

Article

# A Unified Framework for Street-View Panorama Stitching

Li Li <sup>1</sup>, Jian Yao <sup>1,\*</sup>, Renping Xie <sup>1</sup>, Menghan Xia <sup>1</sup> and Wei Zhang <sup>2</sup>

<sup>1</sup> School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; li.li@whu.edu.cn (L.L.); renping.xie@whu.edu.cn (R.X.); menghx@whu.edu.cn (M.X.)

<sup>2</sup> School of Control Science and Engineering, Shandong University, Jinan 250061, China; masakiyo@mail.kitami-it.ac.jp

\* Correspondence: jian.yao@whu.edu.cn; Tel.: +86-27-6877-1218

Academic Editor: Felipe Gonzalez Toro

Received: 8 September 2016; Accepted: 15 December 2016; Published: 22 December 2016

**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 seam line detection and image blending. Since the input images are captured without a precisely common projection center from the scenes with the depth differences with respect to the cameras to different extents, such images cannot be precisely aligned in geometry. Therefore, 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 seam lines between adjacent input images are detected via the graph cut energy minimization framework. At last, the Laplacian pyramid blending algorithm is applied to further eliminate the stitching artifacts along the optimal seam lines. Experimental results on a large set of challenging street-view panoramic images captured from the real world illustrate that the proposed system is capable of creating high-quality panoramas.

**Keywords:** panorama stitching; seam line detection; image warping; graph cuts; image parallax; image blending; color correction

---

## 1. Introduction

Nowadays, with 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 greater. Image stitching is the key technology to produce high-quality panoramic images, which is also an important and classical problem in the fields 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 do not 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 the street-view panoramic images are captured by a panoramic camera mounted on a 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 settings. This paper focuses on creating a visually pleasant street-view panorama by stitching or mosaicking multiple 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 existing in the input aligned panoramic images is to detect the optimal seam lines that avoid crossing the majority of visually obvious objects and most of the overlap regions with low image similarity and large object dislocation. The optimal seam line detection methods search for the seam lines in overlap regions between images where their intensity or gradient differences are not significant. Based on the optimally-detected seam lines, 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 seam line 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 seam lines. 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., the snake model [20], Dijkstra's algorithm [21], dynamic programming [22] and graph cuts [23]. Nowadays, the optimal seam line 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 seam line detection algorithm for mosaicking aerial and panoramic images based on the graph cut energy minimization framework. In this paper, we will apply this algorithm to detect the optimal seam lines.

However, when the geometric misalignments are very large, the stitching artifacts perhaps cannot be completely avoided even though the optimal seam lines are detected, especially for street-view panoramic images among which there always exist geometric misalignments to different extents due to those images being captured from scenes with large depth differences by a 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 existing in the input aligned panoramic images should be eliminated as much as possible before finding the optimal seam lines. 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 that 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, local non-uniform scaling and shearing possibly occurred in the deformed images. The work in [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 may not be smooth across triangle boundaries. The work in [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 on each grid instead of each pixel to reduce the transformation time. At last, it filled the resulting quads

using the bilinear interpolation. The work in [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 at identifying the best radial basis functions for image warping. Our 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. Furthermore, the large color differences may affect the quality of the seam lines. Therefore, we also need to suppress the color differences between warped images before we apply the optimal seam line 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. A few noteworthy parametric approaches are described here. The work in [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. The work in [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 normalized intensity errors for all overlapping pixels. The work in [42] also employed the diagonal model for the color and luminance compensation where the correction coefficients were computed as the ratio of the 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 imaging process. Non-parametric approaches can handle this problem well. Non-parametric approaches do not 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. The work in [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. The work in [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. The work in [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-overlapping regions to overlapping ones.

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

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 seam line detection approach based on the graph cut energy minimization framework to find the optimal seam lines between two overlapped images followed by applying the image blending to eliminate the color transitions along the seam lines. By our proposed unified panorama stitching framework, our system can generate a pleasant street-view panorama as seamlessly 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.



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

## 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 (available at <http://www.panoramatools.com/>), which also serves as the underlying core engine for many image stitching software, such as PTGui (available at <http://www.ptgui.com/>) and Hugin (available at <http://hugin.sourceforge.net/>). However, there always exist large geometric misalignments between these aligned images to different extents because those images were captured from scenes with 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 seam lines are detected for 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 much 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 respect to the original one from the sparse optical flows. Lastly, we warp the two input images based on the dense optical flows, and thus, the geometric misalignments can be greatly lessened. For the case of multiple images, a simple strategy is proposed to first handle the horizontal images and then to 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. Then, we applied the assumption presented in [52] and the epipolar geometrical constraint to remove the outliers. The major steps of the feature matching 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.

---

**Algorithm 1** The proposed feature point matching algorithm.

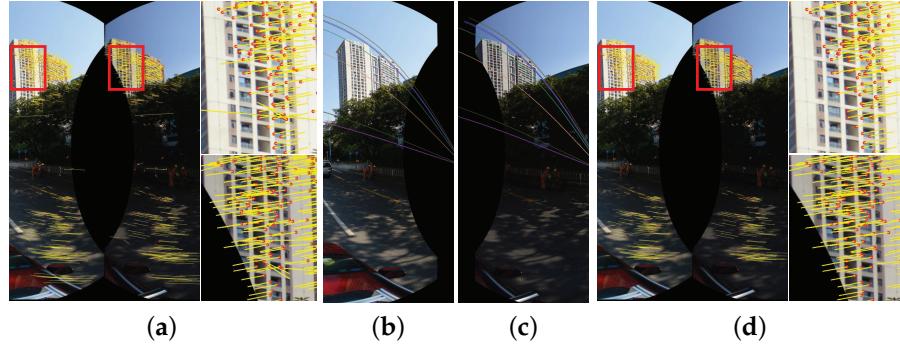
---

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) Find the neighboring inlier matches  $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$  for each match  $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle \in \mathcal{M}_{\text{initial}}$ .
  - (b) Calculate the mean motion  $\mu(\mathbf{f}_p)$  and the standard deviation  $\sigma(\mathbf{f}_p)$  of all matches in  $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$ ;
  - (c) Identify the outliers according to the criterion defined in Equation (3);
  - (d) Further identify the outliers according to the epipolar geometric constraint.
-



**Figure 2.** An illustrative example for feature point matching between two aligned panoramic images: (a) the point correspondences produced by initial matching; (b,c) several epipolar lines detected based on the estimated fundamental matrix; (d) the last point correspondences after outlier detection. 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 respect to another image.

### 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  $\mathbf{d}$  representing 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  $\mathbf{I}'$ , 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  $T_d$  cannot avoid the appearance of high rate outliers when covering 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  $T_r$  as well.

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 may still exist a few outliers in  $\mathcal{M}_{\text{initial}}$ . Of course, we need to filter out those outliers. 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 firstly apply this assumption to identify the outliers. Then, we applied the widely-used constraint of the epipolar geometric to further identify the outliers.

