

Image Stitching Using Structure Deformation

Jiaya Jia, *Member, IEEE*, and Chi-Keung Tang, *Senior Member, IEEE Computer Society*

Abstract—The aim of this paper is to achieve seamless image stitching without producing visual artifact caused by severe intensity discrepancy and structure misalignment, given that the input images are roughly aligned or globally registered. Our new approach is based on structure deformation and propagation for achieving the overall consistency in image structure and intensity. The new stitching algorithm, which has found applications in image compositing, image blending, and intensity correction, consists of the following main processes. Depending on the compatibility and distinctiveness of the 2D features detected in the image plane, single or double optimal partitions are computed subject to the constraints of intensity coherence and structure continuity. Afterwards, specific 1D features are detected along the computed optimal partitions from which a set of sparse deformation vectors is derived to encode 1D feature matching between the partitions. These sparse deformation cues are robustly propagated into the input images by solving the associated minimization problem in gradient domain, thus providing a uniform framework for the simultaneous alignment of image structure and intensity. We present results in general image compositing and blending in order to show the effectiveness of our method in producing seamless stitching results from complex input images.

Index Terms—Image stitching, structure deformation, image alignment.

1 INTRODUCTION

TECHNIQUES in image stitching or blending have been widely applied to generating a natural image composite given a set of globally registered images with limited overlapped region [22], [36]. For image mosaicing applications, global registration is performed based on a variety of predefined camera motion models [33]. For applications aiming to create special effect by engrafting image objects [1], [6], input images are initially registered by manual dragging or assuming static camera configuration. In all these situations, even a small misalignment may cause local intensity or structure inconsistency and produce visual artifacts.

In order to obtain satisfactory results in image stitching, a natural transition from one image to another is required, where both structure and intensity should be aligned or matched within, or possibly beyond, the overlapped area. In this paper, we address the general problem of image stitching in the presence of severe structure and intensity discrepancy and propose a novel technique to simultaneously and globally eliminate misalignment in structure and intensity between the overlapped images.

Previous techniques in image stitching [6], [22], [36] optimize a blending function that minimizes the intensity difference in the vicinity of the overlapped area. There is, however, no guarantee that, after intensity alignment, image features or structures will also be aligned. Structure misalignment causes image ghosting or blurring artifact, where a

salient edge fades out as it enters the overlapped area and fades in just a few pixels away but in a shifted position. To align image features, nonparametric and patch-based techniques have been recently proposed in texture synthesis based on texture deformation [13], [37]. To synthesize a natural texture image, the detected features are matched and deformed inside the overlapped texture samples. Therefore, local structure across patch boundaries can be maintained after synthesis. However, these techniques fail to handle input images with significant color or intensity inconsistency. Moreover, complex global structures and detailed patterns typical of natural images will significantly increase the ambiguity in their 2D patch matching process.

In our experiments, we observe that a successful image-stitching algorithm should not only create a smooth transition within the overlapped region but also preserve the following properties, which are in general agreement with our visual perception:

- **Structure preservation.** The stitched image should not break existing or create new salient structures. A counterexample is shown in Fig. 1a, where the edge of the tower is broken in the overlapped region due to structure misalignment, causing obvious ghosting artifact.
- **Intensity alignment.** Human eyes are sensitive to large intensity change. Unbalanced contrast beyond the overlapped area of a stitched image can be perceptually magnified. An example is shown in Fig. 1b. Although the structure is well aligned and color transition is smooth within the overlapped area, the unnatural color transition from left to right reveals the unmatched intensities inherent in the input images.
- **Image context consideration.** Last but not the least, the context information of objects in the input images should be taken into account during the stitching process. For instance, in Fig. 1c, when the

• J. 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.

• C.-K. Tang is with the Department of Computer Science and Engineering, The Hong Kong University of Science Technology, Clear Water Bay, Hong Kong. E-mail: cktang@cs.ust.hk.

Manuscript received 18 Oct. 2006; revised 7 June 2007; accepted 11 June 2007; published online 21 June 2007.

Recommended for acceptance by C. Taylor.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0735-1006. Digital Object Identifier no. 10.1109/TPAMI.2007.70729.

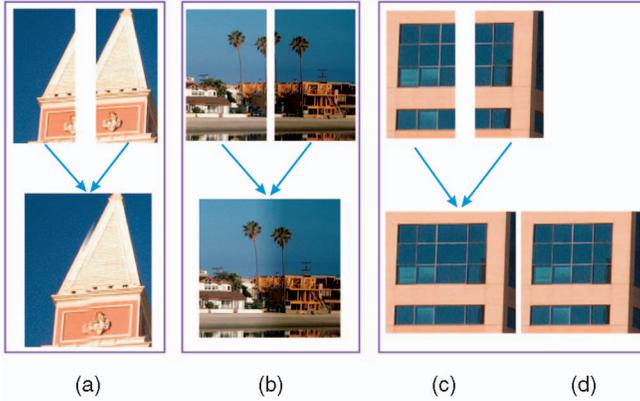


Fig. 1. Typical visual artifact in image stitching. The top row shows the input images to be stitched, whereas the bottom shows the stitching results. (a) One example where the structures of the input images are not correctly aligned. The ghosting artifact is apparent. (b) Color inconsistency in the stitching result. Although the color transition is smooth within the overlapped region, unbalanced intensities are still visible. (c) Object context in image stitching. Given the two input images in the top, the result shown in the bottom is considered visually satisfactory from the point of view of color and structure consistency. However, the windows have the wrong size in the overlapped region. (d) Ground truth, showing that image context, in this case, the uniformity of the window size, should be considered in the stitching process.

images are stitched in a way shown at the bottom, because of the horizontal shift of the second input image, the windows that straddle the overlapped area are widened. Fig. 1d shows the ground truth where the windows are of uniform size.

To address the above issues, we propose a general approach in image stitching based on structure alignment and deformation propagation in natural images. Unlike the previous deformation techniques aiming to align medical images [3] or textures [13], in this paper, our method does not assume camera motion or deformation models. Instead, colors and structures may vary significantly across the images that are problematic to many conventional methods. In our approach, we reduce the ambiguity of deriving structure matching from 2D to 1D, where salient feature detection and matching can be more robustly performed. This is achieved by computing one or two optimal partitions between each pair of the overlapped images, along which structure deformation is performed. In order to achieve smooth and natural deformation, we represent the structure alignment by using feature deformation vectors and propagate them from the optimal partitions toward other pixels in the input images. Such structure deformation propagation is performed in the gradient domain, which globally reduces intensity discrepancy among images.

The rest of our paper is organized as follows: Section 2 reviews related work. Then, in Section 3, we present our algorithms in computing one or two optimal partitions, depending on the compatibility of the detected image features. The preliminary version of our algorithm [18] computes one partition that satisfies the smoothness and alignment constraints. This paper generalizes the notion of searching for two matchable partitions to make the deformation produce more reasonable and meaningful results. In Section 4, the main results from different

applications using our method are shown. The comparisons with previous methods are also given. Finally, we discuss our method and conclude our paper in Section 5.

2 RELATED WORK

Our image stitching aligns not only image intensity but also image structure while preserving the inherent object context. In this section, we review related work in image stitching and structure deformation.

2.1 Image Stitching

Many image registration methods have been developed in recent years. In the presence of significant intensity difference, color blending with the use of a weighting mask over the overlapped area is commonly adopted for generating a smooth intensity transition. For instance, the video mosaics algorithm proposed in [32] estimates the homography matrix for aligning two overlapped images. To reduce visible artifact and local misalignment, the overlapped regions are blended using a bilinear weighting function. In [36], a feather-based algorithm is proposed, which uses averaging and interpolation functions to reduce intensity difference. Unnatural transition, however, is still inevitable since only local operations inside the overlapped regions are performed. Burt and Adelson [6] use a multiresolution spline to perform blending. All these methods only locally blend images in the overlapped areas to transit the images from one lighting environment to another. The local alignment method proposed in [34] performs deghosting, which works well in many situations. However, it requires the recovery of the true 3D ray directions, making it difficult to handle occlusions. Color or intensity difference among images may also make the method susceptible to local minimum. A general review of image alignment and stitching can be found in [33].

Recently, methods in distinctive feature detection and matching have undergone rapid development. Representative methods include Harris corner detector [15], scale invariant SIFT [23], and affine invariant feature detectors [25]. By employing robust feature matching, automatic panorama recognition based on RANSAC is proposed in [4], where multiband blending is introduced to reduce the blurring effect by assigning blending weights to different frequency band. Sand and Teller [30] match video frames using detected features. One of their goals is to find the best matching frames in different videos. The video matching algorithm cannot be directly applied to general image stitching.

Methods using optimal seam are proposed to composite natural or texture images [1], [11], [20]. These methods first compute the color difference in the overlapped area between the two input images. Then, dynamic programming [11] or Graph Cuts [20] is used to compute an optimal partition that produces the least color difference between the two textures/images. In [9], partitions among different motions are also computed. Optimal seam methods do not explicitly consider image features. An ideal partition that does not intersect salient structures may not be found. Another problem of optimal seam is the possible ambiguity associated with the placement of the seam, as shown in Fig. 1c. The stitched

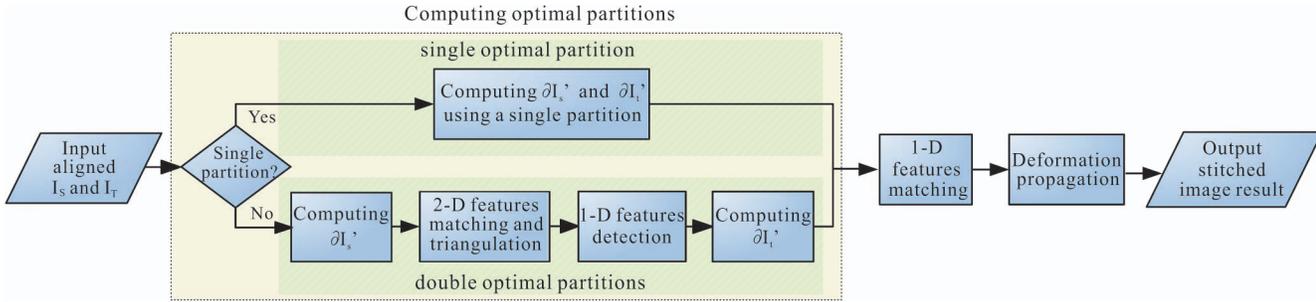


Fig. 2. Overview of our image-stitching algorithm.

image may not be faithful to the original scene in terms of image context.

The two methods described in [16], [17] combine image registration and intensity correction in a single optimization framework. The method in [16] requires that the internal camera parameters be known before the optimization. The tensor voting method described in [17], [19] computes intensity alignment and corrects color globally at all image pixels without any prior knowledge on the camera. However, the structure misalignment is still a problem since structures or features are not explicitly considered in these methods.

Levin et al. [22] propose an image stitching algorithm that operates in the gradient domain and introduce an optimization method based on the gradient strength in the overlapped regions. This method produces good results in the presence of local or global intensity difference between the two input images. However, large structure misalignment cannot be handled.

