

# A Unified Framework for Street-View Panorama Stitching

Li Li, Jian Yao\*, Renping Xie, Menghan Xia

*Computer Vision and Remote Sensing (CVRS) Lab, School of Remote Sensing and Information Engineering,  
Wuhan University, Wuhan, Hubei, P.R. China*

Wei Zhang

*School of Control Science and Engineering, Shandong University, Jinan, Shandong, P.R. China*

---

## Abstract

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

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

---

\*Corresponding author.

Email address: [jian.yao@whu.edu.cn](mailto:jian.yao@whu.edu.cn) (Jian Yao)

URL: <http://cvrs.whu.edu.cn/> (Jian Yao)

<sup>1</sup> **1. Introduction**

<sup>2</sup> Nowadays, as the development of street-view panoramas which provide 360° panoramic  
<sup>3</sup> views along streets in the real world, the demand for high-quality panoramic images gradual-  
<sup>4</sup> ly becomes bigger. Image stitching is the key technology to produce high-quality panoram-  
<sup>5</sup> ic images, which is also an important and classical problem in the field of photogramme-  
<sup>6</sup> try (Du et al., 2008; Chon et al., 2010; Wan et al., 2013; Pan et al., 2014; Pang et al., 2016),  
<sup>7</sup> remote sensing (Kerschner, 2001; Soille, 2006; Yang et al., 2011; Li et al., 2015) and comput-  
<sup>8</sup> er vision (Agarwala et al., 2004; Szeliski, 2006; Brown and Lowe, 2007; Gracias et al., 2009;  
<sup>9</sup> Prados et al., 2012; Philip et al., 2015), which is widely used to merge multiple aligned images  
<sup>10</sup> into a single wide-angle composite image as seamlessly as possible.

<sup>11</sup> In an ideally static scene in which both the geometric misalignments and the photometric  
<sup>12</sup> inconsistencies don't exist or are not obviously visible in overlap regions, the stitched or mo-  
<sup>13</sup> saicked image looks perfect only when the geometric distance criterion is used. However, as we  
<sup>14</sup> know, most of street-view panoramic images are captured by the panoramic camera mounted on  
<sup>15</sup> the mobile platform. Generally, the panoramic camera is comprised of multiple wide-angle or  
<sup>16</sup> fish-eye cameras whose projection centers are slightly different. Therefore, those images cannot  
<sup>17</sup> be precisely aligned in geometry, namely, there exist the geometric deviations for corresponding  
<sup>18</sup> pixels from different images to different extents. In addition, there also exist photometric in-  
<sup>19</sup> consistencies to different extents in overlap regions between adjacent images due to illumination  
<sup>20</sup> variations and/or different exposure setting. This paper focuses on creating a visually pleasant  
<sup>21</sup> street-view panorama by stitching or mosaicking the street-view panoramic images among which  
<sup>22</sup> there may exist the severe geometric misalignments and the strong photometric inconsistencies.

<sup>23</sup> One traditional and efficient way to eliminate the stitching artifacts caused by the large  
<sup>24</sup> geometric misalignments existed in the input aligned panoramic images is to detect the optimal  
<sup>25</sup> seamlines which avoid crossing majority of visually obvious objects and most of overlap regions  
<sup>26</sup> with low image similarity and large object dislocation. The optimal seamline detection methods  
<sup>27</sup> search for the seamlines in overlap regions between images where their intensity or gradient dif-  
<sup>28</sup> ferences are not significant. Based on the optimally detected seamlines, multiple aligned images  
<sup>29</sup> can be mosaicked into a single composite image in which the obvious image parallax caused  
<sup>30</sup> by image misalignments can be magnificently concealed. Many methods (Kerschner, 2001;  
<sup>31</sup> Kwatra et al., 2003; Chon et al., 2010; Yu et al., 2012; Wan et al., 2013; Mills and McLeod,  
<sup>32</sup> 2013; Pan et al., 2014; Pang et al., 2016; Li et al., 2016) regarded the optimal seamline detec-  
<sup>33</sup> tion as an energy optimization problem and solved it by minimizing a specially designed energy

34 function defined to represent the difference between the original images along the seamlines.  
35 For these methods, the key ideas concentrate on how to define the effective energy functions  
36 and how to guarantee the optimality of the solution. The energy functions are often defined by  
37 considering color, gradient and texture, and are optimized via different optimization algorithms,  
38 e.g., snake model (Kass et al., 1988), Dijkstra's algorithm (Dijkstra, 1959), dynamic program-  
39 ming (Bellman, 1957), and graph cuts (Boykov et al., 2001). Nowadays, the optimal seamline  
40 detected by many algorithms can avoid crossing the regions with low image similarity and high  
41 object dislocation. In our previous work presented in (Li et al., 2016), we proposed an efficient  
42 optimal seamline detection algorithm for mosaicking aerial and panoramic images based on the  
43 graph cuts energy minimization framework. In this paper, we will apply this algorithm to detect  
44 the optimal seamlines.

45 However, when the geometric misalignments are very large, the stitching artifacts maybe  
46 cannot be completely avoided even though the optimal seamlines are detected, especially for  
47 street-view panoramic images among which there always exist geometric misalignments at differ-  
48 ent extents due to that those images were captured from the scenes with large depth differences  
49 by the panoramic camera comprised of multiple wide-angle or fish-eye cameras without a pre-  
50 cisely common projection center, which means that the geometric misalignments are different  
51 at different positions. Therefore, the large geometric misalignments existed in the input aligned  
52 panoramic images should be eliminated as much as possible before finding the optimal seam-  
53 lines. In this paper, we creatively propose an image warping algorithm based on the optical flow  
54 field to reduce the geometric misalignments between input panoramic images. Image warping  
55 is a transformation which maps all positions in one image plane to the corresponding ones in  
56 another plane (Glasbey and Mardia, 1998), which has been popularly applied in many fields  
57 of computer vision, such as image morphing (Lee et al., 1996; Liao et al., 2014), image retar-  
58 geting (Liu and Gleicher, 2005; Lin et al., 2013) and image mosaicking (Zaragoza et al., 2013;  
59 Lin et al., 2015). The key technique of image warping is to find the appropriate transforma-  
60 tion functions based on the control conditions and then eliminate the distortions between input  
61 images. One famous image warping algorithm worked based on thin-plate splines (Bookstein,  
62 1989) that attempted to minimize the amount of bending in the deformation. They used the  
63 radial basis functions with thin-plate splines to find a space deformation defined by control  
64 points. However, the local non-uniform scaling and shearing possibly occurred in the deformed  
65 images. Alexa et al. (2000) firstly introduced the concept of as-rigid-as-possible transfor-  
66 mations, which have the property that both local scaling and shearing are very slight. To produce

67 as-rigid-as-possible deformations, Igarashi et al. (2005) proposed a point-based image deformation  
68 technique, which firstly triangulated the input image, and then geometrically minimized the distortion  
69 associated with each triangle. However, this algorithm needs to triangulate the input image at first,  
70 and the results are maybe not smooth across triangle boundaries. Schaefer et al. (2006) provided an image deformation method based on Moving Least Squares (Levin, 1998)  
71 using various classes of linear functions including affine, similarity and rigid transformations. It  
72 first found the deformation functions based on the control points or the line segments, and then  
73 applied the deformation functions onto each grid instead of each pixel to reduce the transformation  
74 time. At last, it filled the resulting quads using the bilinear interpolation. Chen and Geman  
75 (2014) proposed an image warping algorithm based on radial basis functions, which formulated  
76 the image warping problem as the scattered data interpolation problem, and used the radial  
77 basis functions to construct the interpolation. It aimed to identify the best radial basis  
78 functions for image warping. Our used image warping method is similar to this algorithm, but we  
79 used the Multilevel B-Splines Approximation (MBA) (Lee et al., 1997) to solve the scattered  
80 data interpolation problem. Recently, the b-spline approximation technique has been widely  
81 used for image registration (Yao, 2006; Yao and Cham, 2006), image morphing, image warping,  
82 curve/surface fitting and geometric modeling.

84 In addition, due to the differences of both the image capturing viewpoints and the camera  
85 exposure settings, there are large differences of color and brightness between the warped  
86 panoramic images. The large color differences between those images also can cause the stitching  
87 artifacts in the last stitched or mosaicked panorama. Also, the large color differences maybe  
88 affect the quality of the seamlines. So, we also need to suppress the color differences between  
89 warped images before we apply the optimal seamlne detection. Generally, the color correction  
90 approaches can be divided into two broad categories according to (Xu and Mulligan, 2010):  
91 parametric and non-parametric. Panoramic approaches assume that the color relationship be-  
92 tween images can be described by a certain model. Few noteworthy parametric approaches are  
93 described here. Tian et al. (2002) proposed a simple linear model to transform the color of  
94 the source image to the target image. The transformation matrix was estimated by using the  
95 histogram mapping over the overlap regions. Brown and Lowe (2007) applied the gain com-  
96 pensation (i.e., the diagonal model) to reduce color differences between input images. They  
97 computed all gains by minimizing an error function, which is the sum of gain normalised inten-  
98 sity errors for all overlapping pixels. Xiong and Pulli (2010) also employed the diagonal model  
99 for the color and luminance compensation where the correction coefficients were computed as

the ratio of sum of pixel values in the overlap regions. As stated in (Hwang et al., 2014), 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 don't follow any particular model for the color mapping, and most of them use some form of a look-up table to record the mapping of the full range of color levels. As stated in (Xu and Mulligan, 2010), 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. Yamamoto and Oi (2008) proposed to use the joint histogram of correspondences matched using the SIFT features (Lowe, 2004) to correct the color differences. The color mapping function was estimated by using an energy minimization scheme. Fecker et al. (2008) 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. Hwang et al. (2014) presented a nonlinear and nonparametric color transfer framework that operates in a 3D color space. Based on some control corresponding colors in a given image pair, this method used the probabilistic moving least squares to interpolate the transformation functions for each color. We correct the color differences between two images based on the matched extreme points which are extracted from the histograms over the overlap regions. Both the Probability Density Functions (PDFs) and Cumulative Distribution Functions (CDFs) are used to find the reliably matched extreme points. To reduce the gain artifacts in non-overlap regions, we propose to apply the alpha correction method to smooth the transition from non-overlap regions to overlap ones.

Although we propose efficient approaches to correct the color differences and detect the optimal seamlines between warped panoramic images, there maybe also exist some color transitions along the seamlines due to that the color differences cannot be eliminated completely. In order to further conceal these artifacts, the image blending techniques can be further applied along the seamlines. In the last several decades, many image blending algorithms have been proposed to smooth the color differences along the seamlines, such as feathering (Uyttendaele et al., 2001), alpha blending (Porter and Duff, 1984), Laplacian pyramid blending (Burt and Adelson, 1983), poisson blending (Pérez et al., 2003) and gradient domain image blending approach (Levin et al., 2004). In this paper, we simply applied the Laplacian pyramid blending algorithm (Burt and Adelson, 1983) to eliminate the stitching artifacts and generate the last pleasant panorama.



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

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 seamlne detection approach based on the graph cuts energy minimization framework to find the optimal seamlines between two overlapped images followed by applying the image blending to eliminate the color transitions along the seamlines. By our proposed unified panorama stitching framework, our system can generate a pleasant street-view panorama as seamless as possible by stitching multiple panoramic images from the cameras mounted on the mobile platform. Experimental results on challenging

149 street-view panoramic images are reported in Section 5 followed by the conclusions drawn in  
150 Section 6.

