighbors if their distance is smaller than R. We use DLT [28] to fit a homography for all the neighboring feature correspondences, and compute the residual error. If the error is less than a threshold γ, we mark it as an inlier. In our experiments, we generally set $R = 50$ and $\gamma = 5$.

The procedure is shown in Algorithm 1. For image pair (I_i, I_j), we first define the neighboring sets for each feature point in I_i and estimate the corresponding homographies. Each correspondence (p', q') is verified with several homographies since it can be included in different neighboring sets. As long as it fits one homography, this correspondence is recognized as an inlier. After enumerating all feature points in I_i, we obtain the inlier set S_1 for (I_i, I_j). Then we swap I_i and I_j to get another inlier set S_2 by Algorithm 1. The final inlier set is $S_1 \cap S_2$.

As shown in Figure 1, the urban scene contains two major planes. With our local homographies verification, the outliers can be robustly rejected. Figure 2 gives a comparison with the methods using global and local homographies respectively. As shown in (b), the traditional RANSAC approach with global

Algorithm 1 Outliers Rejection with Local Homographies

procedure VERIFY(I_{src}, I_{dst})
  $S_{inlier} := \Phi$
  for all $p \in I_{src}$ **do**
    Solve H for $\{(p', q')|p' \in N(p)\}$
    for all $\{(p', q')|p' \in N(p)\}$ **do**
      if $|H(p') - q'|^2 < \gamma$ **then**
        $S_{inlier} \leftarrow (p', q')$
      end if
    end for
  end for
  return S_{inlier}
end procedure

homography eliminates many feature correspondences from the desktop. In contrast, our method faithfully preserves the inliers on the desktop. The stitching result of this example is shown in Figure 3.

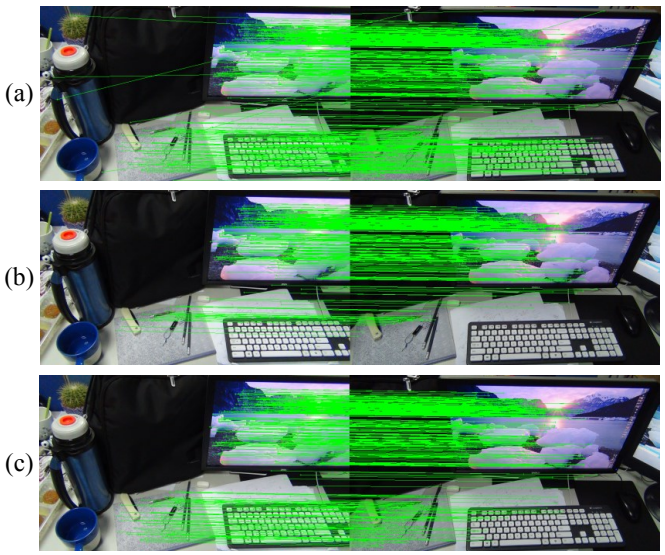


Fig. 2: Outliers rejection comparison. (a) Matched feature correspondences by SIFT. (b) Recognized inliers by RANSAC with global homography. (c) Recognized inliers by our approach.

IV. Energy Function of Image Stitching

After feature matching, we build regular mesh grids for all images, and index the control vertices from 1 to m. Then we put their coordinates into a $2m$ dimensional vector

$$V = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots & x_m & y_m \end{bmatrix}^\top,$$

and optimize V to align the corresponding feature points. Once V is solved, the images can be warped to a reference plane to generate the stitched panorama.

The energy function is defined as

$$E(V) = E_A(V) + \lambda_R E_R(V) + \lambda_S E_S(V) + E_X(V), \quad (1)$$

where $E_A(V)$ is the alignment term, enforcing the corresponding feature points to be warped to the same positions. $E_R(V)$ is the regularization term, encouraging the neighboring vertices to take similar transformation. $E_S(V)$ is the scale term, preventing large image scale change. λ_R and λ_S are the weights, which are usually set to 1 in our experiments. Optionally, $E_X(V)$ is an extra constraint used in cases for stronger regularization. The optimal vertex coordinates $V_{opt} = \arg\min_V E(V)$ are used to manipulate the images for generating a panorama.

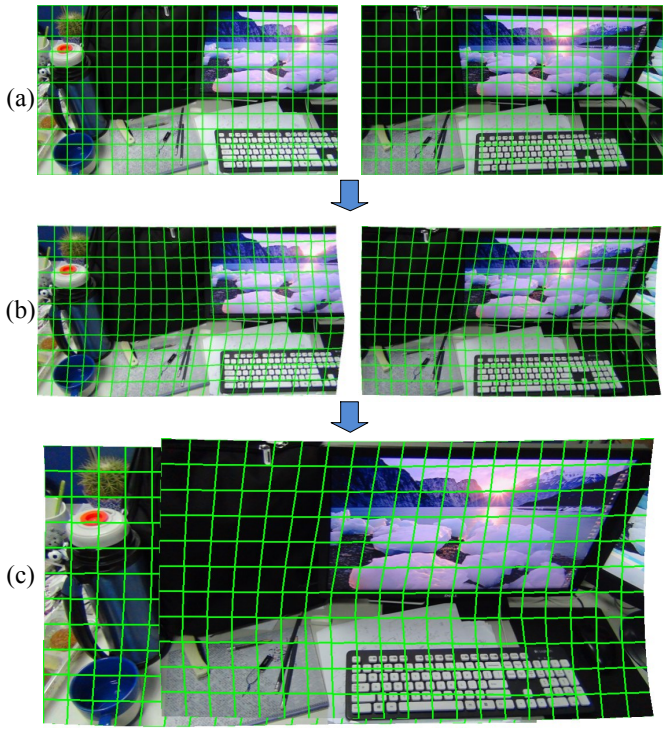


Fig. 3: Mesh based framework. (a) Regular mesh grids on input images. (b) Manipulating images via optimized mesh vertices. (c) Warping the images to a common plane.

In the example of Figure 3, the screen and desktop form two different planes, and our mesh-based model approximately fits two homographies for them, as shown in (b). Compared to the single or dual homography representation, our multi-homography model has more degrees of freedom and can represent warping of a general smooth scene.

In addition, traditional image stitching methods [5], [6], [7] select one input image as the reference view and warp other images to it, which may cause perspective distortion for long sequences. Similar to [4], we project all images onto a common plane. The generated panorama is nearly orthogonal on global aspect while the local perspective property is still preserved. In order to achieve this goal, we contribute a novel scale preservation term, which can constrain the image size to be a constant for ensuring nearly orthogonal but still allows local perspective transformation. Our Laplacian regularization term also can better correct local perspective distortion than the similarity term used in [14], [19], [21].

A. Feature Alignment

We represent each feature point as a weighted sum of their four enclosing control vertices, and minimize the alignment error of the warped points over all features. Similar to [15],

we use bilinear interpolation to calculate the weights on the original meshes, which is equivalent to the barycenter representation.

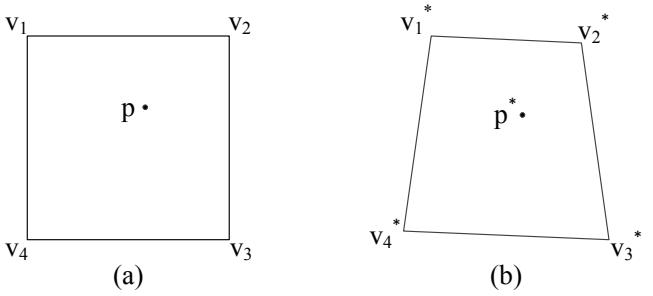


Fig. 4: Feature point interpolation. (a) Original mesh grid and a feature point p. (b) Warped vertices and feature point p^*.