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:

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

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:

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

Here, we use  $\mathbf{m}(\mathbf{f}_p) = (m_p^{(x)}, m_p^{(y)}, m_p^{(x,y)})^\top$  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  $\mathcal{L}(\langle \mathbf{f}_p, \mathbf{f}'_q \rangle) = \text{Inlier}$ , 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,  $\mathcal{L}(\langle \mathbf{f}_p, \mathbf{f}'_q \rangle) = \text{Outlier}$ , 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 Equations (1) and (2). Then, the mean motion  $\boldsymbol{\mu}(\mathbf{f}_p) = (\mu_p^{(x)}, \mu_p^{(y)}, \mu_p^{(x,y)})^\top$  and the standard deviation of all the motions  $\boldsymbol{\sigma}(\mathbf{f}_p) = (\sigma_k^{(x)}, \sigma_k^{(y)}, \sigma_k^{(x,y)})^\top$  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:

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

where  $\text{dist}(\mathbf{m}(\mathbf{f}_p), \boldsymbol{\mu}(\mathbf{f}_p)) = |\mathbf{m}(\mathbf{f}_p) - \boldsymbol{\mu}(\mathbf{f}_p)|$  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.  $\lambda$  is a predefined parameter, and we set as  $\lambda = 3$ .

After that, we applied the epipolar geometric constraint based on the estimated fundamental matrix to further detect the outliers. The fundamental matrix is estimated by using the RANSAC algorithm. As we known, we cannot estimate the fundamental matrix between two panoramic images which have been projected into the spherical coordinate at first. Therefore, we first project the panoramic images into the perspective plane, and the fundamental matrix is estimated in this plane by using the RANSAC algorithm. The epipolar lines are calculated in the perspective plane at first, and then we re-project them into the spherical coordinate to identify the outliers. At last, we can find all the inliers from  $\mathcal{M}_{\text{initial}}$  denoted as of 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 of the integral pixels in  $\bar{\mathbf{I}}$  with respect to  $\mathbf{I}$  and all of the integral pixels in  $\bar{\mathbf{I}}'$  with respect to  $\mathbf{I}'$  based on the disparity vectors of the reliable point matches with respect 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:

$$\begin{cases} \mathbf{d}(\mathbf{x}_m) = \frac{1}{2}(\mathbf{x}_m - \mathbf{x}'_n) = \frac{1}{2}(x_m - x'_n, y_m - y'_n)^\top, \\ \mathbf{d}(\mathbf{x}'_n) = \frac{1}{2}(\mathbf{x}'_n - \mathbf{x}_m) = \frac{1}{2}(x'_n - x_m, y'_n - y_m)^\top, \end{cases} \quad (4)$$

where  $\mathbf{x}_m = (x_m, y_m)^\top$  and  $\mathbf{x}'_n = (x'_n, y'_n)^\top$ . In 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 of the integral pixels  $\{\mathbf{p}\}_{\mathbf{p} \in \bar{\mathbf{I}}}$  and  $\{\mathbf{p}'\}_{\mathbf{p}' \in \bar{\mathbf{I}}'}$  in the warped images  $\bar{\mathbf{I}}$  and  $\bar{\mathbf{I}}'$  with respect to the original images  $\mathbf{I}$  and  $\mathbf{I}'$ , respectively.

## 2.3. Two Image Warping

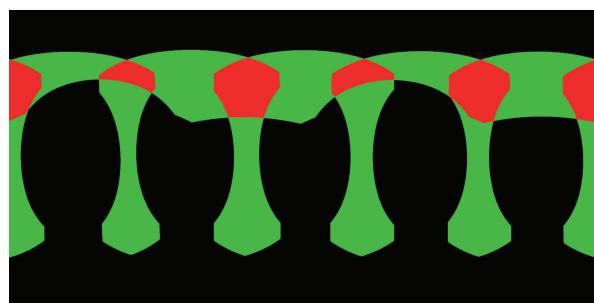
Here, we demonstrate how to generate the warped image  $\bar{\mathbf{I}}$  from the original image  $\mathbf{I}$  based on the dense optical flows  $\mathcal{D}(\bar{\mathbf{I}})$  of  $\bar{\mathbf{I}}$  with respect to  $\mathbf{I}$ , and the generation of the warped image  $\bar{\mathbf{I}}'$  is 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.

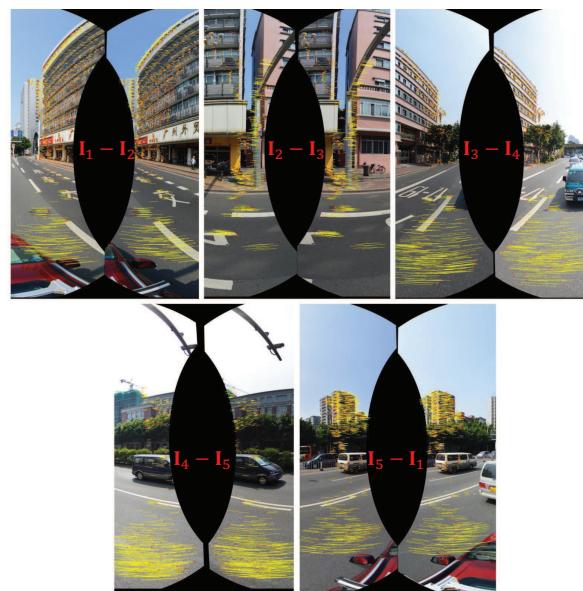
## 2.4. Multiple Image Warping

Until now, we have introduced how to warp two images based on the optical flows. However, 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 five horizontal ones and one 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 3. For this particular case, here we will in detail 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 3. 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

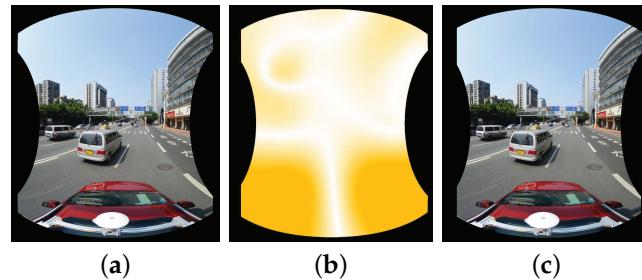
deal with the vertical image pair. For each horizontal image pair, we match them one by one by the method presented in Section 2.1, as an illustrative example shown in Figure 4, from which we can find that one horizontal image is overlapped with two adjacent images in the horizontal direction. For example, for the image  $I_1$ , it overlaps with  $I_2$  and  $I_5$ , respectively, so we need to collect all matching points from these two overlap regions as the control points for warping  $I_1$ . The dense optical flow field of the warped image  $\bar{I}_1$  with respect to the original image  $I_1$  can be approximately interpolated based on those control points via the MBA algorithm. Therefore, five horizontal warped images  $\bar{I}_1$ ,  $\bar{I}_2$ ,  $\bar{I}_3$ ,  $\bar{I}_4$  and  $\bar{I}_5$  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  $I_H$  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  $I_6$  and the horizontal blended image  $I_H$  will be warped according to those matching points as the control ones.



