Article

A Unified Framework for Street-View Panorama Stitching

Li Li [1], Jian Yao [1,*], Renping Xie [1], Menghan Xia [1] and Wei Zhang [2]

[1] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, P.R. China
[2] School of Control Science and Engineering, Shandong University, Jinan, Shandong, P.R. China
* Correspondence: jian.yao@whu.edu.cn; Tel.: +86-27-68771218; Web: http://cvrs.whu.edu.cn/

Version September 7, 2016 submitted to Remote Sens.; Typeset by LaTeX using class file mdpi.cls

Abstract: In this paper, we propose a unified framework to generate a pleasant and high-quality street-view panorama by stitching multiple panoramic images captured from the cameras mounted on the mobile platform. Our proposed framework is comprised of four major steps: image warping, color correction, optimal seamline detection and image blending. Since the input images are captured without a precisely common projection center from the scenes with the depth differences with respective to cameras to different extents, such these images cannot be precisely aligned in geometry. So, an efficient image warping method based on the dense optical flow field is proposed to greatly suppress the influence of large geometric misalignment at first. Then, to lessen the influence of photometric inconsistencies caused by the illumination variations and different exposure settings, we propose an efficient color correction algorithm via matching extreme points of histograms to greatly decrease color differences between warped images. After that, the optimal seamlines between adjacent input images are detected via the graph cuts energy minimization framework. At last, the Laplacian pyramid blending algorithm is applied to further eliminate the stitching artifacts along the optimal seamlines. Experimental results on a large set of challenging street-view panoramic images captured form the real world illustrate that the proposed system is capable of creating high-quality panoramas.

Keywords: Panorama Stitching ; Seamline Detection ; Image Warping ; Graph Cuts ; Image Parallax ; Image Blending ; Color Correction

1 Introduction

Nowadays, as the development of street-view panoramas which provide 360° panoramic views along streets in the real world, the demand for high-quality panoramic images gradually becomes bigger. Image stitching is the key technology to produce high-quality panoramic images, which is also an important and classical problem in the field of photogrammetry [1–5], remote sensing [6–9] and computer vision [10–15], which is widely used to merge multiple aligned images into a single wide-angle composite image as seamlessly as possible.

In an ideally static scene in which both the geometric misalignments and the photometric inconsistencies don't exist or are not obviously visible in overlap regions, the stitched or mosaicked image looks perfect only when the geometric distance criterion is used. However, as we know, most of street-view panoramic images are captured by the panoramic camera mounted on the mobile platform. Generally, the panoramic camera is comprised of multiple wide-angle or fish-eye cameras whose projection centers are slightly different. Therefore, those images cannot be precisely aligned in geometry, namely, there exist the geometric deviations for corresponding pixels from different images to different extents. In addition, there also exist photometric inconsistencies to different extents in overlap regions between adjacent images due to illumination variations and/or different exposure setting. This paper focuses on creating a visually pleasant street-view panorama by stitching or mosaicking the street-view panoramic images among which there may exist the severe geometric misalignments and the strong photometric inconsistencies.

One traditional and efficient way to eliminate the stitching artifacts caused by the large geometric misalignments existed in the input aligned panoramic images is to detect the optimal seamlines which avoid crossing majority of visually obvious objects and most of overlap regions with low image similarity and large object dislocation. The optimal seamline detection methods search for the seamlines in overlap regions between images where their intensity or gradient differences are not significant. Based on the optimally detected seamlines, multiple aligned images can be mosaicked into a single composite image in which the obvious image parallax caused by image misalignments can be magnificently concealed. Many methods [2–6,16–19] regarded the optimal seamline detection as an energy optimization problem and solved it by minimizing a specially designed energy function defined to represent the difference between the original images along the seamlines. For these methods, the key ideas concentrate on how to define the effective energy functions and how to guarantee the optimality of the solution. The energy functions are often defined by considering color, gradient and texture, and are optimized via different optimization algorithms, e.g., snake model [20], Dijkstra's algorithm [21], dynamic programming [22], and graph cuts [23]. Nowadays, the optimal seamline detected by many algorithms can avoid crossing the regions with low image similarity and high object dislocation. In our previous work presented in [19], we proposed an efficient optimal seamline detection algorithm for mosaicking aerial and panoramic images based on the graph cuts energy minimization framework. In this paper, we will apply this algorithm to detect the optimal seamlines.

However, when the geometric misalignments are very large, the stitching artifacts maybe cannot be completely avoided even though the optimal seamlines are detected, especially for street-view panoramic images among which there always exist geometric misalignments at different extents due to that those images were captured from the scenes with large depth differences by the panoramic camera comprised of multiple wide-angle or fish-eye cameras without a precisely common projection center, which means that the geometric misalignments are different at different positions. Therefore, the large geometric misalignments existed in the input aligned panoramic images should be eliminated as much as possible before finding the optimal seamlines. In this paper, we creatively propose an image warping algorithm based on the optical flow field to reduce the geometric misalignments between input panoramic images. Image warping is a transformation which maps all positions in one image plane to the corresponding ones in another plane [24], which has been popularly applied in many fields of computer vision, such as image morphing [25,26], image retargeting [27,28] and image mosaicking [29,30]. The key technique of image warping is to find the appropriate transformation functions based on the control conditions and then eliminate the distortions between input images. One famous image warping algorithm worked based on thin-plate splines [31] that attempted to minimize the amount of bending in the deformation. They used the radial basis functions with thin-plate splines to find a space deformation defined by control points. However, the local non-uniform scaling and shearing possibly occurred in the deformed images. [32] firstly introduced the concept of as-rigid-as-possible transformations, which have the property that both local scaling and shearing are very slight. To produce as-rigid-as-possible deformations, [33] proposed a point-based image deformation technique, which firstly triangulated the input image, and then geometrically minimized the distortion associated with each triangle. However, this algorithm needs to triangulate the input image at first, and the results are maybe not smooth across triangle boundaries. [34] provided an image deformation method based on Moving Least Squares [35] using various classes of linear functions including affine, similarity and rigid transformations. It first found the deformation functions based on the control points or the line segments, and then applied the deformation functions onto each grid instead of each pixel to reduce the transformation time. At last, it filled the resulting quads using the bilinear interpolation. [36] proposed an image warping algorithm based on radial basis functions, which formulated the image warping problem as the scattered data interpolation problem, and used the radial basis functions to construct the interpolation. It aimed to identify the best radial basis functions for image warping. Our used image warping method is similar to this algorithm, but we used the Multilevel B-Splines Approximation (MBA) [37] to solve the scattered data interpolation problem. Recently, the b-spline approximation technique has been widely used for image registration [38,39], image morphing, image warping, curve/surface fitting and geometric modeling.

In addition, due to the differences of both the image capturing viewpoints and the camera exposure settings, there are large differences of color and brightness between the warped panoramic images. The large color differences between those images also can cause the stitching artifacts in the last stitched or mosaicked panorama. Also, the large color differences maybe affect the quality of the seamlines. So, we also need to suppress the color differences between warped images before we apply the optimal seamline detection. Generally, the color correction approaches can be divided into two broad categories according to [40]: parametric and non-parametric. Panoramic approaches assume that the color relationship between images can be described by a certain model. Few noteworthy parametric approaches are described here. [41] proposed a simple linear model to transform the color of the source image to the target image. The transformation matrix was estimated by using the histogram mapping over the overlap regions. [12] applied the gain compensation (i.e., the diagonal model) to reduce color differences between input images. They computed all gains by minimizing an error function, which is the sum of gain normalised intensity errors for all overlapping pixels. [42] also employed the diagonal model for the color and luminance compensation where the correction coefficients were computed as the ratio of sum of pixel values in the overlap regions. As stated in [43], the linear transformation models can provide a simple yet effective way to transform colors, but they have clear limitations in explaining the complicated nonlinear transformations in the the imaging process. Non-parametric approaches can handle this problem well. Non-parametric approaches don't follow any particular model for the color mapping, and most of them use some form of a look-up table to record the mapping of the full range of color levels. As stated in [40], parametric approaches are more effective in extending the color in non-overlap regions without generating gain artifacts, while non-parametric approaches can provide better color matching results. [44] proposed to use the joint histogram of correspondences matched using the SIFT features [45] to correct the color differences. The color mapping function was estimated by using an energy minimization scheme. [46] proposed a color correction approach by using the cumulative color histogram. This method used the cumulative histogram-based mapping to automatically adapt the color of all source images to the reference image. [43] presented a nonlinear and nonparametric color transfer framework that operates in a 3D color space. Based on some control corresponding colors in a given image pair, this method used the probabilistic moving least squares to interpolate the transformation functions for each color. We correct the color differences between two images based on the matched extreme points which are extracted from the histograms over the overlap regions. Both the Probability Density Functions (PDFs) and Cumulative Distribution Functions (CDFs) are used to find the reliably matched extreme points. To reduce the gain artifacts in non-overlap regions, we propose to apply the alpha correction method to smooth the transition from non-overlap regions to overlap ones.

Although we propose efficient approaches to correct the color differences and detect the optimal seamlines between warped panoramic images, there maybe also exist some color transitions along the seamlines due to that the color differences cannot be eliminated completely. In order to further conceal these artifacts, the image blending techniques can be further applied along the seamlines. In the last several decades, many image blending algorithms have been proposed to smooth the color differences along the seamlines, such as feathering [47], alpha blending [48], Laplacian pyramid blending [49], poisson blending [50] and gradient domain image blending approach [51]. In this

Figure 1. Our proposed unified framework for the street-view panorama stitching system.

paper, we simply applied the Laplacian pyramid blending algorithm [49] to eliminate the stitching artifacts and generate the last pleasant panorama.

In this paper, we propose a unified framework for our developed street-view panorama stitching system, as described in Figure 1. First, multiple original images, which were captured from a single panoramic camera comprised of multiple wide-angle or fish-eye cameras (usually digital SLR cameras) without a precisely common projection center, are fed into our stitching system as the input. Therefore, we will align these input images into a common spherical coordinate system based on the found feature correspondences using the existing open-source library. After that, our proposed image warping method based on the dense optical flow field approximately interpolated from the sparse feature matches, which is detailed described in Section 2, is used to greatly reduce the geometric misalignments. Then, an automatic contrast adjustment and an efficient histogram-matching-based color correction approach presented in Section 3 are used to reduce the color differences. Finally, we adopt an efficient seamline detection approach based on the graph cuts energy minimization framework to find the optimal seamlines between two overlapped images followed by applying the image blending to eliminate the color transitions along the seamlines. By our proposed unified panorama stitching framework, our system can generate a pleasant street-view panorama as seamless as possible by stitching multiple panoramic images from the cameras mounted on the mobile platform. Experimental results on challenging street-view panoramic images are reported in Section 5 followed by the conclusions drawn in Section 6.

2 Image Warping

In our developed street-view panorama stitching system, we first check whether all input images are geometrically aligned into a common spherical coordinate system. If not, we will align them by using the open-source library *PanoTools* [1], which is also served as the underlying core engine for many image stitching softwares, such as *PTGui* [2] and *Hugin* [3]. However, there always exist large geometric misalignments between these aligned images at different extents because those images were captured from the scenes of large depth differences by a single panoramic camera comprised of multiple wide-angle or fish-eye cameras without a precisely common projection center. Those geometric misalignments are so large that the stitching artifacts cannot be avoided completely even though the optimal seamlines are detected out for the use of the image stitching. To ensure the high-quality of the last stitched panorama, we propose to apply the image warping technique to eliminate those large geometric misalignments as far as possible. To describe our proposed image warping algorithm more clearly, we first consider a simple case of two aligned images $\mathbf{I}$ and $\mathbf{I}'$ with an overlap. The process of our proposed image warping algorithm is described as follows. Firstly, the corresponding points between two images are found as the control points of image warping, and the sparse optical flows are calculated for those control points. Secondly, the Multilevel B-Splines Approximation (MBA) algorithm [37] is used to approximately interpolate the dense optical flows for all integral pixels in the warped image with respective to the original one from the sparse optical flows. Lastly, we warp the input two images based on the dense optical flows and thus the geometric misalignments can be greatly lessened. For the case of multiple images to be stitched to the last panorama, a simple strategy is proposed to first handle the horizontal images and then deal with the vertical ones.

2.1 Feature Point Matching

To warp two images with large geometric misalignments, we need to find the control points at first. The quality of the warped image mainly depends on the accuracy and densities of control points. In this paper, we apply the feature matching algorithm to robustly find the sparse matching points, namely the control points. The main ideal for feature matching is to first extract local invariant features independently from two images and then characterize them by invariant descriptors. The distance between two descriptor vectors is used to identify candidate matches. However, the nearest neighbors is not always the best match due to occlusion and deformation derived from large viewpoint changes and repeated structures in the scenes. Generally, the epipolar geometrical constraint works well to filter the outliers, but it is not available for the panoramic images aligned in advance. Thus, we need to apply another strategy to filter the outliers. [52] proposed that the motion of one match would be consistent with those of neighbors, and the experimental results presented in [52] sufficiently show that this strategy is simple but effective. Inspired by this idea, we propose a new feature matching algorithm for panoramic images. The major steps of the proposed algorithm include initial matching and outlier detection, which are summarized in Algorithm 1. An example of finding point correspondences between two panoramic images with an overlap is illustrated in Figure 2.

2.1.1 Initial Matching

Given two adjacent images $\mathbf{I}$ and $\mathbf{I}'$ with an overlap, the local invariant features are extracted and described by the SURF algorithm [53]. Let $\mathbf{f} = (\mathbf{x}, \mathbf{d})$ be a feature point where $\mathbf{x} = (x, y)^\top$ denotes the 2D coordinate of this feature point and $\mathbf{d}$ represents its corresponding invariant descriptor vector, and $\mathcal{F} = \{\mathbf{f}_i | \mathbf{f}_i = (\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^{M}$ and $\mathcal{F}' = \{\mathbf{f}'_j | \mathbf{f}'_j = (\mathbf{x}'_j, \mathbf{d}'_j)\}_{j=1}^{N}$ be the feature point sets extracted from $\mathbf{I}$ and $\mathbf{I}'$, respectively, where M and N denote the numbers of the feature points extracted from $\mathbf{I}$ and $\mathbf{I}'$,

1 Available at http://www.panoramatools.com/
2 Available at http://www.ptgui.com/
3 Available at http://hugin.sourceforge.net/

Algorithm 1 The proposed feature point matching algorithm.