As illustrated in Figure 4, there is a feature point p inside the grid whose four vertices are denoted as v_1, v_2, v_3, and v_4. The interpolation weights are computed as

$$\begin{aligned} w_1 &= (v_{3x} - p_x)(v_{3y} - p_y), \\ w_2 &= (p_x - v_{4x})(v_{4y} - p_y), \\ w_3 &= (p_x - v_{1x})(p_y - v_{1y}), \\ w_4 &= (v_{2x} - p_x)(p_y - v_{2y}). \end{aligned} \tag{2}$$

We assume that the interpolation weights are fixed after warping the grids (i.e. assuming affine tranformation for each grid). As demonstrated in our supplementary document[1], this assumption is reasonable in a typical panorama scenario especially when the mesh grid size is small. So we define the alignment term as

$$\begin{aligned} E_A(V) &= \sum_{(p_i,p_j)\in C} \frac{1}{N_{p_i,p_j}} \|p_i^* - q_i^*\|^2 \\ &= \sum_{(p_i,p_j)\in C} \frac{1}{N_{p_i,p_j}} \|W_i V - W_j V\|^2, \end{aligned} \tag{3}$$

where C is a set containing feature correspondences of all image pairs. p_i^* and q_i^* are warped positions of two matched points, whose coordinates can be represented as a weighted sum of the mesh vertices in V. W_i is a sparse $m \times 2$ weight matrix of p_i, formed as

$$\begin{bmatrix} \cdots & w_1 & 0 & \cdots & w_2 & 0 & \cdots & w_3 & 0 & \cdots & w_4 & 0 & \cdots \\ \cdots & 0 & w_1 & \cdots & 0 & w_2 & \cdots & 0 & w_3 & \cdots & 0 & w_4 & \cdots \end{bmatrix},$$

where each column consists of 0 except the four positive values that sum to one. $W_i^\top V$ provides a 2D vector with x and y coordinates of p_i^*. N_{p_i,p_j} is the total number of feature points in the two cells containing p_i and p_j respectively. It is used to normalize the alignment error for different regions and prevent a few grids with rich features from dominating the alignment term. It should be noted that even each grid performs affine transformation, the whole mesh grids can perform perspective alignment well as long as the feature correspondences are accurate.

B. Regularization

The alignment term only affects those grids with feature points. We need a regularization term to propagate the transformation to other regions. In [14], [19], [21], they defined a similarity term to preserve the shape for each of the mesh grids, and achieved good regularization effects. However, this similarity term does not work well in our case. For panoramic stitching, it is not reasonable to enforce similarity constraint, since perspective correction is generally necessary for best alignment. On the other hand, with the local planar assumption, we do prefer those meshes that warp local regions with similar homographies.

As shown in Figure 6, for each vertex v, we can estimate a local homography H with its four neighbors v_1, v_2, v_3, v_4 and their warped positions $v_1^*, v_2^*, v_3^*, v_4^*$. Then we can apply H on the vertex v to get the regular position v'. We would like to minimize the Euclidean distance between v' and the real position v^*.

Again, affine transformation was used to approximately constrain the coherence. So v' was replaced by Av where A is the affine transformation fitting the warping of v_1, v_2, v_3, v_4. With the linearity of affine transformations, we can directly represent v' as a weighted sum of the neighbors instead of solving A. Since we divide the mesh grids evenly, the weights can be set equal. Thus v' is simply the average of $v_1^*, v_2^*, v_3^*, v_4^*$, which leads a Laplacian operator on the mesh grids, i.e. $(v_1^* + v_2^* + v_3^* + v_4^*) - 4v' = 0$. Therefore, our regularization term is defined as

$$E_R(V) = \sum_v \|W_v V - \frac{1}{|N_v|} \sum_{v_i \in N_v} W_{v_i} V\|^2, \tag{4}$$

where N_v is a 4-connected neighboring set of vertex v. For the vertices on the image boundary, we only use 2 horizontal or vertical neighbors. W_v and W_{v_i} are index matrices defined as

$$\begin{bmatrix} 0 & \ldots & 1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & \ldots & 0 \end{bmatrix},$$

which extract x and y coordinates of v, v_i from V, respectively. As a result, $E_R(V)$ enforces the neighboring vertices to favor the same affine transformation.

As shown in Figure 5, given two wide-baseline images, our approach can achieve much better alignment than content-preserving warping [14], which preserves the shape of mesh grids. In order to measure the alignment quality, we average the warped images to obtain the composite image. The area with large alignment error is blurry. For fair comparison, we set the first image as the reference image and warp the other one to it. As shown in Figure 5(c), although the alignment result of [7] is reasonable, there are still misalignment and distortion artifacts due to the insufficient Gaussian smoothing weights.

C. Scale Preservation

The alignment and regularization terms actually form a linear system as $AV = 0$, where $V = 0$ always satisfies this

[1] http://www.cad.zju.edu.cn/home/gfzhang/projects/panorama/pano-supple.pdf

Fig. 5: Panorama construction with 2 wide baseline images. (a) Input images. (b) The averaging result by content-preserving warps [14]. (c) The averaging result by APAP [7]. (d) The averaging result by our approach.

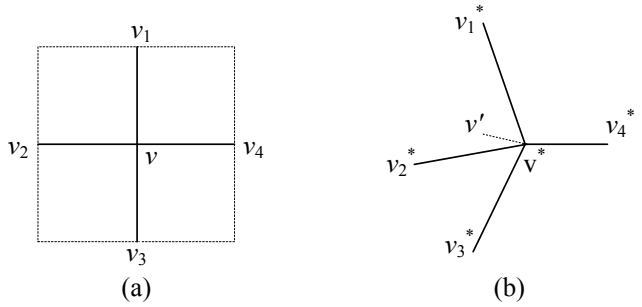


Fig. 6: Regularization Term. (a) Original Vertices. (b) Warped Vertices.

linear system. In order to avoid this degeneration problem, methods [5], [7] were proposed to select one image as the reference view and project other images onto it. This strategy works if there are only $2 \sim 3$ images, as shown in Figure 5. With the increasing field-of-view, the images far from the reference one may be significantly distorted in order to reduce the alignment error. Figure 7 shows a stitching result with 14 images, where the first one was set as the reference view. The right most images are obviously scaled down.

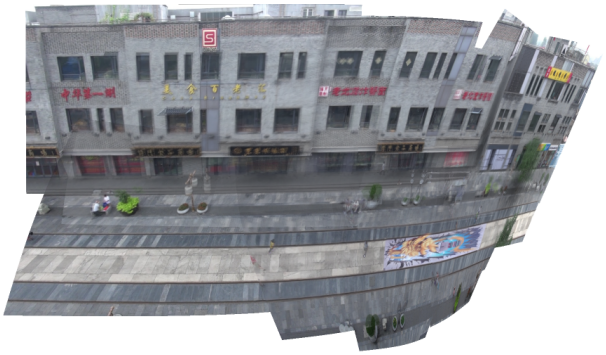


Fig. 7: Image stitching result by fixing the first image.