2.2 Structure Deformation

Structure deformation and alignment are topics of particular interest in medical image registration, especially in nonrigid transformation for registering medical images of different modalities. Bajcsy and Kovacic [3] first propose to use the forces of external stretching and internal smoothness to register medical images. A multiscale technique using a pyramidal representation is applied. In [8], a two-step approach is proposed for nonlinear registration of brain images. In the first step, a one-to-one mapping between corresponding boundaries is established. The second step deforms these boundaries subject to certain criteria. This method requires corresponding boundaries be homothetical to each other, that is, they are related by uniform scaling and length-preserving bending. Elastic registrations [3], [14], [28] are proposed to register medical images, but it is difficult for these methods to handle detailed local features and large luminance changes typical of natural images.

Fluid registration [21], [26] uses the nonrigid method called viscous fluid registration to align medical images. Unlike the elastic models where the desirable deformation may not be obtained because of the internal strain in the elastic continuum, these methods can achieve a desirable deformation since internal forces will disappear over time. However, they may easily introduce blurring and produce nonnegligible distortion.

Recently, feature-matching methods in texture synthesis are proposed in [13], [37]. In [37], binary feature maps are first produced by using a two-pass Canny edge filter. Then, a 2D feature-matching process is applied. This method detects features in the overlapped area in multiple scales. However, the binary feature representation is unsuitable for feature matching in the presence of various feature types, which is common in natural images. These complications may lead to an incorrect warping function. Moreover, since [37] is designed for texture synthesis, there is no provision for correcting intensity misalignment. In [13], a deformation function is introduced to simultaneously maximize color matches while minimizing deformation distortion over the overlapped area. However, the result may be affected by intensity inconsistency between input images.

Our method does not attempt to refine homography estimation in image alignment. Rather, our method deals with intensity inconsistency by considering structure deformation in the gradient domain and by smoothly propagating a set of sparse deformation vectors in the input images. We align the images subject to continuity in not only intensities but also salient structures.

3 OUR IMAGE STITCHING ALGORITHM

For clarity, in this paper, we consider the basic case of stitching two roughly aligned images I_S and I_T with the overlapped area Ω . Our technique can be readily generalized to more input images. In the following, ∇ represents the gradient operator $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$, where ∇_i , $i \in \{x, y\}$, denotes the gradient component.

The essence of our general image stitching approach is to solve the following three subproblems consecutively, which will be described in detail in the following sections. The overview of our method is shown in Fig. 2, where each step is highlighted using a shaded box.

- **Optimal partition computation.** Complex structures in the overlapped regions of the two images make the general image matching and deformation difficult. To address this problem, a subset of salient 1D structures is detected first along the computed optimal partitions between the input images. This step, which will be further divided into more specific ones, is abstracted by the dark yellow box in the top in Fig. 2.
- **One-dimensional feature matching and deformation.** The 1D features detected along the optimal partitions in the overlapped area are matched by

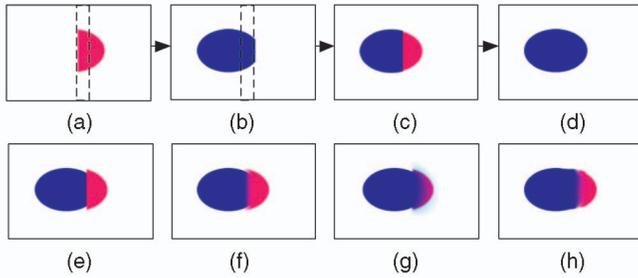


Fig. 3. Toy example. (a) and (b) are two input images. The overlapped regions are indicated by the dashed box. Color and structure misalignment between the images are present. (c) Our result from image space stitching. The edges are aligned. (d) Our result from gradient space stitching, where the structures are properly connected and the color inconsistency is globally corrected. (e) Result using the optimal seam method in [11]. The structure misalignment is inevitable. (f) Result obtained using feathering. (g) Result obtained using the optimal seam operated in the gradient domain. (h) Result from the method in [37]. The warping method cannot produce smooth transition in the presence of significant intensity differences.

minimizing an energy function. We associate each matched 1D feature pair with a corresponding deformation vector and produce a sparse matching set in the image plane.

- **Deformation propagation.** The computed deformation vectors are propagated from the detected 1D features toward all other pixels in the target image to produce a smooth transition in structure and intensity. A deformed gradient map is computed, followed by the reconstruction of the color image result.

Fig. 3 shows a simple illustration where two objects, with color and structure discrepancy, as shown in Figs. 3a and 3b, are to be stitched. Fig. 3c shows the result using our approach in the color image space. Even without considering color blending, our structure deformation can already align the overlapped elliptical structures. Fig. 3d shows that when our method is applied in the gradient domain, after image reconstruction, structures are smoothly aligned while color difference between the input images is reduced. Here, we also show the results obtained using other methods. More comparisons on natural images are given in Section 4. Fig. 3e shows the result using the optimal seam method [11]. Because of the salient structure misalignment, no matter how the seam is constructed, the edge break-up cannot be avoided. Fig. 3f shows the feathering result, which still contains evident structure and color misalignment. Fig. 3g demonstrates the stitching result generated from the optimal seam in the gradient domain. Without the explicit feature alignment, visual artifacts due to edge discontinuity and color inconsistency are apparent. Fig. 3h is the result produced by our implementation of the texture deformation method [37], which is designed for synthesizing texture images. Without the consideration of intensity dissimilarity, the result is unsatisfactory.

In the following, we describe in detail each computation step overviewed in Fig. 2.

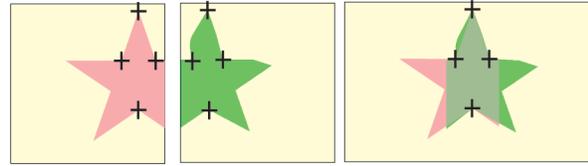


Fig. 4. Image matching using distinctive features. The two images on the left are to be stitched. The correctly matched feature (corner) point pairs are marked in the two images using “+” within the overlapped area. The rightmost image shows the registration result. Although the feature points are aligned well, the edges delimited by the matched feature points are misaligned because of the noticeable difference in local geometry inadequately handled by, for instance, thin-plate spline registration. Without the exact knowledge of the image transformation, 2D deformation is problematic even in this simple example.

3.1 Optimal Partition Computation

In image stitching, directly matching and aligning all pixels within the overlapped region implies a quadratic search space in the total number of pixels. The large search space causes ambiguities in the matching process and makes the general alignment process difficult.

In image registration or morphing [14], [29], a sparse set of matched points or lines is either manually drawn by the user or computed from the images. Afterward, interpolation or approximation using, for example, thin-plate splines or radial basis functions, is applied to all other pixels to smoothly propagate the matching. This approach is limited in deforming general images, because the underlying transformation may not be coincident with the adapted model or function. For instance, if the unknown distortion within the overlapped regions is nonlinear, there may not exist any registration model that adequately describes the transformation. This leads to the problem that even if a large set of points is correctly matched, it is still difficult to appropriately propagate the matching information to all other nondistinctive features such as edges and uniform textures. Fig. 4 shows one simple example in which the feature (corner) points are correctly registered (highlighted using the black crosses in the two input images on the left), there still exists an apparent structure misalignment in the stitching result (shown in the rightmost image), simply because the transformation model does not match the unknown local geometry between the edges of the two input images.

In this section, we shall describe our approach in computing an optimal partition, or two matchable partitions, to robustly align selected 1D features along them. In our method, even if the transformation model between the images is unknown, we are still able to compute robust matching for all the necessary pixels in order to produce seamless image stitching. Either one of the techniques, referred to as a *single optimal partition (SOP)* or a *double optimal partition (DOP)*, is employed, depending on the compatibility and distinctiveness of the feature points detected in the overlapped regions of the two images.

Single Optimal Partition (SOP). There are two situations where an SOP will be applied to minimize structure misalignment. One case is that the overlapped area Ω between the input images only contains textureless regions, where no sufficient distinctive 2D features can be reliably matched. The other situation is that the input images capture different

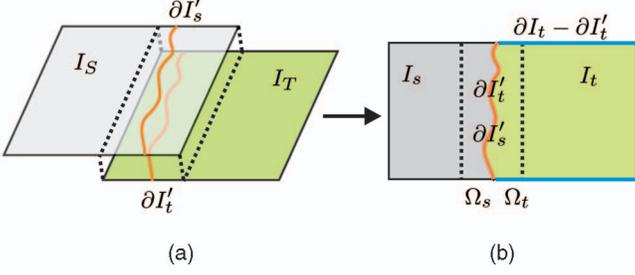


Fig. 5. Single optimal partition (SOP). (a) Images I_S and I_T are overlapped. The SOP divides the overlapped region into Ω_s and Ω_t . (b) The partitioned images are I_s and I_t . The partition boundary $\partial I'_s$ and $\partial I'_t$ (shown in orange) exactly overlap. They are respectively formed by the pixels in I_S and I_T along the cut. The blue boundary of I_t is $\partial I_t - \partial I'_t$.

scenes and inherently contain no matchable features. The latter situation happens in unconventional image compositing, for which we shall show examples in Section 4.

Here, we formulate the partitioning problem as one of the labeling and adopt the Graph Cuts method to find an optimal solution. We define the gradient alignment cost $S(p, q)$ between any adjacent pixels p and q as the sum of the computed values in the red, green, blue (rgb) color channels:

$$S(p, q) = \sum_{r,g,b} ((1 - \beta)S_m + \beta S_d), \quad (1)$$

where S_m and S_d are two costs measuring the gradient smoothness and similarity between the neighboring pixels, which will be defined shortly. β is a weight used to balance the relative influence of the two costs, which is set to 0.3 in our experiments. Before the above computation, we assume that both images have already been smoothed by Gaussian filtering. S_m is defined as

$$S_m(p, q) = \|\nabla_x I_S(p)\| + \|\nabla_x I_S(q)\| + \|\nabla_x I_T(p)\| + \|\nabla_x I_T(q)\|,$$

where $\|\nabla \cdot\|$ denotes the norm of the gradient for each pixel. I_S and I_T are the input images with the overlapped area Ω . Thus, S_m takes gradient smoothness into account, which effectively avoids the partition from breaking object edges in both input images. S_d is defined as

$$S_d(p, q) = \|\nabla_x I_S(p) - \nabla_x I_T(p)\| + \|\nabla_x I_S(q) - \nabla_x I_T(q)\| + \|\nabla_y I_S(p) - \nabla_y I_T(p)\| + \|\nabla_y I_S(q) - \nabla_y I_T(q)\|, \quad (2)$$

where each term above represents the gradient-level similarity at the same pixel location in the overlapped area. S_d penalizes pixel dissimilarity in the gradient domain.

$S(p, q)$, combining S_m and S_d , enables the Graph Cuts method to produce a good initial partition with maximum continuity in the gradient domain. Fig. 5 illustrates that the cut divides Ω into Ω_s and Ω_t , which generates two new regions I_s and I_t (colored in gray and dark green, respectively) to be stitched. The set of pixels along the optimal cut in images I_S and I_T define the optimal partitions, denoted by $\partial I'_s$ and $\partial I'_t$, respectively. $\partial I'_s$ and $\partial I'_t$ exactly overlap. The set of boundary pixels of I_t excluding $\partial I'_t$ is denoted by $\partial I_t - \partial I'_t$. $\partial I'_t$ and $\partial I_t - \partial I'_t$ are, respectively, indicated using orange and blue curves, shown in Fig. 5b.