1. ***Initial Matching***
   (a) Extract and describe two sets of local invariant features from two overlapped images $\mathbf{I}$ and $\mathbf{I}'$ by using the SURF algorithm, respectively;
   (b) Find the initial point matches $\mathcal{M}_{\text{initial}}$ between $\mathbf{I}$ and $\mathbf{I}'$ according to the conditions listed in Section 2.1.1.
2. ***Outlier Detection***
   (a) Set the value of λ used in Eq. (3) as λ_{max};
   (b) Find the neighboring inlier matches $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$ for each match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle \in \mathcal{M}_{\text{initial}}$, and then calculate the mean motion $\boldsymbol{\mu}(\mathbf{f}_p)$ and the standard deviation $\boldsymbol{\sigma}(\mathbf{f}_p)$ of all matches in $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$;
   (c) Sort all matches in $\mathcal{M}_{\text{initial}}$ in the decreasing order according to their costs defined in Eq. (4), and only check whether the top N_t matches are inliers or outliers in each iteration;
   (d) Iterate the steps (b)-(c) until the maximum number of iterations is reached or no more outliers can be found in the current iteration.;
   (e) Decrease the value of λ with the step λ_{step} and iterate the steps (b)-(d) until the smallest value λ_{min} is reached.

respectively. Generally, for one feature point $\mathbf{f}_p$ in $\mathcal{F}$, the feature point $\mathbf{f}'_q$ with the nearest Euclidean distance $d(\mathbf{f}_p, \mathbf{f}'_q) = \min_{\mathbf{f}'_j \in \mathcal{F}'} ||\mathbf{d}_p - \mathbf{d}'_j||$ which is not larger than a predefined threshold T_d can be regarded as the corresponding matching point of $\mathbf{f}_p$. However, this simple strategy has some drawbacks in the context of feature matching. This mainly because that the distance values between different corresponding pairs may vary in a relatively large range, so any permissive distance threshold T_d can not avoid the appearance of high rate outliers when covers most of the good correspondences. Thus, we propose to modify the matching strategy as follows. In this paper, we accept two feature points $\mathbf{f}_p$ and $\mathbf{f}'_q$ as a potential match only when they satisfy the following conditions:

- The feature points $\mathbf{f}_p \in \mathcal{F}$ and $\mathbf{f}'_q \in \mathcal{F}'$ are the nearest neighbors of each other. Namely, for the feature point $\mathbf{f}_p$, $\mathbf{f}'_q$ is its nearest neighbor in $\mathcal{F}'$. At the same time, for the feature point $\mathbf{f}'_q$, $\mathbf{f}_p$ is its nearest neighbor in $\mathcal{F}$.
- The Euclidean descriptor vector distance $d(\mathbf{f}_p, \mathbf{f}'_q)$ between two feature points $\mathbf{f}_p$ and $\mathbf{f}'_q$ is not larger than T_d, i.e., $d(\mathbf{f}_p, \mathbf{f}'_q) = \|\mathbf{d}_p - \mathbf{d}'_q\| \leq T_d$.
- We represent the nearest distance between $\mathbf{f}_p$ and $\mathcal{F}'$ as $d_1(\mathbf{f}_p, \mathcal{F}') = d(\mathbf{f}_p, \mathbf{f}'_q) = \min_{\mathbf{f}'_j \in \mathcal{F}'} ||\mathbf{d}_p - \mathbf{d}'_j||$ and the next distance as $d_2(\mathbf{f}_p, \mathcal{F}') = \min_{\mathbf{f}'_j \in \mathcal{F}', \mathbf{f}'_j \neq \mathbf{f}'_q} ||\mathbf{d}_p - \mathbf{d}'_j||$, respectively. The distance ratio $r(\mathbf{f}_p, \mathcal{F}') = d_1(\mathbf{f}_p, \mathcal{F}')/d_2(\mathbf{f}_p, \mathcal{F}')$ should be smaller than the predefined threshold T_r. Similarly, for the feature point $\mathbf{f}'_q$, the distance ratio $r(\mathbf{f}'_q, \mathcal{F}) = d_1(\mathbf{f}'_q, \mathcal{F})/d_2(\mathbf{f}'_q, \mathcal{F})$ should be smaller than T_r too.

By this matching strategy, we obtain a set of initial matches denoted as $\mathcal{M}_{\text{initial}} = \{\langle \mathbf{f}_p, \mathbf{f}'_q \rangle | \mathbf{f}_p \in \mathcal{F}, \mathbf{f}'_q \in \mathcal{F}'\}$.

2.1.2 Outlier Detection

After initial matching, there maybe still exist a few outliers in $\mathcal{M}_{\text{initial}}$. Of course, we need to filter out those outliers. The widely used constraint of the epipolar geometric constraint cannot be efficiently used in panoramic images, especially when the panoramic images have been aligned into a common spherical coordinate system in advance. According to the assumption proposed by [52] that the matches in a small neighborhood tend to have the consistent location changes (i.e., motions). In this paper, we also apply this assumption to identify the outliers.

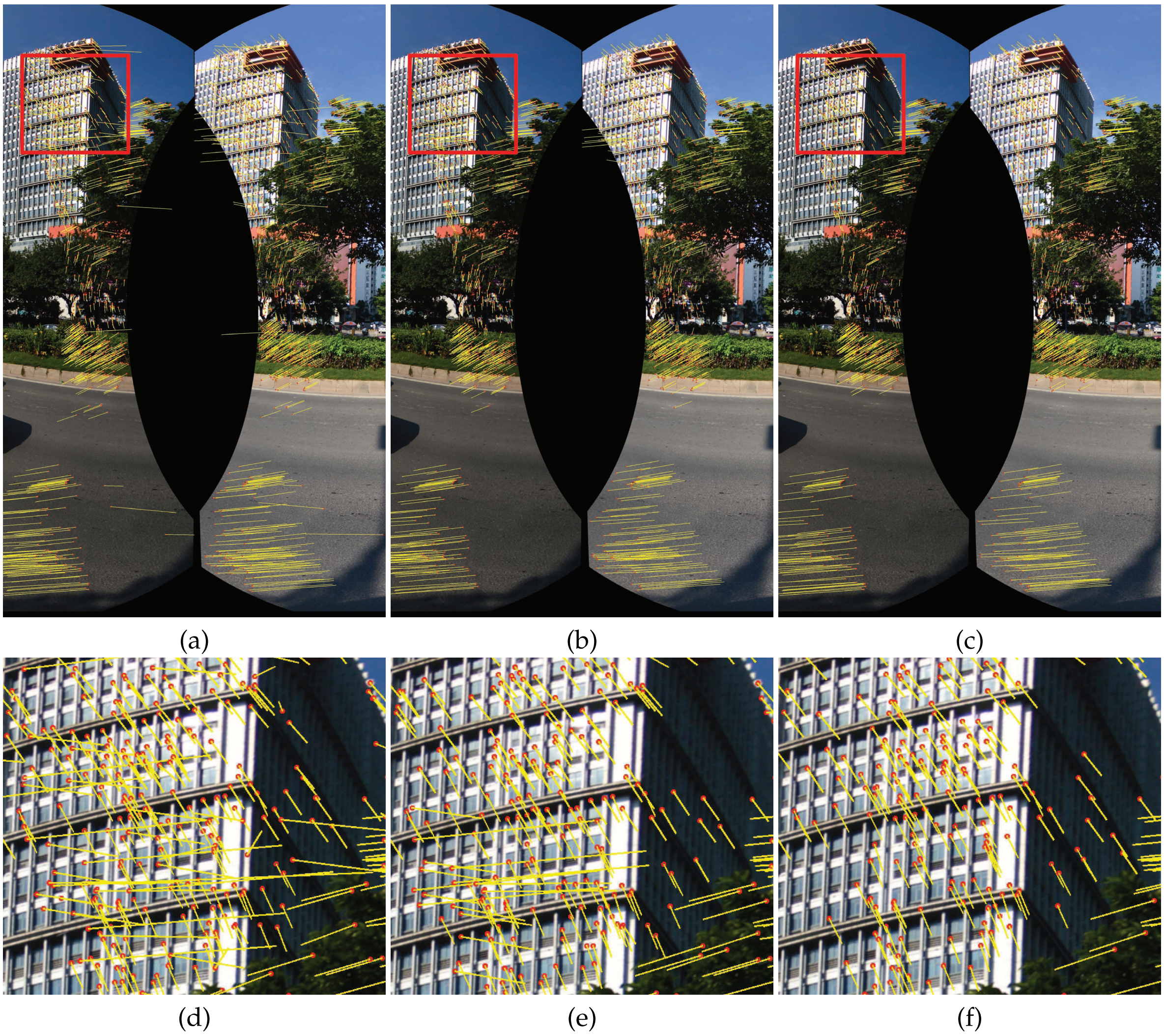


Figure 2. An illustrative example for feature point matching between two aligned panoramic images: (a) the point correspondences produced by initial matching; (b)-(c) the filtered point correspondences after the first iteration ($\lambda = 6$) and the last one ($\lambda = 3$) of outlier detection, respectively. The local detailed regions of (a)-(c) are presented in (d)-(f), respectively. The red circles denote the positions of the matched points in the current image points and the yellow lines represent the optical flows (i.e., motions) of the matched points in the current image with respective to another image.

Given a match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle$, the motions from $\mathbf{f}_p$ to $\mathbf{f}'_q$ along the horizontal direction and the vertical one are calculated, respectively, as follows:

$$\begin{cases} m_p^{(x)} = x'_q - x_p, \\ m_p^{(y)} = y'_q - y_p, \end{cases} \tag{1}$$

where $\mathbf{f}_p = (x_p, y_p)^\top$ and $\mathbf{f}'_q = (x'_q, y'_q)^\top$. Thus, the magnitude value of the motion vector $(m_p^{(x)}, m_p^{(y)})^\top$ can be calculated as:

$$m_p^{(x,y)} = \sqrt{\left(m_p^{(x)}\right)^2 + \left(m_p^{(y)}\right)^2}. \tag{2}$$

Here, we use $\mathbf{m}(\mathbf{f}_p) = \left(m_p^{(x)}, m_p^{(y)}, m_p^{(x,y)}\right)^\top$ to represent all the three motion components of the match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle$.

At first, we assign the labels of all matches as *Inlier*, namely, for each match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle \in \mathcal{M}_{\text{initial}}$, the label $\mathcal{L}(\langle \mathbf{f}_p, \mathbf{f}'_q \rangle) = \textit{Inlier}$, and then we iteratively find the outliers. For each match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle \in \mathcal{M}_{\text{initial}}$, we find K_n ($K_n = 60$ was used in this paper) neighboring match points of $\mathbf{f}_p$ from $\mathcal{F}$ denoted as the set $\mathcal{N}(\mathbf{f}_p)$. Then we collect all matches whose labels are *Inlier* from $\mathcal{N}(\mathbf{f}_p)$ as a new set $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$. If the number of inliers in $\mathcal{N}(\mathbf{f}_p)$, namely, the size of $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$, is less than K_i ($K_i = 10$ was used in this paper), we directly label this match as an *Outlier*, namely, $\mathcal{L}(\langle \mathbf{f}_p, \mathbf{f}'_q \rangle) = \textit{Outlier}$, otherwise, we determine whether this match is an inlier by checking whether it has the consistent motion with its neighbors $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$. For each match $\langle \mathbf{f}_m, \mathbf{f}'_n \rangle \in \mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$, the motion $\mathbf{m}(\mathbf{f}_m)$ from $\mathbf{f}_m$ to $\mathbf{f}'_n$ can be calculated according to both Eq. (1) and Eq. (2). Then, the mean motion $\boldsymbol{\mu}(\mathbf{f}_p) = \left(\mu_p^{(x)}, \mu_p^{(y)}, \mu_p^{(x,y)}\right)^\top$ and the standard deviation of all the motions $\boldsymbol{\sigma}(\mathbf{f}_p) = \left(\sigma_k^{(x)}, \sigma_k^{(y)}, \sigma_k^{(x,y)}\right)^\top$ of all match points in $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$ can be determined easily. According to the following measurement proposed by [52], the label of the match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle$ can be determined as follows:

$$\mathcal{L}(\langle \mathbf{f}_p, \mathbf{f}'_q \rangle) = \begin{cases} \textit{Inlier}, & \textit{dist}(\mathbf{m}(\mathbf{f}_p), \boldsymbol{\mu}(\mathbf{f}_p)) \leq \lambda \times \boldsymbol{\sigma}(\mathbf{f}_p), \\ \textit{Outlier}, & \text{Otherwise}, \end{cases} \tag{3}$$

where $\textit{dist}(\mathbf{m}(\mathbf{f}_p), \boldsymbol{\mu}(\mathbf{f}_p)) = |\mathbf{m}(\mathbf{f}_p) - \boldsymbol{\mu}(\mathbf{f}_p)|$ denotes the absolute distances in three components between the motion $\mathbf{m}(\mathbf{f}_p)$ of the match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle$ and the mean motion $\boldsymbol{\mu}(\mathbf{f}_p)$ of its neighbor matches, and λ is a predefined parameter. However, this measurement has two following drawbacks. To overcome these two drawbacks, we propose the corresponding strategies.