To address this problem, the scale constraint should be applied to all images equally. The scale of an image can be measured by its four edges, since the inner area can be interpolated once the edge scales are decided. We estimate a scaling factor for each image according to the feature points. Specifically, for a matched image pair (I_i, I_j), we build a convex polygon P_i on the feature points from I_i and find its corresponding polygon P_j on image I_j. Then the relative scaling factor γ_{ij} is defined using the ratio of polygon perimeters

$$\gamma_{ij} = \frac{e_{P_i}}{e_{P_j}},$$

where e_{P_i} and e_{P_j} are the perimeters of P_i and P_j respectively. We estimate the absolute scaling factor for each image by solving

$$\arg\min_{s} \sum_{(i,j)\in C_I} |\gamma_{ij} s_j - s_i|^2, \quad s.t. \sum_{i\in I} s_i = N_I,$$

where N_I denotes the number of images and C_I is the set of matched image pairs. The obtained scaling factors agree with the relative ratios while the sum of all scales is preserved.

With the scaling factors, we add a constraint for each image. The scale preserving term can be defined as

$$E_S(V) = \sum_{I_i\in I} |S(I_i^*) - s_i S(I_i)|^2, \quad S(I_i) = \begin{bmatrix} \|B_t\| + \|B_b\| \\ \|B_l\| + \|B_r\| \end{bmatrix}, \tag{5}$$

where I_i^* and I_i are the i-th warped and original images respectively. S is a scale measurement for images defined as a 2D vector. B_t, B_b, B_l and B_r are the top, bottom, left and right edges of image I_i and can be represented with the vertices V. For example, the length of edge B_t can be represented as a nonlinear function of V:

$$\|B_t\| = \sqrt{(W_{tl}V - W_{tr}V)^\top (W_{tl}V - W_{tr}V)},$$

where W_{tl} and W_{tr} are index matrices for the top-left and top-right vertices. $\|B_t\|$, $\|B_l\|$ and $\|B_r\|$ are similarly defined.

We define $S(I_i)$ as a 2D vector, because the vertical and horizontal edges should be considered independently, which is better than summing vertical and horizontal edges to only consider the total length. If we only consider the total length of four image edges, the horizontal image edges may become very small and vertical image edges may become very large, resulting in a long and narrow image. If we directly constrain each image edge to be constant, the freedom degree will be too small to correct perspective distortion. In contrast, preserving vertical and horizontal scales independently can

allow more freedom degree to correct perspective distortion and simultaneously avoid unatural distortion.

By constraining the image size to be nearly constant, the images are nearly orthogonally projected to a reference plane. In addition, our feature alignment and regularization terms favor as-perspective-as-possible alignment. Our scale preserving term $E_S(V)$ not only can constrain the image size but also allows perspective correction in local regions. Therefore, by combining these terms together, our method can construct a visually plausible panorama which is locally as-perspective-as-possible but nearly orthogonal on global aspect, as shown in Figure 8 (a). Since $E_S(V)$ is nonlinear, we propose an iterative approach to optimize it, which will be described in Section V.

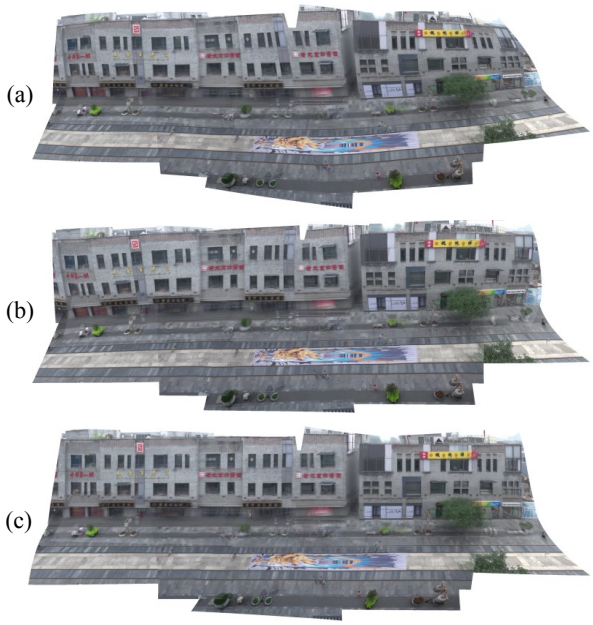


Fig. 8: Image stitching with different prior constraints. (a) The averaging result by solving $E(V) = \lambda_A E_A(V) + \lambda_R E_R(V) + \lambda_S E_S(V)$. (b) The averaging result by further incorporating the line preserving term. (c) The averaging result by further incorporating the orientation term.

D. Extra Constraints

Our mesh-based model allows for incorporating extra constraints conveniently. For special cases of urban scenes and closed-loop camera motion, we incorporate one or multiple of the following priors to achieve even better results.

a) Line Preserving Constraint: In order to further reduce the distortion, we introduce a line preserving term, which prevents the line segments from bending. We use the method of [29] to automatically extract line segments, and denote the set of lines as L. For a line segment l in L, we evenly sample a few points $\{p_1, p_2, ..., p_n\}$ so that each grid contains at least one point. To make l straight, we claim that all segments on the line are with the same direction, leading to the energy function

$$E_{line}(V) = \lambda_{line} \sum_{l \in L} \sum_{i=1}^{n-1} ([a_l, b_l]_\perp \cdot (W_{p_i} V - W_{p_{i+1}} V)), \quad (6)$$

where $[a_l, b_l]_\perp$ is the orthogonal direct of l and the coordinates of p_i can be represented by a linear interpolation of its enclosing vertices, as in Eq. (3). λ_{line} is a weight and usually set to 1 in our experiments. We update a_l and b_l iteratively during optimization. Figure 9 shows the detected line segments. Incorporating the line preserving term, the stitching result is improved as shown in Figure 8(b).

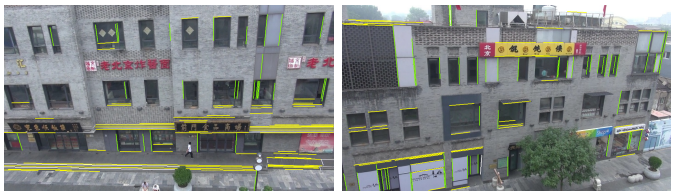


Fig. 9: Detected line segments.

b) Orientation Constraint: Urban scenes generally contain a few vanishing lines, which are either vertical or horizontal. While enforcing their straightness, we also constrain the orientation at the same time. After detecting line segments, we can divide them into vertical and horizontal categories L_V and L_H (colored in green and yellow respectively in Figure 9). We use RANSAC [27] to estimate the vanishing points and eliminate outliers. The lines intersecting the same vanishing point correspond to either the horizontal lines or vertical lines. By assuming that the images are taken horizontally, we can compute their angles with the horizontal direction, and recognize the lines with smaller angles as horizontal lines. Denoting p and q as two end points of such a line segment, they should have the same x or y coordinates depending on their category. The orientation term is defined as

$$E_O(V) = \lambda_O \Big(\sum_{l \in L_V} |(W_{p_x} - W_{q_x})V|^2 + \sum_{l \in L_H} |(W_{p_y} - W_{q_y})V|^2 \Big), \quad (7)$$

where W_{p_x} and W_{p_y} are the interpolation weight vectors of p in x and y coordinates respectively. λ_O is a weight and set to 1 in our experiments. Mesh V warps p to position $(W_{p_x}V, W_{p_y}V)$. Figure 8(c) shows the result with the orientation term.

c) Loop Closure Constraint: For capturing a full panoramic view, the camera needs to rotate 360°so that the first image has overlapping content with the last one. However, the alignment term can not be applied directly to the tail images because the feature points of the first and last images are aligned with an unknown offset.