TABLE 1
Double Optimal Partitioning: Steps

1.	Compute the optimal partition $\partial I'_s$ in image I_S using smoothness constraint.
2.	Detect and match 2-D distinctive features in input images I_S and I_T . Triangulate I_S and I_T using the matched 2-D feature points.
3.	Detect 1-D features along $\partial I'_s$.
4.	Compute $\partial I'_t$ by employing constraints derived respectively in steps 1-3.

Comparing to the optimal seam methods proposed in [11], [20], our new cost function takes into account both gradient smoothness and gradient similarity. Our partition favors smooth area in both images, which effectively reduces structure complexity along the partition and in turn reduces the matching ambiguity in the subsequent steps. Comparison of our stitching results with those produced by other optimal seam methods is presented in Section 4.

Double optimal partitions (DOP) with feature analysis. If there are sufficient distinctive 2D features in the overlapped area Ω and the two images are matchable, we employ the features as context constraint in computing optimal partitions. The corresponding partitions in the two images may not exactly overlap due to geometric distortion or transformation.

We propose to compute respectively two distinctive and matchable partitions in the two input images. Without assuming known distortion among the images, we first compute the optimal partition $\partial I'_s$ in image I_S . Then, a set of constraints with respect to image color, feature, and structure are employed in generating a matchable $\partial I'_t$ in image I_T . Our unified framework consists of several steps. They are listed in Table 1 and will be further motivated and described in the following sections. Later, we will show in Fig. 9 a running example with the intermediate results produced in different steps. Similar to computing SOP, the input images I_S and I_T are assumed to have been smoothed by Gaussian filtering.

3.1.1 Computing Partition $\partial I'_s$

The color deviation along the optimal partition $\partial I'_s$ should be small in order to make the following matching and deformation between the two partitions simple and robust. Accordingly, in the computation, we minimize the color differences of pixels along the partition. Denoting these pixels $\{I_s(0), I_s(1), \dots, I_s(n-1)\}$ in sequence order, where $I_s(i) \in N(I_s(i+1))$, n is the total number of the pixels, and $N(\cdot)$ is the set of the four nearest neighbors, we propose to minimize the following objective function:

$$f_s = \sum_{0 \leq i < n-1} \|\phi(i, s)\|^2, \quad (3)$$

where

$$\phi(i, s) = \begin{cases} 0 & \|I_s(i) - I_s(i+1)\| < \kappa \\ I_s(i) - I_s(i+1) & \|I_s(i) - I_s(i+1)\| \geq \kappa. \end{cases} \quad (4)$$

κ is a predefined threshold to suppress the small intensity change along the partition and to encourage the partition to pass through smooth regions. This minimization problem can be solved using dynamic programming by traversing all

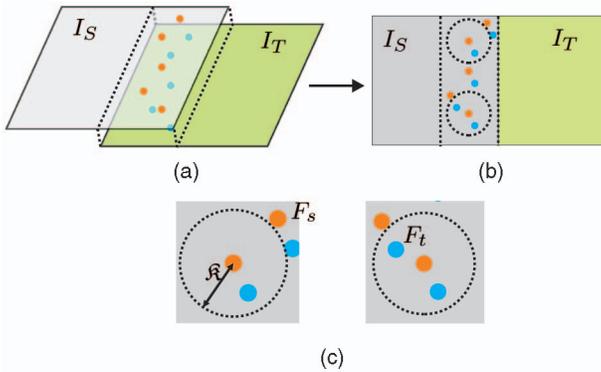


Fig. 6. Robust feature matching. (a) Detected features in the two images are shown in orange and blue, respectively. (b) In the matching process, only the features within a defined window are considered matchable. (c) Magnified regions of (b). By taking into consideration the window size \mathcal{K} and the feature descriptors, matching ambiguities among all features are largely reduced.

possible positions for $I_s(0)$ and $I_s(n-1)$, which is similarly employed in [11].

3.1.2 Two-Dimensional Feature Detection, Matching, and Triangulation

Since the two input images are assumed to be matchable, we detect and match distinctive 2D features in the input images in order to constrain the respective positions of the partitions.

Many feature detectors and descriptors have been proposed. In our method, we first detect the corners using the Harris-Laplacian detector [24] followed by nonmaximal suppression to reduce congregated features. To make the estimation robust, we only compute the most salient features in two images. This can be achieved by computing and comparing the “corner strength” [5] for all features. Then, we represent the features by the local image descriptors proposed by Lowe [23] and match them accordingly. Suppose that $\{F_s^1, \dots, F_s^m\}$ and $\{F_t^1, \dots, F_t^n\}$ are the descriptors for the detected features in the two images within the overlapped regions, the features $\{F_s^i, F_t^j\}$ are matched if the following criteria are satisfied:

$$\|F_s^i - F_t^j\| < \varepsilon \text{ and } |P(F_s^i) - P(F_t^j)| < \mathcal{K}, \quad (5)$$

where ε is a small threshold, $P(F)$ is the pixel position for feature F in the respective images, and \mathcal{K} represents the window size. $|P(F_s^i) - P(F_t^j)| < \mathcal{K}$ requires the distance of the matched 2D feature points should not be far away in the image plane, since the images are already roughly aligned. Thus, in our method, only a few most similar 2D feature points are detected and matched. Fig. 6 illustrates one scenario of the matching process. The final matched features are represented as $\mathbf{F}_s = \{F_s^{m_1}, \dots, F_s^{m_k}\}$ and $\mathbf{F}_t = \{F_t^{n_1}, \dots, F_t^{n_k}\}$, respectively, where k is the total number of matched features.

With the matched \mathbf{F}_s and \mathbf{F}_t , one naive method to generate the partitions is directly connecting part of the matched 2D feature points in \mathbf{F}_s and \mathbf{F}_t , which unfortunately produces undesirable results. This is because feature points represent corners connected to image edges. Connecting the 2D features makes the partition cross unmatched edges, thus conflicting the smoothness constraint described in the

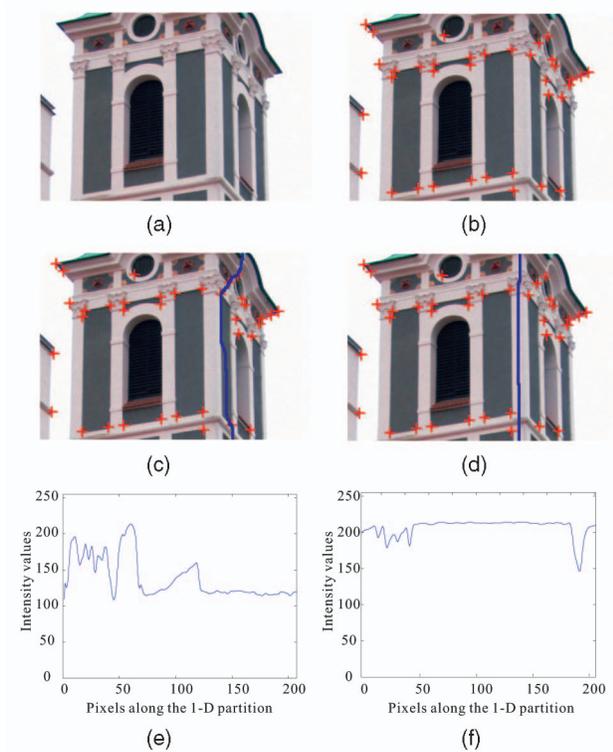


Fig. 7. Two examples of partition computation, respectively, constrained by the detected features. (a) Input image. (b) Using the method described, detected distinctive 2D features are detected and labeled by a red “+.” (c) Partition connecting several feature points inevitably crosses image edges. (d) Partition passing through smoother regions does not cross the 2D features. (e) Intensity curve along the partition in (c) shows complex fluctuation. (f) The intensity curve along the partition in (d) is relatively smoother, making more robust the subsequent partition matching and deformation processes because of its simpler structure.

previous section and, consequently, increase the difficulties in the following processes of partition matching and deformation. Fig. 7 compares two cases where Fig. 7c shows the partition connecting a set of 2D feature points. This partition intersects quite a number of unmatched edges, making the shape of the pixel intensity curve complex along it, as shown in Fig. 7e. A partition passing through smooth regions does not cross these 2D feature points, as shown in Fig. 7d, where the corresponding intensity curve shown in Fig. 7f has less fluctuation compared to that in Fig. 7e.

Thus, in our approach, we do not directly connect the 2D feature points in constructing DOP. Instead, we triangulate them to constrain the generation of optimal partition $\partial I'_t$. The benefit is twofold. First, by knowing which segments (triangles) that $\partial I'_s$ passes through, we can compute $\partial I'_t$ with similar coverage. Second, within the corresponding triangles intersected by $\partial I'_s$, we can estimate the relative distance ratios defined by the partitioning points and the closest features. The corresponding ratios defined by the matchable $\partial I'_t$ and the closest features should be similar.

Now, given the matched features, our goal is to compute two *topologically equivalent* triangulations for the two input images. In particular, if $\{F_s^i, F_t^j\}$ and $\{F_s^{i+1}, F_t^{j+1}\}$ are matched feature points, and F_s^i and F_s^{i+1} are connected by a triangle edge in I_s , then F_t^j and F_t^{j+1} are also connected by a triangle edge in I_T . These topologically equivalent triangulations are said to be *compatible* [35] if there is no edge crossing or triangle

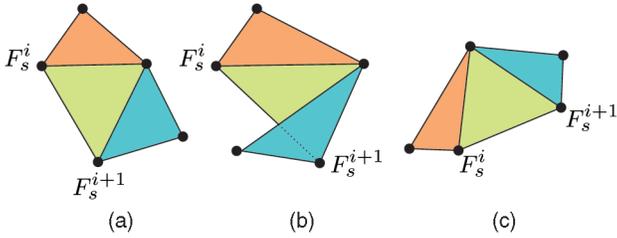


Fig. 8. Triangulation compatibility. (a) Input triangulation. (b) Incompatible triangulation with (a) since some edges cross each other. (c) A compatible triangulation with (a).

fold over. One example of triangulation (in)compatibility is shown in Fig. 8. Unfortunately, the problem of determining whether two point sets are compatible is believed to be NP-hard [2]. In [31], it shows that if Steiner points (extra points) are allowed, any two sets of N points may be made compatible by adding $O(N^2)$ points.

In our method, it is not possible to add more points since our working set consists of the most similar features, which

are already computed and matched. Fortunately, it is possible to remove some of them to achieve compatibility. In the following, we define the triangulation in I_S as $T_S = T(V_S, \mathcal{E}_S)$, where $V_S = \{F_S^{m_1}, \dots, F_S^{m_k}\}$, and \mathcal{E}_S is an edge subset of connected vertex pairs $\{F_S^i, F_S^j\}$, and propose an algorithm shown in Table 2 to compute compatible triangulations.

In essence, we compute two compatible triangulations by incrementally removing problematic points. We use Delaunay triangulation [12] because it maximizes the minimum angle and makes triangles shape more uniform. This is important in partition computation and distance ratio measurement. In practice, our triangulation algorithm converges rapidly, thanks to the initial rough alignment of the images. The matched features are not far away in the overlapped regions. In all our experiments, the number of iterations is always less than 8. We show in Figs. 9d and 9e the computed compatible triangulations given the input images shown in Figs. 9a and 9b.