**Figure 3.** 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.



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



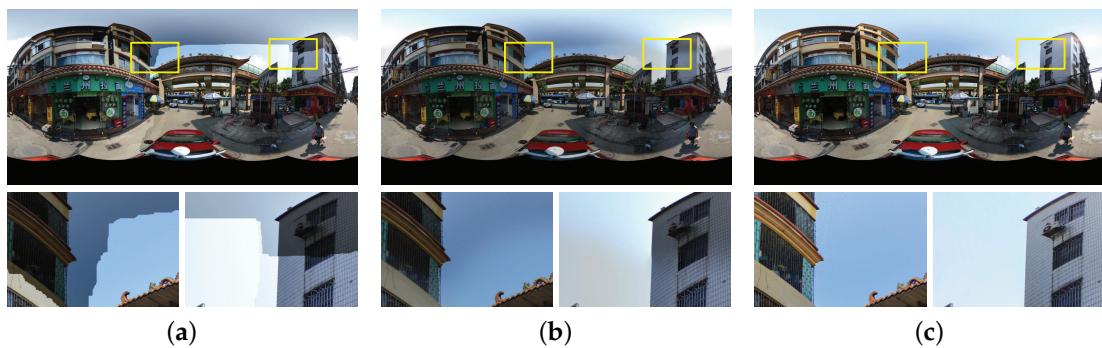
**Figure 5.** An illustrative example of image warping: (a) the original aligned image; (b) the dense optical flows approximately interpolated by the Multilevel B-splines Approximation (MBA) algorithm; (c) the last warped image. In (b), the deeper orange means larger disparity.

## 2.5. Summary

In this section, we proposed a new image warping method to eliminate the large geometrical misalignments between adjacent aligned images. This method includes three stages: feature point matching, approximate interpolation of dense optical flows and image warping. In the first stage, we first used the SURF algorithm [53] to find the initial matching points between two images and then applied the assumption presented in [52] and the epipolar constraint to eliminate outliers. The sparse optical flows can be calculated between two matched feature points. In the second stage, we proposed to apply the dense optical flows to simulate the non-rigid deformations between aligned images. The dense optical flows will be interpolated via the MBA algorithm [37] from sparse ones. In the last stage, we proposed to warp the initially aligned images based on the dense optical fields, and we also proposed a strategy to handle multiple image warping. The key contribution of this section is that we proposed to use the optical flows to simulate and eliminate the large non-rigid geometrical misalignments between input aligned street-view images. This method is simple and completed by combining some conventional algorithms, such as SURF, MBA, and so on. However, to the best of the authors' knowledge, no one has applied this efficient strategy to handle the large geometrical misalignments between street-view panoramic images before color correction and image mosaicking.

## 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 this easily by smoothing the color along the seam lines. However, it does not work well for input images with very large color differences. The simple image blending may not be able to efficiently conceal the artifacts if we do not magnificently correct color differences between images in advance, which results in low-quality panoramic images, as the illustrative example shown in Figure 6. In addition, the large color differences maybe also affect the quality of the detected seam lines. Thus, in this paper, we propose to reduce the color differences between warped images before the optimal seam lines are detected. Generally, the color differences should be also corrected before the image warping step to ensure the quality of feature matching results. However, our adopted SURF feature matching algorithm is robust enough to the large photometric inconsistencies, so there is no obvious influence on our algorithm if we apply the color correction after image warping.



**Figure 6.** An example of our proposed color correction strategy used to improve the panorama stitching quality before applying the image blending. (a) The initial stitched image; (b) Only image blending used; (c) Image blending after color correction.

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 the same as 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 of 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:

$$I_{\min} = I_{\lceil N \times c \% \rceil} \text{ and } I_{\max} = I_{\lceil N \times (1 - c \% ) \rceil}, \quad (5)$$

where  $\lceil \Delta \rceil$  denotes the upper integer of a real value  $\Delta$  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:

$$I' = \begin{cases} 0, & I \leq V_{\min}, \\ 255 \times \frac{I - V_{\min}}{V_{\max} - V_{\min}}, & V_{\min} < I < V_{\max}, \\ 255, & I \geq V_{\max}. \end{cases} \quad (6)$$

In the same way, all of 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 being 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 exists 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  $\check{C}$  mean 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  $\{\mathbf{P}_A^i\}_{i=1}^{N_A}$  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:

$$Cost(\mathbf{P}_A^i, \mathbf{P}_B^j) = \frac{F_A^i + F_B^j}{2F_{\max}} \times \frac{\min(F_A^i, F_B^j)}{\max(F_A^i, F_B^j)} \times \frac{\max(\hat{C}_A^i - \check{C}_A^i, \hat{C}_B^j - \check{C}_B^j)}{\max(\hat{C}_A^i, \hat{C}_B^j) - \min(\check{C}_A^i, \check{C}_B^j)}, \quad (7)$$

where  $F_{\max}$  is the maximal frequency of all of 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 generate higher costs, which may be peak 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  $\mathbf{P}_B^j$ . The last term is applied to ensure that the accumulative values of two possibly matched extreme points  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  are approximate. From this term, we can find that if the small ranges 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 being close to one. In contrast, if the numerator is smaller and the denominator is larger, this term will approach to zero. 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:

$$\begin{cases} \frac{\min(F_A^i, F_B^j)}{\max(F_A^i, F_B^j)} < \theta_f, \\ \check{C}_A^i > \hat{C}_B^j + \theta_c \times C_{\max}, \\ \check{C}_B^j > \hat{C}_A^i + \theta_c \times C_{\max}, \end{cases} \quad (8)$$

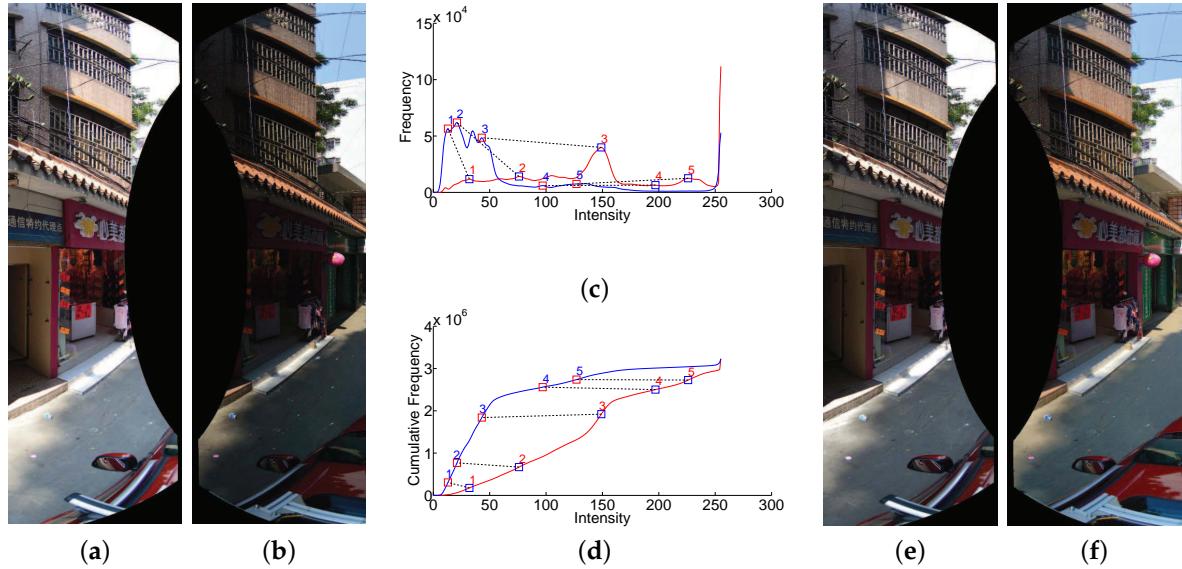
where  $\theta_f$  and  $\theta_c$  are two empirical thresholds ( $\theta_f = 0.25$  and  $\theta_c = 0.02$  were used in this paper)  $C_{\max}$  is the maximal value of CDF, namely the valid pixel number of overlap regions. The matching cost  $Cost(\mathbf{P}_A^i, \mathbf{P}_B^j)$  is set to zero, 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 of 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 two input images have large color differences in overlap regions, as shown in Figure 7a,b. We find 5 matched extreme points in the PDF of one channel. Based on those correspondences, the large color differences can be

eliminated, as shown in Figure 7e,f. From this example, we can find that our proposed approach can handle the images with large color differences very well.