One drawback is that the inliers around the outliers may also be labeled as *Outlier*. As shown in Figure 3, apparently, the black point with the inconsistent motion with its neighbors is the outlier, we can remove it easily according to the measurement defined in Eq. (3). However, due to the existence of this outlier, the inliers (marked in red points) around it maybe also have large deviations with respective to the corresponding mean motions. So, those inliers may also be regarded as outliers. But, if we remove the black point as an outlier at first, the deviations with respective to the mean motions of the rest red points will be decreased dramatically and can all be labeled as *Inlier* certainly. Thus, for each match $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle$, we first evaluate its cost of assigning this match as *Outlier* as follows:

$$\begin{aligned} \textit{Cost}(\mathbf{f}_p) \quad = \quad & \frac{\mu_p^{(x)} - \mu_{\text{min}}^{(x)}}{\mu_{\text{max}}^{(x)} - \mu_{\text{min}}^{(x)}} + \frac{\mu_p^{(y)} - \mu_{\text{min}}^{(y)}}{\mu_{\text{max}}^{(y)} - \mu_{\text{min}}^{(y)}} + \frac{\mu_p^{(x,y)} - \mu_{\text{min}}^{(x,y)}}{\mu_{\text{max}}^{(x,y)} - \mu_{\text{min}}^{(x,y)}} + \\ & \frac{\sigma_p^{(x)} - \sigma_{\text{min}}^{(x)}}{\sigma_{\text{max}}^{(x)} - \sigma_{\text{min}}^{(x)}} + \frac{\sigma_p^{(y)} - \sigma_{\text{min}}^{(y)}}{\sigma_{\text{max}}^{(y)} - \sigma_{\text{min}}^{(y)}} + \frac{\sigma_p^{(x,y)} - \sigma_{\text{min}}^{(x,y)}}{\sigma_{\text{max}}^{(x,y)} - \sigma_{\text{min}}^{(x,y)}}, \end{aligned} \tag{4}$$

where $\mu_{\text{min}}^{(x)} = \min\limits_p \mu_p^{(x)}$, $\sigma_{\text{min}}^{(x)} = \min\limits_p \sigma_p^{(x)}$, $\mu_{\text{max}}^{(x)} = \max\limits_p \mu_p^{(x)}$ and $\sigma_{\text{max}}^{(x)} = \max\limits_p \sigma_p^{(x)}$ denote the minimum and maximal mean and standard deviations in the x component of all match points in $\mathcal{M}_{\text{initial}}$, respectively, and others have the same meanings. Apparently, the black point (outlier) shown in Figure 3 has a bigger cost, and the red points (inliers) have smaller costs. Then, all match points are sorted in the decreasing order according to their costs defined in Eq. (4). In each iteration, we only check whether the top N_t matches are inliers or outliers according to the measurement defined in Eq. (3), where $N_t = \rho \times N_{\text{inlier}}(\mathcal{M}_{\text{initial}})$, ρ is the predefined proportion parameter ($\rho = 0.025$ was used in this paper) and $N_{\text{inlier}}(\mathcal{M}_{\text{initial}})$ denotes the number of matches with the label *Inlier* in $\mathcal{M}_{\text{initial}}$. We end up the iterations until the maximum number of iterations is reached or no more matches can be labeled as *Outlier*. By this way, the ouliers with larger deviations with respective to neighboring points will be robustly filtered out step-by-step.

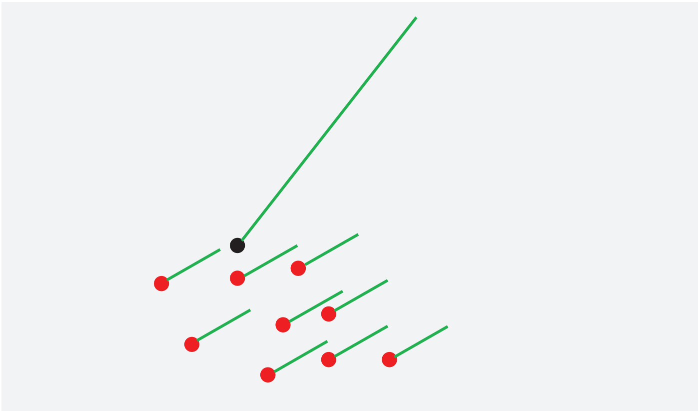


Figure 3. An visual example of outliner detection. The black point means an outlier, the red points mean inliers, and the green lines represent the motions of corresponding match points.

Another drawback is that the value of the parameter λ in Eq. (3) is difficult to be determined. If the value of λ is small, many inliers may be assigned as *Outlier*. In contrast, if the value of λ is big, many outliers may be assigned as *Inlier*. Thus, in this paper, we iteratively decrease the value of λ from λ_{max} to λ_{min} with the step λ_{step} ($\lambda_{\text{max}} = 6$ and $\lambda_{\text{min}} = 3$, $\lambda_{\text{step}} = 3$ were used in this paper). Namely, we first set $\lambda = \lambda_{\text{max}}$, and perform the outlier detection process until no more outliers can be found. Then, we iteratively decrease the value of λ with a step λ_{step}, and repeat the outlier detection process until the value of λ reaches to λ_{min}. At last, we can find all inliers from $\mathcal{M}_{\text{initial}}$ denoted as the set $\mathcal{M}_{\text{inlier}} = \{\langle \mathbf{f}_m, \mathbf{f}'_n \rangle | \mathbf{f}_m \in \mathcal{F}, \mathbf{f}'_n \in \mathcal{F}'\}$.

2.2 Approximate Interpolation of Dense Optical Flows

Let $\bar{\mathbf{I}}$ and $\bar{\mathbf{I}}'$ be the warped images of two adjacent images $\mathbf{I}$ and $\mathbf{I}'$, respectively. The aim of our proposed image warping algorithm is to ensure that the geometric alignments between the warped images $\bar{\mathbf{I}}$ and $\bar{\mathbf{I}}'$ become smaller. To achieve this objective, we propose to approximately interpolate the optical flows of all the integral pixels in $\bar{\mathbf{I}}$ with respective to $\mathbf{I}$ and all the integral pixels in $\bar{\mathbf{I}}'$ with respective to $\mathbf{I}'$ based on the disparity vectors of the reliable point matches with respective to each other as the control points. Firstly, we calculate the disparity vectors $\mathbf{d}(\mathbf{x}_m)$ and $\mathbf{d}(\mathbf{x}'_n)$ of each reliable point match $\langle \mathbf{x}_m, \mathbf{x}'_n \rangle$ in $\mathcal{M}_{\text{inlier}}$ from the warped images to the original ones as follows:

$$\begin{cases} \mathbf{d}(\mathbf{x}_m) = \dfrac{1}{2}(\mathbf{x}_m - \mathbf{x}'_n) = \dfrac{1}{2}(x_m - x'_n, y_m - y'_n)^\top, \\ \mathbf{d}(\mathbf{x}'_n) = \dfrac{1}{2}(\mathbf{x}'_n - \mathbf{x}_m) = \dfrac{1}{2}(x'_n - x_m, y'_n - y_m)^\top, \end{cases} \tag{5}$$

where $\mathbf{x}_m = (x_m, y_m)^\top$ and $\mathbf{x}'_n = (x'_n, y'_n)^\top$. By this way, we expect to warp the images $\mathbf{I}$ and $\mathbf{I}'$ based on the half offsets of real disparity vectors to reduce the warping distortion.

Secondly, we propose to approximately interpolate the optical flows of all integral pixels in the warped images $\bar{\mathbf{I}}$ and $\bar{\mathbf{I}}'$ based on the disparity vectors $\{\mathbf{d}(\mathbf{x}_m)\}_{\mathbf{x}_m \in \mathcal{M}_{\text{inlier}}}$ and $\{\mathbf{d}(\mathbf{x}'_n)\}_{\mathbf{x}'_n \in \mathcal{M}_{\text{inlier}}}$ of the control points $\{\mathbf{x}_m - \mathbf{d}(\mathbf{x}_m)\}_{\mathbf{x}_m \in \mathcal{M}_{\text{inlier}}}$ and $\{\mathbf{x}'_n - \mathbf{d}(\mathbf{x}'_n)\}_{\mathbf{x}'_n \in \mathcal{M}_{\text{inlier}}}$, respectively. This problem can be formulated as the scattered data interpolation problem. Due to the sparsity of the control points, in this paper we adapt to apply the Multilevel B-Splines Approximation (MBA) [37] to solve this problem, which has been widely used for image registration, image morphing, image warping, curve/surface fitting and geometric modeling. By this MBA interpolation, we separately interpolate the horizontal and vertical components of optical flows (i.e., disparity vectors) of all the integral pixels in $\bar{\mathbf{I}}$ and $\bar{\mathbf{I}}'$, respectively. In this way, we finally obtain the dense optical flows $\mathcal{D}(\bar{\mathbf{I}}) = \{\tilde{\mathbf{d}}(\mathbf{p})\}_{\mathbf{p} \in \bar{\mathbf{I}}}$ and

$\mathcal{D}(\bar{\mathbf{I}}') = \{\tilde{\mathbf{d}}(\mathbf{p}')\}_{\mathbf{p}'\in\bar{\mathbf{I}}'}$ of all the integral pixels $\{\mathbf{p}\}_{\mathbf{p}\in\mathbf{I}}$ and $\{\mathbf{p}'\}_{\mathbf{p}'\in\mathbf{I}'}$ in the warped images $\bar{\mathbf{I}}$ and $\bar{\mathbf{I}}'$ with respective to the original images $\mathbf{I}$ and $\mathbf{I}'$, respectively.

2.3 Two Image Warping

Here, we demonstrate how to generate the warped image $\bar{\mathbf{I}}$ from the original image $\mathbf{I}$ based on the dense optical flows $\mathcal{D}(\bar{\mathbf{I}})$ of $\bar{\mathbf{I}}$ with respective to $\mathbf{I}$, and the generation of the warped image $\bar{\mathbf{I}}'$ is similar. For each pixel $\mathbf{p} \in \bar{\mathbf{I}}$, we can easily calculate its corresponding 2D position in $\mathbf{I}$ based on its approximately interpolated optical flow (i.e., disparity vector) $\tilde{\mathbf{d}}(\mathbf{p})$ as $\mathbf{p} + \tilde{\mathbf{d}}(\mathbf{p})$. Then, we use the bilinear interpolation algorithm to interpolate the intensity of the corresponding point $\mathbf{p} + \tilde{\mathbf{d}}(\mathbf{p})$ in $\mathbf{I}$ as the intensity of the integral pixel $\mathbf{p} \in \bar{\mathbf{I}}$.

According to the above image warping procedure, we can obtain two warped images from two input panoramic images with the overlap. The geometric misalignments between warped images become smaller than those between the original images after warping correction.

2.4 Multiple Image Warping

Until now, we have introduced how to warp two images based on the optical flows. But, we need to warp multiple input images to generate the last panorama. In the experimental results presented in this paper, the input images are comprised of 5 horizontal ones and 1 vertical one, which are represented as $(\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3, \mathbf{I}_4, \mathbf{I}_5, \mathbf{I}_6)$ whose correspondingly warped images are represented as $(\bar{\mathbf{I}}_1, \bar{\mathbf{I}}_2, \bar{\mathbf{I}}_3, \bar{\mathbf{I}}_4, \bar{\mathbf{I}}_5, \bar{\mathbf{I}}_6)$, and the overlap relationship of those images is shown in Figure 9. For this particular case, here we will detailedly introduce how to warp these six images for producing the last panorama before color correction. Other cases of multiple images can be handled in a similar way. For multiple input panoramic images, we first collect all image pairs according to their overlap relationship as shown in Figure 9. Obviously, there are five image pairs along the horizontal direction, and one image pair along the vertical direction. We first handle the horizontal image pairs and then deal with the vertical image pair. For each horizontal image pair, we match them one by one by the method presented in Section 2.1, as an illustrative example shown in Figure 4, from which we can find that one horizontal image is overlapped with two adjacent images in the horizontal direction. For example, for the image $\mathbf{I}_1$, it overlaps with $\mathbf{I}_2$ and $\mathbf{I}_5$, respectively, so we need to collect all matching points from these two overlap regions as the control points for warping $\mathbf{I}_1$. The dense optical flow field of the warped image $\bar{\mathbf{I}}_1$ with respective to the original image $\mathbf{I}_1$ can be approximately interpolated based on those control points via the MBA algorithm. Therefore, five horizontal warped images $\bar{\mathbf{I}}_1$, $\bar{\mathbf{I}}_2$, $\bar{\mathbf{I}}_3$, $\bar{\mathbf{I}}_4$ and $\bar{\mathbf{I}}_5$ can be generated by warping their corresponding original images according to the method presented in Section 2.3, respectively. Figure 5 shows an example for warping one horizontal image. After that, we generate the bottom blended image $\mathbf{I}_H$ by blending all horizontal warped images according to the proposed color correction method presented in Section 3 and the adopted image mosaicking strategy described in Section 4. Finally, to produce the last panorama, the top image $\mathbf{I}_6$ and the horizontal blended image $\mathbf{I}_H$ will be warped according to those matching points as the control ones.