In practice we align edges instead of points, so that the unknown offset can be eliminated. We propose

$$E_{loop}(V) = \lambda_L \sum_{(e_i, e_j) \in C_e} \|e_i - e_j\|^2, \quad (8)$$

where C_e is a set of corresponding edges matched between the first and last images. $e_i = p_i - q_i = (W_{p_i} - W_{q_i})V$ and $e_j = p_j - q_j = (W_{p_j} - W_{q_j})V$. p_i and q_i are two ending points of edge e_i. W_{p_i} and W_{q_i} are the weight matrices of p_i and q_i respectively. λ_L is a weight and set to 1000 to enforce hard constraint if there is a loop closure.

If we connect each point pair for n points, there will be n^2 edges. For reducing the complexity, we randomly shuffle the feature points and connect the neighboring ones. Figure 14 shows an example for generating a 360°panoramic mosaic.

With the loop closure constraint, the left and right most images become more consistent, so that they can be aligned well if we project them onto a cylinder.

V. Optimization

Since the energy function defined in (1) is not quadratic, we propose an iterative approach to optimize it. Specifically, only the scale and line preservation terms (i.e. E_S and E_{line}) are non-quadratic. We replace these terms by their linear approximation in each step, and update the result iteratively.

A. Linear Approximation

As defined in Eq. (5), E_S is non-linear because the scale function S needs to compute the length of edges. In each iteration, we denote the direction of B_t as a normalized vector B_t^*, and assume that B_t^* does not change much in the next iteration. The length then can be approximated as $\|B_t\| = {B_t^*}^\top B_t$, leading to

$$E_{S1}(V) = \sum_{I_i \in I} (|{B_t^*}^\top B_t + {B_b^*}^\top B_b - 2s_i W|^2 + |{B_l^*}^\top B_l + {B_r^*}^\top B_r - 2s_i H|^2),$$

where W and H are the original width and height of the image, corresponding to the two components from $S(I_i)$ in Eq. (5).

Since we assume that the edge direction does not change much, we regularize it by introducing

$$E_{S2}(V) = \sum_{i \in I} (\left|{B_t'}^\top B_t\right|^2 + \left|{B_b'}^\top B_b\right|^2 + \left|{B_l'}^\top B_l\right|^2 + \left|{B_r'}^\top B_r\right|^2),$$

where B_t', B_b', B_l', B_r' are orthogonal normalized vectors of B_t^*, B_b^*, B_l^*, B_r^* respectively. E_{S2} penalizes the rotation of edges and enforces smooth update.

During each iteration, (5) is replaced to

$$E_S'(V) = E_{S1}(V) + \lambda E_{S2}(V),$$

where λ is a weight trading off the robustness and convergence speed. We set λ to 0.2 in our experiments, and the function converges quickly (generally fewer than 10 iterations).

Similarly, the line preserving term E_{line} is not quadratic because the direction vector $[a_l, b_l]$ is unknown. We linearly approximate it by assuming that the lines change smoothly. In each iteration, we estimate the direction based on the current solution. By fixing a_l and b_l in Eq. (6), E_{line} becomes a quadratic function for us to optimize and update iteratively.

B. Efficient Optimization

With the above linear approximation, we can optimize Eq. (1) efficiently. In each iteration, we solve a linear system of

$$\begin{bmatrix} A_A \\ A_R \\ A_S \\ A_X \end{bmatrix} V = \begin{bmatrix} 0 \\ 0 \\ b_S \\ b_X \end{bmatrix}$$

where A_A, A_R, A_S, A_X and 0, 0, b_S, b_X are Jacobian matrices and residual errors of the alignment, regularization, scale preserving, and extra terms respectively.

The left side of the equation is a $n \times 2m$ matrix with n much larger than $2m$, since we have much more constraints than the number of vertices (m). We convert the stacked matrices into the summation format

$$(A_A^T A_A + ... + A_X^T A_X)V = A_S^T b_S + A_X^T b_X, \qquad (9)$$

making the matrix size reduce to $2m \times 2m$. Since these matrices are rather sparse, we can utilize the sparsity to significantly reduce the computational complexity.

Except for the scale and line preserving terms, other terms are all quadratic, thus their Jacobian matrices and residual errors are constant during the process. We update A_S, b_S and A_{line}, b_{line} in each iteration, so that the computation can be significantly reduced. For "Urban1" example with 14 images, it takes 2.30 seconds to initialize the matrix and 0.21 seconds to update the matrix in each iteration. The whole optimization[2] takes 15.4 seconds in total with three iterations.

C. Rapid Interactive Refinement

Since our term updating and optimizatin is rather efficient, our system provides an efficient interaction tool to allow the user to correct residual image distortion and improve the alignment in interactive time. The user can draw a few lines, then the corresponding line preserving constraints are added into the energy fucntion and the solution is updated immediately by solving Eq. (9). The updating time is generally $1 \sim 5$ seconds, which can provide instant feedback for the user to conveniently refine the alignment. Please refer to our supplementary video[3] to watch the real-time interactions and instant refinement.

VI. Seamless Composition

After solving (1), we warp input images to a common coordinate system. For overlapping regions, a simple average may cause blurring. Graph cuts have been used [8] to find seams between images so that pixels on the two sides of the seam are consistent or continuous.

In previous approaches, color difference is commonly used as reference. In our wide-baseline cases, alignment errors can be large and the misaligned pixels might have similar colors. We propose combining the alignment error and color difference to generate a better condition.

A. Alignment Score

Given a pair of overlapping images I_i and I_j, we measure alignment errors for all matched feature points and convert them to $[0, 1]$ scores through the Gaussian function

$$s_{p,q} = \exp(-\frac{\|\Psi_i(p) - \Psi_j(q)\|^2}{\sigma_1^2}),$$

where (p, q) is a pair of corresponding feature points from I_i and I_j respectively. Ψ_i and Ψ_j are the warping functions

[2]We use Cholesky decomposition to analytically solve the linear system. If we use conjugate gradient algorithm to iteratively update the solution in each iteration, the optimization speed could be even much quicker.

[3]http://www.cad.zju.edu.cn/home/gfzhang/projects/panorama/pano-video.wmv

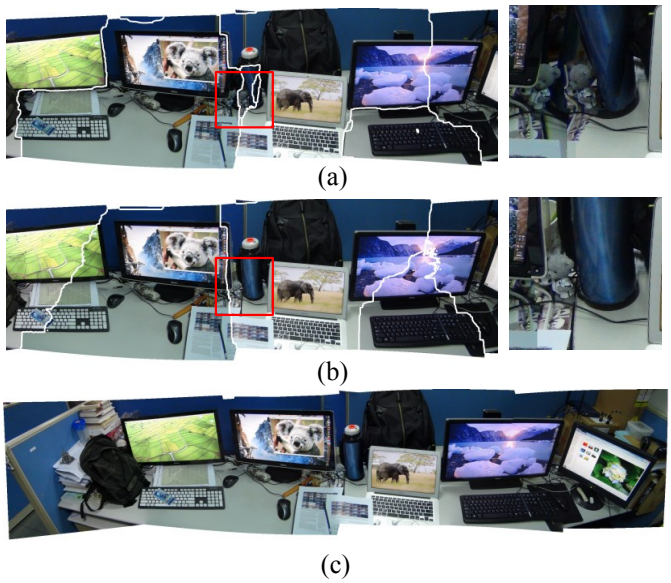


Fig. 10: Seamless Composition. (a) Graph cuts result with the traditional smoothness term only incorporating color difference. (b) Our result. (c) Our final result with gradient domain blending.

corresponding to I_i and I_j, respectively. σ_1 is set to $0.003D$ where D denotes image diagonal length. For features with the alignment error larger than $0.01D$, we assume they are not reliable and ignore them in following process.