**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.

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  $\kappa$  is a given threshold in advance ( $\kappa = 0.1$  was used in this paper).

### 3.4. Correcting Color Difference

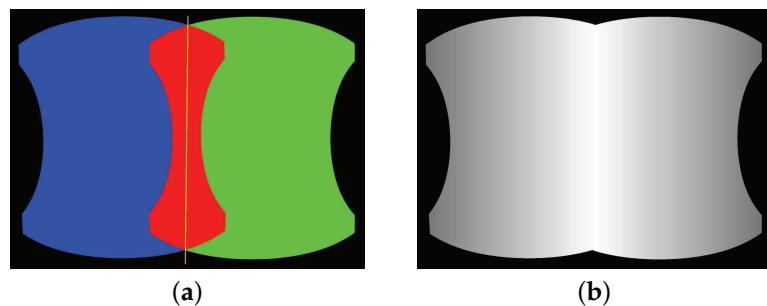
The extracted matching points in the overlap image regions are then applied to correct the intensities of two adjacent images, including the pixels in non-overlap regions. Let  $\{Q_A^k\}_{k=1}^N$  and  $\{Q_B^k\}_{k=1}^N$  be the final matching points in CDFs in the overlap regions **A** and **B** with  $N$  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  $\hat{L}_A^k = \hat{L}_B^k = (L_A^k + L_B^k)/2$ . 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:

$$\hat{L}_A(\mathbf{p}) = \hat{L}_A^l + (L_A(\mathbf{p}) - L_A^l) \frac{\hat{L}_A^u - \hat{L}_A^l}{L_A^u - L_A^l}, \quad (9)$$

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

$$L'_A(\mathbf{p}) = (1 - \alpha(\mathbf{p}))L_A(\mathbf{p}) + \alpha(\mathbf{p})\hat{L}_A(\mathbf{p}), \quad (10)$$

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 relates to the distance between the pixel  $\mathbf{p}$  and the center line of the overlap image region, which ranges between zero and one as shown in Figure 8, where the smaller the distance to the center line, the larger the  $\alpha$  is. All the pixels in other images will be processed in the same way.



**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.

### 3.5. Summary

In this section, we proposed a new color correction method to reduce the color differences between warped images. This method also includes three stages: automatic contrast adjustment, histogram extreme point detection and matching and color difference correction. In the first stage, we applied the traditional method to adjust the image contrast. This stage can be regarded as the pre-processing. The second stage is the key contribution of the proposed color correction method. In this stage, we proposed a new cost function to evaluate the similarity between two extreme points extracted from two PDFs and then find all matched extreme points iteratively. In the last stage, based on the matched histogram extreme points, the color differences can be corrected easily. The key contribution of this section is that we proposed a new color correction method via matching extreme points of histograms. This method is effective and efficient, which can handle large color differences between two images.

## 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 seam lines in the overlap image regions between warped images to magnificently conceal the parallax. In this paper, the optimal seam lines between color corrected images will be efficiently extracted using the graph cuts-based seam line detection algorithm presented in [19]. For multiple images, we apply the traditional *frame-to-frame optimization* strategy to efficiently find all optimal seam

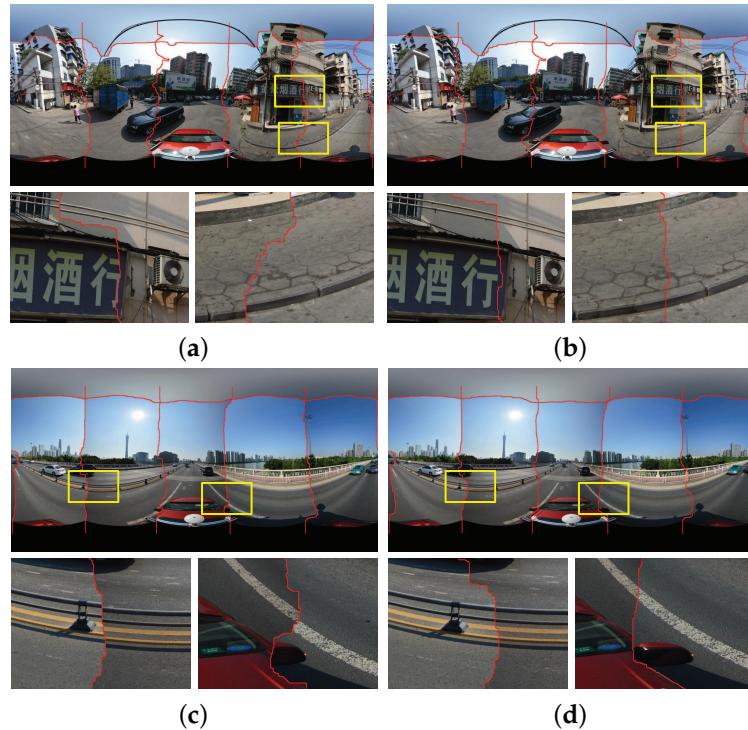
lines. The details of this strategy are described in Section 3.1 of [19]. Furthermore, the Laplacian pyramid blending [49] algorithm will be further applied to eliminate the stitching artifacts caused by small color differences along the seam lines.

## 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 this 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  $12,000 \times 6000$  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 3. Our algorithms in this paper were implemented with C++ under Windows and tested in a computer with an Intel Core i7-4770 at 3.4 GHz and the 16 GB RAM memory. Due to the limit of pages, more experimental results and analysis are presented at <http://cvrs.whu.edu.cn/projects/PanoStitching/>. In addition, the stitching framework provided by this paper has been adopted by *IShowChina*(<http://www.ishowchina.com/>) to produce the street-view panoramas for many cities in China. Nowadays, the street-view maps of some cities are already available on *IShowChina*, such as Beijing(<http://map.ishowchina.com/index.html#c=110000&s=005200-0-201308190236430390&heading=127&pitch=-3&zoom=0>) and Macao(<http://map.ishowchina.com/index.html#c=440400&s=005223-0-201401210342400290&heading=353&pitch=10&zoom=0>). The large set of panoramic images generated by our proposed framework have proved that our framework is robust and effective.

### 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 Figure 9a,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 Figure 9a,b, we observed that the stitching artifacts caused by the geometric dislocation in the panorama, as shown in Figure 9a, stitched without the use of image warping algorithm are more obvious than the panorama, as shown in Figure 9b, stitched with its use. Noticeably, the stitching artifacts caused by geometric dislocation become smaller as expected when the image warping algorithm was applied, as shown in Figure 9b. While not using the image warping algorithm, the geometric dislocation is very large, as shown in Figure 9a. 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 17.89 s in the above experiment, only the elapsed time in six optimal seamlines detection is included. However, with its use, our algorithm took around 70.93 s 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.



**Figure 9.** 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,d)) or not (Left: (a,c), namely, the stitching results of [19]). The red lines stand for the detected optimal seamlines between images.

The comparative experimental results on another group of panoramic images are presented in Figure 9c,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 13.19 s and 56.77 s, respectively.