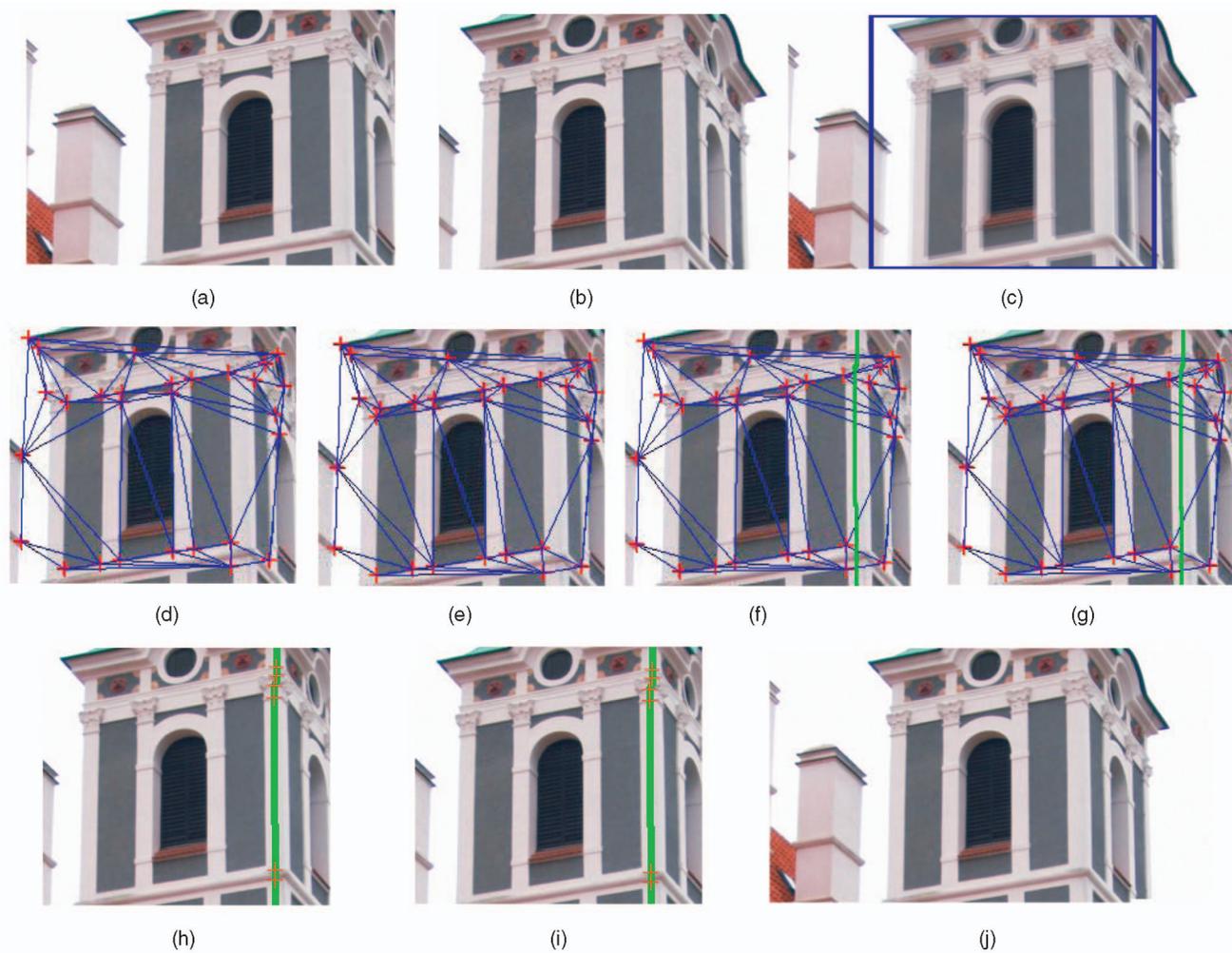


Fig. 9. A running example for demonstrating the computation of DOPs. (a) and (b) are I_S and I_T , respectively. (c) is the initial blending showing structure misalignment. (d) and (e) are the compatible triangulations, respectively, computed in the overlapped area of I_S and I_T . (f) and (g) show the optimal partitions, respectively, computed in the two input images, which are highlighted in green. They pass between the detected 2D features. (h) and (i) show the detected 1D features along the partitions using red crosses. By feature matching, deformation propagation, and image reconstruction, we produce seamless stitching result shown in (j).

TABLE 2
Computing Compatible Triangulations

Initialization:

1. Compute Delaunay triangulation \mathcal{T}_s using the feature points in I_s and initialize $\zeta_s = \zeta_s^+ = \emptyset$.
2. Randomly select three vertices that construct a triangle in \mathcal{T}_s , put the vertices and their internal edges into ζ_s , and then compute an initial triangulation \mathcal{T}_t in I_T using the feature points and edges mapped from ζ_s .

Computation:

3. Select one point $F_s^i \notin \zeta_s$ s.t. $\exists F_s^j \in \zeta_s : \{F_s^i, F_s^j\} \in \mathcal{E}_s$. Add F_s^i and all edges connecting F_s^i and the points in ζ_s to ζ_s^+ .
4. Compute a triangulation \mathcal{T}_t^+ in I_T using the feature points and edges mapped from ζ_s^+ .
5. If the triangulation of \mathcal{T}_t^+ is invalid, i.e., there exist crossed edges or triangle fold-over as shown in Fig. 8, remove F_s^i and the corresponding edges from ζ_s and jump to step 1.
6. Otherwise, set $\zeta_s = \zeta_s^+$ and $\mathcal{T}_t = \mathcal{T}_t^+$. If $\zeta_s \neq \mathcal{T}_s$, jump to step 3. Otherwise, stop and output the triangulations \mathcal{T}_s and \mathcal{T}_t in the two images.

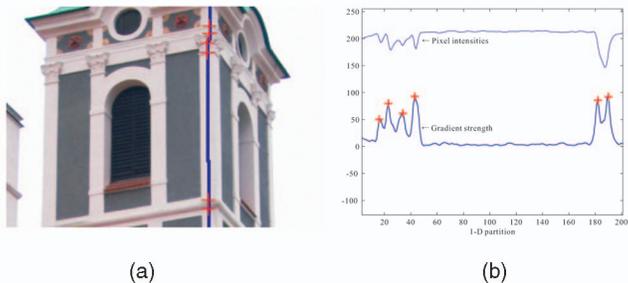


Fig. 10. One-dimensional features along the partition. (a) The input image shown in Fig. 7. The blue line is the partition. The red crosses indicate the 1D features detected by using gradient strength along the partition. (b) Plot of pixel intensity and gradient strength along the partition. The local maxima of the gradient strength are highlighted using the red crosses. They map to the 1D features along the partition in (a).

TABLE 3
1D Feature Detection Along a Partition

1. Compute the gradient strength for all pixels along $\partial I'_s$.
2. Select the pixels with gradient strength larger than a threshold. In our experiments, we set the it to be 50.
3. Perform 1-D non-maximal suppression on the selected pixels to find the most salient 1-D features f_s^k .
4. Compute the signed strength $g_s(k)$ for each 1-D feature point. The absolute feature strength is the projected gradient strength on the partition $\partial I'_s$, while its corresponding sign is set in a way as shown in Fig. 11.

3.1.3 One-Dimensional Feature Detection along the Partitions

In the image plane, 2D distinctive feature points represent corners or edge joints. Similarly, along 1D partitions, there also exist features indicating abrupt change in intensity, as shown in Fig. 10 using red crosses. These 1D features are most noticeable, if they are not well aligned during the partition matching process. In this section, we propose an algorithm to detect 1D features along $\partial I'_s$. A similar algorithm will be used to compute matchable 1D features along $\partial I'_t$ in later sections.

We propose a 1D feature detection algorithm to robustly estimate the strongest gradient along the partitions. The detailed steps are described in Table 3. In the last step, we assign each detected 1D feature point a direction (polarity)

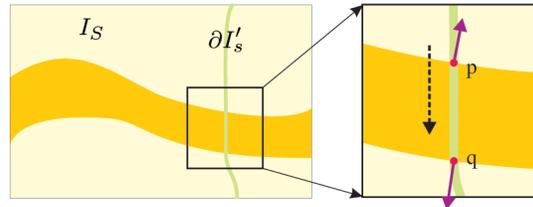


Fig. 11. Direction of 1D features along the optimal partition. The light green curve is $\partial I'_s$. p and q are the curve features since they are salient structure points, shown in the magnified view on the right. We assign the direction of $\partial I'_s$, shown as the black dashed arrow on the right. The gradient directions of p and q are illustrated by the purple arrows. The corresponding gradient direction projected onto the curve is negative for p and positive for q .

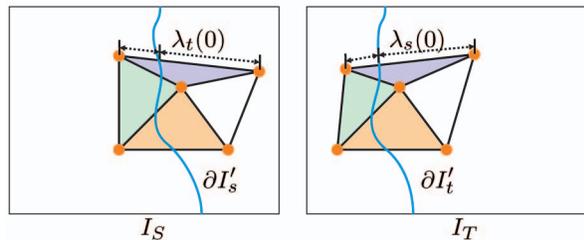


Fig. 12. Triangulation constraint. The two input images are triangulated with compatible triangulation using five points. Since the optimal partition in I_s intersects three triangles, as shown on the left, $\partial I'_t$ should also intersect the corresponding three triangles. The corresponding ratios (for instance, $\lambda_s(0)$ and $\lambda_t(0)$ shown in the figure) by which the partitions divide the edges should also be similar.

to represent the gradient projected onto the partition, which takes a value either negative or positive, as illustrated in Fig. 11.

The 1D features f_s^k detected along $\partial I'_s$ are ordered. We show in Fig. 9h, the running example, the 1D detected features along the partition $\partial I'_s$ using red crosses given the input image I_s in Fig. 9a.

3.1.4 Computing Partition $\partial I'_t$

Taking the output from all the previous steps, we have obtained a set of constraints to characterize the partition $\partial I'_t$. We summarize these constraints as follows:

- **Smoothness constraint.** We have computed the partition in I_s by minimizing $f_s = \sum_{0 \leq i < n} \|\phi(i, s)\|^2$. Likewise, a matchable $\partial I'_t$ should also satisfy the smoothness constraint.
- **Triangulation constraint.** We have triangulated the overlapped regions in the two images into \mathcal{T}_s and \mathcal{T}_t , which satisfy the compatibility requirement, and obtained a subset of triangles in \mathcal{T}_s intersected by $\partial I'_s$. The triangulation constraint favors a matchable $\partial I'_t$ that intersects the corresponding triangles in \mathcal{T}_t . One example is shown in Fig. 12, where $\partial I'_s$ and $\partial I'_t$, respectively, intersect three corresponding triangles.
- **Structure constraint.** In our 1D feature detection along $\partial I'_s$, a set of feature points with signed feature strength is detected. These features characterize the inherent structure of $\partial I'_s$ with which $\partial I'_t$ should exactly match. Specifically, the 1D features detected along $\partial I'_t$ should be distributed in a similar way as those along $\partial I'_s$.