151 **2. Image Warping**

152 In our developed street-view panorama stitching system, we first check whether all input im-  
153 ages are geometrically aligned into a common spherical coordinate system. If not, we will align  
154 them by using the open-source library *PanoTools*<sup>1</sup>, which is also served as the underlying core  
155 engine for many image stitching softwares, such as *PTGui*<sup>2</sup> and *Hugin*<sup>3</sup>. However, there always  
156 exist large geometric misalignments between these aligned images at different extents because  
157 those images were captured from the scenes of large depth differences by a single panoramic  
158 camera comprised of multiple wide-angle or fish-eye cameras without a precisely common pro-  
159 jection center. Those geometric misalignments are so large that the stitching artifacts cannot be  
160 avoided completely even though the optimal seamlines are detected out for the use of the image  
161 stitching. To ensure the high-quality of the last stitched panorama, we propose to apply the  
162 image warping technique to eliminate those large geometric misalignments as far as possible.  
163 To describe our proposed image warping algorithm more clearly, we first consider a simple case  
164 of two aligned images  $\mathbf{I}$  and  $\mathbf{I}'$  with an overlap. The process of our proposed image warping al-  
165 gorithm is described as follows. Firstly, the corresponding points between two images are found  
166 as the control points of image warping, and the sparse optical flows are calculated for those  
167 control points. Secondly, the Multilevel B-Splines Approximation (MBA) algorithm (Lee et al.,  
168 1997) is used to approximately interpolate the dense optical flows for all integral pixels in the  
169 warped image with respective to the original one from the sparse optical flows. Lastly, we warp  
170 the input two images based on the dense optical flows and thus the geometric misalignments  
171 can be greatly lessened. For the case of multiple images to be stitched to the last panorama, a  
172 simple strategy is proposed to first handle the horizontal images and then deal with the vertical  
173 ones.

174 *2.1. Feature Point Matching*

175 To warp two images with large geometric misalignments, we need to find the control points  
176 at first. The quality of the warped image mainly depends on the accuracy and densities of

---

<sup>1</sup> Available at <http://www.panoramatoools.com/>

<sup>2</sup> Available at <http://www.ptgui.com/>

<sup>3</sup> Available at <http://hugin.sourceforge.net/>

control points. In this paper, we apply the feature matching algorithm to robustly find the sparse matching points, namely the control points. The main ideal for feature matching is to first extract local invariant features independently from two images and then characterize them by invariant descriptors. The distance between two descriptor vectors is used to identify candidate matches. However, the nearest neighbors is not always the best match due to occlusion and deformation derived from large viewpoint changes and repeated structures in the scenes. Generally, the epipolar geometrical constraint works well to filter the outliers, but it is not available for the panoramic images aligned in advance. Thus, we need to apply another strategy to filter the outliers. Yao and Cham (2007) proposed that the motion of one match would be consistent with those of neighbors, and the experimental results presented in (Yao and Cham, 2007) sufficiently show that this strategy is simple but effective. Inspired by this idea, we propose a new feature matching algorithm for panoramic images. The major steps of the proposed algorithm include initial matching and outlier detection, which are summarized in Algorithm 1. An example of finding point correspondences between two panoramic images with an overlap is illustrated in Figure 2.

### 2.1.1. Initial Matching

Given two adjacent images  $\mathbf{I}$  and  $\mathbf{I}'$  with an overlap, the local invariant features are extracted and described by the SURF algorithm (Bay et al., 2006). Let  $\mathbf{f} = (\mathbf{x}, \mathbf{d})$  be a feature point where  $\mathbf{x} = (x, y)^\top$  denotes the 2D coordinate of this feature point and  $\mathbf{d}$  represents its corresponding invariant descriptor vector, and  $\mathcal{F} = \{\mathbf{f}_i | \mathbf{f}_i = (\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^M$  and  $\mathcal{F}' = \{\mathbf{f}'_j | \mathbf{f}'_j = (\mathbf{x}'_j, \mathbf{d}'_j)\}_{j=1}^N$  be the feature point sets extracted from  $\mathbf{I}$  and  $\mathbf{I}'$ , respectively, where  $M$  and  $N$  denote the numbers of the feature points extracted from  $\mathbf{I}$  and  $\mathbf{I}'$ , respectively. Generally, for one feature point  $\mathbf{f}_p$  in  $\mathcal{F}$ , the feature point  $\mathbf{f}'_q$  with the nearest Euclidean distance  $d(\mathbf{f}_p, \mathbf{f}'_q) = \min_{\mathbf{f}'_j \in \mathcal{F}'} \|\mathbf{d}_p - \mathbf{d}'_j\|$  which is not larger than a predefined threshold  $T_d$  can be regarded as the corresponding matching point of  $\mathbf{f}_p$ . However, this simple strategy has some drawbacks in the context of feature matching. This mainly because that the distance values between different corresponding pairs may vary in a relatively large range, so any permissive distance threshold  $T_d$  can not avoid the appearance of high rate outliers when covers most of the good correspondences. Thus, we propose to modify the matching strategy as follows. In this paper, we accept two feature points  $\mathbf{f}_p$  and  $\mathbf{f}'_q$  as a potential match only when they satisfy the following conditions:

- The feature points  $\mathbf{f}_p \in \mathcal{F}$  and  $\mathbf{f}'_q \in \mathcal{F}'$  are the nearest neighbors of each other. Namely, for the feature point  $\mathbf{f}_p$ ,  $\mathbf{f}'_q$  is its nearest neighbor in  $\mathcal{F}'$ . At the same time, for the feature

---

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

---

### 1. Initial Matching

- (a) Extract and describe two sets of local invariant features from two overlapped images  $\mathbf{I}$  and  $\mathbf{I}'$  by using the SURF algorithm, respectively;
- (b) Find the initial point matches  $\mathcal{M}_{\text{initial}}$  between  $\mathbf{I}$  and  $\mathbf{I}'$  according to the conditions listed in Section 2.1.1.

### 2. Outlier Detection

- (a) Set the value of  $\lambda$  used in Eq. (3) as  $\lambda_{\max}$ ;
  - (b) Find the neighboring inlier matches  $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$  for each match  $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle \in \mathcal{M}_{\text{initial}}$ , and then calculate the mean motion  $\boldsymbol{\mu}(\mathbf{f}_p)$  and the standard deviation  $\sigma(\mathbf{f}_p)$  of all matches in  $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$ ;
  - (c) Sort all matches in  $\mathcal{M}_{\text{initial}}$  in the decreasing order according to their costs defined in Eq. (4), and only check whether the top  $N_t$  matches are inliers or outliers in each iteration;
  - (d) Iterate the steps (b)-(c) until the maximum number of iterations is reached or no more outliers can be found in the current iteration.;
  - (e) Decrease the value of  $\lambda$  with the step  $\lambda_{\text{step}}$  and iterate the steps (b)-(d) until the smallest value  $\lambda_{\min}$  is reached.
- 

209 point  $\mathbf{f}'_q$ ,  $\mathbf{f}_p$  is its nearest neighbor in  $\mathcal{F}$ .

- 210 • The Euclidean descriptor vector distance  $d(\mathbf{f}_p, \mathbf{f}'_q)$  between two feature points  $\mathbf{f}_p$  and  $\mathbf{f}'_q$  is  
211 not larger than  $T_d$ , i.e.,  $d(\mathbf{f}_p, \mathbf{f}'_q) = \|\mathbf{d}_p - \mathbf{d}'_q\| \leq T_d$ .
- 212 • 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  
213 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  $T_r$ .  
214 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  
215 be smaller than  $T_r$  too.  
216
- 217 By this matching strategy, we obtain a set of initial matches denoted as  $\mathcal{M}_{\text{initial}} = \{\langle \mathbf{f}_p, \mathbf{f}'_q \rangle | \mathbf{f}_p \in \mathcal{F}, \mathbf{f}'_q \in \mathcal{F}'\}$ .

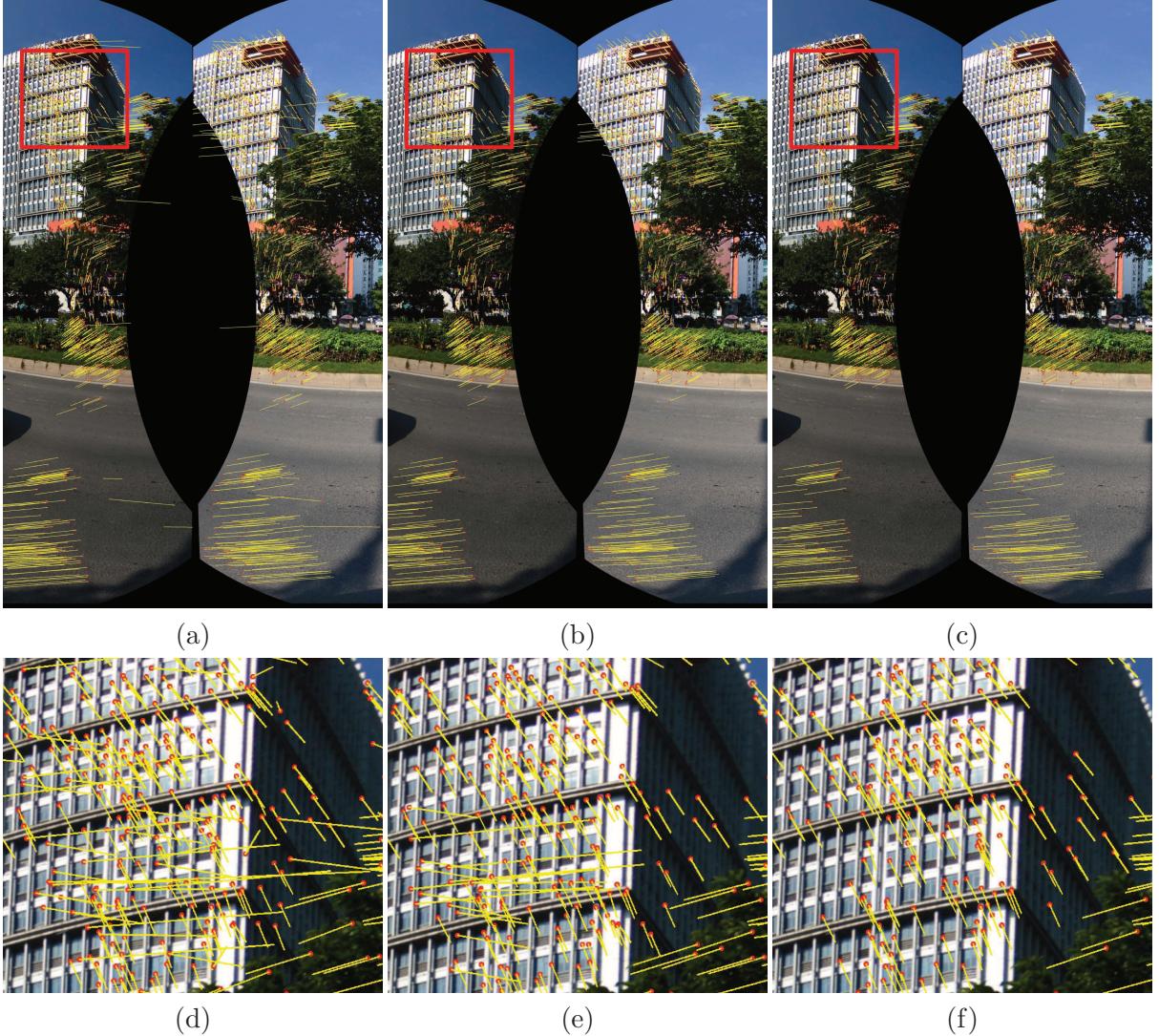


Figure 2: An illustrative example for feature point matching between two aligned panoramic images: (a) the point correspondences produced by initial matching; (b)-(c) the filtered point correspondences after the first iteration ( $\lambda = 6$ ) and the last one ( $\lambda = 3$ ) of outlier detection,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

219    2.1.2. *Outlier Detection*

220    After initial matching, there may still exist a few outliers in  $\mathcal{M}_{\text{initial}}$ . Of course, we need to  
 221    filter out those outliers. The widely used constraint of the epipolar geometric constraint cannot  
 222    be efficiently used in panoramic images, especially when the panoramic images have been aligned  
 223    into a common spherical coordinate system in advance. According to the assumption proposed  
 224    by Yao and Cham (2007) that the matches in a small neighborhood tend to have the consistent

225 location changes (i.e., motions). In this paper, we also apply this assumption to identify the  
 226 outliers.