With the feature alignment scores, we produce a dense score map on I_i. The contribution of feature p to pixel x depends on distance from p to x as

$$w_{p,x} = \exp(-\frac{\|p-x\|^2}{\sigma_2^2}).$$

σ_2 should be related to the alignment score, since a well aligned feature point propagates better than those with larger alignment errors. For rotational camera motion or a locally planar scene, pixels surrounding the feature points are very likely to be also good. We generally set σ_2 to $0.4D \cdot s_{p,q}$ in our experiments.

Then we define the alignment score map for image I_i as

$$S_{I_i}(x) = \frac{\sum_p w_{p,x}^2 s_{p,q}}{\sum_p w_{p,x}}.$$

Finally we repeat the same process on I_j to generate S_{I_j}, warp the score maps according to the optimized mesh, and average them as the final map as

$$S_{align} = \frac{1}{2}(\Psi_i(S_{I_i}) + \Psi_j(S_{I_j})). \quad (10)$$

B. Color Score

We also use the color difference as a measure of consistency. A Gaussian function is adopted to smooth the energy. The color distance is normalized as

$$S_{color}(x) = \exp(-\frac{|\Psi(I_i)(x) - \Psi(I_j)(x) - \mu|^2}{\sigma^2}), \quad (11)$$

where $\Psi(I_i)$ and $\Psi(I_j)$ are the warped images. μ and σ are the mean and standard deviation of the L2 distance, which are estimated with the overlapping region.

With the Gaussian function, misaligned pixels with large color difference do not provide absurdly large costs. With increasing color distances, the color score moves close to 0. Misaligned pixels are thus assigned with small scores, no matter how different the colors are.

Conditions such as lighting and exposure affect the global luminance of images. With normalization factors μ and σ, this can be corrected in a certain degree. And the global color difference can be finally resolved by the gradient domain blending technique [22].

C. Graph-Cut Optimization

We combine the alignment score (10) and color score (11), and convert them to a function

$$E_{(i,j)}(x) = \max(0, \min(1.5 - S_{align} - S_{color}, 1)). \quad (12)$$

Since $S_{align} \in [0,1]$ and $S_{color} \in [0,1]$, the value of $-S_{align} - S_{color}$ is in the range of [-2, 0]. We adopt the formula in (12) to truncate the value and only choose the medium range [0, 1.0], which can avoid the influence of extreme cases. Now $E_{(i,j)}(x)$ describes the consistency of image pair (I_i, I_j) at pixel x. Given a seam connecting I_i and I_j, the total consistency is defined as the accumulated $E_{ij}(x)$ over the seam pixels.

Similar to previous methods, we optimize the function via graph cuts [30] as

$$E_{cut}(p, L) = \sum_p E_d(p, L_p) + \lambda_s \sum_{(p,q)\in N} E_s(p, q, L_p, L_q), \quad (13)$$

where E_d is the data term defined by the availability of pixels, E_s is the smoothness term preferring well aligned regions, and N is the set of neighboring pixels. λ_s is a smoothness weight and set to 256 in our experiments.

The data term E_d is defined as

$$E_d(p, L_p) = \begin{cases} 0, & x \in \hat{I}_{L_p} \\ \eta, & otherwise \end{cases}$$

where $\hat{I}_{L_p}$ is a warped mask of the image with index L_p. If a pixel is available in the warped L_p-th image, its cost is 0, otherwise it is set to a very large penalty η to avoid being labelled with this image.

The smoothness term E_s is defined as the sum of consistency scores on the neighboring pixels:

$$E_s(p, q, L_p, L_q) = E_{(L_p,L_q)}(p) + E_{(L_p,L_q)}(q).$$

The final labeling problem is solved by minimizing the energy. We use graph cuts algorithm [30] to efficiently solve this energy function, and then further apply gradient domain blending [22] to correct color.

As shown in Figure 10 (a), the blue pot is misaligned due to the lack of reliable features. Since the color appears similar, the traditional seam cut method [31] choose a seam through this area. With our new energy function, such a seam causes much larger cost thus was prohibited. The result shown in (b) demonstrate the effectiveness of our method.