3 Color Correction

The large geometric misalignments can be efficiently eliminated by our proposed image warping algorithm, but there also exist the color differences between the warped images, so the stitching artifacts are still visible. Generally, the image blending technique can solve it easily by smoothing the color along the seamlines. However, it does not work well for input images with very large color differences. The simple image blending maybe can not efficiently conceal the artifacts if we don't magnificently correct color differences between images in advance, which results in low-quality

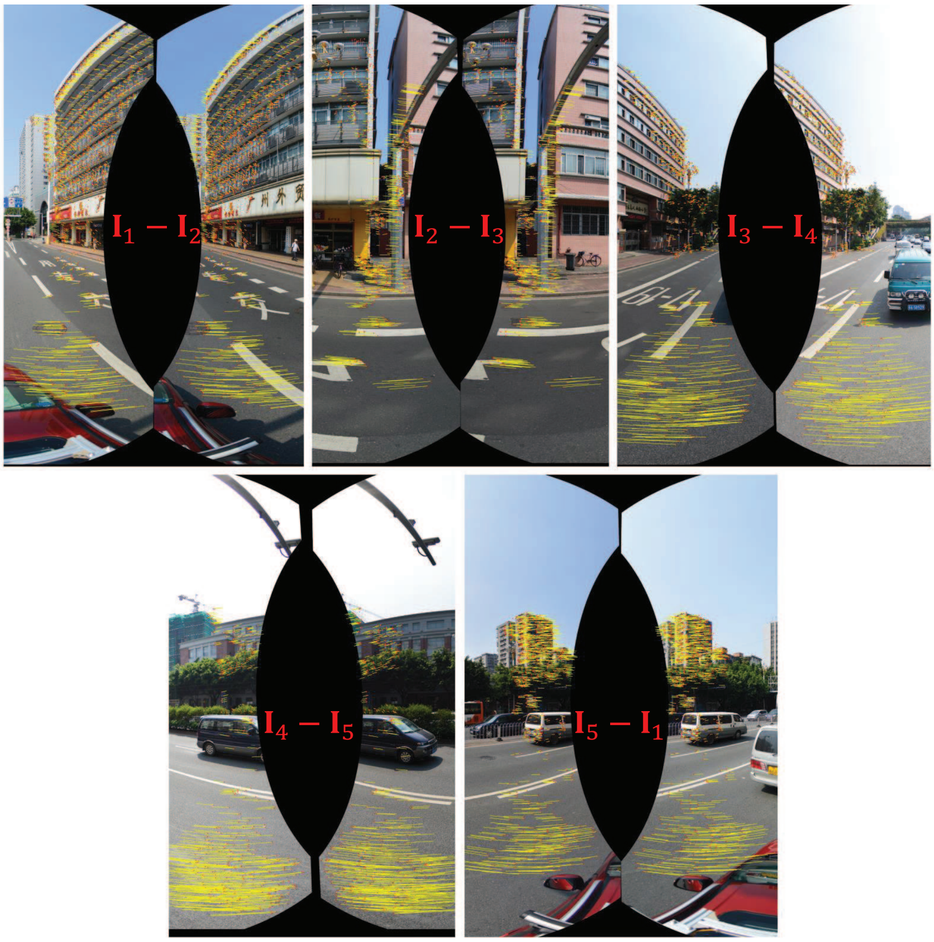
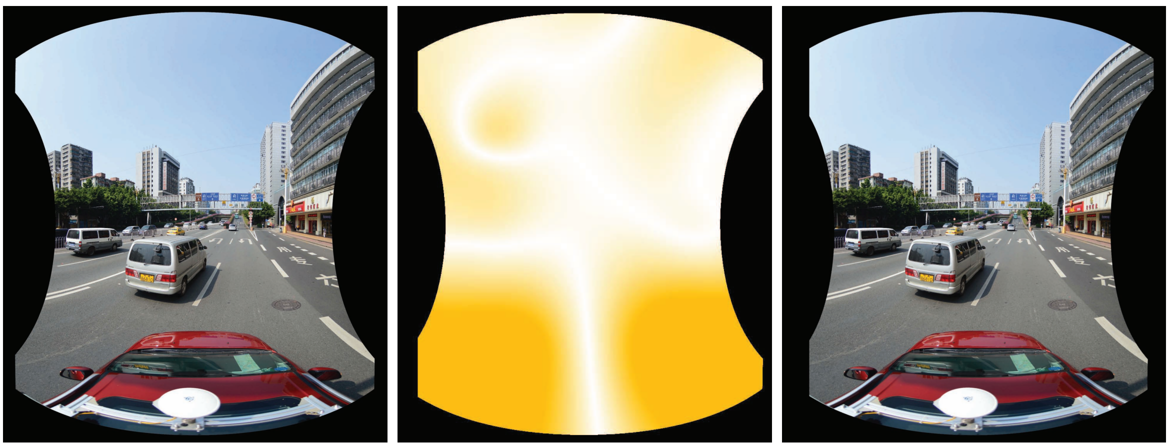


Figure 4. The feature matching results of all five horizontal image pairs in the overlap regions.



(a) (b) (c)

Figure 5. An illustrative example of image warping: (a) the original aligned image; (b) the dense optical flows approximately interpolated by the MBA algorithm; (c) the last warped image. In (b), the deeper orange means the larger disparity.

panoramic images, as an illustrative example shown in Figure 6. In addition, the large color differences maybe also affect the quality of the detected seamlines. Thus, in this paper, we propose to reduce the color differences between warped images before the optimal seamlines are detected.

(a) the initial stitched image (b) only image blending used (c) image blending after color correction

Figure 6. An example of our proposed color correction strategy used to improve the panorama stitching quality before apply the image blending.

Generally, the color differences should be also corrected before the image warping step to ensure the quality of feature matching results. But our adopted SURF feature matching algorithm is robust enough to the large photometric inconsistencies, so there are no obvious influence on our algorithm if we apply the color correction after image warping.

In this paper, we first apply the automatic contrast adjustment to reduce the brightness differences between images and then propose a novel and efficient color correction algorithm via matching extreme points of intensity histograms to further reduce the color differences. For the overlap image regions between two images, we construct their own Probability Density Functions (PDFs) and Cumulative Distribution Functions (CDFs) with respect to the intensity histograms in the three HSV channels, respectively. One way to eliminate color differences is to ensure that the three CDFs of the overlap regions in the first image in the three HSV channels are approximately same to those CDFs of the overlap regions in the second image, respectively. Obviously, we can correct the CDFs based on several uniformly spaced knots as [54] did. However, due to the existence of geometric misalignments, the scenes presented by two images in the overlap regions are not completely consistent. To solve this problem, we replace the knots by the matched extreme points extracted from the two PDFs. If the number of matched extreme points is not sufficient, we will suitably introduce those uniformly space knots. At last, the intensities of all the pixels in the two images are modified afterwards based on the matched extreme points extracted from the PDFs, not only for the pixels in the overlap regions, but also in the non-overlap regions.

3.1 Automatic Contrast Adjustment

At first, in order to make sure that multiple images have the similar contrast, which can produce satisfactory blending results, the three RGB channels of individual images are automatically adjusted in contrast. The histograms of a color image are calculated firstly in each of the three RGB channels, respectively. Let $\mathbf{I}$ be a single-channel image and $\mathbb{I} = \{I_k\}_{k=1}^{N}$ be a set of one dimensional sorted intensities of all valid pixels in $\mathbf{I}$ in the ascending order where N denotes the total number of valid pixels in $\mathbf{I}$ and I_k represents the intensity of the k-th sorted pixel in $\mathbf{I}$. The minimal and maximal intensities I_{min} and I_{max} in $\mathbf{I}$ are defined, respectively, as follows:

$$I_{\text{min}} = I_{\lceil N \times c\% \rceil} \text{ and } I_{\text{max}} = I_{\lceil N \times (1-c\%) \rceil}, \tag{6}$$

where $\lceil \Delta \rceil$ denotes the upper integer of a real value Δ and c is a small percentage value in the range of $(0, 50)$ ($c = 0.1$ was empirically used in this paper), which can be used to skip over a part of the real minimal and maximal intensities due to the fact that these pixels may be caused by noises and information lacking in most cases. The minimal and maximal intensity values of the R, G and B channels of a color image are denoted as $R_{\min}$, $G_{\min}$, $B_{\min}$, $R_{\max}$, $G_{\max}$, and $B_{\max}$, respectively. The minimal and maximal intensity values of the whole color image are defined as $V_{\min} = \min(R_{\min}, G_{\min}, B_{\min})$ and $V_{\max} = \max(R_{\max}, G_{\max}, B_{\max})$, respectively. Therefore, any intensity I of the R, G and B channels of a color image will be modified as:

$$I' = \begin{cases} 0, & I \leq V_{\min}, \\ 255 \times \frac{I - V_{\min}}{V_{\max} - V_{\min}}, & V_{\min} < I < V_{\max}, \\ 255, & I \geq V_{\max}. \end{cases} \tag{7}$$

In the same way, all the images to be used for creating a panorama will be automatically adjusted in contrast, which will slightly reduce the brightness differences between images.

3.2 Finding Extreme Points

After applying the automatic contrast adjustment on the multiple panoramic images, we propose to further reduce the color differences between panoramic images by matching extreme points of histograms. For the statistic analysis, only valid pixels in the overlap regions between two images are considered. Let $\mathbf{A}$ and $\mathbf{B}$ be the overlap image regions in two images, respectively. To make a better description of the information hidden behind the image, we convert $\mathbf{A}$ and $\mathbf{B}$ from the original RGB color space to the HSV color space, respectively. For each channel of $\mathbf{A}$ and $\mathbf{B}$, we calculate their PDFs and CDFs, which are denoted as $\mathbf{PDF_A}$, $\mathbf{PDF_B}$, $\mathbf{CDF_A}$, and $\mathbf{CDF_B}$, respectively.

To robustly find extreme points in both $\mathbf{PDF_A}$ and $\mathbf{PDF_B}$, these two PDFs are smoothed first by a Gaussian function to suppress possible noise. The initial local extreme points can be easily obtained from the smoothed $\mathbf{PDF_A}$ and $\mathbf{PDF_B}$. In an ideal situation, the extreme points should be uniformly distributed in the color space. However, most of the extreme points are relatively centralized in some cases, which will lead to the information redundancy due to that multiple extreme points are selected out to represent the similar image statistical information. To avoid the situation mentioned above, we further checkout all initial extreme points by the local window suppression. Let $\{L_A^i\}_{i=1}^K$ be the intensities of K extreme points $\{P_A^i\}_{i=1}^K$ in $\mathbf{PDF_A}$, which are sorted in the ascending order. Given an extreme point P_A^i, we generate a neighborhood range $[L_A^i - w, L_A^i + w]$ centered on the corresponding intensity L_A^i with the size of $(2w+1)$. We set $w = 2$ if not specifically stated in this paper. If there exist more than one extreme points located in this neighborhood range, the extreme point with the highest frequency in $\mathbf{PDF_A}$ will be retained and other extreme points will be removed. All initial extreme points are checked in this way and the retained extreme points are used for the following matching. The final extreme points extracted from $\mathbf{PDF_A}$ and $\mathbf{PDF_B}$ are represented as $\{\mathbf{P}_A^i\}_{i=1}^{N_A}$ and $\{\mathbf{P}_B^j\}_{j=1}^{N_B}$, where N_A and N_B are the numbers of extreme points in $\mathbf{PDF_A}$ and $\mathbf{PDF_B}$, respectively. For each extreme point $\mathbf{P}$, it consists of 4 components according to $\mathbf{P} = \{F, L, \hat{C}, \check{C}\}$ where F denotes the frequency of this point in PDF, L represents the corresponding intensity, and $\hat{C}$ and $\check{C}$ means the cumulative values of the intensities $(L + \varepsilon)$ and $(L - \varepsilon)$ in CDF (we set $\varepsilon = 2$ if not specifically stated in this paper).

3.3 Matching Extreme Points

The extreme points can sufficiently reflect image statistical characteristics. To efficiently adjust the color differences, one way is to ensure that the intensities of corresponding extreme points are the same. Thus, we should match the extreme points firstly. To reliably match these extreme points

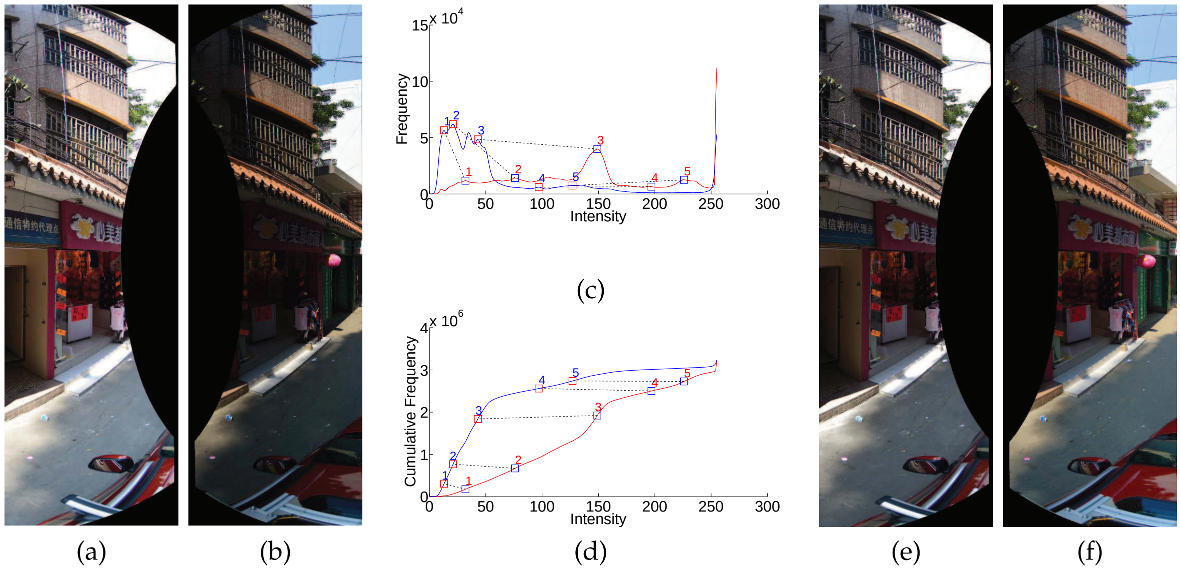


Figure 7. A visual example of our proposed color correction approach: (a)-(b) the overlap image regions of the input left and right images, respectively; (c)-(d) the curves of PDF and CDF in one channel where the red curves stand for the left image and the blue ones stand for the right image, and the matched peaks are marked by the same number and connected by the black dotted lines; (e)-(f) the corrected left and right images, respectively.