227 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  
 228 vertical one are calculated, respectively, as follows:

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

229 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  
 230  $(m_p^{(x)}, m_p^{(y)})^\top$  can be calculated as:

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

231 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  
 232 match  $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle$ .

233 At first, we assign the labels of all matches as *Inlier*, namely, for each match  $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle \in$   
 234  $\mathcal{M}_{\text{initial}}$ , the label  $\mathcal{L}(\langle \mathbf{f}_p, \mathbf{f}'_q \rangle) = \text{Inlier}$ , and then we iteratively find the outliers. For each match  
 235  $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle \in \mathcal{M}_{\text{initial}}$ , we find  $K_n$  ( $K_n = 60$  was used in this paper) neighboring match points  
 236 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*  
 237 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  
 238  $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$ , is less than  $K_i$  ( $K_i = 10$  was used in this paper), we directly label this match as  
 239 an *Outlier*, namely,  $\mathcal{L}(\langle \mathbf{f}_p, \mathbf{f}'_q \rangle) = \text{Outlier}$ , otherwise, we determine whether this match is an  
 240 inlier by checking whether it has the consistent motion with its neighbors  $\mathcal{N}_{\text{inlier}}(\mathbf{f}_p)$ . For each  
 241 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  
 242 both Eq. (1) and Eq. (2). Then, the mean motion  $\boldsymbol{\mu}(\mathbf{f}_p) = (\mu_p^{(x)}, \mu_p^{(y)}, \mu_p^{(x,y)})^\top$  and the standard  
 243 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  
 244 determined easily. According to the following measurement proposed by [Yao and Cham \(2007\)](#),  
 245 the label of the match  $\langle \mathbf{f}_p, \mathbf{f}'_q \rangle$  can be determined as follows:

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

246 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  
 247 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 match-  
 248 es, and  $\lambda$  is a predefined parameter. However, this measurement has two following drawbacks.  
 249 To overcome these two drawbacks, we propose the corresponding strategies.

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

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

259 where  $\mu_{\min}^{(x)} = \min_p \mu_p^{(x)}$ ,  $\sigma_{\min}^{(x)} = \min_p \sigma_p^{(x)}$ ,  $\mu_{\max}^{(x)} = \max_p \mu_p^{(x)}$  and  $\sigma_{\max}^{(x)} = \max_p \sigma_p^{(x)}$  denote the  
 260 minimum and maximal mean and standard deviations in the  $x$  component of all match points in  
 261  $\mathcal{M}_{\text{initial}}$ , respectively, and others have the same meanings. Apparently, the black point (outlier)  
 262 shown in Figure 3 has a bigger cost, and the red points (inliers) have smaller costs. Then, all  
 263 match points are sorted in the decreasing order according to their costs defined in Eq. (4). In  
 264 each iteration, we only check whether the top  $N_t$  matches are inliers or outliers according to the  
 265 measurement defined in Eq. (3), where  $N_t = \rho \times N_{\text{inlier}}(\mathcal{M}_{\text{initial}})$ ,  $\rho$  is the predefined proportion  
 266 parameter ( $\rho = 0.025$  was used in this paper) and  $N_{\text{inlier}}(\mathcal{M}_{\text{initial}})$  denotes the number of matches  
 267 with the label *Inlier* in  $\mathcal{M}_{\text{initial}}$ . We end up the iterations until the maximum number of  
 268 iterations is reached or no more matches can be labeled as *Outlier*. By this way, the outliers with  
 269 larger deviations with respective to neighboring points will be robustly filtered out step-by-step.  
 270

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

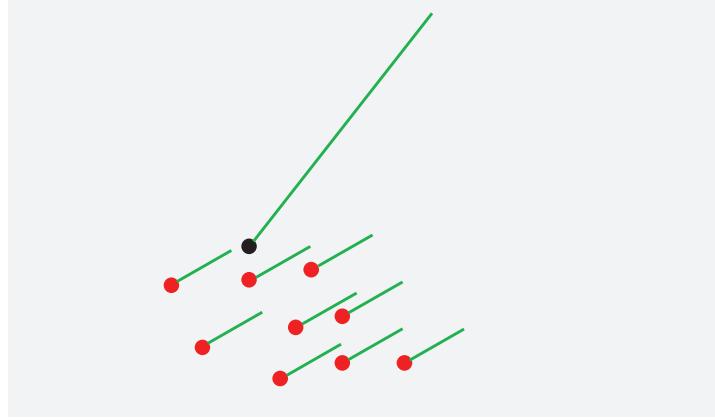


Figure 3: An visual example of outlier detection. The black point means an outlier, the red points mean inliers, and the green lines represent the motions of corresponding match points.

## 2.2. Approximate Interpolation of Dense Optical Flows

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

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

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

Secondly, we propose to approximately interpolate the optical flows of all integral pixels in the warped images  $\bar{\mathbf{I}}$  and  $\bar{\mathbf{I}}'$  based on the disparity vectors  $\{\mathbf{d}(\mathbf{x}_m)\}_{\mathbf{x}_m \in \mathcal{M}_{\text{inlier}}}$  and  $\{\mathbf{d}(\mathbf{x}'_n)\}_{\mathbf{x}'_n \in \mathcal{M}_{\text{inlier}}}$  of the control points  $\{\mathbf{x}_m - \mathbf{d}(\mathbf{x}_m)\}_{\mathbf{x}_m \in \mathcal{M}_{\text{inlier}}}$  and  $\{\mathbf{x}'_n - \mathbf{d}(\mathbf{x}'_n)\}_{\mathbf{x}'_n \in \mathcal{M}_{\text{inlier}}}$ , respectively. This problem can be formulated as the scattered data interpolation problem. Due to the sparsity of the control points, in this paper we adapt to apply the Multilevel B-Splines Approximation (MBA) (Lee et al., 1997) 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

296 interpolation, we separately interpolate the horizontal and vertical components of optical flows  
297 (i.e., disparity vectors) of all the integral pixels in  $\bar{\mathbf{I}}$  and  $\bar{\mathbf{I}}'$ , respectively. In this way, we finally  
298 obtain the dense optical flows  $\mathcal{D}(\bar{\mathbf{I}}) = \{\tilde{\mathbf{d}}(\mathbf{p})\}_{\mathbf{p} \in \bar{\mathbf{I}}}$  and  $\mathcal{D}(\bar{\mathbf{I}}') = \{\tilde{\mathbf{d}}(\mathbf{p}')\}_{\mathbf{p}' \in \bar{\mathbf{I}}'}$  of all the integral  
299 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 respective to the original images  
300  $\mathbf{I}$  and  $\mathbf{I}'$ , respectively.

301 *2.3. Two Image Warping*

302 Here, we demonstrate how to generate the warped image  $\bar{\mathbf{I}}$  from the original image  $\mathbf{I}$  based on  
303 the dense optical flows  $\mathcal{D}(\bar{\mathbf{I}})$  of  $\bar{\mathbf{I}}$  with respective to  $\mathbf{I}$ , and the generation of the warped image  $\bar{\mathbf{I}}'$   
304 is similar. For each pixel  $\mathbf{p} \in \bar{\mathbf{I}}$ , we can easily calculate its corresponding 2D position in  $\mathbf{I}$  based  
305 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  
306 use the bilinear interpolation algorithm to interpolate the intensity of the corresponding point  
307  $\mathbf{p} + \tilde{\mathbf{d}}(\mathbf{p})$  in  $\mathbf{I}$  as the intensity of the integral pixel  $\mathbf{p} \in \bar{\mathbf{I}}$ .

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

311 *2.4. Multiple Image Warping*

312 Until now, we have introduced how to warp two images based on the optical flows. But,  
313 we need to warp multiple input images to generate the last panorama. In the experimental  
314 results presented in this paper, the input images are comprised of 5 horizontal ones and 1  
315 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  
316 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  
317 Figure 9. For this particular case, here we will detailedly introduce how to warp these six images  
318 for producing the last panorama before color correction. Other cases of multiple images can be  
319 handled in a similar way. For multiple input panoramic images, we first collect all image pairs  
320 according to their overlap relationship as shown in Figure 9. Obviously, there are five image  
321 pairs along the horizontal direction, and one image pair along the vertical direction. We first  
322 handle the horizontal image pairs and then deal with the vertical image pair. For each horizontal  
323 image pair, we match them one by one by the method presented in Section 2.1, as an illustrative  
324 example shown in Figure 4, from which we can find that one horizontal image is overlapped  
325 with two adjacent images in the horizontal direction. For example, for the image  $\mathbf{I}_1$ , it overlaps  
326 with  $\mathbf{I}_2$  and  $\mathbf{I}_5$ , respectively, so we need to collect all matching points from these two overlap  
327 regions as the control points for warping  $\mathbf{I}_1$ . The dense optical flow field of the warped image

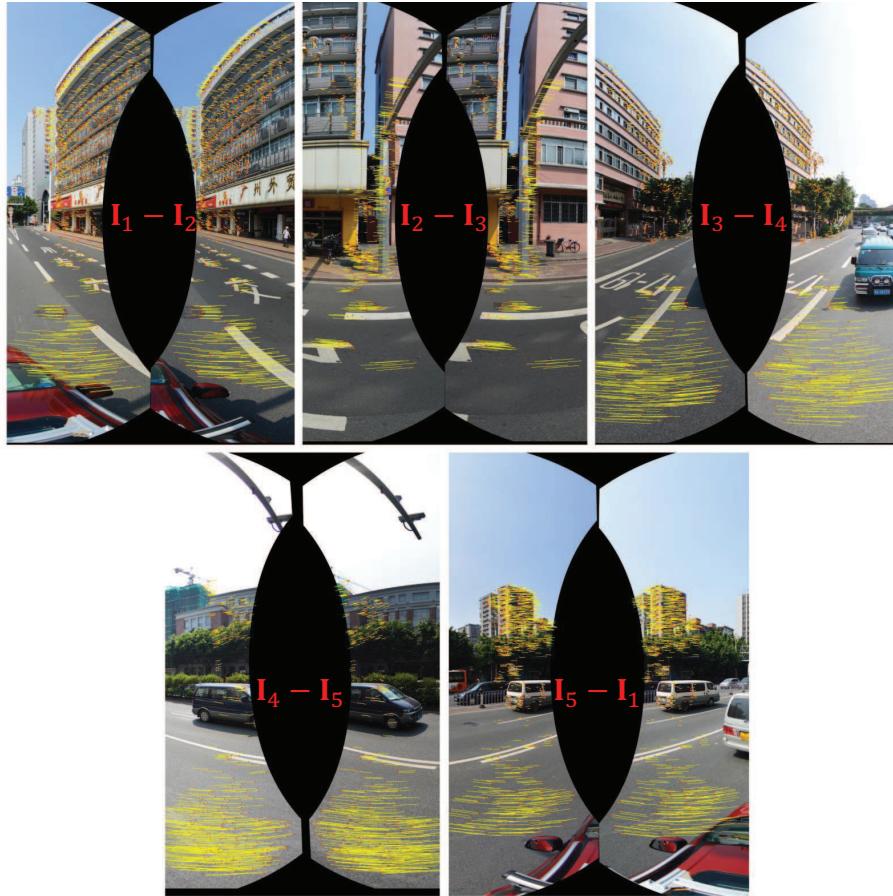


Figure 4: The feature matching results of all five horizontal image pairs in the overlap regions.