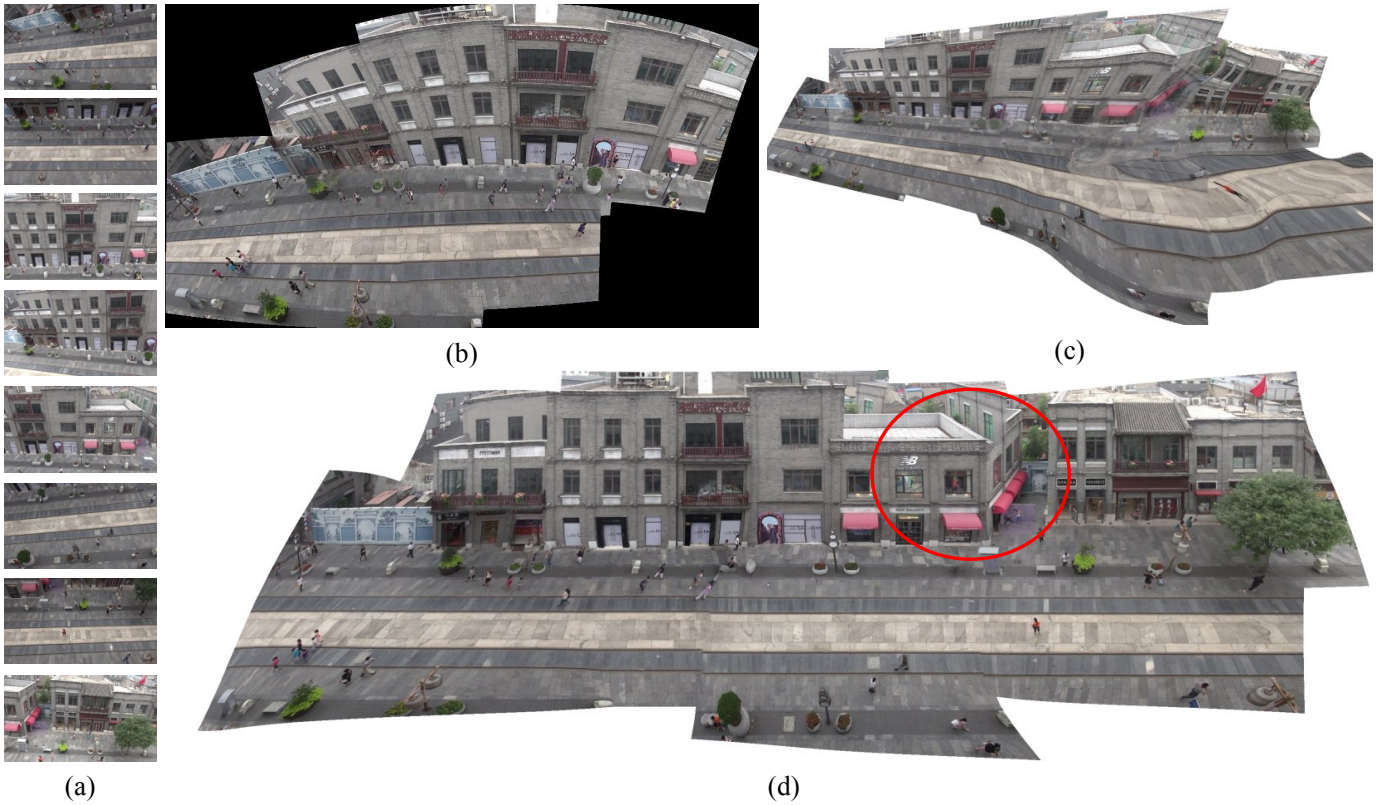


Fig. 11: Image stitching with "Urban2" dataset including 8 wide-baseline images. (a) Input images. (b) Panorama generated by AutoStitch. (c) Panorama generated by APAP. (d) Panorama generated by our approach.

Fig. 12: Long scene example. (a) The averaging of 107 stitiched images by the method of [4]. (b) The averaging of 13 stitiched images by our approach. (c) The final result of [4]. (d) Our final result.

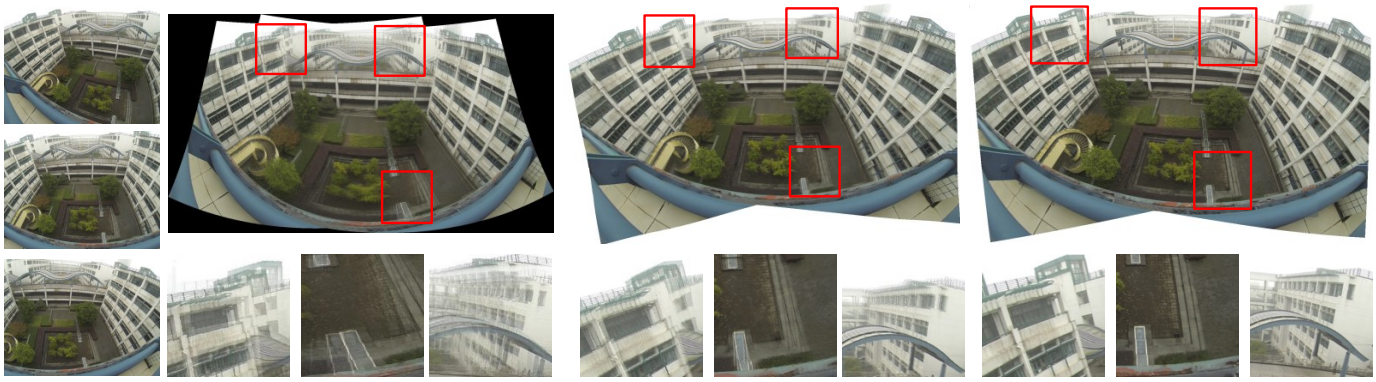


Fig. 13: Image stitching with radial distortion. (a) Three images captured with GoPro Hereo3. (b) The averaging result generated by AutoStitch. (c) The averaging result generated by APAP. (d) The averaging result generated by our approach.

Datasets	Urban1 Fig. 1	Urban2 Fig. 11	Globe	Campus Fig. 14	Desktop Fig. 10
Number	14	8	24	15	6
Matching	2.1s	1.7s	5.6s	4.0s	1.2s
Stitching	15.4s	5.2s	62.0s	48.3s	5.1s
Blending	0.95s	0.4s	1.61s	0.87s	0.18s
Seam Cut	79.3s	19.4s	215.3s	79.7s	16.6s
Fusion	59.3s	33.7s	76.3s	51.4s	18.3s
APAP	176.1s	65.4s	503.4s	-	155.7s
AutoStitch	∼6s	∼5s	∼12s	∼8s	∼3s

TABLE I: The running time of some test datasets.

VII. Experiments

To evaluate the performance of the proposed method, we have conducted experiments on several challenging wide-baseline image datasets, including urban image datasets, indoor image datasets, wide-angle image datasets. If there is no special mention, the results by our method are generated fully automatically without user interactions. The timing statistics are shown in Table I, which are conducted on a desktop PC with an Intel i5-4590 CPU@3.30GHz and a GeForce GTX 760 display card. We use SiftGPU [32] to perform feature matching with outlier rejection, which is rather fast and only takes $1 \sim 6$ seconds in our datasets. Other modules of our system are implemented without GPU acceleration. For each image, it takes about 0.2 second to extract line segments if line preserving constraint is used. Our stitching optimization is also very efficient, which is an order of magnitude faster than APAP [7]. For seamless composition, our graph-cut optimization takes 79.3 seconds, and gradient domain fusion takes 59.3 seconds for "Urban1" example in Figure 1. It should be noted that both APAP and AutoStitch [33] use simple blending techniques without global optimization, where the composition time is quite close to that of our simple blending listed in Table I.

A. Results of Urban Image Datasets

Figure 11 shows an urban scene example with 8 wide-baseline images, where the building and street form two dominant planes. AutoStitch does not find many correspondences under the perspective assumption. APAP [7] constructs a complete panorama, but suffers from inevitable distortion due to the lack of prior constraint. The same correspondences are used for APAP and our approaches for fair comparison. Our mesh-based model generates a dual-homography panorama, as shown in (d). All methods cannot handle strong occlusions. Figure 1 shows another example with the input of 14 images. The results of AutoStitch and APAP are contained in the supplementary document.

We also test our approach using the long sequences from [4]. Figure 12 gives a comparison, where (a) and (b) show the averaging results aligned by the method of [4] and ours respectively. Compared to [4], our method does not require 3D information and can work with much sparser images. We choose only 13 images from the 107 images and achieve comparable result with [4]. Similar to [4], for this example, we use view selection strokes to guide composition. We do not apply other manual work, such as the inpainting strokes employed in [4] to remove power lines from the sky.

B. Results of Wide-Angle and Loop-Closing Images

With adaptive homographies, our method also can handle images with significant radial distortion. For the example shown in Figure 13, we capture 3 images by GoPro Hero3 camera. Due to radial distortion, AutoStitch and APAP do not work well as shown in Figures 13 (b) and (c). The image stitching result by our method contains less ghost artifacts, indicating low alignment errors.

Figure 14 shows a 360°panorama example. The input images are also with significant radial distortion. With the loop closure term in Eq. (8), the left and right most images become more consistent with each other. They are aligned when projecting onto a cylindrical surface. We note the smoothly variant transformation assumption makes the right most highlight not aligned very well. Due to the perspective assumption, APAP result is not suitable for this projection.

Besides panoramic mosaics, our approach can also be applied to texture unfolding for simple objects. The supplementary document shows an example where the desktop globe is unfolded to a world map.