$\{\mathbf{P}_A^i\}_{i=1}^{N_A}$ and $\{\mathbf{P}_B^j\}_{j=1}^{N_B}$, we define a cost function to measure the matching similarity of two extreme points $\mathbf{P}_A^i$ and $\mathbf{P}_B^j$ as:

$$Cost(\mathbf{P}_A^i, \mathbf{P}_B^j) = \frac{F_A^i + F_B^j}{2F_{\max}} \times \frac{\min(F_A^i, F_B^j)}{\max(F_A^i, F_B^j)} \times \frac{\max(\hat{C}_A^i - \check{C}_A^i, \hat{C}_B^j - \check{C}_B^j)}{\max(\hat{C}_A^i, \hat{C}_B^j) - \min(\check{C}_A^i, \check{C}_B^j)}, \tag{8}$$

where $F_{\max}$ is the maximal frequency of all the extreme points in both $\mathbf{PDF_A}$ and $\mathbf{PDF_B}$. The above cost function judges the two extreme points from the view of both PDF and CDF. The first term $\frac{F_A^i + F_B^j}{2F_{\max}}$ indicates that those possibly matched extreme points with the higher frequencies generates higher costs, which may be peaked out first in the following matching selection strategy. The second term $\frac{\min(F_A^i, F_B^j)}{\max(F_A^i, F_B^j)}$ indicates that there are the similar frequencies for two possibly matched extreme points $\mathbf{P}_A^i$ and $\mathbf{P}_B^i$. The last term is applied to ensure that the accumulative values of two possibly matched extreme points $\mathbf{P}_A^i$ and $\mathbf{P}_B^i$ are approximate. From this term, we can find that if the small range of cumulative values of two extreme points are similar, the numerator $\max(\hat{C}_A^i - \check{C}_A^i, \hat{C}_B^j - \check{C}_B^j)$ is close to the denominator $\max(\hat{C}_A^i, \hat{C}_B^j) - \min(\check{C}_A^i, \check{C}_B^j)$, which results in that this term is close to 1. In contrast, if the numerator is smaller and the denominator is larger, this term will approach to 0. In summary, if the frequencies of two extreme points are larger and more similar, and the accumulative values of those points are more approximate, their matching cost is bigger. In contrast, it is smaller. The higher the cost function value is, the more likely these two extreme points are matched. Based on this cost definition, a $N_A \times N_B$ matching cost matrix $\mathbf{M} = [M_{ij}]_{N_A \times N_B}$ is created. In order to efficiently eliminate the impossibly matched extreme points, we empirically designed three hard conditions from the view of both PDF and CDF to check whether two extreme points $\mathbf{P}_A^i$ and $\mathbf{P}_B^j$ are possibly matched as follows:

$$\begin{cases} \dfrac{\min(F_A^i, F_B^j)}{\max(F_A^i, F_B^j)} < \theta_f, \\ \check{C}_A^i > \hat{C}_B^j + \theta_c \times C_{\max}, \\ \check{C}_B^j > \hat{C}_A^i + \theta_c \times C_{\max}, \end{cases} \tag{9}$$

where θ_f and θ_c are two empirical thresholds ($\theta_f = 0.25$ and $\theta_c = 0.02$ were used in this paper), $C_{\max}$ is the maximal value of CDF, namely, the valid pixel number of overlap regions. The matching cost $Cost(\mathbf{P}_A^i, \mathbf{P}_B^j)$ is set to zero, i.e., $M_{ij} = Cost(\mathbf{P}_A^i, \mathbf{P}_B^j) = 0$, if at least one of the above three conditions is not met, namely, $\mathbf{P}_A^i$ and $\mathbf{P}_B^j$ are not possibly matched. From the view of PDF, the first condition indicates that the frequencies of the two possibly matched extreme points $\mathbf{P}_A^i$ and $\mathbf{P}_B^j$ should be a relatively small difference. From the view of CDF, the second and third conditions indicate that $\mathbf{P}_A^i$ and $\mathbf{P}_B^j$ are possibly matched if their corresponding CDF values are approximate. According to the above three hard conditions, the matching cost matrix $\mathbf{M}$ will be updated, in which all the zero elements indicate that they are not possibly matched.

Based on the computed matching cost matrix $\mathbf{M}$, we propose an efficient iterative strategy to find the matched extreme points as the following steps:

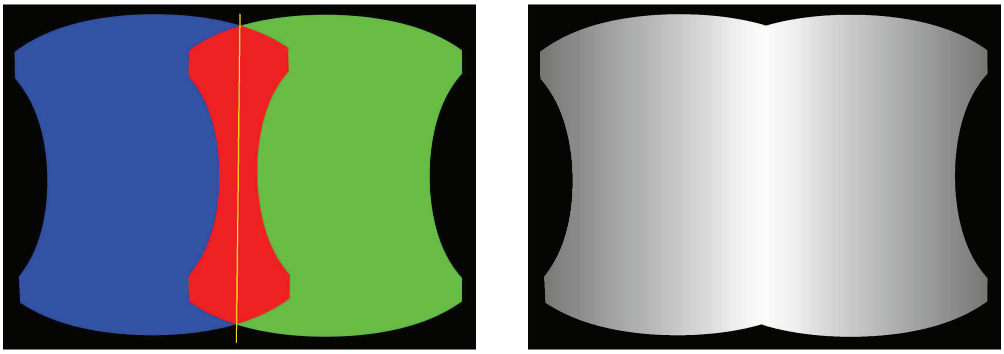
- Step 1: Finding the highest non-zero cost element M_{ij} from the matrix $\mathbf{M}$ and its corresponding extreme points $\mathbf{P}_A^i$ and $\mathbf{P}_B^j$ is selected out as a reliable extreme point match.
- Step 2: Updating the matrix $\mathbf{M}$ by removing the i-th row and the j-th column due to that $\mathbf{P}_A^i$ and $\mathbf{P}_B^j$ have been successfully matched.
- Step 3: Performing the above two steps iteratively until the updated matrix $\mathbf{M}$ is empty or there exists no non-zero element in $\mathbf{M}$.

By the above iterative strategy, a set of reliable extreme point matches will be found. In Figure 7, we have shown a visual example of our proposed color correction approach. The input two images have large color differences in overlap regions, as shown in Figures 7(a)-(b). We find 5 matched extreme points in PDF of one channel. Based on those correspondences, the large color differences can be eliminated, as shown in Figures 7(e)-(f). From this example, we can find that our proposed approach can handle the images with large color differences very well.

Sometimes, no match or too few matches can be reliably found via the above matching strategy in the whole CDF range or some relatively large CDF range. In this case, we will introduce more matches with the help of both $\mathbf{CDF_A}$ and $\mathbf{CDF_B}$, which are selected from H uniformly distributed points $\{C_A^k\}_{k=1}^H$ and $\{C_B^k\}_{k=1}^H$ from $\mathbf{CDF_A}$ and $\mathbf{CDF_B}$, respectively, but not from the previously found extreme points. The same number of sampling points in $\mathbf{CDF_A}$ and $\mathbf{CDF_B}$ are uniformly selected in accordance with the cumulative density values. In our experiments, the percentages of sampling intervals were used as $[0.1, 0.3, 0.5, 0.7, 0.9]$. If there exists no extreme point match found in the ranges $[C_A^k - \kappa C_{\max}, C_A^k + \kappa C_{\max}]$ and $[C_B^k - \kappa C_{\max}, C_B^k + \kappa C_{\max}]$, the current sampling points C_A^k and C_B^k will be added into the matching set as a new point match, where κ is a given threshold in advance ($\kappa = 0.1$ was used in this paper).

3.4 Correcting Color Difference

The extracted matching points in the overlap image regions are then applied to correct the intensities of two adjacent images, including the pixels in non-overlap regions. Let $\{Q_A^k\}_{k=1}^N$ and $\{Q_B^k\}_{k=1}^N$ be the final matching points in CDFs in the overlap regions $\mathbf{A}$ and $\mathbf{B}$ with N point matches. Based on the matching results, the intensities of the matching points Q_A^k and Q_B^k are modified to $(L_A^k + L_B^k)/2$ where L_A^k and L_B^k denote the intensities of k-th match (Q_A^k, Q_B^k) in CDFs, respectively. In this way, the intensities of $\{(Q_A^k, Q_B^k)\}_{k=1}^N$ are corrected to $\{(\hat{L}_A^k, \hat{L}_B^k)\}_{k=1}^N$, respectively, where



(a) (b)

Figure 8. An illustration of the alpha weighting fusion map of two adjacently warped images: (a) two overlapped images represented by the blue and green regions, respectively, with the overlap image region marked in red and the center line marked in yellow; (b) the normalized alpha weighting fusion map for two images where the brighter regions indicate higher values.

$\hat{L}^k_A = \hat{L}^k_B = (L^k_A + L^k_B)/2$. Based on these corrections, the intensity of any pixel in both **A** and **B** will be adjusted linearly. For example, given a pixel $\mathbf{p} \in \mathbf{A}$ whose intensity $L_A(\mathbf{p})$ will be linearly corrected as:

$$\hat{L}_A(\mathbf{p}) = \hat{L}^l_A + (L_A(\mathbf{p}) - L^l_A)\frac{\hat{L}^u_A - \hat{L}^l_A}{L^u_A - L^l_A}, \tag{10}$$

where $L_A(\mathbf{p}) \in [L^u_A, L^l_A]$, L^u_A and L^l_A denote the intensities of two matching points in **A** that are closest to $L_A(\mathbf{p})$, and the $\hat{L}^u_A$ and $\hat{L}^l_A$ are the corresponding corrected intensities. In order to produce a smooth and gradual transition from non-overlap regions to overlap ones, the alpha correction method is conducted as:

$$L'_A(\mathbf{p}) = (1 - \alpha(\mathbf{p}))L_A(\mathbf{p}) + \alpha(\mathbf{p})\hat{L}_A(\mathbf{p}), \tag{11}$$

where $L'_A(\mathbf{p})$ denotes the finally fused intensity of the pixel $\mathbf{p}$, $L_A(\mathbf{p})$ is the original intensity of the pixel $\mathbf{p}$ while $\hat{L}_A(\mathbf{p})$ is the corrected intensity of the corresponding pixel based on the above mentioned correction method, and $\alpha(\mathbf{p})$ is a function that related to the distance between the pixel $\mathbf{p}$ and the center line of the overlap image region, which ranges between 0 and 1 as shown in Figure 8 where the smaller the distance to the center line is, the larger the α is. All the pixels in another image will be processed in the same way.

4 Image Mosaicking

Although the large geometric misalignments and photometric inconsistencies have been greatly reduced through our proposed image warping and color correction algorithms,respectively, there always exist small geometric misalignments and color differences between adjacent images. To stitch the color corrected panoramic images into the single composite panorama, we also need to find the optimal seamlines in the overlap image regions between warped images to magnificently conceal the parallax. Furthermore, an efficient image blending algorithm will be further applied to eliminate the stitching artifacts caused by small color differences along the seamlines.

4.1 Optimal Seamline Detection

In this paper, the optimal seamlines between color corrected images will be efficiently extracted using the graph-cuts-based seamline detection algorithm presented in [19]. This novel algorithm is used to efficiently detect optimal seamlines for mosaicking street-view panoramic images without precisely common center in a two-label graph cuts energy minimization framework. This algorithm magnificently fuses the information of image color, gradient, and texture complexity into the data and smooth energy terms in graph cuts to effectively ensure that the seamlines are optimally detected in the laterally continuous regions with high image similarity and low object dislocation to magnificently conceal the parallax between images. For multiple images, we apply the traditional *frame-to-frame optimization* strategy to efficiently find all optimal seamlines. The details of this strategy are described in Section 3.1 of [19]. The experimental results on a large set of images reported in [19] have demonstrated that this algorithm is capable of creating high-quality seamlines for multiple image mosaicking, while not crossing majority of visually obvious foreground objects and most of overlap regions with low image similarity to effectively conceal the image parallax at different extents.

4.2 Image Blending

Although the major color differences are eliminated between input images by applying our proposed color correction strategy presented in Section 3, there still exist the artifacts along the seamlines due to that the color differences can not be removed completely via color correction. Thus, a good image blending algorithm is needed to generate the last pleasant panorama. To quickly blend the color corrected images after detecting the optimal seamlines, the transition smoothing methods (also known as feathering [47] or alpha blending methods [48]) can be used to minimize the visibility of seamlines by smoothing the common overlapping regions of the combined images. However, to produce a more pleasant panorama, in this paper, we use the Laplacian pyramid blending [49] to stitch multiple color corrected images at one time.

5 Experimental Results

Extensive experiments on representative street-view panoramic images were conducted to comprehensively evaluate the performance of our proposed unified framework for street-view panorama stitching. In our paper, all used street-view panoramic images were captured from the real world scenes by an integrated multi-camera equipment with six Nikon D7100 cameras of 24 million pixels with wide-angle lenses mounted on a mobile vehicle platform. Six camera images were aligned into a common spherical coordinate system with the image size of 12000×6000 pixels. Due to that the projection centers of these six cameras are not precisely the same, there always exist large geometrical misalignments at different extents between the adjacently aligned images, especially in the image regions close to the camera centers. The overlap relationship of those six panoramic images is shown in Figure 9. Our algorithms in this paper were implemented with C++ under Windows and tested in a computer with an Intel Core i7-4770 at 3.4GHz and the 16GB RAM memory. Due to the limit of pages, more experimental results and analysis are presented at http://cvrs.whu.edu.cn/projects/PanoStitching/.