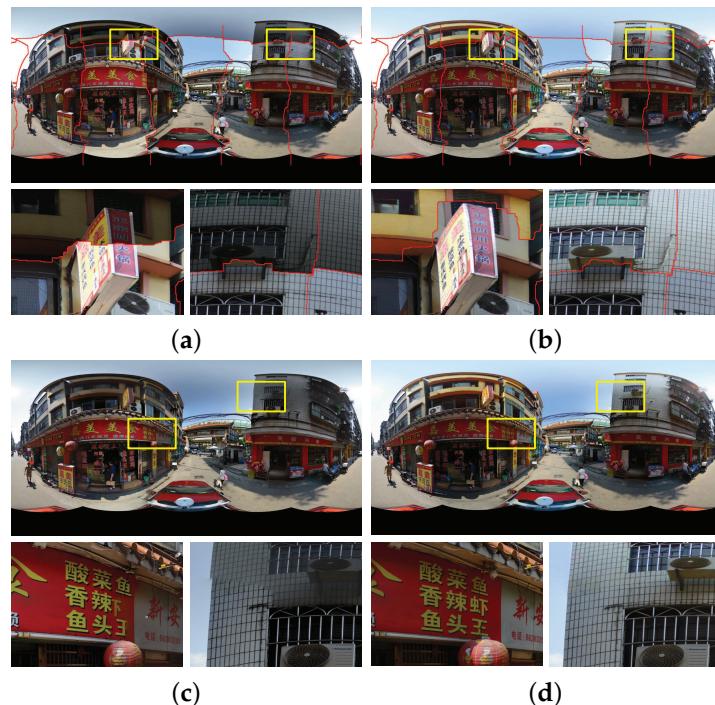
From the above experimental results on two groups of panoramic images, we observed that our proposed image warping algorithm can effectively eliminate the stitching artifacts caused by the geometrical dislocations and can also slightly improve the quality of the found optimal seamlines to some extent.

### 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* (Available at <http://enblend.sourceforge.net/>) which are popularly used to generate the street-view panorama by stitching the registered panoramic images.

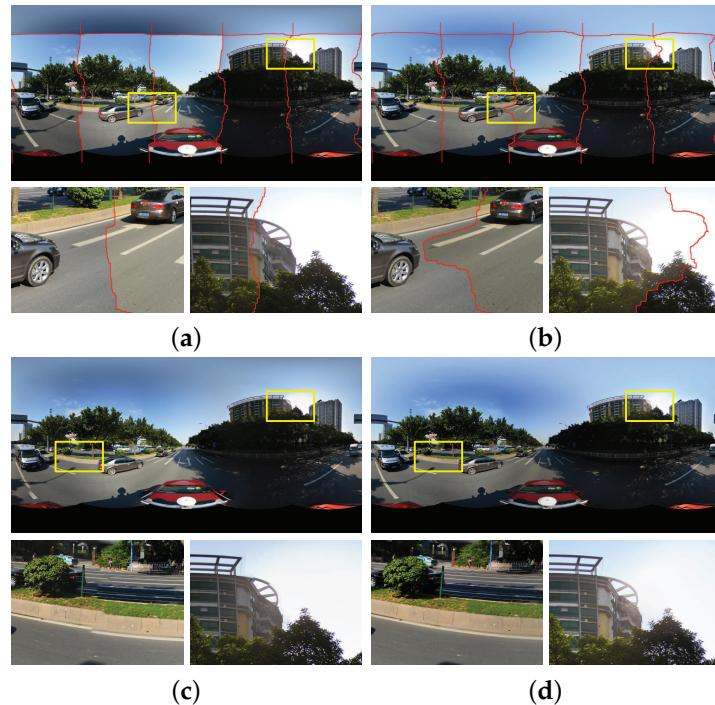
Figure 10 shows the experimental results on the first group of panoramic images. The panorama stitching results without and with the use of color correction are shown in Figure 10a,b, respectively. From the whole stitching results and especially the detailed local regions shown in Figure 10a,b, we can 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 Figure 10a,b. In the aspect of computational cost, without the use of the color correction, our algorithm took around 18.01 s to find all six optimal seamlines. With its use, our algorithm took around 33.52 s to

correct the color differences and find the optimal seamlines, means that the color correction algorithm took around 15.51 s. To generate the last panorama, the Laplacian pyramid blending algorithm was further applied, whose generated result is shown in Figure 10d. And in Figure 10c, we also present the last panorama generated by *Enblend*. From 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 10c but they almost disappeared in our produced panorama, as shown in Figure 10d. In the aspect of computational cost, the Laplacian pyramid blending algorithm took 35.56 s.



**Figure 10.** 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).

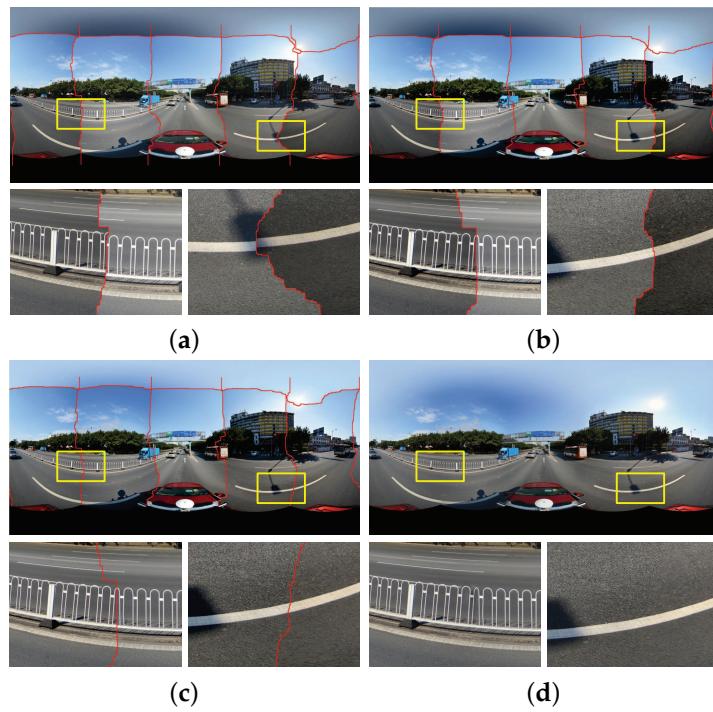
The experimental results on another group of panoramic images are presented in Figure 11 and the similar conclusion can be drawn. The large color differences were greatly reduced by our proposed color correction algorithm, especially in the regions of sky and the tall buildings, and the quality of the 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 17.33, 13.77 and 35.61 s, respectively.



**Figure 11.** 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 12. At first, Figure 12a 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 Figure 12b,c, respectively. The last blended panorama generated by our proposed system is shown in Figure 12d 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.