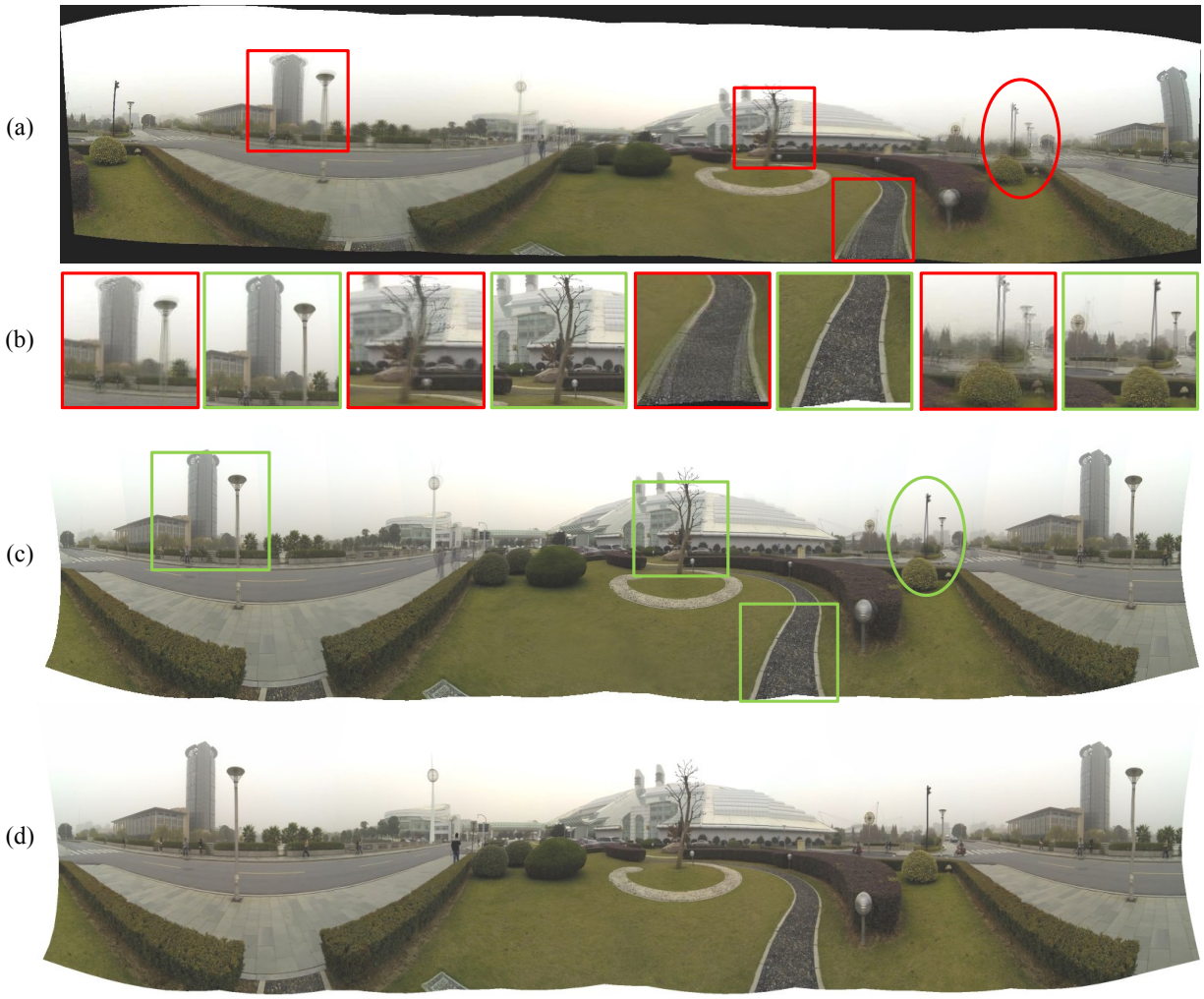


Fig. 14: 360°panoramic mosaic with radial distorted images. (a) The averaging result generated by AutoStitch. (b) Highlights. (c) The averaging result generated by our approach. (d) Our final result with seamless composition.

C. Application for Self-Shots

In the self-shots application on mobile phones, panoramic stitching is a useful feature to generate images with a large field of view. However, it is difficult to keep the camera center static. If the camera is close to faces, the introduced parallax can be rather large. Figure 15 shows an example, where AutoStitch causes misalignment. APAP performs better with the multi-homography model. Our result is with the decent quality.

D. Quantitative Evaluation

We follow the method of [7] to evaluate results quantitatively. For pairwise stitching, we quantify the alignment error of the estimated warp $f : R^2 \rightarrow R^2$ by the root mean squared error (RMSE) of corresponding feature points, where $RMSE(f) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||f(x_i) - x_i||^2}$. We randomly partition all feature matches into "training" and "testing" sets with equal sizes. We use the training set to optimize the warp, and evaluate RMSE on both sets.

We also compare pixel-wise difference quantitatively. Following [34], [7], we define a pixel x as an outlier if there is no similar pixel (intensity difference less than 10 gray levels) within the 4-pixel radius of the warped point. The percentage of outliers in the overlapped area is calculated similarly. For each datum, we repeat this process 20 iterations, and use the average of the results. In each iteration we use the same feature matches on both methods. For our wide-baseline images pairs, since the number of the matched features is already small, we use all matches and evaluate the whole RMSE and outlier percentage.

For fair comparison, we select the first frame as reference, same as that in [7]. In this case, most prior constraints are unnecessary. So we only use feature alignment and regularization terms to construct the energy function, i.e. $E(V) = E_A(V) + \lambda_R E_R(V)$. Table II shows the average RMSE (in pixels) and outlier percentage on different image pairs. "apartment", "railtracks", "conssite", and "garden" are from [7]. "carpark" and "temple" are from [5]. "chess" is from [6]. For APAP, we use the implementation provided by the authors. In most image pairs, our method yields lower errors than APAP [7]. More comparison examples are shown in the

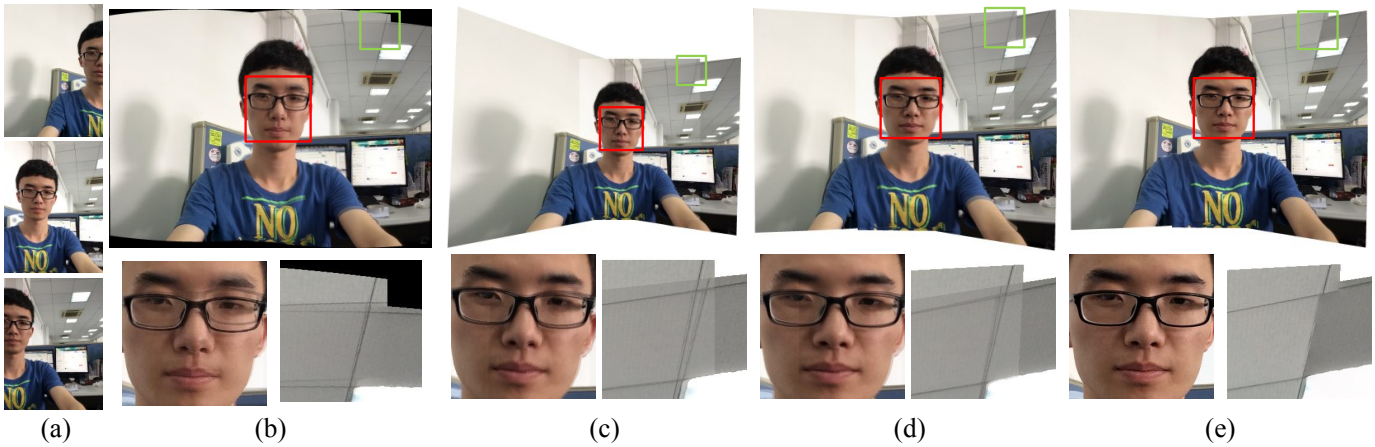


Fig. 15: Self-shot example. (a) Self-shot images. (b) The averaging result by AutoStitch. (c) The averaging result by APAP. (d) The averaging result by our approach. (e) Our final result with seamless composition.