5.1 Image Warping

In this section, we conducted the experiments on two groups of panoramic images to prove the effectiveness and superiority of our proposed image warping algorithm described in Section 2. The panorama stitching results without and with the use of our proposed image warping algorithm in the first group of six panoramic images are shown in Figures 10(a) and (b), respectively. We can find that the whole seamlines in two panoramas cross the similar regions with the high image similarity. However, from the whole stitching results and especially the detailed local regions shown in Figures 10(a) and (b), we observed that the stitching artifacts caused by the geometric dislocation in the panorama, as shown in Figure 10(a), stitched without the use of image warping algorithm are more obvious than the panorama, as shown in Figure 10(b), stitched with its use. Noticeably, the stitching artifacts caused by geometric dislocation become smaller as expected when the image

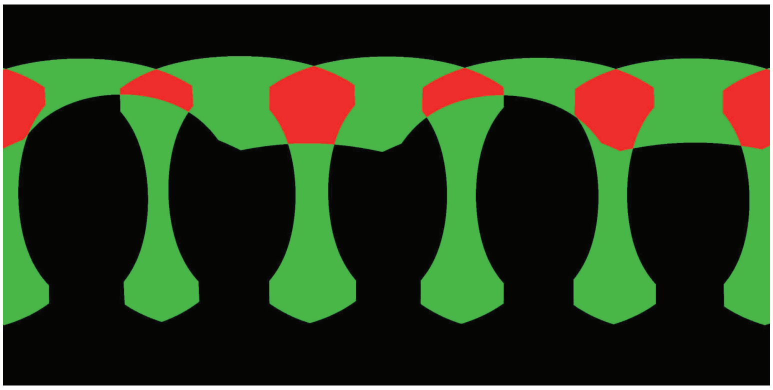


Figure 9. The image overlap regions of six geometrically aligned and warped images in the 360° street-view panoramic view where the black, the green and the red stand for the no-overlapped, two-overlapped, multi-overlapped image regions, respectively.

warping algorithm was applied, as shown in Figure 10(b). While not using the image warping algorithm, the geometric dislocation is very large, as shown in Figure 10(a). For example, in the first enlarged local region, the seamline crossed the text without the used of image warping, and it avoided crossing the text when the image warping was used. In the second enlarged local region, although two seamlines crossed the road with pavement stairs, we can find that the dislocation is almost invisible in the pavement stairs when the image warping was used, but it is so obvious without the use of the image warping. In the aspect of computational cost, without the use of image warping, our algorithm took around 17.89s in the above experiment, only the elapsed time in six optimal seamlines detection is included. However, with its use, our algorithm took around 70.93s consisting of all the elapsed times in the image warping and the optimal seamline detection. From this comparison, we observed that the seamline detection is efficient, but the image warping is relatively time-consuming. This is mainly because that we need to find the inlier matches for all image pairs at first and then interpolate the dense optical flows by MBA for each image, which is time-consuming. But our proposed image warping algorithm can significantly improve the quality of the last stitched panorama.

The comparative experimental results on another group of panoramic images are presented in Figures 10(c) and (d), respectively, and the similar conclusions can be drawn. The computational times of our algorithm without the use of image warping and with its use are 13.19s and 56.77s, respectively.

From the above experimental results on two groups of panoramic images, we observed that our proposed image warping algorithm can effectively eliminate the stitching artifacts caused by the geometrical dislocations and can also slightly improve the quality of the found optimal seamlines to some extent.

5.2 Color Correction and Image Blending

In this section, we conducted the experiments in two group panoramic images to prove that our proposed color correction algorithm can magnificently reduce the large color differences between the warped images. In addition, we also presented the last panoramas generated by our proposed system and compared them with the open-source software *Enblend* [4] which are popularly used to generate the street-view panorama by stitching the registered panoramic images.

[4] Available at http://enblend.sourceforge.net/.

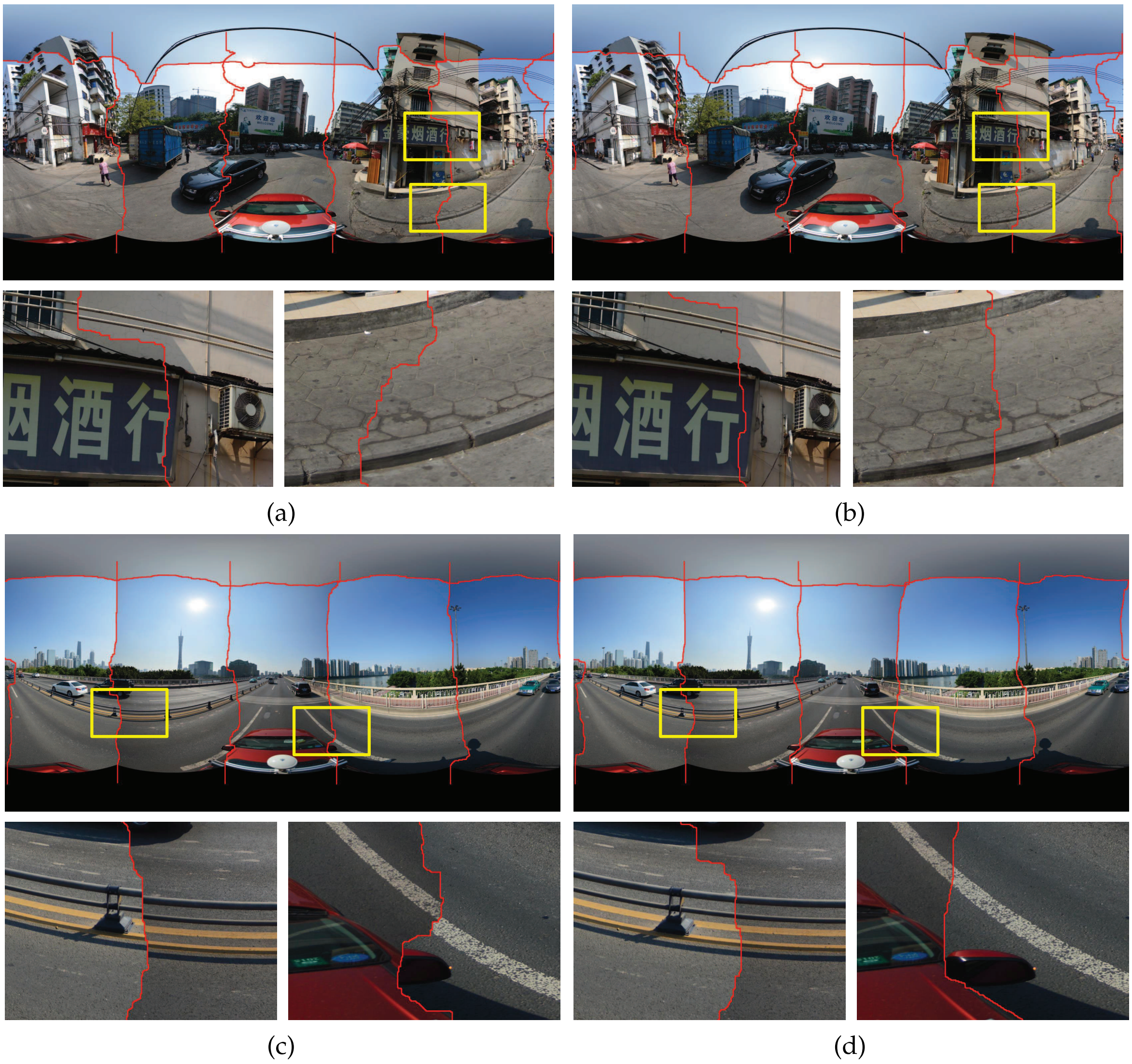


Figure 10. Visual comparison of the stitching results with the optimal seamlines in two groups of six panoramic images when our proposed image warping algorithm was used (Right:(b) and (d)) or not (Left: (a) and (c), namely, the stitching results of [19]). The red lines stand for the detected optimal seamlines between images.

Figure 11 shows the experimental results on the first group of panoramic images. The panorama stitching results without and with the use of color correction are shown in Figures 11(a) and (b), respectively. From the whole stitching results and especially the detailed local regions shown in Figures 11(a) and (b), we can find that color differences between the warped images were significantly reduced and are almost invisible. In addition, the quality of the detected optimal seamlines was improved as expected when the color correction algorithm was used due to that the color differences were greatly reduced before the seamlines were found. For example, the seamline rounded the advertising board instead of crossing it when the color correction algorithm was used, as shown in the detailed image regions in Figures 11(a) and (b). In the aspect of computational cost, without the use of the color correction, our algorithm took around 18.01s to find all six optimal seamlines. With its use, our algorithm took around 33.52s to correct the color differences and find the optimal seamlines, means that the color correction algorithm took around 15.51s. To generate the last panorama, the Laplacian pyramid blending algorithm was further applied, whose generated result is shown in Figure 11(d). And in Figure 11(c), we also present the last panorama generated by *Enblend*. From

Figure 11. Visual comparison in the first group of six panoramic images: (a)-(b) the stitching results with the optimal seamlines when the our proposed color correction was used (b) or not (a); (c)-(d) the last panoramas generated by *Enblend* (c) and our proposed stitching system (d).

the visual comparison, we can observe that our proposed stitching system with image warping and color correction obviously outperforms *Enblend*. Noticeably, the stitching artifacts caused by geometric misalignments and photometric inconsistencies still exist in the panorama generated by *Enblend*, as shown in Figure 11(c) but they almost disappeared in our produced panorama, as shown in Figure 11(d). In the aspect of computational cost, the Laplacian pyramid blending algorithm took 35.56s.

The experimental results on another group of panoramic images are presented in Figure 12 and the similar conclusion can be drawn. The large color differences were greatly reduced by our proposed color correction algorithm, especially in the regions of sky and the tall buildings, and the quality of the detected seamlines was slightly improved to some extent. The seamlines bypass the buildings and the white lane when the color differences were corrected for the warped images. Likewise, the stitching artifacts existed in the panorama produced by *Enblend* disappeared in the panorama generated by our proposed system. The elapsed times in color correction, optimal seamline detection and image blending are 17.33s, 13.77s, and 35.61s, respectively.

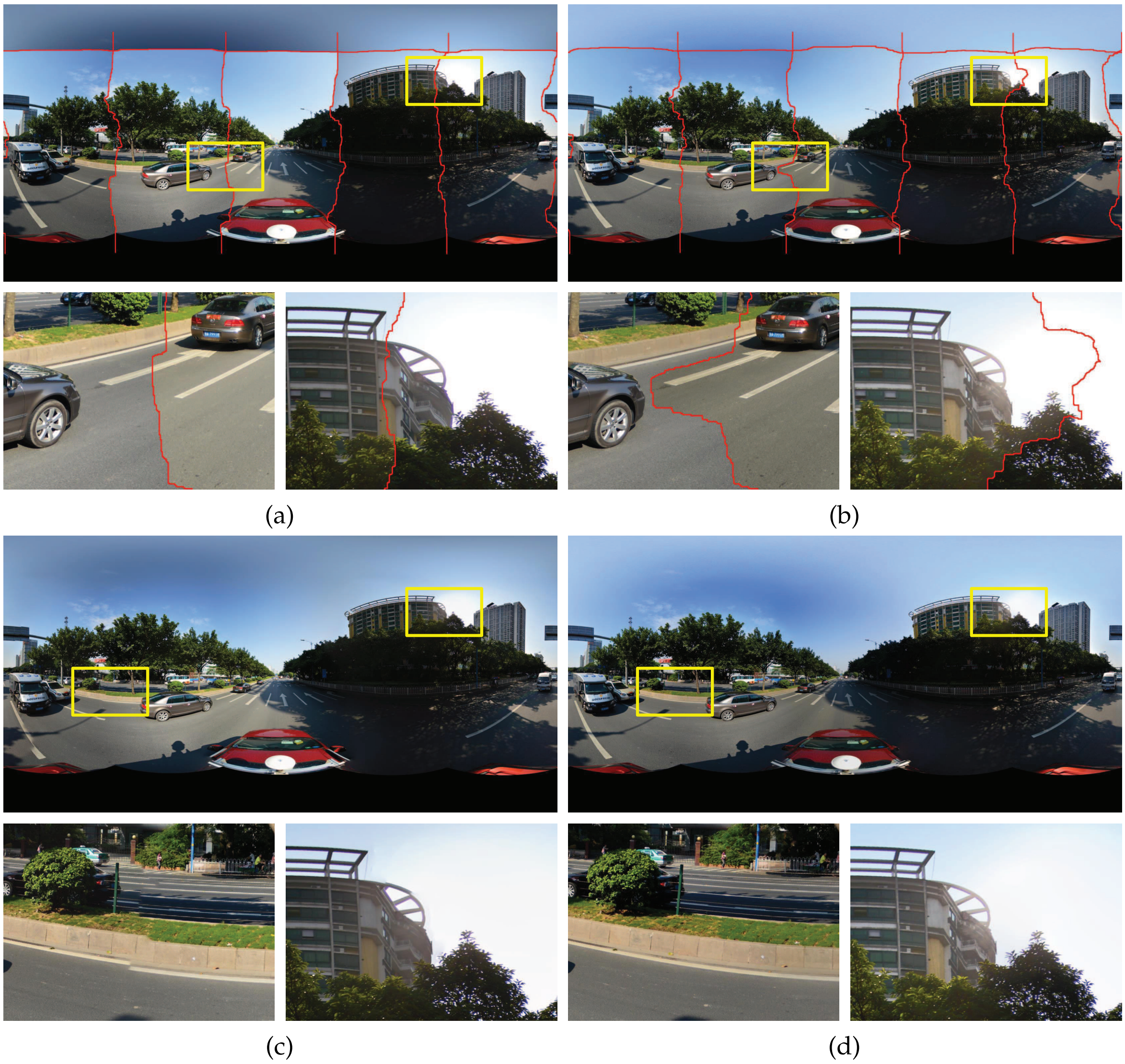


Figure 12. Visual comparison in the second group of six panoramic images: (a)-(b) the stitching results with the optimal seamlines when the our proposed color correction was used (b) or not (a); (c)-(d) the last panoramas generated by *Enblend* (c) and our proposed stitching system (d).