328  $\bar{I}_1$  with respect to the original image  $I_1$  can be approximately interpolated based on those  
329 control points via the MBA algorithm. Therefore, five horizontal warped images  $\bar{I}_1$ ,  $\bar{I}_2$ ,  $\bar{I}_3$ ,  $\bar{I}_4$   
330 and  $\bar{I}_5$  can be generated by warping their corresponding original images according to the method  
331 presented in Section 2.3, respectively. Figure 5 shows an example for warping one horizontal  
332 image. After that, we generate the bottom blended image  $I_H$  by blending all horizontal warped  
333 images according to the proposed color correction method presented in Section 3 and the adopted  
334 image mosaicking strategy described in Section 4. Finally, to produce the last panorama, the  
335 top image  $I_6$  and the horizontal blended image  $I_H$  will be warped according to those matching  
336 points as the control ones.

### 337 3. Color Correction

338 The large geometric misalignments can be efficiently eliminated by our proposed image  
339 warping algorithm, but there also exist the color differences between the warped images, so the

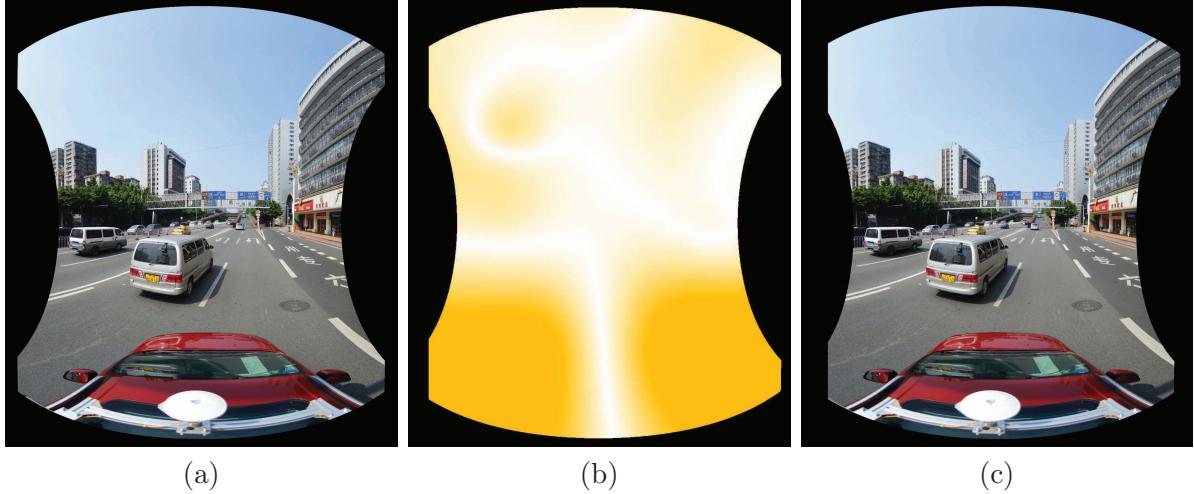


Figure 5: An illustrative example of image warping: (a) the original aligned image; (b) the dense optical flows approximately interpolated by the MBA algorithm; (c) the last warped image. In (b), the deeper orange means the larger disparity.

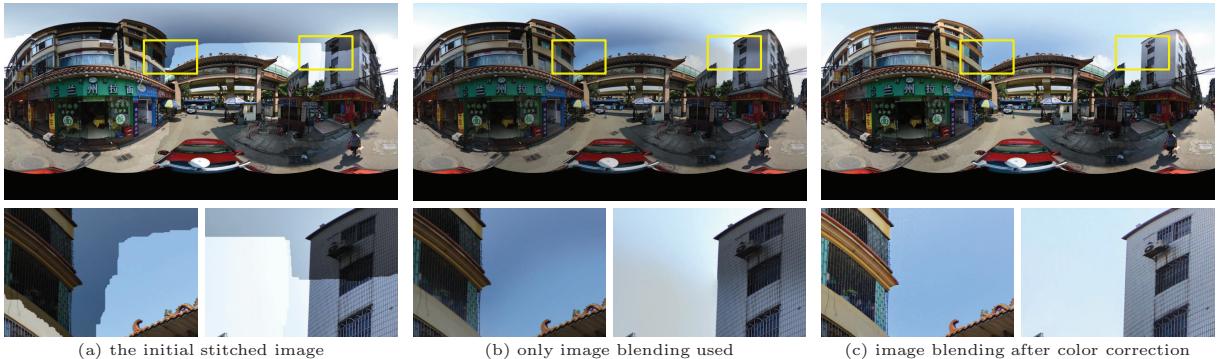


Figure 6: An example of our proposed color correction strategy used to improve the panorama stitching quality before applying the image blending.

340 stitching artifacts are still visible. Generally, the image blending technique can solve it easily  
 341 by smoothing the color along the seamlines. However, it does not work well for input images  
 342 with very large color differences. The simple image blending maybe can not efficiently conceal  
 343 the artifacts if we don't magnificently correct color differences between images in advance,  
 344 which results in low-quality panoramic images, as an illustrative example shown in Figure 6.  
 345 In addition, the large color differences maybe also affect the quality of the detected seamlines.  
 346 Thus, in this paper, we propose to reduce the color differences between warped images before the  
 347 optimal seamlines are detected. Generally, the color differences should be also corrected before  
 348 the image warping step to ensure the quality of feature matching results. But our adopted  
 349 SURF feature matching algorithm is robust enough to the large photometric inconsistencies,  
 350 so there are no obvious influence on our algorithm if we apply the color correction after image

351 warping.

352 In this paper, we first apply the automatic contrast adjustment to reduce the brightness  
353 differences between images and then propose a novel and efficient color correction algorithm  
354 via matching extreme points of intensity histograms to further reduce the color differences.  
355 For the overlap image regions between two images, we construct their own Probability Density  
356 Functions (PDFs) and Cumulative Distribution Functions (CDFs) with respect to the intensity  
357 histograms in the three HSV channels, respectively. One way to eliminate color differences is to  
358 ensure that the three CDFs of the overlap regions in the first image in the three HSV channels  
359 are approximately same to those CDFs of the overlap regions in the second image, respectively.  
360 Obviously, we can correct the CDFs based on several uniformly spaced knots as [HaCohen et al.](#)  
361 ([2013](#)) did. However, due to the existence of geometric misalignments, the scenes presented  
362 by two images in the overlap regions are not completely consistent. To solve this problem, we  
363 replace the knots by the matched extreme points extracted from the two PDFs. If the number  
364 of matched extreme points is not sufficient, we will suitably introduce those uniformly space  
365 knots. At last, the intensities of all the pixels in the two images are modified afterwards based  
366 on the matched extreme points extracted from the PDFs, not only for the pixels in the overlap  
367 regions, but also in the non-overlap regions.

368 *3.1. Automatic Contrast Adjustment*

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

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

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

382 and  $B_{\max}$ , respectively. The minimal and maximal intensity values of the whole color image  
 383 are defined as  $V_{\min} = \min(R_{\min}, G_{\min}, B_{\min})$  and  $V_{\max} = \max(R_{\max}, G_{\max}, B_{\max})$ , respectively.  
 384 Therefore, any intensity  $I$  of the  $R$ ,  $G$  and  $B$  channels of a color image will be modified as:

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

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

### 387 3.2. Finding Extreme Points

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

396 To robustly find extreme points in both  $\mathbf{PDF_A}$  and  $\mathbf{PDF_B}$ , these two PDFs are smoothed  
 397 first by a Gaussian function to suppress possible noise. The initial local extreme points can be  
 398 easily obtained from the smoothed  $\mathbf{PDF_A}$  and  $\mathbf{PDF_B}$ . In an ideal situation, the extreme points  
 399 should be uniformly distributed in the color space. However, most of the extreme points are  
 400 relatively centralized in some cases, which will lead to the information redundancy due to that  
 401 multiple extreme points are selected out to represent the similar image statistical information.  
 402 To avoid the situation mentioned above, we further checkout all initial extreme points by the  
 403 local window suppression. Let  $\{L_A^i\}_{i=1}^K$  be the intensities of  $K$  extreme points  $\{P_A^i\}_{i=1}^K$  in  
 404  $\mathbf{PDF_A}$ , which are sorted in the ascending order. Given an extreme point  $P_A^i$ , we generate a  
 405 neighborhood range  $[L_A^i - w, L_A^i + w]$  centered on the corresponding intensity  $L_A^i$  with the size  
 406 of  $(2w + 1)$ . We set  $w = 2$  if not specifically stated in this paper. If there exist more than  
 407 one extreme points located in this neighborhood range, the extreme point with the highest  
 408 frequency in  $\mathbf{PDF_A}$  will be retained and other extreme points will be removed. All initial  
 409 extreme points are checked in this way and the retained extreme points are used for the following  
 410 matching. The final extreme points extracted from  $\mathbf{PDF_A}$  and  $\mathbf{PDF_B}$  are represented as

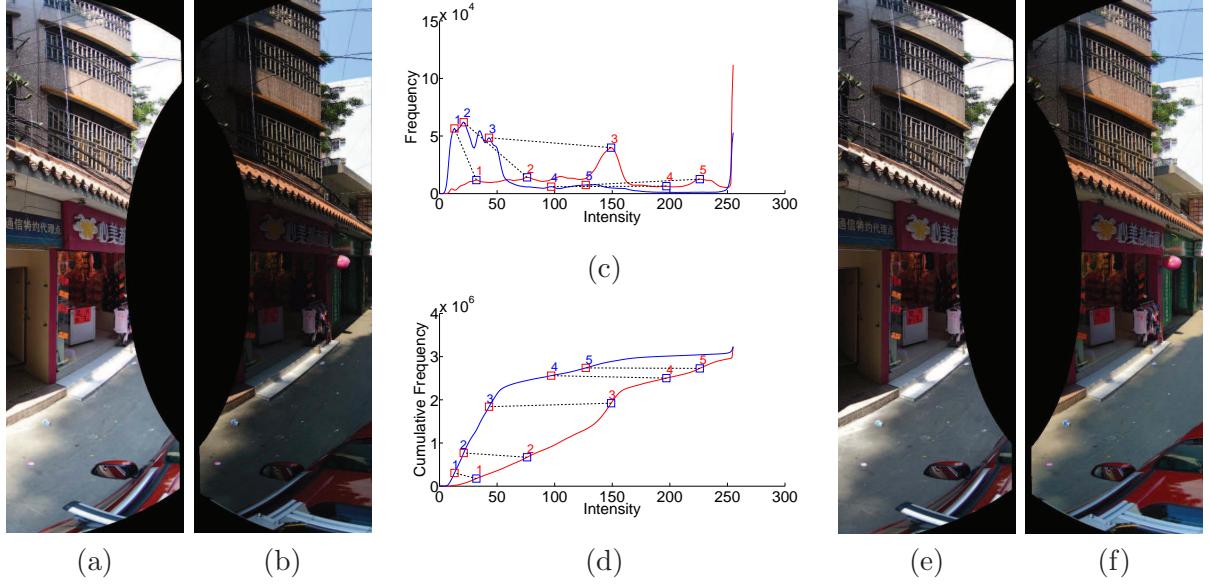


Figure 7: A visual example of our proposed color correction approach: (a)-(b) the overlap image regions of the input left and right images, respectively; (c)-(d) the curves of PDF and CDF in one channel where the red curves stand for the left image and the blue ones stand for the right image, and the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

411  $\{\mathbf{P}_A^i\}_{i=1}^{N_A}$  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  
412  $\mathbf{PDF}_B$ , respectively. For each extreme point  $\mathbf{P}$ , it consists of 4 components according to  $\mathbf{P} =$   
413  $\{F, L, \hat{C}, \check{C}\}$  where  $F$  denotes the frequency of this point in PDF,  $L$  represents the corresponding  
414 intensity, and  $\hat{C}$  and  $\check{C}$  means the cumulative values of the intensities  $(L + \varepsilon)$  and  $(L - \varepsilon)$  in  
415 CDF (we set  $\varepsilon = 2$  if not specifically stated in this paper).