The smoothness constraint requires that $\partial I'_t$ satisfies: $\min \sum_{0 \leq i < m} \|\phi(i, t)\|^2$, where $\phi(\cdot)$ is similarly defined as that in (4), except that we replace I_S by I_T . We formulate the smoothness energy E_1 as

$$E_1 = \sum_i \|\phi(i, t)\|^2, \quad (6)$$

where pixel i is within the space of all pixels along the partition.

The triangulation constraint requires that $\partial I'_s$ and $\partial I'_t$ intersect exactly the same set of corresponding triangles. Specifically, suppose that $\partial I'_s$ intersects an edge subset $\mathcal{E}'_s \subseteq \mathcal{E}_s$ and partitions each edge in \mathcal{E}'_s into two segments with ratios $\lambda_s(0), \lambda_s(1), \dots$, respectively. To be matchable to $\partial I'_s$, $\partial I'_t$ should intersect the same corresponding edge subset \mathcal{E}'_t and partition the edges with similar ratio $\lambda_t(0), \lambda_t(1), \dots$. One example is shown in Fig. 12. We formulate the triangulation energy E_2 as

$$E_2 = \sum_j (\lambda_s(j) - \lambda_t(j))^2 \quad (7)$$

s.t. $|\mathcal{E}'_t| = |\mathcal{E}'_s|$ & $\forall \{F'_s, F'_s\} \in \mathcal{E}'_s : \{F'_t, F'_t\} \in \mathcal{E}'_t$,

where $|\cdot|$ denotes the cardinality of a set.

To formulate the structure constraint, we first detect the salient 1D features along $\partial I'_t$ using the same algorithm shown in Table 3, except that we replace all instances of $\partial I'_s$ by $\partial I'_t$. To achieve robust matching, the number of detected features in $\partial I'_s$ and $\partial I'_t$ should be the same, where the respective strength should also be similar in value. In addition, the variance of the geometric distance between the matched features should be small. Let $d(f'_t, f'_s)$ be the euclidean distance between any two corresponding features f'_t and f'_s in the overlapped regions. The structure energy term E_3 is defined as

$$E_3 = \sum_k ((g_s(k) - g_t(k))^2 + \gamma (d(f'_t, f'_s) - \bar{d})^2) \quad (8)$$

s.t. $|g_t| = |g_s|$,

where γ is the normalization term, $g_s(k)$ and $g_t(k)$ are the respective signed gradient strength for 1D features along $\partial I'_s$ and $\partial I'_t$, and \bar{d} is the mean of $d(f'_t, f'_s)$.

The diversity of the forms of the three energy terms in the above equations makes it difficult to simultaneously minimize them once and for all. Alternatively, we introduce an iterative algorithm to estimate $\partial I'_t$ by first minimizing (6) to filter out nonsmooth partitions and then computing and comparing the defined matching scores from E_1 to E_3 to obtain the optimal partition $\partial I'_t$. We describe the algorithm as follows:

1. In the overlapped region, we compute all possible partitions $\partial I'_t$ connecting any two pixels on the upper and lower boundaries in Ω to minimize (6) of E_1 . If there are n pixels on the upper and lower boundaries in Ω , respectively, the shortest path algorithm [10] can be applied to compute n^2 most smooth partitions with different start and end points with computation complexity $O(n^2)$. We then set $h = 0$.

2. Along the computed h th partition in I_T , detect the 1D features and compute the triangles that the h th partition intersects.
 - a. If $|g_s| \neq |g_t|$ or $|\mathcal{E}'_t| \neq |\mathcal{E}'_s|$, we set its error $err(h) = \infty$ according to (7) and (8).
 - b. Otherwise, we compute the error $err(h)$ using

$$err(h) = E_1 + \alpha E_2 + \beta E_3$$

$$= \sum_i \|\phi(i)\|^2 + \alpha \sum_j (\lambda_s(j) - \lambda_t(j))^2 + \beta \sum_k ((g_s(k) - g_t(k))^2 + \gamma (d(f'_t, f'_s) - \bar{d})^2), \quad (9)$$

where α and β are weights.

- c. $h++$. If $h < n^2$, go to Step 2.
3. Set $\partial I'_t$ to be the partition with error $err^* = \min err(i)$, where $0 \leq i < n^2$.

In all our experiments, the above algorithm works well and produces visually matchable partition $\partial I'_t$ in the target image. Another possible way to minimize E_1 , E_2 , and E_3 simultaneously is by formulating it as a maximum a posteriori (MAP) problem where the smoothness energy E_1 is the log prior and the other two energy terms (E_2 and E_3) are the log likelihood. It can be solved by applying *Markov Chain Monte Carlo* in a Bayesian inference framework [7] over a large solution space. Since there is no exact time complexity estimation, defining an appropriate proposal to speedup the convergence is an issue.

After we have obtained $\partial I'_s$ and $\partial I'_t$, the images I_S and I_T are similarly partitioned into two parts, where I_s and I_t are those parts to be stitched. Referring to that in Fig. 9g shows the result of the partition $\partial I'_t$ computed using the above algorithm. The triangulation and smoothness constraints are well satisfied. Figs. 9h and 9i show that $\partial I'_t$ and $\partial I'_s$ contain the matchable 1D features and satisfy the structure constraint. Fig. 9j shows the final result after structure deformation, to be introduced in the following sections. Notice that the visual artifact is largely reduced.

3.2 One-Dimensional Feature Matching

In DOP estimation, $\partial I'_s$ and $\partial I'_t$ are computed separately. Using the algorithm described above, the matched partitions have exactly the same number of detected 1D features distributed in a similar order. Thus, the 1D features can be readily matched by forming a one-to-one mapping $f'_s \rightarrow f'_t$. In an SOP estimation, we can also detect the 1D features along $\partial I'_s$ and $\partial I'_t$ using the same algorithm described in Table 3. Similarly, the feature strength can also be computed. Suppose that there are, respectively, m and n detected feature points along $\partial I'_t$ and $\partial I'_s$, where m and n may not be equal, in the rest of this section, we propose a general 1D feature-matching approach for the partitions generated using SOP estimation.

Without losing generality, we assume that $n \geq m$, that is, $\partial I'_s$ has more detected features. We propose to match each 1D feature in $\partial I'_t$ to a distinctive feature in $\partial I'_s$ in monotonic

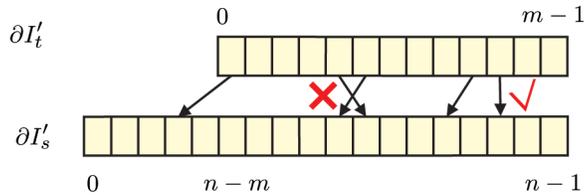


Fig. 13. Feature matching in queues of different lengths. In this example, $\partial I'_t$ has m detected features in the queue, whereas $\partial I'_s$ has n features, $n > m$ as shown. Therefore, there is no one-to-one mapping between the two queues. We propose to match each feature in $\partial I'_t$ to a distinctive in $\partial I'_s$ in a monotonic order. Crossed feature mapping such as the one highlighted in red is not allowed.

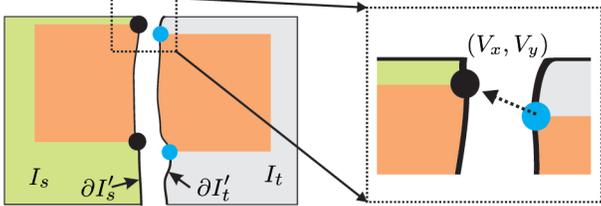


Fig. 14. After 1D feature matching, similar 1D features (pairs of black and blue dots) along the two partitions in the images should be transformed to align the two partitions. One example is shown in the zoom-in view on the right, where features in I_t and I_s are aligned according to the direction (V_x, V_y) , as indicated by the dashed arrow in image plane.

order. One example is shown in Fig. 13. This problem can be formulated as

$$E' = \min \sum_{0 \leq i < m} (g_t(i) - g_s(k_i))^2 \quad (10)$$

$$\text{s.t. } 0 \leq k_0 < k_1 < \dots < k_{m-1} < n.$$

We propose a dynamic programming algorithm to solve (10). Before the description, we denote

$$E'_{a,b} = \min \sum_{0 \leq i < a} (g_t(i) - g_s(k_i))^2 \text{ s.t. } 0 \leq k_0 < \dots < k_{a-1} < b,$$

which is to minimize the energy in matching the first a features in $\partial I'_t$ to the first b features in $\partial I'_s$ in the respective queues, where $b \geq a$. Therefore, $E'_{m,n} = E'$.

In the top-down approach, we have

$$E'_{m,n} = \min_{m-1 \leq i < n} \left(E'_{m-1,i} + \min_{i \leq j < n} (g_t(m-1) - g_s(j))^2 \right),$$

$$\vdots$$

$$E'_{2,3} = \min_{1 \leq i < 3} \left(E'_{1,i} + \min_{i \leq j < 3} (g_t(1) - g_s(j))^2 \right),$$

$$E'_{2,2} = (g_t(0) - g_s(0))^2 + (g_t(1) - g_s(1))^2,$$

$$E'_{1,1} = (g_t(0) - g_s(0))^2.$$

This implies that the dynamic programming using an array storage can efficiently solve the matching problem.

Now, we have constructed a feature mapping, which is *injective* since there may exist unmatched features along one partition. There are two possible ways to further handle them: leaving them unmatched or matching them with the nearest similar features on the other partition without having feature mappings crossing each other. The first solution may still cause structure misalignment in the final

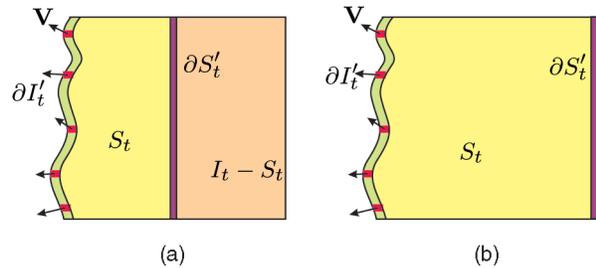


Fig. 15. Boundary condition for deformation propagation. (a) and (b) show that $S_t = \Omega_t$ and $S_t = I_t$, respectively. The small red squares are the 1D features whose deformation vectors are marked by arrows. Along $\partial S'_t$, we set the deformation vector to 0 to avoid unnecessary deformation in $I_t - S_t$.

stitched image, whereas the latter method produces a *surjective* mapping and makes 1D features merge during the transformation, which is employed in our approach.

3.3 Deformation Propagation

For each matched 1D feature pair $(f_t^{k_1}, f_s^{k_2})$ along the two partitions, we construct a *deformation vector*:

$$\mathbf{V}(f_t^{k_1}) = \{V_x(f_t^{k_1}), V_y(f_t^{k_1}), V_{\|\nabla\|}(f_t^{k_1})\},$$

where V_x and V_y are the x and y components of the vector pointing from $f_t^{k_1}$ to $f_s^{k_2}$ (the matched feature pixels) in the image plane, as shown on the right in Fig. 14. $V_{\|\nabla\|}$ measures the strength difference in the gradient map and is computed by

$$V_{\|\nabla\|}(f_t^{k_1}) = \|\nabla I_t(f_t^{k_1})\| - \|\nabla I_s(f_s^{k_2})\|. \quad (11)$$

Therefore, the *deformation vector* consists of both the geometric and strength differences between the matched 1D features, respectively, in the image plane and gradient map.