5.3 Image Stitching

To illustrate the effectiveness of our proposed framework for street-view panorama stitching, we presented the last panoramas stitched by different combination of optimal seamline detection (S), image warping (W), color correction (C) and image blending (B) algorithms in Figure 13. At first, Figure 13(a) shows the panorama generated by the optimal seamline detection algorithm presented by [19], from which we can find that there are many stitching artifacts caused by geometric misalignments and photometric inconsistencies in the last stitching image, especially obvious in the detailed local regions. For example, the white lanes on the road were broken due to the large geometric dislocations. In addition, there also exist large color differences along the optimal seamlines. Our proposed image warping and color correction algorithm can eliminate large geometric misalignments and photometric inconsistencies, as shown in Figures 13(b) and (c), respectively. The last blended panorama generated by our proposed system is shown in Figure 13(d) from which we can observe that the last stitched panoramic image is pleasant and high-quality, which can meet the application requirement of the street-view map.

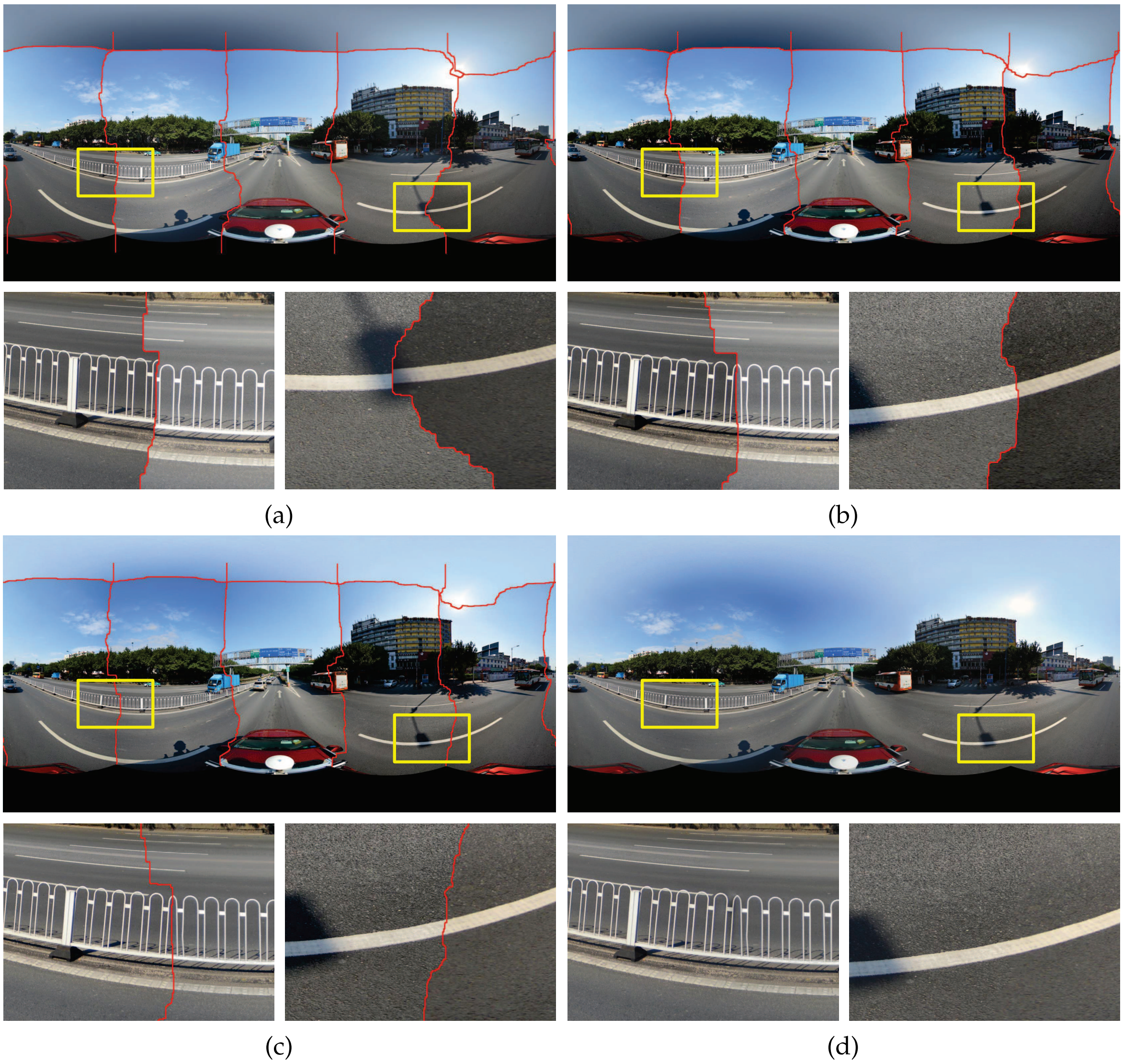


Figure 13. The stitching results with different combination of optimal seamline detection (S), image warping (W), color correction (C) and image blending (B) algorithms: (a) S (the result generated by [19]); (b) $W + S$; (c) $W + C + S$; (d) $W + C + S + B$. The computational times of (a)-(d) are 18.00s, 69.08s, 86.63s and 123.68s, respectively.

5.4 Comparative Results

At last, to prove that our approach is superior and can generate high-quality panoramas, we compared our proposed approach with the Xiong and Pulli's approach [42]. We used two representative groups of panoramic images for visual comparison. The color differences in the first group of images are relatively small but large in the second group. Because the Xiong and Pulli's approach has not eliminated the influence of large geometric misalignments between aligned images, so we used the warped images generated by our image warping algorithm as the input ones for comparing two approaches. In addition, their approach applied the Poisson blending algorithm to generate the last blended image, however, our approach used the Laplacian pyramid blending algorithm. To evaluate the last blended panoramas generated by two approaches fairer, we replaced the Poisson blending algorithm in the tested Xiong and Pulli's approach with the Laplacian pyramid blending algorithm.

Figure 14 shows the stitching results of the first group of images with relatively small color differences. Figures 14(a) and (b) illustrate the stitching results just with the detected seamlines of the Xiong and Pulli's approach and our approach without the use of color correction, respectively. From these two figures, we can observe that the seamlines detected by our approach are better than those detected by their approach. For example, the seamlines detected by their approach crossed the tall building, but our approach avoided crossing it. Figures 14(c) and (d) illustrate the stitching results of two approaches with the use of color correction, respectively, from which we observed that both of two approaches can eliminate the small color differences effectively. Figures 14(e) and (f) show the last blended panoramas generated by two approaches, respectively, from which we found that there is some petty ghost on the top of the tallest building in the second enlarged region shown in Figure 14(e), which disappeared in the panorama generated by our approach, as shown in Figure 14(f). This is mainly because the horizontal seamline between bottom and top input images detected by the Xiong and Pulli's approach is close to this building, as shown in Figure 14(c). In conclusion, if the color differences between input images are small, both of two approaches can generate high-quality panoramas.

Figure 15 shows the stitching results of the second group of images with very large color differences. Figures 14(a) and (b) show the stitching results of the Xiong and Pulli's approach and our approach without the use of color correction, respectively, from which we observed that our approach also generated more high-quality seamlines than their approach. Figures 14(c) and (d) present the stitching results of two approaches with the use of color correction, respectively. From the visual comparison, we observed that our proposed color correction algorithm obviously outperformed than the algorithm presented in [42], especially obvious in two locally enlarged regions. For example, in the first enlarged region (from left to right), the detected seamline divides the building into two parts, one comes from the top input image which is dark, and another comes from the bottom input image which is relatively lighter. After color correction, the top image is also very dark in the result generated by the Xiong and Pulli's approach and the color differences along the seamline are also very large. In addition, due to that the top image is too dark, many detailed informations cannot be pleasantly observed. But, in our result, the color of the top image is similar with the bottom one, and more detailed informations of this region can be clearly observed. Figures 14(e) and (f) show the last blended panoramas generated by two approaches, respectively. In the second enlarged region of Figure 15(e), we found that there are some very obvious ghosts on the top of the building, which disappeared in the panorama generated by our approach, as shown in Figure 15(f). In addition, in Figure 15(e), the color of top sky regions almost is white, which is not pleasant. However, in the last panorama generated by our approach, the color of those regions is slightly bluish, which is more reasonable and pleasant, as shown in Figure 15(f). In conclusion, if the color differences between input images are large, our approach can also generate high-quality panoramas, but the results generated by the Xiong and Pulli's approach are not so good.

In the aspect of computational times of two approaches, the average times on two groups of images are presented in Table 1, from which we can find that our approach is a litter bit more time-consuming than their approach. This is mainly because their approach applied dynamic programming to detect the optimal seamlines but we used graph cuts, which is more time-consuming than dynamic programming. We also observed that the computation times of our proposed color correction algorithm and their algorithm are 16.83s and 16.08s, respectively, which are almost the same. But our color correction algorithm is more effective than their algorithm.

6 Conclusion

In this paper, we proposed a unified framework for street-view panorama stitching system which is comprised of image warping, color correction, optimal seamline detection and image blending for stitching or mosaicking a set of geometrically aligned street-view panoramic images with large

Table 1. The computational times of our proposed approach and the approach proposed in [42].

	Optimal Seamline Detection	Color Correction	Image Blending	#Total
Our Proposed ãĂĂApproach (s)	16.92	16.83	36.60	70.36
Xiong and Pulli's Appproach (s)	13.77	16.08	36.72	66.5815

geometric misalignments and photometric inconsistencies into a visual-appealing and informative wide-angle composite image. The contributions in this paper are summarized as follows:

- We creatively proposed a novel image warping method based on the dense optical flows to greatly reduce the large geometric misalignment existed in the input images as much as possible. Experimental results have demonstrated the superiority of our proposed image warping method, which can efficiently and greatly eliminate the influence of the large geometric misalignment.
- We proposed a novel color correction and image blending method to further reduce the color differences between panoramic images based on extreme point matching of histograms of the overlapped image regions of two involved images via both probability density functions and cumulative distribution functions. Experimental results on representative street-view panoramic images have proved that our proposed color correction method is capable of eliminating the large color differences between adjacent images captured in different illumination conditions and/or different exposure settings, which obviously outperforms the open-source software *Enblend* and the approach proposed by [42].
- We proposed a unified framework for street-view panorama stitching system. Even thought there are large geometrical misalignments and photometric inconsistencies in the input aligned images, our system can also generate pleasant and high-quality panoramas.

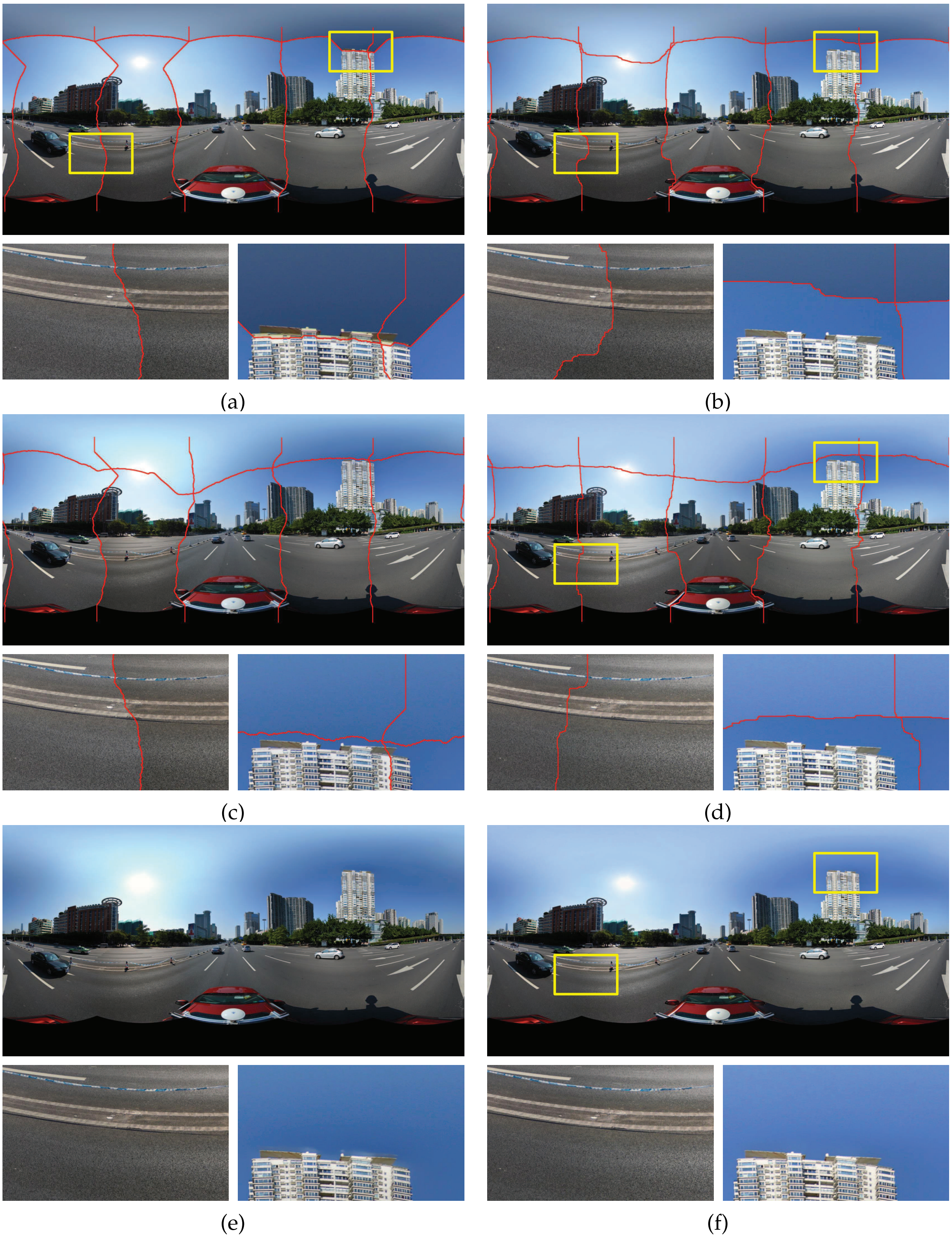
Nevertheless, the proposed system may be improved in the future in the following ways. First, when detecting the optimal seamlines, the superpixel segmentation can be introduced to greatly improve the optimization efficiency by decreasing the number of elements in graph cuts, and the scene understanding or parsing can also be applied in some particular image data. For example, the roads can be detected out for guiding the seamlines. Second, the whole image mosaicking method can be improved to handle more different types of images, not only street-view panoramic ones, but also aerial and oblique ones. At last, the parallel optimization strategy is expected to be developed to more efficiently generate the last panorama.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (Project No. 41571436), the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027), the National Natural Science Foundation of China (Project No. 41271431), and the Jiangsu Province Science and Technology Support Program, China (Project No. BE2014866).