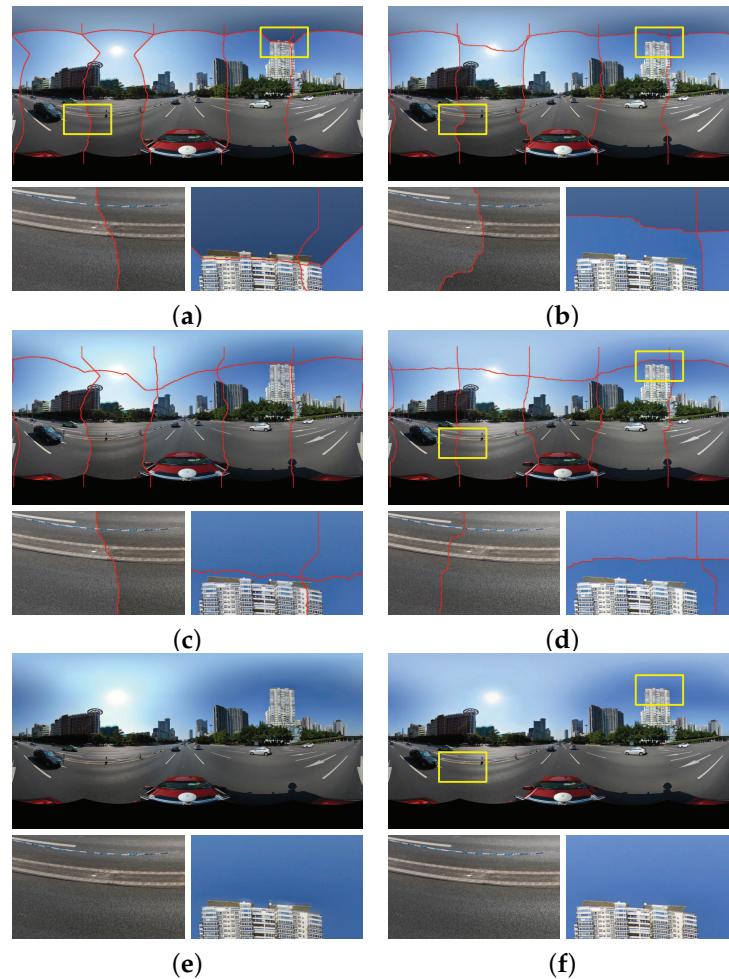


**Figure 12.** The stitching results with different combination of optimal seamlne detection ( $S$ ), image warping ( $W$ ), color correction ( $C$ ) and image blending ( $B$ ) algorithms: (a)  $S$  (the result generated by [19]); (b)  $W + S$ ; (c)  $W + C + S$ ; (d)  $W + C + S + B$ . The computational times of (a–d) are 18.00 s, 69.08 s, 86.63 s and 123.68 s, 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 another algorithm, the open source library and the open tools. At first, we compared our proposed framework with the Xiong and Pulli's approach [42] step by step. 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 more fair, we replaced the Poisson blending algorithm in the tested Xiong and Pulli's approach with the Laplacian pyramid blending algorithm.

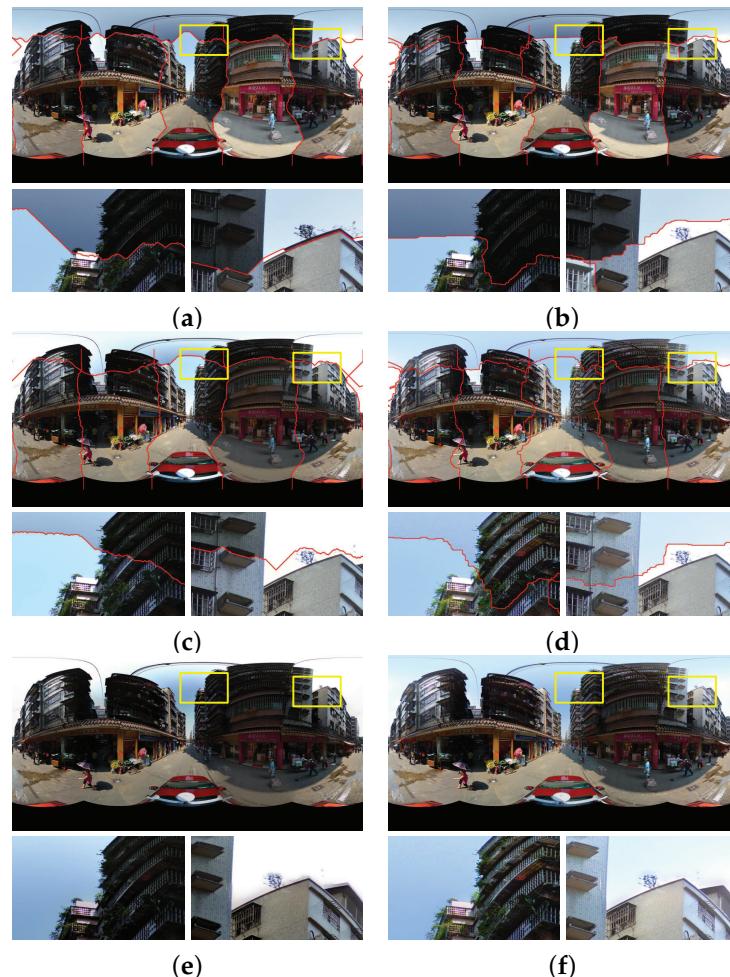
Figure 13 shows the stitching results of the first group of images with relatively small color differences. Figure 13a,b illustrate the stitching results just with the detected seamlines of the Xiong and Pulli's approach and our approach without the use of color correction, respectively. Figure 13c,d 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. Figure 13e,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 13e, which disappeared in the panorama generated by our approach, as shown in Figure 13f. This is mainly because the horizontal seamlne between bottom and top input images detected by the Xiong and Pulli's approach is close to this building, as shown in Figure 13c. In conclusion, if the color differences between input images are small, both of two approaches can generate high-quality panoramas.



**Figure 13.** 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.

Figure 14 shows the stitching results of the second group of images with very large color differences. Figure 13a,b show the stitching results of the Xiong and Pulli's approach and our approach without the use of color correction, respectively. Figure 13c,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. Figure 13e,f show the last blended panoramas generated by two approaches, respectively. In the second enlarged region of Figure 14e, 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 14f. In addition, in Figure 14e, 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 14f.

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.