For clarity of depiction, in the following, we describe our structure deformation method by matching features from I_t to I_s . The analogous problems that I_s is deformed to match I_t , or both I_s and I_t are deformed to match their mean respective feature points, are similar.

To smoothly propagate deformation vectors from the sparse 1D features along $\partial I'_t$ to part of or all other pixels in image I_t , we define the *deformation area* S_t to represent the region being affected by the deformation propagation. We list two configurations in producing seamless results as follows: 1) full propagation, where $S_t = I_t$ to globally diffuse the deformation and 2) partial propagation, where $S_t = \Omega_t$ to locally deform the image. They are alternatively used in our method in different situations.

Taking Fig. 15 as an example, the yellow region in Figs. 15a and 15b are the *deformation areas*, where $S_t = \Omega_t$ and $S_t = I_t$, respectively. The sparse 1D feature points are illustrated using small red squares with associated deformation vectors \mathbf{V} . In Fig. 15a, $\partial S'_t$ is the boundary separating S_t and $I_t - S_t$. Thus, in order not to propagate the deformation outside of S_t , we set the deformation vectors to be $\mathbf{0} = \{0, 0, 0\}$ for all pixels along $\partial S'_t$. For Fig. 15b, $S_t = I_t$, so $\partial S'_t$ is outermost boundary, colored in dark purple. In both cases, we have

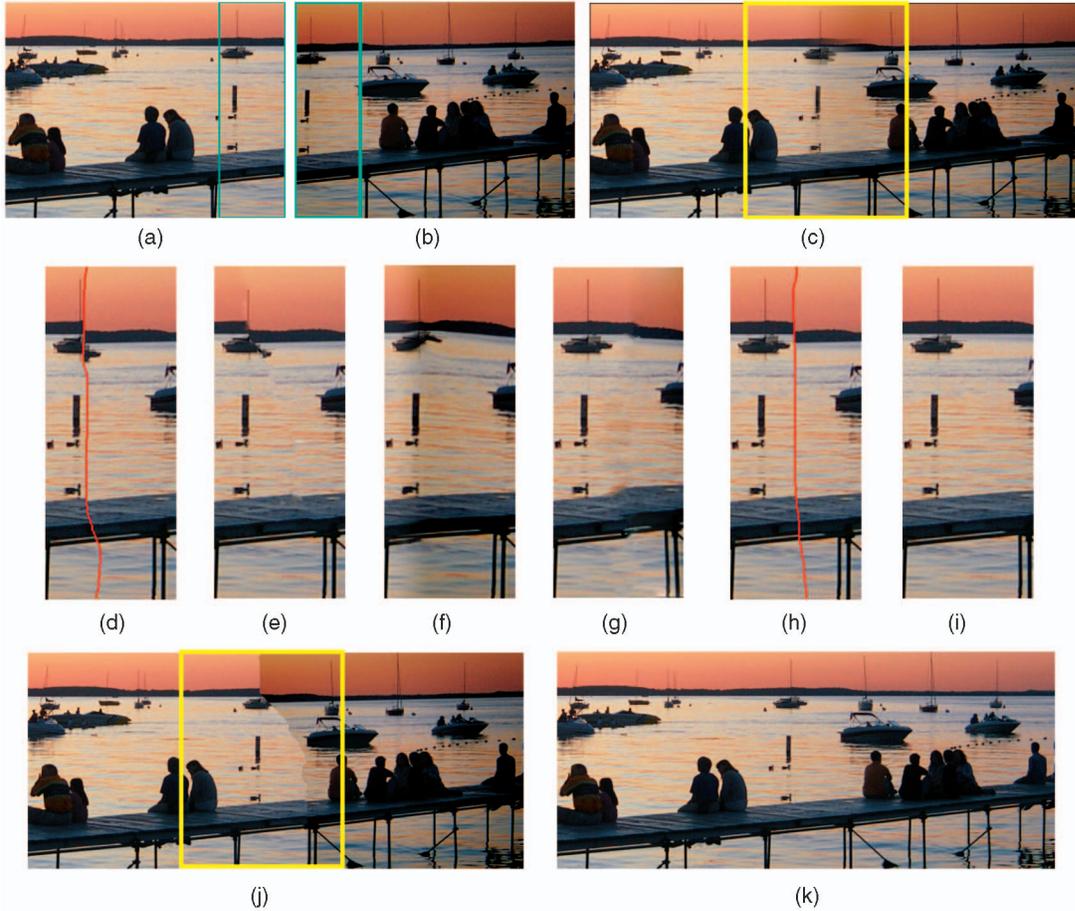


Fig. 16. Sunset. (a) and (b) are the two registered images. The green boxes indicate the overlapped area. (c) Feathering result from the initial alignment. The ghosting artifact is evident because of local structure and intensity misalignment. (d), (e), (f), (g), (h), and (i) are zoom-in views of the results generated by (d) the optimal seam method operated in the gradient domain. The seam is highlighted in red. (e) GIST1 [22]. (f) Structure deformation in [37]. (g) Feature matching in [13]. (h) Our method using SOP. (i) Magnified view of our result, where we set $S_t = I_t$. (j) The result from the optimal seam method. (k) is our image stitching result, where intensity and structures are globally aligned.

$$\mathbf{V}(p) = \mathbf{0} \quad \forall p \in \partial S'_t.$$

Given the sparsely assigned deformation vectors for the features and pixels along $\partial S'_t$ to smoothly propagate the deformation inside image I_t , we propose to solve the minimization problem

$$\mathbf{V}^* = \arg \min_V \int_{p \in S_t} \|\nabla \mathbf{V}\|^2 dp \quad (12)$$

by using conjugate gradients. After the optimization, each pixel in S_t is associated with a deformation vector.

Finally, using the propagated deformation vectors in S_t , we perform an inverse mapping with bilinear interpolation in the gradient domain in S_t to construct the deformed gradient map. The final image is obtained by solving the Poisson equations on the deformed gradient map.

To summarize this section, our double-optimal-partition method appropriately incorporates the three stitching properties described in Section 1. Specifically, the *structure connectivity* is preserved in our stitching process by introducing the structure constraint in Section 3.1.4 and structure deformation in Section 3.3. The *intensity alignment* is achieved by the operations in gradient domain: After solving the Poisson equations, global color consistency can be obtained. The *image context* is considered by incorporating the feature

points in computing $\partial I'_t$. The sparsely matched features offer the necessary image topology information, and the triangulation constraint in Section 3.1.4 requires that the partitions be in relatively similar positions in the aligned images.

4 RESULTS

In this section, we show that our method is capable of generating natural image stitching results for a variety of scenes. Comparison with other methods using our implementation is also given.

4.1 Image Stitching Using Single Optimal Partition (SOP)

We first show some image stitching examples using SOP. In Figs. 16a and 16b, we show two overlapped images of a sunset scene. Precise alignment is difficult because of the local displacement and the small overlapped region shown inside the green boxes in the figure. Fig. 16c is the feathering result obtained from initial alignment, where the ghosting artifact is significant because of intensity discrepancy between the images. In Figs. 16d, 16e, 16f, 16g, 16h, and 16i, we compare our result with those generated using previous methods. Fig. 16d is the result obtained by the optimal seam method operated in gradient domain. Although the color discrepancy is alleviated, it does not help in solving the problem of

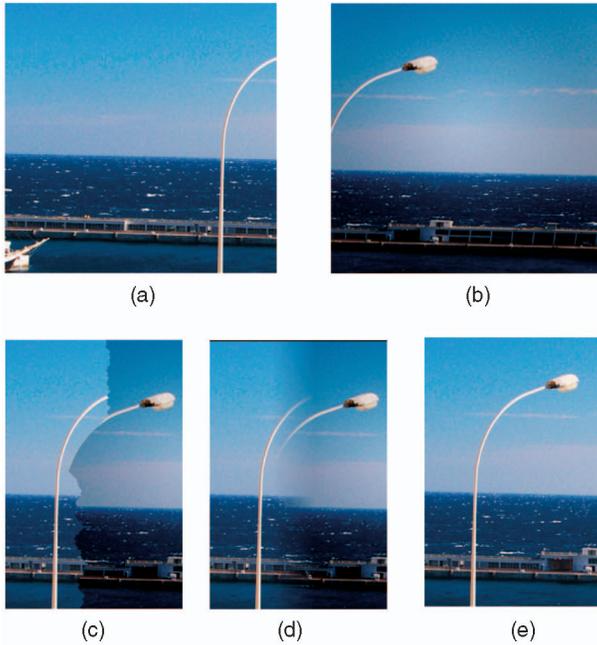


Fig. 17. Lamp post. (a) and (b) are the two registered images. (c) and (d) are zoom-in views of the results generated by (c) the optimal seam method and (d) feathering. Notice the visible artifact in both color and structure. (e) Our result using SOP. The structures are globally aligned and deformed with minimum distortion under our computation framework.

structure misalignment. Fig. 16e shows the result computed by GIST1 [22], where the imposed smoothness constraint cannot eliminate the structure misalignment. Fig. 16f is the structure deformation result obtained in [37]. Matching of the extracted 2D features cannot adequately handle complex structures such as the boat, causing incorrect warping result and local distortion as shown. Fig. 16g is the feature-matching result generated in [13]. The significant difference in global image intensity makes their optimization method susceptible to local minima, which causes inaccurate alignment.

In Fig. 16h, SOP feature matching is applied. By taking structure smoothness into consideration, the matching process is robust. Fig. 16i shows a magnified view of the overlapped area in our result after image deformation. Fig. 16j is the full result generated by the optimal seam method [11]. Since no pixels are similar in intensity, an obvious seam is produced (indicated by the thin red curve). Fig. 16k is obtained by our method. Because the deformation is properly and smoothly propagated toward the interior of the images, structures are properly aligned, and intensities are better matched.

Fig. 17 shows one example where the two input images in Figs. 17a and 17b have a large discrepancy in both color and structures. Fig. 17c shows the zoom-in view of the result generated by the optimal seam method. Fig. 17d shows a magnified view of the feathering result. Fig. 17e shows our result using SOP. Our method can handle color and structure deformation within the same framework.

4.2 Image Stitching Using Double Optimal Partitions (DOP)

We have shown in Fig. 9 one example using the detected 2D features to constrain the partitions constructed separately in two images.