### 416 3.3. Matching Extreme Points

417 The extreme points can sufficiently reflect image statistical characteristics. To efficiently  
418 adjust the color differences, one way is to ensure that the intensities of corresponding extreme  
419 points are the same. Thus, we should match the extreme points firstly. To reliably match  
420 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  
421 similarity of two extreme points  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  as:

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

422 where  $F_{\max}$  is the maximal frequency of all the extreme points in both  $\mathbf{PDF}_A$  and  $\mathbf{PDF}_B$ .  
423 The above cost function judges the two extreme points from the view of both PDF and CDF.  
424 The first term  $\frac{F_A^i + F_B^j}{2F_{\max}}$  indicates that those possibly matched extreme points with the higher  
425 frequencies generates higher costs, which may be peaked out first in the following matching

426 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  
 427 for two possibly matched extreme points  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$ . The last term is applied to ensure that  
 428 the accumulative values of two possibly matched extreme points  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  are approximate.  
 429 From this term, we can find that if the small range of cumulative values of two extreme points  
 430 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  $\max(\hat{C}_A^i, \hat{C}_B^j) -$   
 431  $\min(\check{C}_A^i, \check{C}_B^j)$ , which results in that this term is close to 1. In contrast, if the numerator is smaller  
 432 and the denominator is larger, this term will approach to 0. In summary, if the frequencies of  
 433 two extreme points are larger and more similar, and the accumulative values of those points  
 434 are more approximate, their matching cost is bigger. In contrast, it is smaller. The higher the  
 435 cost function value is, the more likely these two extreme points are matched. Based on this cost  
 436 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  
 437 eliminate the impossibly matched extreme points, we empirically designed three hard conditions  
 438 from the view of both PDF and CDF to check whether two extreme points  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  are  
 439 possibly matched as follows:

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

440 where  $\theta_f$  and  $\theta_c$  are two empirical thresholds ( $\theta_f = 0.25$  and  $\theta_c = 0.02$  were used in this paper),  
 441  $C_{\max}$  is the maximal value of CDF, namely, the valid pixel number of overlap regions. The  
 442 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  
 443 above three conditions is not met, namely,  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  are not possibly matched. From the view  
 444 of PDF, the first condition indicates that the frequencies of the two possibly matched extreme  
 445 points  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  should be a relatively small difference. From the view of CDF, the second  
 446 and third conditions indicate that  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  are possibly matched if their corresponding CDF  
 447 values are approximate. According to the above three hard conditions, the matching cost matrix  
 448  $\mathbf{M}$  will be updated, in which all the zero elements indicate that they are not possibly matched.

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

- 451 • Step 1: Finding the highest non-zero cost element  $M_{ij}$  from the matrix  $\mathbf{M}$  and its corre-  
 452 sponding extreme points  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  is selected out as a reliable extreme point match.
- 453 • Step 2: Updating the matrix  $\mathbf{M}$  by removing the  $i$ -th row and the  $j$ -th column due to

454 that  $\mathbf{P}_A^i$  and  $\mathbf{P}_B^j$  have been successfully matched.

- 455 • Step 3: Performing the above two steps iteratively until the updated matrix  $\mathbf{M}$  is empty  
 456 or there exists no non-zero element in  $\mathbf{M}$ .

457 By the above iterative strategy, a set of reliable extreme point matches will be found. In  
 458 Figure 7, we have shown a visual example of our proposed color correction approach. The input  
 459 two images have large color differences in overlap regions, as shown in Figures 7(a)-(b). We find  
 460 5 matched extreme points in PDF of one channel. Based on those correspondences, the large  
 461 color differences can be eliminated, as shown in Figures 7(e)-(f). From this example, we can  
 462 find that our proposed approach can handle the images with large color differences very well.

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

474 *3.4. Correcting Color Difference*

475 The extracted matching points in the overlap image regions are then applied to correct the  
 476 intensities of two adjacent images, including the pixels in non-overlap regions. Let  $\{Q_A^k\}_{k=1}^N$   
 477 and  $\{Q_B^k\}_{k=1}^N$  be the final matching points in CDFs in the overlap regions **A** and **B** with  
 478  $N$  point matches. Based on the matching results, the intensities of the matching points  $Q_A^k$   
 479 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  
 480  $(Q_A^k, Q_B^k)$  in CDFs, respectively. In this way, the intensities of  $\{(Q_A^k, Q_B^k)\}_{k=1}^N$  are corrected to  
 481  $\{(\hat{L}_A^k, \hat{L}_B^k)\}_{k=1}^N$ , respectively, where  $\hat{L}_A^k = \hat{L}_B^k = (L_A^k + L_B^k)/2$ . Based on these corrections, the  
 482 intensity of any pixel in both **A** and **B** will be adjusted linearly. For example, given a pixel  
 483  $\mathbf{p} \in \mathbf{A}$  whose intensity  $L_A(\mathbf{p})$  will be linearly corrected as:

$$\hat{L}_A(\mathbf{p}) = \hat{L}_A^l + (L_A(\mathbf{p}) - L_A^l) \frac{\hat{L}_A^u - \hat{L}_A^l}{L_A^u - L_A^l}, \quad (10)$$

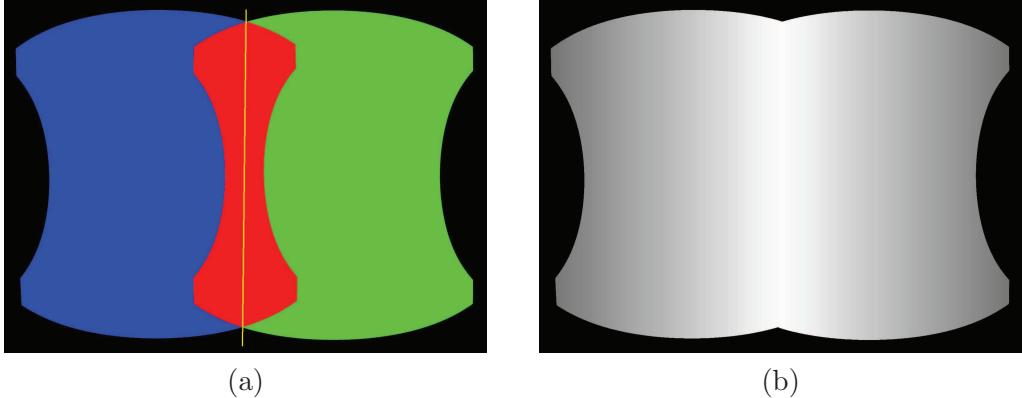


Figure 8: An illustration of the alpha weighting fusion map of two adjacently warped images: (a) two overlapped images represented by the blue and green regions, respectively, with the overlap image region marked in red and the center line marked in yellow; (b) the normalized alpha weighting fusion map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

484 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  $\mathbf{A}$  that  
485 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  
486 to produce a smooth and gradual transition from non-overlap regions to overlap ones, the alpha  
487 correction method is conducted as:

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

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

#### 494 4. Image Mosaicking

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

502    4.1. Optimal Seamline Detection

503    In this paper, the optimal seamlines between color corrected images will be efficiently ex-  
504 tracted using the graph-cuts-based seamline detection algorithm presented in (Li et al., 2016).  
505    This novel algorithm is used to efficiently detect optimal seamlines for mosaicking street-view  
506 panoramic images without precisely common center in a two-label graph cuts energy minimiza-  
507 tion framework. This algorithm magnificently fuses the information of image color, gradient,  
508 and texture complexity into the data and smooth energy terms in graph cuts to effectively  
509 ensure that the seamlines are optimally detected in the laterally continuous regions with high  
510 image similarity and low object dislocation to magnificently conceal the parallax between im-  
511 ages. For multiple images, we apply the traditional *frame-to-frame optimization* strategy to  
512 efficiently find all optimal seamlines. The details of this strategy are described in Section 3.1  
513 of (Li et al., 2016). The experimental results on a large set of images reported in (Li et al.,  
514 2016) have demonstrated that this algorithm is capable of creating high-quality seamlines for  
515 multiple image mosaicking, while not crossing majority of visually obvious foreground objects  
516 and most of overlap regions with low image similarity to effectively conceal the image parallax  
517 at different extents.

518    4.2. Image Blending

519    Although the major color differences are eliminated between input images by applying our  
520 proposed color correction strategy presented in Section 3, there still exist the artifacts along the  
521 seamlines due to that the color differences can not be removed completely via color correction.  
522 Thus, a good image blending algorithm is needed to generate the last pleasant panorama. To  
523 quickly blend the color corrected images after detecting the optimal seamlines, the transition  
524 smoothing methods (also known as feathering (Uyttendaele et al., 2001) or alpha blending meth-  
525 ods (Porter and Duff, 1984)) can be used to minimize the visibility of seamlines by smoothing  
526 the common overlapping regions of the combined images. However, to produce a more pleasant  
527 panorama, in this paper, we use the Laplacian pyramid blending (Burt and Adelson, 1983) to  
528 stitch multiple color corrected images at one time.

529    5. Experimental Results

530    Extensive experiments on representative street-view panoramic images were conducted to  
531 comprehensively evaluate the performance of our proposed unified framework for street-view  
532 panorama stitching. In our paper, all used street-view panoramic images were captured from the

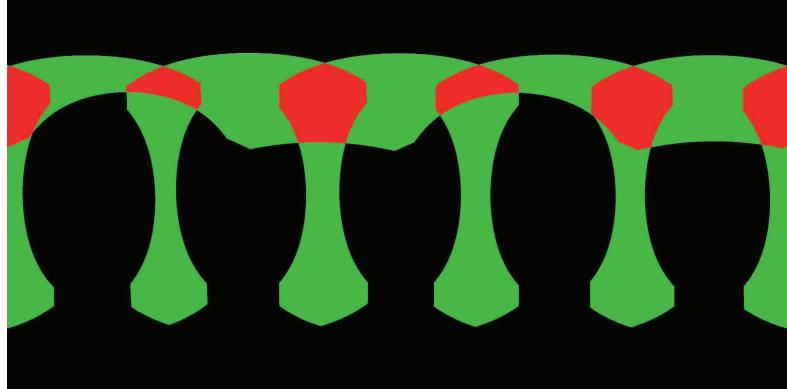


Figure 9: The image overlap regions of six geometrically aligned and warped images in the 360° street-view panoramic view where the black, the green and the red stand for the no-overlapped, two-overlapped, multi-overlapped image regions, respectively.

real world scenes by an integrated multi-camera equipment with six Nikon D7100 cameras of 24 million pixels with wide-angle lenses mounted on a mobile vehicle platform. Six camera images were aligned into a common spherical coordinate system with the image size of  $12000 \times 6000$  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 9. Our algorithms in this paper were implemented with C++ under Windows and tested in a computer with an Intel Core i7-4770 at 3.4GHz and the 16GB RAM memory. Due to the limit of pages, more experimental results and analysis are presented at <http://cvrs.whu.edu.cn/projects/PanoStitching/>.