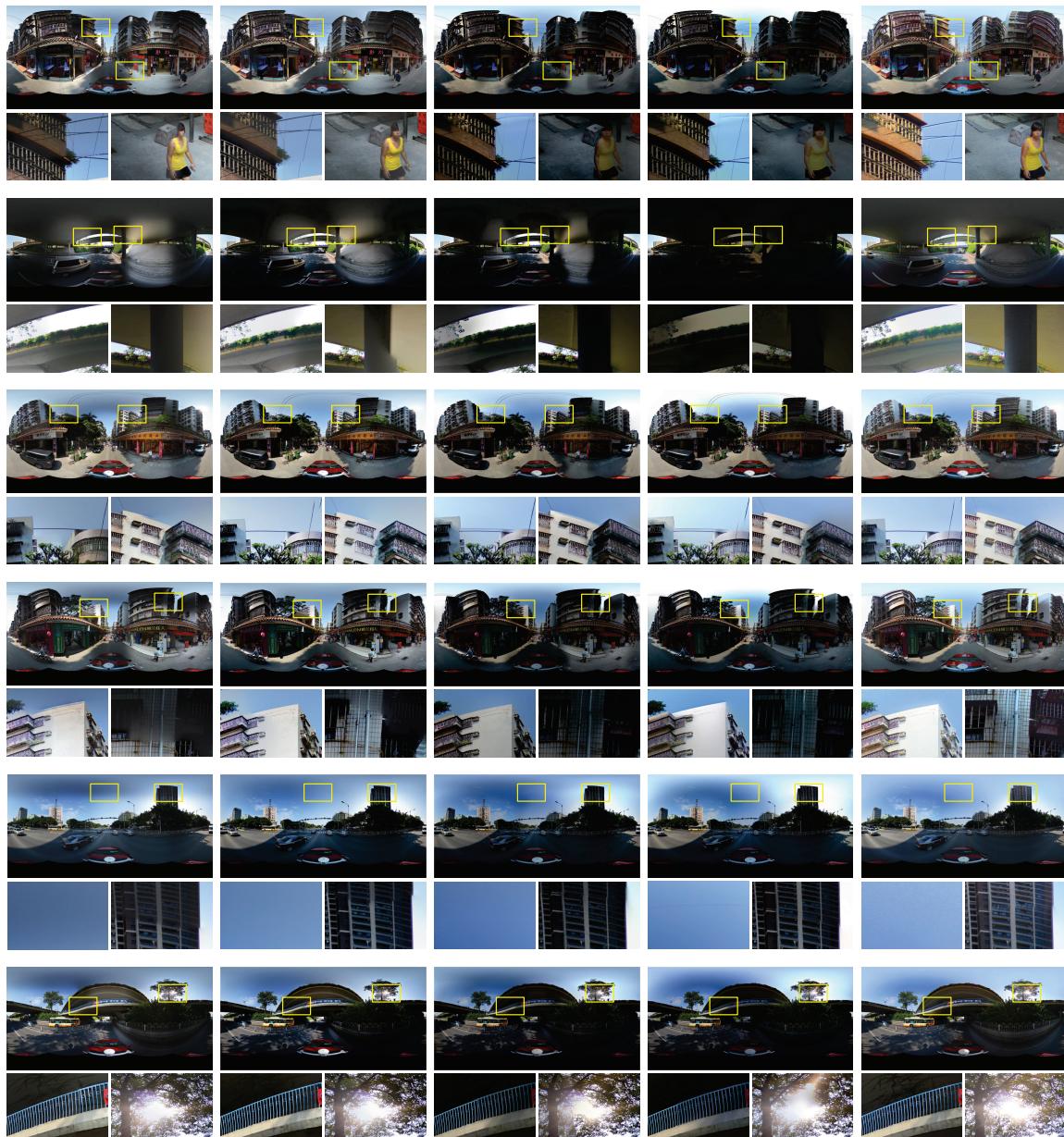
**Figure 14.** Visual comparison between our approach in the left column and the Xiong and Pulli's approach in the right column on the second group of images with large color differences: (a,b) the results without the use of color correction; (c,d) the results with the use of color correction; (e,f)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

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 16.83 s and 16.08 s, respectively, which are almost the same. But our color correction algorithm is more effective than their algorithm.

**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 Approach (s)	16.92	16.83	36.60	70.36
Xiong and Pulli's Approach (s)	13.77	16.08	36.72	66.5815

In addition, we presented more comparative results in Figure 15. We also compared our proposed approach with two widely used open tools *Enblend* and *Smartblend* (Available at <http://wiki.panotools.org/SmartBlend>), and also with the stitching functions provided by OpenCV (Available at <http://opencv.org/>). Similar with the experiments presented in Figures 13 and 14, we also used the warped images generated by our image warping algorithm as the input images. From those more visually comparative results, we also can find that our proposed system can generate the best high-quality street-view panoramas.



**Figure 15.** Six groups of panoramic images generated by five different tools or approaches. Form left to right: *Smartblend*, *Enblend*, OpenCV stitching functions, Xiong and Pulli's approach and our proposed approach. The faces in the first group of images were blurred due to privacy protection.

At last, to strongly prove the superiority of our proposed approach, we applied the Oriented Gradients Image Assessment (OG-IQA) algorithm [55] to quantitatively evaluate the qualities of the final panoramic images stitched by different tools and approaches. Here, we evaluated the panoramic images presented in Figures 13 and 14, and six groups of panoramic images presented in Figure 15.

The results of quality assessment are presented in Table 2. In Table 2, we used the Group 1 to Group 6 to represent the panoramic images in the first row to the last row of Figure 15. From these results, we observed that our approach performed best in most of cases, except of Group 4. But, in Group 4, the quality score of our approach is only litter less than the score of *Enblend*.

**Table 2.** The quantitative quality assessments of five tested tools and approaches.

	Figure 13	Figure 14	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
<i>Smartblend</i>	-	-	0.6517	0.6576	0.6011	0.6887	0.6022	0.6498
<i>Enblend</i>	-	-	0.7207	0.6773	0.7076	<b>0.7511</b>	0.7359	0.7341
OpenCV	-	-	0.6371	0.6279	0.5880	0.6026	0.5522	0.6203
Xiong and Pulli's Approach	0.5469	0.4693	0.5527	0.4283	0.5306	0.5265	0.6173	0.6075
Our Proposed Approach	<b>0.7055</b>	<b>0.6817</b>	<b>0.7248</b>	<b>0.7585</b>	<b>0.7144</b>	0.7333	<b>0.7481</b>	<b>0.7469</b>

## 6. Conclusions

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 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.
- 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. Our proposed framework outperforms the open-source tools *Enblend*, *Smartblend*, OpenCV stitching functions and the approach proposed by [42].

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.

**Acknowledgments:** 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), the Jiangsu Province Science and Technology Support Program, China (Project No. BE2014866), and LIESMARS Special Research Funding.