Another example is shown in Figs. 18a and 18b, where the two input images I_S and I_T to be stitched. Color and structure distortion are present in the overlapped area, which makes alignment fail to produce satisfactory result, as shown in Fig. 18c. In Figs. 18d, 18e, 18f, 18g, 18h, 18i, and 18j, intermediate results in the overlapped region from different steps in the DOP process are shown. We first produce partition $\partial I'_s$, as shown on the left in Fig. 18d. Fig. 18e shows the plot of gradient strength along the partition. Using our method, three 1D feature points are detected in this case, which are highlighted using red crosses. The 2D features in the overlapped region are then detected in both images, as shown in Fig. 18f. Fig. 18g shows the result in which the two regions are compatibly triangulated. Combining all the constraints described in the paper, the optimal partition $\partial I'_t$ is computed as shown in the right subfigure in Fig. 18h, which corresponds well to $\partial I'_s$ in terms of shape and the triangles intersected. Fig. 18i shows the gradient strength along the computed partition $\partial I'_t$, where the detected 1D features are also matchable to those in $\partial I'_s$ shown in Fig. 18e. The deformation vectors for the sparse 1D features are illustrated in Fig. 18j. Fig. 18k shows a failure example using optimal seam compositing, where salient structure misalignment is visible. Fig. 18l shows our image stitching result. By performing structure deformation in the image gradient domain, structure and color misalignment is avoided.

4.3 Image Stitching in General Image Compositing

Our method can be readily applied to unconventional image stitching with arbitrary overlapped areas to generate special effect. In Fig. 19a, we only use a single input image, where the lower brush is copied to align with the upper one. The user draws the mask, as shown in the yellow region in Fig. 19b. The feathering result is shown in Fig. 19c, where misalignment is obvious. Fig. 19d shows the result obtained by the optimal seam method. The corresponding pixels along the boundary differ significantly in intensity, resulting in the obvious seam as shown. Fig. 19e shows the result of the GIST1 method [22], which cannot eliminate misalignment in this example. Fig. 19f shows the result obtained using the deformation method in [37]. Since this example contains complex features at multiple scales, their proposed warping function mixes up these features and causes alignment error. Shown in Fig. 19g is the result produced by direct Poisson blending [27]. Our result is shown in Fig. 19h, where structures and intensities are seamlessly aligned along the stitching boundary, and the sparse deformation vectors are smoothly propagated into the interior of the pasted region.

Fig. 20 shows another example demonstrating that our method can automatically align salient structures for seamless image composition, whereas previous techniques employing interactive photo editing tools still require user to perform manual adjustment. Figs. 20a and 20b show, respectively, the input source and target images. Fig. 20c shows the target scene on which we want to composite the source object. Fig. 20d shows the result after a set of typical operations, where the user first manually selects the position that the object cut-out will be placed then applies the optimal seam method to compute the partition where color inconsistency is minimized and, finally, uses Poisson blending to composite the object. Even after all these operations, since there exist inherent misaligned structures, visual artifacts such as broken edges are still visible, as shown in the magnified view in Fig. 20e. Our method does not require the user to carefully adjust where the object cut-out should be

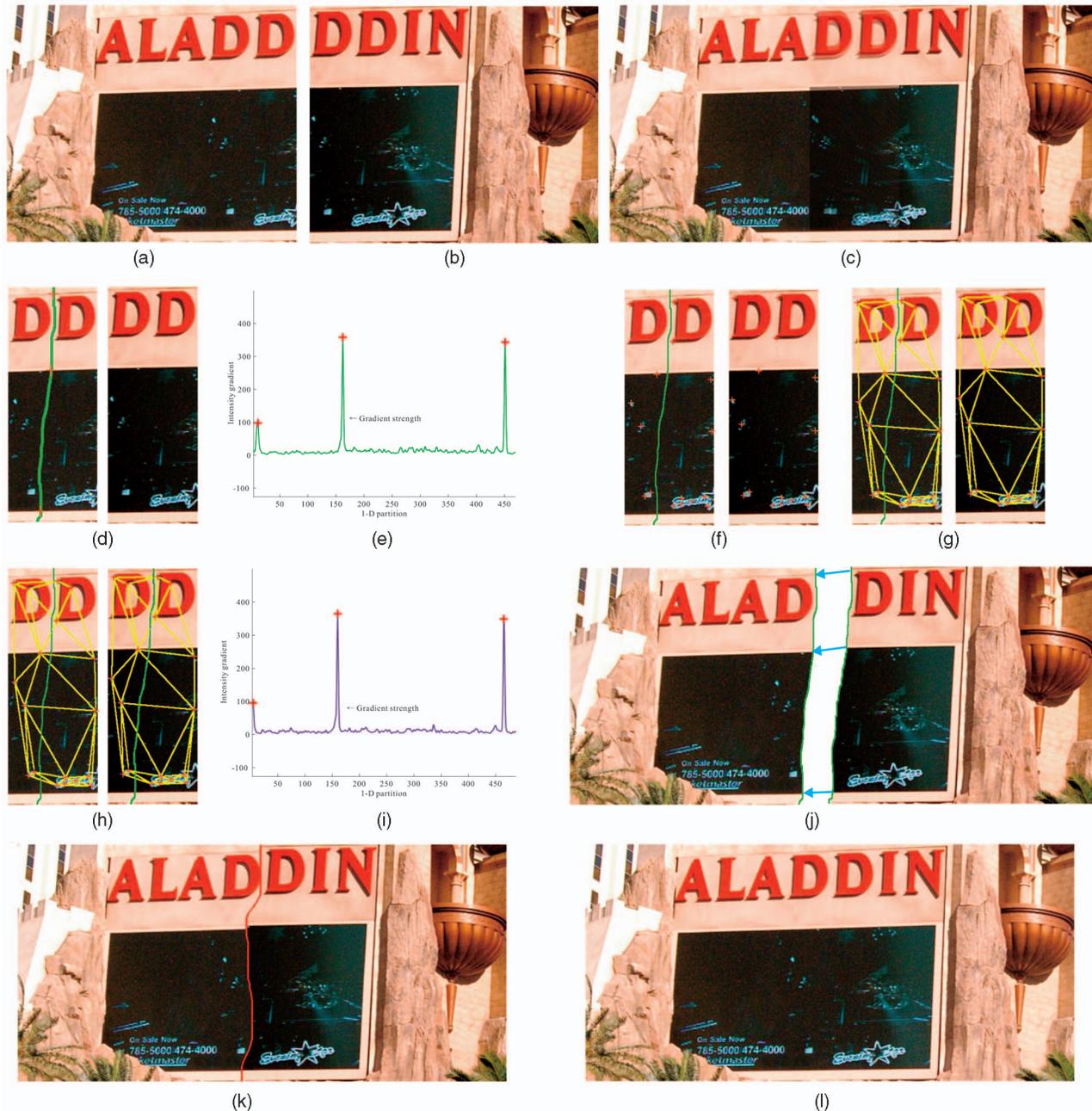


Fig. 18. DOPs. (a) and (b) are input images I_S and I_T , respectively. Because of image distortion, the initial alignment of the two images introduces visual artifact, as shown in (c). We apply DOP. The steps are shown in (d) to (j). (d) shows the computed $\partial I'_s$. (e) illustrates the detected 1D features along $\partial I'_s$, denoted by crosses. To compute a matchable partition $\partial I'_t$, 2D matched features are first detected in the overlapped area (f), followed by constructing compatible triangulations in the two images in (g). Combining all constraints, the partition $\partial I'_t$ is computed, as shown on the right in (h). (i) shows the detected 1D features along $\partial I'_t$. (j) shows the deformation vectors computed on the corresponding 1D features. (k) shows the result computed using the optimal seam method [11], where structure mismatching is apparent. (l) is our result after structure deformation. Both structure and color are well aligned.

placed to achieve maximum alignment. Using our automatic 1D feature detection, matching and deformation, as shown in Fig. 20f, structure misalignment is eliminated in our result. Magnified views are shown in Fig. 20e.

5 CONCLUSION AND DISCUSSION

In this paper, we have proposed a novel image stitching approach by image deformation, where the overlapped regions may contain significant intensity inconsistency and structure misalignment. Instead of generating precise

alignment, considering all salient features in a 2D image plane or using a predefined model in image alignment, we propose to match only the necessary features along one or two optimal partitions and use them to construct a sparse set of deformation vectors. This reduces the misalignment problem caused by complex 2D structure and color, especially for input images with significant mismatches in the overlapped area. From the 1D sparse features detected along the partition(s), we propagate the deformation into the target image smoothly. Structure deformation and color correction are simultaneously achieved within the same framework operating in

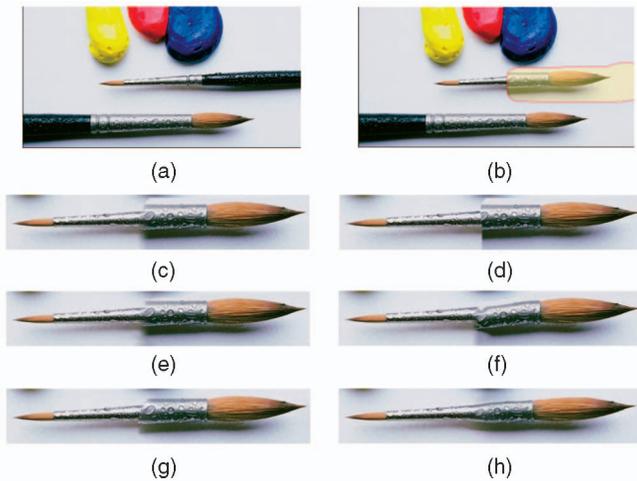


Fig. 19. Brushes. (a) Input image. (b) Part of the lower brush is copied and pasted onto the upper one, as shown inside the yellow region. (c) Feathering result. (d) Optimal seam result. (e) GIST1 result [22]. (f) Result of structure deformation in [37]. (g) Result by direct Poisson blending [27]. (h) Our result. The complexity in feature matching is reduced to 1D, allowing the sparse deformation vectors to be robustly propagated into the interior of the image to enforce the necessary structure continuity and smoothness.

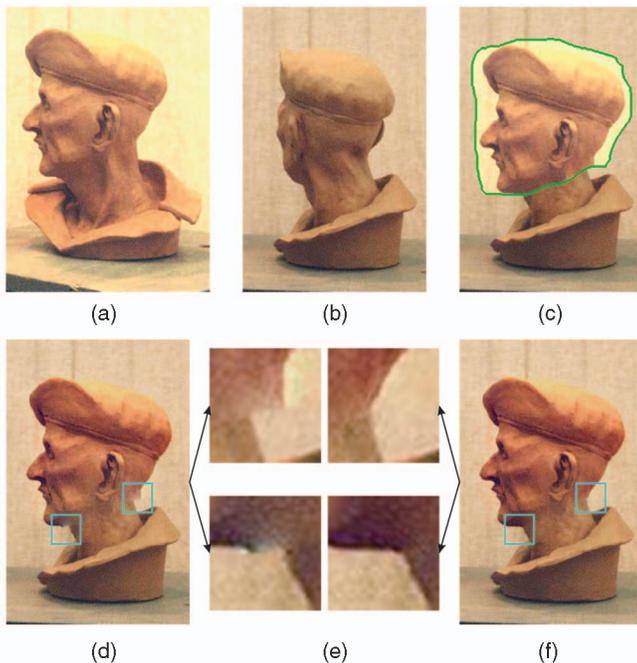


Fig. 20. Bust. (a) and (b) show that, respectively, the source and target images. (c) shows part of bust in (a) is to be composited to the target image. In traditional image editing, a user needs to carefully align the source object and the target image in order to produce a seamless result. (d) shows that in this example, even with careful manual alignment and optimal seam computation followed by Poisson blending, the result still contains artifact because of structure discontinuity. Magnified views are shown in (e). (f) shows our result where all pertinent structures are seamlessly aligned.

the image gradient domain. In experiments, we also observe that commonly used methods such as blending or optimal seam cannot always produce seamless results. Our method, when applied to image composition, can automatically search for matchable features and align them by deformation. This largely alleviates the users from carefully and manually matching structures along the optimal boundaries.