### 5.1. Image Warping

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

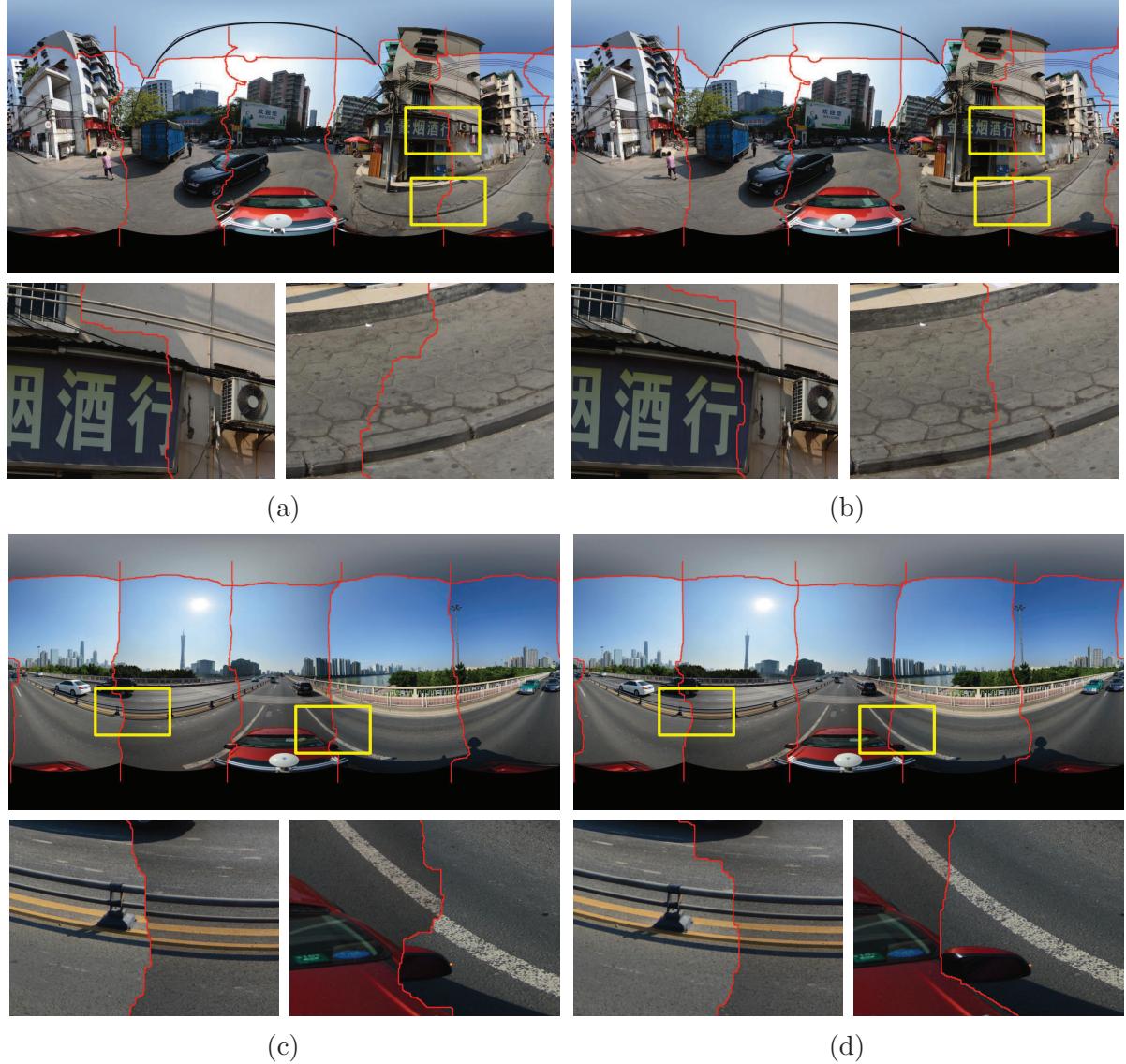


Figure 10: Visual comparison of the stitching results with the optimal seamlines in two groups of six panoramic images when our proposed image warping algorithm was used (Right:(b) and (d)) or not (Left: (a) and (c), namely, the stitching results of ([Li et al., 2016](#))). The red lines stand for the detected optimal seamlines between images.

554 smaller as expected when the image warping algorithm was applied, as shown in Figure 10(b).  
 555 While not using the image warping algorithm, the geometric dislocation is very large, as shown  
 556 in Figure 10(a). For example, in the first enlarged local region, the seamline crossed the text  
 557 without the used of image warping, and it avoided crossing the text when the image warping  
 558 was used. In the second enlarged local region, although two seamlines crossed the road with  
 559 pavement stairs, we can find that the dislocation is almost invisible in the pavement stairs when  
 560 the image warping was used, but it is so obvious without the use of the image warping. In the as-

pect of computational cost, without the use of image warping, our algorithm took around 17.89s in the above experiment, only the elapsed time in six optimal seamlines detection is included. However, with its use, our algorithm took around 70.93s consisting of all the elapsed times in the image warping and the optimal seamline detection. From this comparison, we observed that the seamline detection is efficient, but the image warping is relatively time-consuming. This is mainly because that we need to find the inlier matches for all image pairs at first and then interpolate the dense optical flows by MBA for each image, which is time-consuming. But our proposed image warping algorithm can significantly improve the quality of the last stitched panorama.

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

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

## 5.2. Color Correction and Image Blending

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

Figure 11 shows the experimental results on the first group of panoramic images. The panorama stitching results without and with the use of color correction are shown in Figures 11(a) and (b), respectively. From the whole stitching results and especially the detailed local regions shown in Figures 11(a) and (b), we can find that color differences between the warped images were significantly reduced and are almost invisible. In addition, the quality of the detected optimal seamlines was improved as expected when the color correction algorithm was used due to that the color differences were greatly reduced before the seamlines were found.

---

<sup>4</sup> Available at <http://enblend.sourceforge.net/>.

592 For example, the seamlne rounded the advertising board instead of crossing it when the color  
593 correction algorithm was used, as shown in the detailed image regions in Figures 11(a) and  
594 (b). In the aspect of computational cost, without the use of the color correction, our algorithm  
595 took around 18.01s to find all six optimal seamlines. With its use, our algorithm took around  
596 33.52s to correct the color differences and find the optimal seamlines, means that the color  
597 correction algorithm took around 15.51s. To generate the last panorama, the Laplacian pyra-  
598 mid blending algorithm was further applied, whose generated result is shown in Figure 11(d).  
599 And in Figure 11(c), we also present the last panorama generated by *Enblend*. From the vi-  
600 sual comparison, we can observe that our proposed stitching system with image warping and  
601 color correction obviously outperforms *Enblend*. Noticeably, the stitching artifacts caused by  
602 geometric misalignments and photometric inconsistencies still exist in the panorama generated  
603 by *Enblend*, as shown in Figure 11(c) but they almost disappeared in our produced panorama,  
604 as shown in Figure 11(d). In the aspect of computational cost, the Laplacian pyramid blending  
605 algorithm took 35.56s.

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

### 614 5.3. Image Stitching

615 To illustrate the effectiveness of our proposed framework for street-view panorama stitching,  
616 we presented the last panoramas stitched by different combination of optimal seamlne detection  
617 ( $S$ ), image warping ( $W$ ), color correction ( $C$ ) and image blending ( $B$ ) algorithms in Figure 13.  
618 At first, Figure 13(a) shows the panorama generated by the optimal seamlne detection algorith-  
619 m presented by Li et al. (2016), from which we can find that there are many stitching artifacts  
620 caused by geometric misalignments and photometric inconsistencies in the last stitching image,  
621 especially obvious in the detailed local regions. For example, the white lanes on the road were  
622 broken due to the large geometric dislocations. In addition, there also exist large color differ-  
623 ences along the optimal seamlines. Our proposed image warping and color correction algorithm

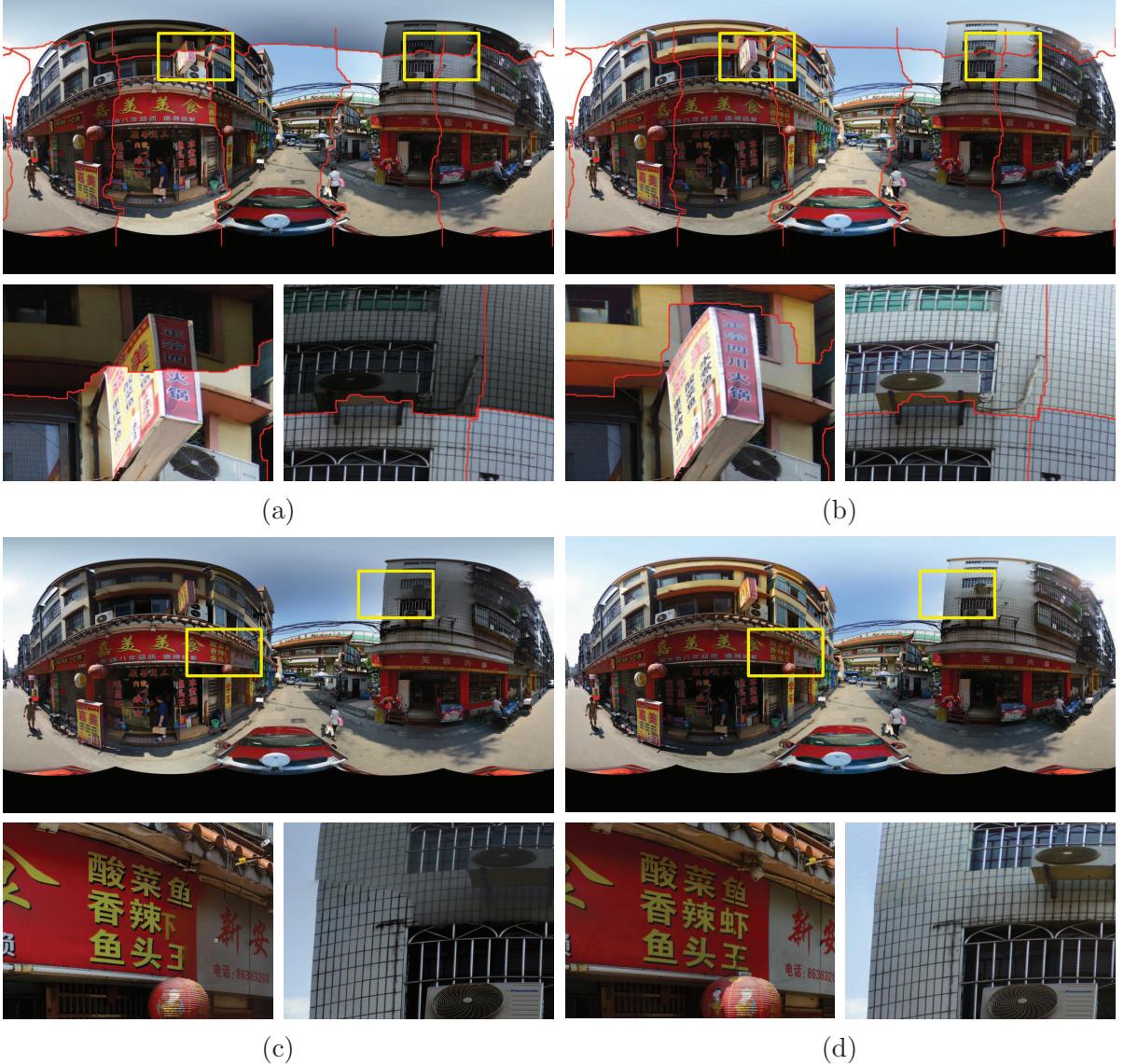


Figure 11: Visual comparison in the first group of six panoramic images: (a)-(b) the stitching results with the optimal seamlines when the our proposed color correction was used (b) or not (a); (c)-(d) the last panoramas generated by *Enblend* (c) and our proposed stitching system (d).