Bibliography

1. Du, Q.; Raksuntorn, N.; Orduyilmaz, A.; Bruce, L.M. Automatic registration and mosaicking for airborne multispectral image sequences. *Photogrammetric Engineering & Remote Sensing* **2008**, *74*, 169–181.
2. Chon, J.; Kim, H.; Lin, C.S. Seam-line determination for image mosaicking: A technique minimizing the maximum local mismatch and the global cost. *ISPRS Journal of Photogrammetry and Remote Sensing* **2010**, *65*, 86–92.



(a) (b) (c) (d) (e) (f)

Figure 14. Visual comparison between our approach in the left column and the Xiong and Pulli's approach in the right column on the first group of images with relatively small color differences: (a)-(b) the results without the use of color correction; (c)-(d) the results with the use of color correction; (e)-(f) the last generated panoramas.

3. Wan, Y.; Wang, D.; Xiao, J.; Lai, X.; Xu, J. Automatic determination of seamlines for aerial image mosaicking based on vector roads alone. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *76*, 1–10.

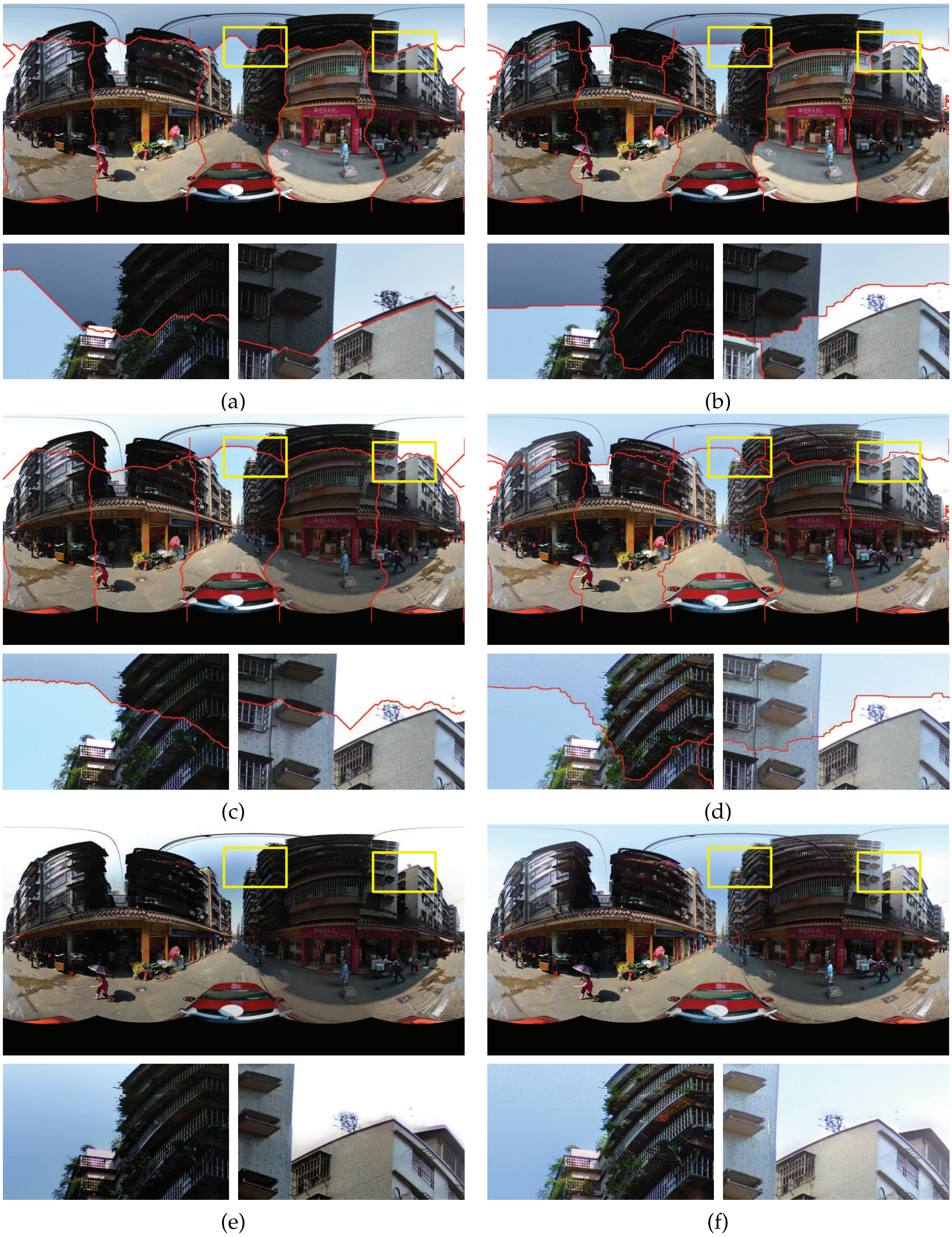


Figure 15. Visual comparison between our approach in the left column and the Xiong and Pulli's approach in the right column on the second group of images with large color differences: (a)-(b) the results without the use of color correction; (c)-(d) the results with the use of color correction; (e)-(f) the last generated panoramas.

4. Pan, J.; Zhou, Q.; Wang, M. Seamline Determination Based on Segmentation for Urban Image Mosaicking. *IEEE Transactions on Geoscience and Remote Sensing Letters* **2014**, *11*, 1335–1339.

5. Pang, S.; Sun, M.; Hu, X.; Zhang, Z. SGM-based seamline determination for urban orthophoto mosaicking. *ISPRS Journal of Photogrammetry and Remote Sensing* **2016**, *112*, 1–12.
6. Kerschner, M. Seamline detection in colour orthoimage mosaicking by use of twin snakes. *ISPRS Journal of Photogrammetry and Remote Sensing* **2001**, *56*, 53–64.
7. Soille, P. Morphological image compositing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2006**, *28*, 673–683.
8. Yang, Y.; Gao, Y.; Li, H.; Han, Y. An algorithm for remote sensing image mosaic based on valid area. International Symposium on Image and Data Fusion (ISIDF), 2011.
9. Li, X.; Hui, N.; Shen, H.; Fu, Y.; Zhang, L. A robust mosaicking procedure for high spatial resolution remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing* **2015**, *109*, 108–125.
10. Agarwala, A.; Dontcheva, M.; Agrawala, M.; Drucker, S.; Colburn, A.; Curless, B.; Salesin, D.; Cohen, M. Interactive Digital Photomontage. *ACM Transactions on Graphics (TOG)* **2004**, *23*, 294–302.
11. Szeliski, R. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* **2006**, *2*, 1–104.
12. Brown, M.; Lowe, D.G. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision* **2007**, *74*, 59–73.
13. Gracias, N.; Mahoor, M.; Negahdaripour, S.; Gleason, A. Fast image blending using watersheds and graph cuts. *Image and Vision Computing* **2009**, *27*, 597–607.
14. Prados, R.; Garcia, R.; Gracias, N.; Escartin, J.; Neumann, L. A novel blending technique for underwater gigamosaicing. *IEEE Journal of Oceanic Engineering* **2012**, *37*, 626–644.
15. Philip, S.; Summa, B.; Tierny, J.; Bremer, P.T.; Pascucci, V. Distributed Seams for Gigapixel Panoramas. *IEEE Transactions on Visualization and Computer Graphics* **2015**, *21*, 350–362.
16. Kwatra, V.; Schödl, A.; Essa, I.; Turk, G.; Bobick, A. Graphcut Textures Image and Video Synthesis Using Graph Cuts. *ACM Transactions on Graphics (TOG)* **2003**, *22*, 277–286.
17. Yu, L.; Holden, E.J.; Dentith, M.C.; Zhang, H. Towards the automatic selection of optimal seam line locations when merging optical remote-sensing images. *International Journal of Remote Sensing* **2012**, *33*, 1000–1014.
18. Mills, S.; McLeod, P. Global seamline networks for orthomosaic generation via local search. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *75*, 101–111.
19. Li, L.; Yao, J.; Lu, X.; Tu, J.; Shan, J. Optimal seamline detection for multiple image mosaicking via graph cuts. *ISPRS Journal of Photogrammetry and Remote Sensing* **2016**, *113*, 1–16.
20. Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: active contour models. *International Journal of Computer Vision* **1988**, *1*, 321–331.
21. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numerische Mathematik* **1959**, *1*, 269–271.
22. Bellman, R. *Dynamic Programming*; Princeton University Press, Princeton, NJ, 1957.
23. Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2001**, *23*, 1222–1239.
24. Glasbey, C.A.; Mardia, K.V. A review of image-warping methods. *Journal of Applied Statistics* **1998**, *25*, 155–171.
25. Lee, S.Y.; Chwa, K.Y.; Hahn, J.K.; Shin, S.Y. Image morphing using deformation techniques. *Journal of Visualization and Computer Animation* **1996**, *7*, 3–23.
26. Liao, J.; Lima, R.S.; Nehab, D.; Hoppe, H.; Sander, P.V.; Yu, J. Automating image morphing using structural similarity on a halfway domain. *ACM Transactions on Graphics (TOG)* **2014**, *33*, 1–12.
27. Liu, F.; Gleicher, M. Automatic image retargeting with fisheye-view warping. ACM Symposium on User Interface Software and Technology, 2005, pp. 153–162.
28. Lin, S.S.; Yeh, I.C.; Lin, C.H.; Lee, T.Y. Patch-based image warping for content-aware retargeting. *IEEE Transactions on Multimedia* **2013**, *15*, 359–368.
29. Zaragoza, J.; Chin, T.J.; Brown, M.; Suter, D. As-projective-as-possible image stitching with moving DLT. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2339–2346.
30. Lin, C.C.; Pankanti, S.U.; Ramamurthy, K.N.; Aravkin, A.Y. Adaptive as-natural-as-possible image stitching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1155–1163.
31. Bookstein, F.L. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1989**, *11*, 567–585.

32. Alexa, M.; Cohen-Or, D.; Levin, D. As-Rigid-As-Possible Shape Interpolation. Internationl Conference on Computer Graphics and Interactive Techniques Conference (SIGGRAPH), 2000, pp. 157–164.
33. Igarashi, T.; Moscovich, T.; Hughes, J.F. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)* **2005**, *24*, 1134–1141.
34. Schaefer, S.; McPhail, T.; Warren, J. Image deformation using moving least squares. ACM transactions on graphics (TOG), 2006, Vol. 25, pp. 533–540.
35. Levin, D. The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society* **1998**, *67*, 1517–1531.
36. Chen, T.L.; Geman, S. Image warping using radial basis functions. *Journal of Applied Statistics* **2014**, *41*, 242–258.
37. Lee, S.; Wolberg, G.; Shin, S.Y. Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics* **1997**, *3*, 228–244.
38. Yao, J. Modeling and Rendering from Multiple Views. PhD thesis, The Chinse University of Hong Kong, 2006.
39. Yao, J.; Cham, W.K. 3D modeling and rendering from multiple wide-baseline images by match propagation. *Signal Processing: Image Communication* **2006**, *21*, 506–518.
40. Xu, W.; Mulligan, J. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 263–270.
41. Tian, G.Y.; Gledhill, D.; Taylor, D.; Clarke, D. Colour correction for panoramic imaging. International Conference on Information Visualisation, 2002, Vol. 215, pp. 483–488.
42. Xiong, Y.; Pulli, K. Fast panorama stitching for high-quality panoramic images on mobile phones. *IEEE Transactions on Consumer Electronics* **2010**, *56*, 298–306.
43. Hwang, Y.; Lee, J.Y.; Kweon, I.S.; Kim, S.J. Color transfer using probabilistic moving least squares. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3342–3349.
44. Yamamoto, K.; Oi, R. Color correction for multi-view video using energy minimization of view networks. *International Journal of Automation and Computing* **2008**, *5*, 234–245.
45. Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* **2004**, *60*, 91–110.
46. Fecker, U.; Barkowsky, M.; Kaup, A. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Transactions on Circuits and Systems for Video Technology* **2008**, *18*, 1258–1267.
47. Uyttendaele, M.; Eden, A.; Skeliski, R. Eliminating ghosting and exposure artifacts in image mosaics. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2001, Vol. 2, pp. II–509.
48. Porter, T.; Duff, T. Compositing Digital Images. *ACM SIGGRAPH Computer Graphics* **1984**, *18*, 253–259.
49. Burt, P.J.; Adelson, E.H. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)* **1983**, *2*, 217–236.
50. Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. *ACM Transactions on Graphics (TOG)* **2003**, *22*, 313–318.
51. Levin, A.; Zomet, A.; Peleg, S.; Weiss, Y. Seamless image stitching in the gradient domain. European Conference on Computer Vision (ECCV), 2004.
52. Yao, J.; Cham, W.K. Robust multi-view feature matching from multiple unordered views. *Pattern Recognition* **2007**, *40*, 3081–3099.
53. Bay, H.; Tuytelaars, T.; Gool, L.V. SURF: Speeded Up Robust Features. *Computer Vision & Image Understanding* **2006**, *110*, 404–417.
54. HaCohen, Y.; Shechtman, E.; Goldman, D.B.; Lischinski, D. Optimizing color consistency in photo collections. *ACM Transactions on Graphics (TOG)* **2013**, *32*, 38:1–38:10.

© 2016 by the authors. Submitted to *Remote Sens.* for possible open access publication under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/)