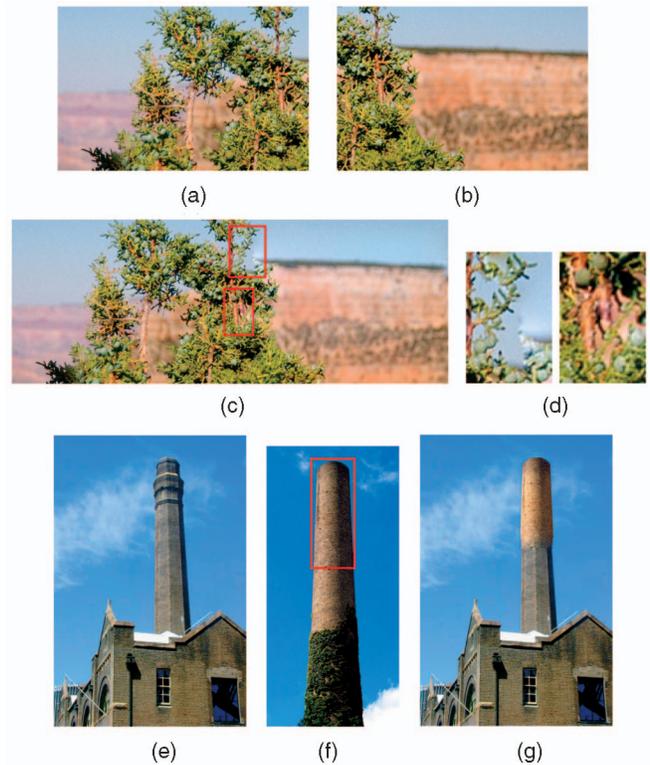


Fig. 21. Two difficult examples. (a) and (b) are two input images to be stitched. They contain complex structures and features. (c) shows our stitching result. (d) The magnified region still contains errors. (e) and (f) show two input images. We graft the chimney in the red rectangle onto (e). (g) shows our result. Although the structure is aligned well, it does not look natural.

Our method provides a principled and effective way to address the general problem of natural image stitching. SOP or DOP are adopted in different situations to achieve seamless stitching. Generally speaking, SOP is applicable when very few feature points are found in the overlapped regions, whereas the DOP method is appropriate for more general scenes.

Due to complexity of natural images, the following factors may influence our stitching quality. First, in our experiments, when DOP is applied, we need to employ existing feature detectors and descriptors [24] to match features. The SIFT detector is not invariant to affine transformation and partial occlusion and may produce erroneous matching results in difficult images. We show one example in Figs. 21a, 21b, 21c, and 21d, where the input images contain very complex structures (the canyon) and textures (the plant). In this example, the features cannot be matched well. The magnified regions in Fig. 21d illustrates the misaligned structures. Second, our method stitches images in the image gradient domain, it does not guarantee to produce the best visual effect if the source and target images are very different in appearance. One example is shown in Figs. 21e, 21f, and 21g, where we composite part of the chimney in Fig. 21f to the image in Fig. 21e. Although the main structures are aligned well, the stitching result is not visually natural because the textures of the two chimneys are very dissimilar.

In the future, we shall investigate other image matching criteria to handle the above difficult examples. Moreover, extending this method to videos and multiple images is another possible direction.

ACKNOWLEDGMENTS

The authors would like to thank the associate editor and all the reviewers for their constructive comments to improve the final manuscript. This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project 412206 and Project 620005).

REFERENCES

- [1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive Digital Photomontage," *Proc. ACM SIGGRAPH '04*, vol. 23, no. 3, pp. 294-302, 2004.
- [2] B. Aronov, R. Seidel, and D.L. Souvaine, "On Compatible Triangulations of Simple Polygons," *Computational Geometry: Theory and Applications*, pp. 27-35, 1993.
- [3] R. Bajcsy and S. Kovacic, "Multiresolution Elastic Matching," *Computer Vision, Graphics and Image Processing*, vol. 46, pp. 1-21, 1989.
- [4] M. Brown and D.G. Lowe, "Recognising Panoramas," *Proc. Int'l Conf. Computer Vision*, pp. 1218-1225, 2003.
- [5] M. Brown, R. Szeliski, and S. Winder, "Multi-Image Matching Using Multi-Scale Oriented Patches," *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, pp. 510-517, 2005.
- [6] P.J. Burt and E.H. Adelson, "A Multiresolution Spline with Applications to Image Mosaics," *ACM Trans. Graphics*, vol. 2, no. 4, pp. 217-236, Oct. 1983.
- [7] M.-H. Chen, Q.-M. Shao, and J.G. Ibrahim, *Monte Carlo Methods in Bayesian Computation*. Springer, 2000.
- [8] C. Davatzikos, J.L. Prince, and R.N. Bryan, "Image Registration Based on Boundary Mapping," *IEEE Trans. Medical Imaging*, vol. 15, pp. 112-115, 1996.
- [9] J. Davis, "Mosaics of Scenes with Moving Objects," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 1998.
- [10] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, 1959.
- [11] A.A. Efros and W.T. Freeman, "Image Quilting for Texture Synthesis and Transfer," *Proc. ACM SIGGRAPH '01*, pp. 341-346, Aug. 2001.
- [12] D. Eppstein, "The Farthest Point Delaunay Triangulation Minimizes Angles," *CGTA: Computational Geometry: Theory and Applications*, vol. 1, 1992.
- [13] H. Fang and J.C. Hart, "Textureshop: Texture Synthesis as a Photograph Editing Tool," *Proc. ACM SIGGRAPH '04*, 2004.
- [14] M. Fornefett, K. Rohr, and H.S. Stiehl, "Radial Basis Functions with Compact Support for Elastic Registration of Medical Images," *Image and Vision Computing*, vol. 19, pp. 87-96, 2001.
- [15] C. Harris and M.J. Stephens, "A Combined Corner and Edge Detector," *Proc. Alvey Vision Conf.*, pp. 147-152, 1988.
- [16] D. Hasler and S. Susstrunk, "Colour Handling in Panoramic Photography," *Proc. SPIE*, Jan. 2001.
- [17] J. Jia and C.-K. Tang, "Image Registration with Global and Local Luminance Alignment," *Proc. Int'l Conf. Computer Vision*, vol. I, pp. 156-163, 2003.
- [18] J. Jia and C.-K. Tang, "Eliminating Structure and Intensity Misalignment in Image Stitching," *Proc. Int'l Conf. Computer Vision*, vol. II, pp. 1651-1658, 2005.
- [19] J. Jia and C.-K. Tang, "Tensor Voting for Image Correction by Global and Local Intensity Alignment," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 36-50, Jan. 2005.
- [20] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *ACM Trans. Graphics, Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '03)*, vol. 22, no. 3, pp. 277-286, July 2003.
- [21] H. Lester and S.R. Arridge, "Summarising Fluid Registration by Thin-Plate Spline Warps with Many Landmarks," *Proc. Medical Image Understanding and Analysis*, 1997.
- [22] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless Image Stitching in the Gradient Domain," *Proc. European Conf. Computer Vision*, May 2004.
- [23] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 90-110, 2004.
- [24] K. Mikolajczyk and C. Schmid, "Indexing Based on Scale Invariant Interest Points," *Proc. Int'l Conf. Computer Vision*, vol. I, pp. 525-531, 2001.
- [25] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A Comparison of Affine Region Detectors," *Int'l J. Computer Vision*, vol. 65, no. 1/2, pp. 43-72, 2005.
- [26] B.-N. Morten and C. Gramkow, "Fast Fluid Registration of Medical Images," *Proc. Visualization in Biomedical Computing*, vol. 1131, pp. 267-276, 1996.
- [27] P. Pérez, M. Gangnet, and A. Blake, "Poisson Image Editing," *ACM Trans. Graphics, Proc. ACM SIGGRAPH '03*, vol. 22, no. 3, pp. 313-318, 2003.
- [28] K. Rohr, H.S. Stiehl, R. Sprengel, T.M. Buzug, J. Weese, and M.H. Kuhn, "Landmark-Based Elastic Registration Using Approximating Thin-Plate Splines," *IEEE Trans. Medical Imaging*, vol. 20, no. 6, pp. 526-534, 2001.
- [29] D. Ruprecht and H. Müller, "Image Warping with Scattered Data Interpolation," *IEEE Computer Graphics and Applications*, vol. 15, 1995.
- [30] P. Sand and S. Teller, "Video Matching," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 592-599, 2004.
- [31] D.L. Souvaine and R. Wenger, "Constructing Piecewise Linear Homeomorphisms," DIMACS Technical Report 94-52, Rutgers Univ., 1994.
- [32] R. Szeliski, "Video Mosaics for Virtual Environments," *IEEE Computer Graphics and Applications*, pp. 22-30, Mar. 1996.
- [33] R. Szeliski, "Image Alignment and Stitching: A Tutorial," Technical Report MSR-TR-2004-92, Microsoft Research, Dec. 2004.
- [34] R. Szeliski and H.-Y. Shum, "Construction of Panoramic Image Mosaics with Global and Local Alignment," *Int'l J. Computer Vision*, vol. 36, no. 2, pp. 101-130, 2000.
- [35] A. Tal and G. Elber, "Image Morphing with Feature Preserving Texture," *Proc. Conf. European Assoc. for Computer Graphics*, pp. 339-348, 1999.
- [36] M. Uyttendaele, A. Eden, and R. Szeliski, "Eliminating Ghosting and Exposure Artifacts in Image Mosaics," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001.
- [37] Q. Wu and Y. Yu, "Feature Matching and Deformation for Texture Synthesis," *Proc. ACM SIGGRAPH '04*, vol. 23, no. 3, pp. 362-365, 2004.



Jiaya Jia received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2004. He joined the Department of Computer Science and Engineering at the Chinese University of Hong Kong in September 2004, where he is currently an assistant professor. His research interests include computer vision and graphics, pattern recognition, and image/video processing. He was the program committee member of the International Conference on Computer Vision (ICCV), Computer Vision and Pattern Recognition (CVPR), and Asian Conference on Computer Vision (ACCV). He is a member of the IEEE.



Chi-Keung Tang received the MS and PhD degrees in computer science from the University of Southern California (USC), Los Angeles, in 1999 and 2000, respectively. Since 2000, he has been with the Department of Computer Science, Hong Kong University of Science and Technology (HKUST), where he is currently an associate professor. He is an adjunct researcher at the Visual Computing Group, Microsoft Research Asia. His research interests include computer vision and computer graphics. He was on the program committees of the International Conference on Computer Vision (ICCV), Computer Vision and Pattern Recognition (CVPR), European Conf. Computer Vision (ECCV), and the International Conference on Pattern Recognition (ICPR) and served as an area chair for the Asian Conference on Computer Vision (ACCV) 2006 (Hyderabad) and ICCV 2007 (Rio de Janeiro). He is a senior member of the IEEE Computer Society.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.