624 can eliminate large geometric misalignments and photometric inconsistencies, as shown in Fig-  
 625 ures 13(b) and (c), respectively. The last blended panorama generated by our proposed system  
 626 is shown in Figure 13(d) from which we can observe that the last stitched panoramic image is  
 627 pleasant and high-quality, which can meet the application requirement of the street-view map.

#### 628 5.4. Comparative Results

629 At last, to prove that our approach is superior and can generate high-quality panoramas, we  
 630 compared our proposed approach with the Xiong and Pulli’s approach (Xiong and Pulli, 2010).

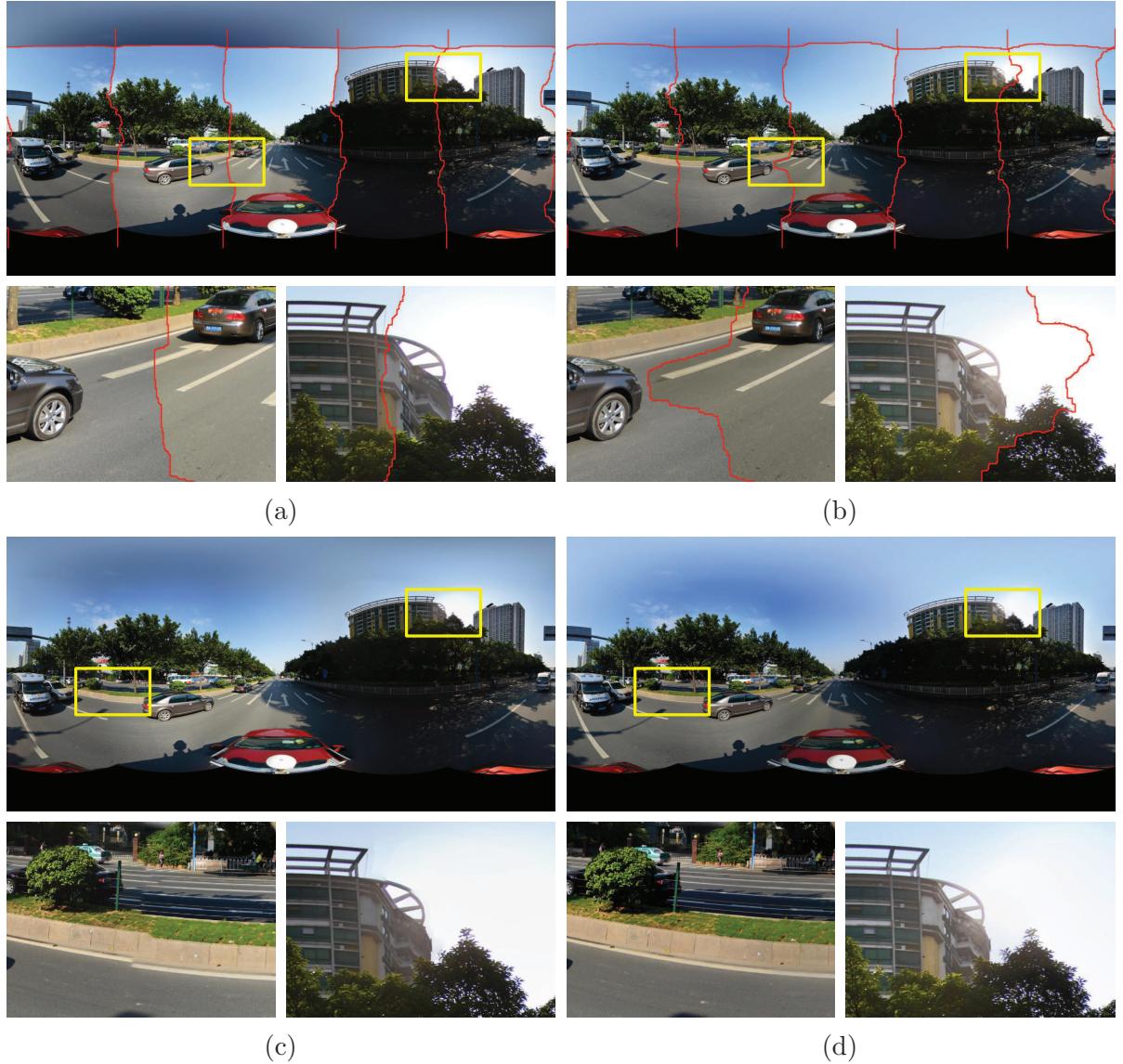


Figure 12: Visual comparison in the second group of six panoramic images: (a)-(b) the stitching results with the optimal seamlines when the our proposed color correction was used (b) or not (a); (c)-(d) the last panoramas generated by *Enblend* (c) and our proposed stitching system (d).

631 We used two representative groups of panoramic images for visual comparison. The color dif-  
 632 ferences in the first group of images are relatively small but large in the second group. Because  
 633 the Xiong and Pulli's approach has not eliminated the influence of large geometric misalign-  
 634 ments between aligned images, so we used the warped images generated by our image warping  
 635 algorithm as the input ones for comparing two approaches. In addition, their approach applied  
 636 the Poisson blending algorithm to generate the last blended image, however, our approach used  
 637 the Laplacian pyramid blending algorithm. To evaluate the last blended panoramas generated

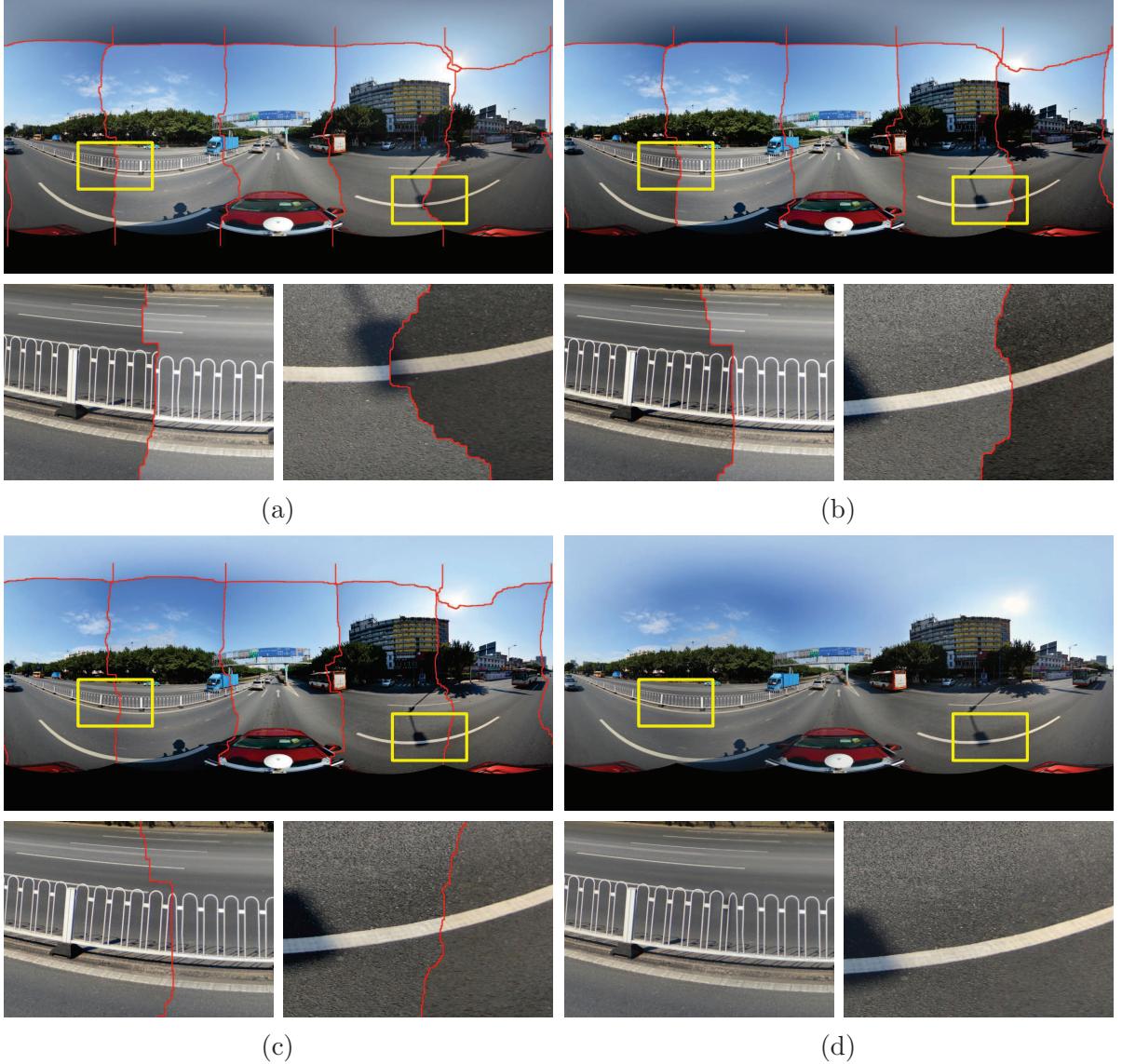


Figure 13: The stitching results with different combination of optimal seamlne detection ( $S$ ), image warping ( $W$ ), color correction ( $C$ ) and image blending ( $B$ ) algorithms: (a)  $S$  (the result generated by [Li et al. \(2016\)](#)); (b)  $W + S$ ; (c)  $W + C + S$ ; (d)  $W + C + S + B$ . The computational times of (a)-(d) are 18.00s, 69.08s, 86.63s and 123.68s, respectively.

638 by two approaches fairer, we replaced the Poisson blending algorithm in the tested Xiong and  
 639 Pulli's approach with the Laplacian pyramid blending algorithm.

640 Figure 14 shows the stitching results of the first group of images with relatively small color  
 641 differences. Figures 14(a) and (b) illustrate the stitching results just with the detected seamlnes  
 642 of the Xiong and Pulli's approach and our approach without the use of color correction, respec-  
 643 tively. From these two figures, we can observe that the seamlnes detected by our approach

are better than those detected by their approach. For example, the seamlines detected by their approach crossed the tall building, but our approach avoided crossing it. Figures 14(c) and (d) illustrate the stitching results of two approaches with the use of color correction, respectively, from which we observed that both of two approaches can eliminate the small color differences effectively. Figures 14(e) and (f) show the last blended panoramas generated by two approaches, respectively, from which we found that there is some petty ghost on the top of the tallest building in the second enlarged region shown in Figure 14(e), which disappeared in the panorama generated by our approach, as shown in Figure 14(f). This is mainly because the horizontal seamline between bottom and top input images detected by the Xiong and Pulli's approach is close to this building, as shown in Figure 14(c). In conclusion, if the color differences between input images are small, both of two approaches can generate high-quality panoramas.