Datasets		TR	TE	%outliers
apartment	-APAP	1.26	1.82	2.86
	-OURS	0.84	1.72	2.63
carpark	-APAP	0.78	0.90	8.12
	-OURS	0.24	0.71	7.02
chess	-APAP	1.43	3.57	25.64
	-OURS	1.08	3.33	24.29
conssite	-APAP	0.43	0.62	9.30
	-OURS	0.31	0.63	9.95
garden	-APAP	0.88	1.04	12.42
	-OURS	0.41	0.94	12.94
railtracks	-APAP	1.10	1.47	15.44
	-OURS	0.45	1.26	15.39
temple	-APAP	0.79	0.90	11.44
	-OURS	0.24	0.81	11.04

TABLE II: Average RMSE (TR: training set error, TE: testing set error).

supplementary document[4].

VIII. Discussion and Conclusions

We have presented a new image stitching approach for wide-baseline images. With the flexibility of a mesh-based model, our method can accommodate moderate deviation from the planar structures. By combining feature alignment, regularization, scale preservation and other extra constraints, a visually plausbile multi-viewpoint panorama is achieved without explicit 3D reconstruction.

Our approach still has some limitations. If a straight line spans across multiple images, our method can only preserve the local straightness in each image. This problem can be addressed either by performing line matching or manually specifying feature matches along the lines if the corresponding matches are not automatically found. In addition, if the input multiple images contain significant occlusion – one region appears in one image but is occluded in others – the occluded parts may not be aligned correctly, such as the highlighted red circle region in Figure 11(d). This problem can be alleviated with user interaction and seam cut. Our future work will be using the multi-homography model to support the discontinuity representation around occlusion boundaries, which may require accurate segmentation. In addition, if the scene contains many complex structures and viewpoints change too much, correspondences may not be automatically established.

[4]http://www.cad.zju.edu.cn/home/gfzhang/projects/panorama/pano-supple.pdf

Acknowledgments

The authors would like to thank all the reviewers for their constructive comments to improve this paper. Hujun Bao is the corresponding author. This work is partially supported by National Science and Technology Support Plan Project (No. 2012BAH35B02), NSF of China (Nos. 61232011 and 61272048), the Fundamental Research Funds for the Central Universities (2015XZZX005-05), a Foundation for the Author of National Excellent Doctoral Dissertation of PR China (No. 201245), a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 413113), and a research grant from Huawei Technologies Co., Ltd.

References

[1] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *SIGGRAPH*, 1997, pp. 251–258.
[2] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 1, 2006.
[3] 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.
[4] A. Agarwala, M. Agrawala, M. F. Cohen, D. Salesin, and R. Szeliski, "Photographing long scenes with multi-viewpoint panoramas," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 853–861, 2006.
[5] J. Gao, S. J. Kim, and M. S. Brown, "Constructing image panoramas using dual-homography warping," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 49–56.
[6] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L. F. Cheong, "Smoothly varying affine stitching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 345–352.
[7] 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.
[8] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *ACM Transactions on Graphics*, vol. 22, no. 3. ACM, 2003, pp. 277–286.
[9] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 519–528.

[10] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, 2009.

[11] H.-H. Vu, R. Keriven, P. Labatut, and J.-P. Pons, "Towards high-resolution large-scale multi-view stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1430–1437.

[12] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, 2010.

[13] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Mach. Vis. Appl.*, vol. 23, no. 5, pp. 903–920, 2012.

[14] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," *ACM Transactions on Graphics*, vol. 28, no. 3, 2009.

[15] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 78, 2013.

[16] Y. Guo, F. Liu, J. Shi, Z.-H. Zhou, and M. Gleicher, "Image retargeting using mesh parametrization," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 856–867, 2009.

[17] W. Hu, Z. Luo, and X. Fan, "Image retargeting via adaptive scaling with geometry preservation," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, pp. 70–81, 2014.

[18] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized scale-and-stretch for image resizing," in *ACM Transactions on Graphics*, vol. 27, no. 5. ACM, 2008, p. 118.

[19] G. Zhang, M. Cheng, S. Hu, and R. R. Martin, "A shape-preserving approach to image resizing," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1897–1906, 2009.

[20] C.-H. Chang and Y.-Y. Chuang, "A line-structure-preserving approach to image resizing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1075–1082.

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

[22] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313–318, 2003.

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

[24] C.-H. Chang, Y. Sato, and Y.-Y. Chuang, "Shape-preserving half-projective warps for image stitching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3254–3261.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[26] G. Yu and J.-M. Morel, "ASIFT: An Algorithm for Fully Affine Invariant Comparison," *Image Processing On Line*, vol. 1, 2011.

[27] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[28] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[29] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[30] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[31] A. Agarwala, M. Dontcheva, M. Agrawala, S. M. Drucker, A. Colburn, B. Curless, D. Salesin, and M. F. Cohen, "Interactive digital photomontage," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 294–302, 2004.

[32] C. Wu, "SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)," http://cs.unc.edu/~ccwu/siftgpu, 2007.

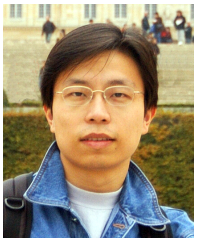
[33] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[34] W.-Y. Lin, L. Liu, Y. Matsushita, K.-L. Low, and S. Liu, "Aligning images in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Guofeng Zhang received the BS and PhD degrees in Computer Science from Zhejiang University, in 2003 and 2009, respectively. He was a recipient of the National Excellent Doctoral Dissertation Award and the Excellent Doctoral Dissertation Award of the China Computer Federation. He is currently an associate professor at State Key Laboratory of CAD&CG, Zhejiang University. His research interests include structure-from-motion, 3D reconstruction, augmented reality, video segmentation and editing. He is a member of IEEE.

Yi He received the BS degree in Software School from Tongji University in 2009, and the Master degree in Computer Science from Zhejiang University in 2015. His research interests include computer vision and image processing.

Weifeng Chen received the B.E. degree in Computer Science from Zhejiang University in 2014. He is now pursuing a Ph.D. degree in Computer Science in the University of Michigan, Ann Arbor. His research interests include computer vision and image processing.



Jiaya Jia received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2004 and is currently a professor in Department of Computer Science and Engineering, The Chinese University of Hong Kong (CUHK). He heads the research group focusing on computational photography, machine learning, practical optimization, and low- and high-level computer vision. He currently serves as an associate editor for the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) and served as an area chair for ICCV and CVPR. He was also on the technical paper program committees of SIGGRAPH, ICCP, and 3DV for several times, and cochaired the Workshop on Interactive Computer Vision, in conjunction with ICCV 2007. He received the Young Researcher Award 2008 and Research Excellence Award 2009 from CUHK. He is a senior member of the IEEE.

Hujun Bao received the BS and PhD degrees in applied mathematics from Zhejiang University in 1987 and 1993, respectively. He is currently a Cheung Kong professor at State Key Laboratory of CAD&CG, Zhejiang University. His main research interest is computer graphics and computer vision, including geometry and vision computing, realtime rendering and mixed reality. He is a member of IEEE.