**Author Contributions:** Li Li and Jian Yao conducted the algorithm design, and Li Li wrote the paper and performed all the experiments under the supervision of Jian Yao. Renping Xie, Menghan Xia and Wei Zhang worked on the image alignment and contributed to prepare and analyze the experimental data. All authors were involved in modifying the paper, the literature review and the discussion of the results.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Du, Q.; Raksuntorn, N.; Orduyilmaz, A.; Bruce, L.M. Automatic registration and mosaicking for airborne multispectral image sequences. *Photogramm. Eng. Remote Sens.* **2008**, *74*, 169–181.
- 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 J. Photogramm. Remote Sens.* **2010**, *65*, 86–92.
- Wan, Y.; Wang, D.; Xiao, J.; Lai, X.; Xu, J. Automatic determination of seamlines for aerial image mosaicking based on vector roads alone. *ISPRS J. Photogramm. Remote Sens.* **2013**, *76*, 1–10.
- Pan, J.; Zhou, Q.; Wang, M. Seamline Determination Based on Segmentation for Urban Image Mosaicking. *IEEE Trans. Geosci. Remote Sens. Lett.* **2014**, *11*, 1335–1339.
- Pang, S.; Sun, M.; Hu, X.; Zhang, Z. SGM-based seamline determination for urban orthophoto mosaicking. *ISPRS J. Photogramm. Remote Sens.* **2016**, *112*, 1–12.
- Kerschner, M. Seamline detection in colour orthoimage mosaicking by use of twin snakes. *ISPRS J. Photogramm. Remote Sens.* **2001**, *56*, 53–64.
- Soille, P. Morphological image compositing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 673–683.
- Yang, Y.; Gao, Y.; Li, H.; Han, Y. An algorithm for remote sensing image mosaic based on valid area. In Proceedings of the International Symposium on Image and Data Fusion (ISIDF), Tengchong, China, 9–11 August 2011.
- Li, X.; Hui, N.; Shen, H.; Fu, Y.; Zhang, L. A robust mosaicking procedure for high spatial resolution remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2015**, *109*, 108–125.
- Agarwala, A.; Dontcheva, M.; Agrawala, M.; Drucker, S.; Colburn, A.; Curless, B.; Salesin, D.; Cohen, M. Interactive Digital Photomontage. *ACM Trans. Graph. (TOG)* **2004**, *23*, 294–302.
- Szeliski, R. Image alignment and stitching: A tutorial. *Found. Trends® Comput. Graph. Vis.* **2006**, *2*, 1–104.
- Brown, M.; Lowe, D.G. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.* **2007**, *74*, 59–73.
- Gracias, N.; Mahoor, M.; Negahdaripour, S.; Gleason, A. Fast image blending using watersheds and graph cuts. *Image Vis. Comput.* **2009**, *27*, 597–607.
- Prados, R.; Garcia, R.; Gracias, N.; Escartin, J.; Neumann, L. A novel blending technique for underwater gigamosaicing. *IEEE J. Ocean. Eng.* **2012**, *37*, 626–644.
- Philip, S.; Summa, B.; Tierny, J.; Bremer, P.T.; Pascucci, V. Distributed Seams for Gigapixel Panoramas. *IEEE Trans. Vis. Comput. Graph.* **2015**, *21*, 350–362.
- Kwatra, V.; Schödl, A.; Essa, I.; Turk, G.; Bobick, A. Graphcut Textures Image and Video Synthesis Using Graph Cuts. *ACM Trans. Graph. (TOG)* **2003**, *22*, 277–286.
- 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. *Int. J. Remote Sens.* **2012**, *33*, 1000–1014.
- Mills, S.; McLeod, P. Global seamline networks for orthomosaic generation via local search. *ISPRS J. Photogramm. Remote Sens.* **2013**, *75*, 101–111.
- Li, L.; Yao, J.; Lu, X.; Tu, J.; Shan, J. Optimal seamline detection for multiple image mosaicking via graph cuts. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 1–16.
- Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active contour models. *Int. J. Comput. Vis.* **1988**, *1*, 321–331.
- Dijkstra, E.W. A note on two problems in connexion with graphs. *Numerische Mathematik* **1959**, *1*, 269–271.
- Bellman, R. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
- Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239.
- Glasbey, C.A.; Mardia, K.V. A review of image-warping methods. *J. Appl. Stat.* **1998**, *25*, 155–171.
- Lee, S.Y.; Chwa, K.Y.; Hahn, J.K.; Shin, S.Y. Image morphing using deformation techniques. *J. Vis. Comput. Animat.* **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 Trans. Graph. (TOG)* **2014**, *33*, 1–12.
27. Liu, F.; Gleicher, M. Automatic image retargeting with fisheye-view warping. In Proceedings of the ACM Symposium on User Interface Software and Technology, Seattle, WA, USA, 23–26 October 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 Trans. Multimed.* **2013**, *15*, 359–368.
29. Zaragoza, J.; Chin, T.J.; Brown, M.; Suter, D. As-projective-as-possible image stitching with moving DLT. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2339–2346.
30. Lin, C.C.; Pankanti, S.U.; Ramamurthy, K.N.; Aravkin, A.Y. Adaptive as-natural-as-possible image stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1155–1163.
31. Bookstein, F.L. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 567–585.
32. Alexa, M.; Cohen-Or, D.; Levin, D. As-Rigid-As-Possible Shape Interpolation. In Proceedings of the International Conference on Computer Graphics and Interactive Techniques Conference (SIGGRAPH), New Orleans, LA, USA, 23–28 July 2000; pp. 157–164.
33. Igarashi, T.; Moscovich, T.; Hughes, J.F. As-rigid-as-possible shape manipulation. *ACM Trans. Graph. (TOG)* **2005**, *24*, 1134–1141.
34. Schaefer, S.; McPhail, T.; Warren, J. Image deformation using moving least squares. *ACM Trans. Graph. (TOG)* **2006**, *25*, 533–540.
35. Levin, D. The approximation power of moving least-squares. *Math. Comput. Am. Math. Soc.* **1998**, *67*, 1517–1531.
36. Chen, T.L.; Geman, S. Image warping using radial basis functions. *J. Appl. Stat.* **2014**, *41*, 242–258.
37. Lee, S.; Wolberg, G.; Shin, S.Y. Scattered data interpolation with multilevel B-splines. *IEEE Trans. Vis. Comput. Graph.* **1997**, *3*, 228–244.
38. Yao, J. Modeling and Rendering from Multiple Views. Ph.D. Thesis, The Chinse University of Hong Kong, Hong Kong, China, 2006.
39. Yao, J.; Cham, W.K. 3D modeling and rendering from multiple wide-baseline images by match propagation. *Signal Process.: Image Commun.* **2006**, *21*, 506–518.
40. Xu, W.; Mulligan, J. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 263–270.
41. Tian, G.Y.; Gledhill, D.; Taylor, D.; Clarke, D. Colour correction for panoramic imaging. In Proceedings of the International Conference on Information Visualisation, London, UK, 10–12 July 2002; Volume 215, pp. 483–488.
42. Xiong, Y.; Pulli, K. Fast panorama stitching for high-quality panoramic images on mobile phones. *IEEE Trans. Consum. Electron.* **2010**, *56*, 298–306.
43. Hwang, Y.; Lee, J.Y.; Kweon, I.S.; Kim, S.J. Color transfer using probabilistic moving least squares. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 3342–3349.
44. Yamamoto, K.; Oi, R. Color correction for multi-view video using energy minimization of view networks. *Int. J. Autom. Comput.* **2008**, *5*, 234–245.
45. Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
46. Fecker, U.; Barkowsky, M.; Kaup, A. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1258–1267.
47. Uyttendaele, M.; Eden, A.; Skeliski, R. Eliminating ghosting and exposure artifacts in image mosaics. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 8–14 December 2001; Volume 2, pp. 509–516.
48. Porter, T.; Duff, T. Compositing Digital Images. *ACM SIGGRAPH Comput. Graph.* **1984**, *18*, 253–259.
49. Burt, P.J.; Adelson, E.H. A multiresolution spline with application to image mosaics. *ACM Trans. Graph. (TOG)* **1983**, *2*, 217–236.
50. Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. *ACM Trans. Graph. (TOG)* **2003**, *22*, 313–318.

51. Levin, A.; Zomet, A.; Peleg, S.; Weiss, Y. Seamless image stitching in the gradient domain. In Proceedings of the European Conference on Computer Vision (ECCV), Prague, Czech Republic, 11–14 May 2004.
52. Yao, J.; Cham, W.K. Robust multi-view feature matching from multiple unordered views. *Pattern Recognit.* **2007**, *40*, 3081–3099.
53. Bay, H.; Tuytelaars, T.; Gool, L.V. SURF: Speeded up Robust Features. *Comput. Vis. Image Underst.* **2006**, *110*, 404–417.
54. HaCohen, Y.; Shechtman, E.; Goldman, D.B.; Lischinski, D. Optimizing color consistency in photo collections. *ACM Trans. Graph. (TOG)* **2013**, *32*, 38:1–38:10.
55. Liu, L.; Hua, Y.; Zhao, Q.; Huang, H.; Bovik, A.C. Blind image quality assessment by relative gradient statistics and adaboosting neural network. *Signal Process.: Image Commun.* **2015**, *40*, 1–15.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).