Figure 15 shows the stitching results of the second group of images with very large color differences. Figures 14(a) and (b) show the stitching results of the Xiong and Pulli's approach and our approach without the use of color correction, respectively, from which we observed that our approach also generated more high-quality seamlines than their approach. Figures 14(c) and (d) present the stitching results of two approaches with the use of color correction, respectively. From the visual comparison, we observed that our proposed color correction algorithm obviously outperformed than the algorithm presented in (Xiong and Pulli, 2010), especially obvious in two locally enlarged regions. For example, in the first enlarged region (from left to right), the detected seamline divides the building into two parts, one comes from the top input image which is dark, and another comes from the bottom input image which is relatively lighter. After color correction, the top image is also very dark in the result generated by the Xiong and Pulli's approach and the color differences along the seamline are also very large. In addition, due to that the top image is too dark, many detailed informations cannot be pleasantly observed. But, in our result, the color of the top image is similar with the bottom one, and more detailed informations of this region can be clearly observed. Figures 14(e) and (f) show the last blended panoramas generated by two approaches, respectively. In the second enlarged region of Figure 15(e), we found that there are some very obvious ghosts on the top of the building, which disappeared in the panorama generated by our approach, as shown in Figure 15(f). In addition, in Figure 15(e), the color of top sky regions almost is white, which is not pleasant. However, in the last panorama generated by our approach, the color of those regions is slightly bluish, which is more reasonable and pleasant, as shown in Figure 15(f). In conclusion, if the color differences between input images are large, our approach can also generate high-quality

Table 1: The computational times of our proposed approach and the approach proposed in (Xiong and Pulli, 2010).

	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

677 panoramas, but the results generated by the Xiong and Pulli’s approach are not so good.

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

## 686 6. Conclusion

687 In this paper, we proposed a unified framework for street-view panorama stitching system  
688 which is comprised of image warping, color correction, optimal seamline detection and image  
689 blending for stitching or mosaicking a set of geometrically aligned street-view panoramic images  
690 with large geometric misalignments and photometric inconsistencies into a visual-appealing and  
691 informative wide-angle composite image. The contributions in this paper are summarized as  
692 follows:

- 693 • We creatively proposed a novel image warping method based on the dense optical flows to  
694 greatly reduce the large geometric misalignment existed in the input images as much as  
695 possible. Experimental results have demonstrated the superiority of our proposed image  
696 warping method, which can efficiently and greatly eliminate the influence of the large  
697 geometric misalignment.
- 698 • We proposed a novel color correction and image blending method to further reduce the col-  
699 or differences between panoramic images based on extreme point matching of histograms of

700 the overlapped image regions of two involved images via both probability density functions  
701 and cumulative distribution functions. Experimental results on representative street-view  
702 panoramic images have proved that our proposed color correction method is capable of  
703 eliminating the large color differences between adjacent images captured in different illu-  
704 mination conditions and/or different exposure settings, which obviously outperforms the  
705 open-source software *Enblend* and the approach proposed by [Xiong and Pulli \(2010\)](#).

- 706 • We proposed a unified framework for street-view panorama stitching system. Even thought  
707 there are large geometrical misalignments and photometric inconsistencies in the input  
708 aligned images, our system can also generate pleasant and high-quality panoramas.

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

## 717 Acknowledgment

718 This work was partially supported by the National Natural Science Foundation of China  
719 (Project No. 41571436), the Hubei Province Science and Technology Support Program, China  
720 (Project No. 2015BAA027), the National Natural Science Foundation of China (Project No.  
721 41271431), and the Jiangsu Province Science and Technology Support Program, China (Project  
722 No. BE2014866).

## 723 References

- 724 Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen,  
725 M., 2004. Interactive digital photomontage. ACM Transactions on Graphics (TOG) 23 (3), 294–302.
- 726 Alexa, M., Cohen-Or, D., Levin, D., 2000. As-rigid-as-possible shape interpolation. In: Internationl  
727 Conference on Computer Graphics and Interactive Techniques Conference (SIGGRAPH). pp. 157–  
728 164.

- 729 Bay, H., Tuytelaars, T., Gool, L. V., 2006. SURF: Speeded up robust features. Computer Vision & Image  
730 Understanding 110 (3), 404–417.
- 731 Bellman, R., 1957. Dynamic Programming. Princeton University Press, Princeton, NJ.
- 732 Bookstein, F. L., 1989. Principal warps: Thin-plate splines and the decomposition of deformations. IEEE  
733 Transactions on Pattern Analysis and Machine Intelligence 11 (6), 567–585.
- 734 Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE  
735 Transactions on Pattern Analysis and Machine Intelligence 23 (11), 1222–1239.
- 736 Brown, M., Lowe, D. G., 2007. Automatic panoramic image stitching using invariant features. International  
737 Journal of Computer Vision 74 (1), 59–73.
- 738 Burt, P. J., Adelson, E. H., 1983. A multiresolution spline with application to image mosaics. ACM  
739 Transactions on Graphics (TOG) 2 (4), 217–236.
- 740 Chen, T.-L., Geman, S., 2014. Image warping using radial basis functions. Journal of Applied Statistics  
741 41 (2), 242–258.
- 742 Chon, J., Kim, H., Lin, C.-S., 2010. Seam-line determination for image mosaicking: A technique min-  
743 imizing the maximum local mismatch and the global cost. ISPRS Journal of Photogrammetry and  
744 Remote Sensing 65 (1), 86–92.
- 745 Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. Numerische Mathematik 1 (1),  
746 269–271.
- 747 Du, Q., Rakshit, N., Orduyilmaz, A., Bruce, L. M., 2008. Automatic registration and mosaicking  
748 for airborne multispectral image sequences. Photogrammetric Engineering & Remote Sensing 74 (2),  
749 169–181.
- 750 Fecker, U., Barkowsky, M., Kaup, A., 2008. Histogram-based prefiltering for luminance and chrominance  
751 compensation of multiview video. IEEE Transactions on Circuits and Systems for Video Technology  
752 18 (9), 1258–1267.
- 753 Glasbey, C. A., Mardia, K. V., 1998. A review of image-warping methods. Journal of Applied Statistics  
754 25 (2), 155–171.
- 755 Gracias, N., Mahoor, M., Negahdaripour, S., Gleason, A., 2009. Fast image blending using watersheds  
756 and graph cuts. Image and Vision Computing 27 (5), 597–607.
- 757 HaCohen, Y., Shechtman, E., Goldman, D. B., Lischinski, D., 2013. Optimizing color consistency in  
758 photo collections. ACM Transactions on Graphics (TOG) 32 (4), 38:1–38:10.

- 759 Hwang, Y., Lee, J.-Y., Kweon, I. S., Kim, S. J., 2014. Color transfer using probabilistic moving least  
760 squares. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3342–3349.
- 761 Igarashi, T., Moscovich, T., Hughes, J. F., 2005. As-rigid-as-possible shape manipulation. ACM trans-  
762 actions on Graphics (TOG) 24 (3), 1134–1141.
- 763 Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: active contour models. International Journal of  
764 Computer Vision 1 (4), 321–331.
- 765 Kerschner, M., 2001. Seamlne detection in colour orthoimage mosaicking by use of twin snakes. ISPRS  
766 Journal of Photogrammetry and Remote Sensing 56 (1), 53–64.
- 767 Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A., 2003. Graphcut textures image and video synthesis  
768 using graph cuts. ACM Transactions on Graphics (TOG) 22 (3), 277–286.
- 769 Lee, S., Wolberg, G., Shin, S. Y., 1997. Scattered data interpolation with multilevel B-splines. IEEE  
770 Transactions on Visualization and Computer Graphics 3 (3), 228–244.
- 771 Lee, S.-Y., Chwa, K.-Y., Hahn, J. K., Shin, S. Y., 1996. Image morphing using deformation techniques.  
772 Journal of Visualization and Computer Animation 7 (1), 3–23.
- 773 Levin, A., Zomet, A., Peleg, S., Weiss, Y., 2004. Seamless image stitching in the gradient domain. In:  
774 European Conference on Computer Vision (ECCV).
- 775 Levin, D., 1998. The approximation power of moving least-squares. Mathematics of Computation of the  
776 American Mathematical Society 67 (224), 1517–1531.
- 777 Li, L., Yao, J., Lu, X., Tu, J., Shan, J., 2016. Optimal seamlne detection for multiple image mosaicking  
778 via graph cuts. ISPRS Journal of Photogrammetry and Remote Sensing 113, 1–16.
- 779 Li, X., Hui, N., Shen, H., Fu, Y., Zhang, L., 2015. A robust mosaicking procedure for high spatial  
780 resolution remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing 109, 108–  
781 125.
- 782 Liao, J., Lima, R. S., Nehab, D., Hoppe, H., Sander, P. V., Yu, J., 2014. Automating image morphing  
783 using structural similarity on a halfway domain. ACM Transactions on Graphics (TOG) 33 (5), 1–12.
- 784 Lin, C.-C., Pankanti, S. U., Ramamurthy, K. N., Aravkin, A. Y., 2015. Adaptive as-natural-as-possible  
785 image stitching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp.  
786 1155–1163.
- 787 Lin, S.-S., Yeh, I.-C., Lin, C.-H., Lee, T.-Y., 2013. Patch-based image warping for content-aware retar-  
788 geting. IEEE Transactions on Multimedia 15 (2), 359–368.

- 789 Liu, F., Gleicher, M., 2005. Automatic image retargeting with fisheye-view warping. In: ACM Symposium  
790 on User Interface Software and Technology. pp. 153–162.
- 791 Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Com-  
792 puter Vision 60 (60), 91–110.
- 793 Mills, S., McLeod, P., 2013. Global seamline networks for orthomosaic generation via local search. ISPRS  
794 Journal of Photogrammetry and Remote Sensing 75 (1), 101–111.
- 795 Pan, J., Zhou, Q., Wang, M., 2014. Seamline determination based on segmentation for urban image  
796 mosaicking. IEEE Transactions on Geoscience and Remote Sensing Letters 11 (8), 1335–1339.
- 797 Pang, S., Sun, M., Hu, X., Zhang, Z., 2016. SGM-based seamline determination for urban orthophoto  
798 mosaicking. ISPRS Journal of Photogrammetry and Remote Sensing 112, 1–12.
- 799 Pérez, P., Gangnet, M., Blake, A., 2003. Poisson image editing. ACM Transactions on Graphics (TOG)  
800 22 (3), 313–318.
- 801 Philip, S., Summa, B., Tierny, J., Bremer, P.-T., Pascucci, V., 2015. Distributed seams for gigapixel  
802 panoramas. IEEE Transactions on Visualization and Computer Graphics 21 (3), 350–362.
- 803 Porter, T., Duff, T., 1984. Compositing digital images. ACM SIGGRAPH Computer Graphics 18 (3),  
804 253–259.
- 805 Prados, R., Garcia, R., Gracias, N., Escartin, J., Neumann, L., 2012. A novel blending technique for  
806 underwater gigamosaicing. IEEE Journal of Oceanic Engineering 37 (4), 626–644.
- 807 Schaefer, S., McPhail, T., Warren, J., 2006. Image deformation using moving least squares. In: ACM  
808 transactions on graphics (TOG). Vol. 25. pp. 533–540.
- 809 Soille, P., 2006. Morphological image compositing. IEEE Transactions on Pattern Analysis and Machine  
810 Intelligence 28 (5), 673–683.
- 811 Szeliski, R., 2006. Image alignment and stitching: A tutorial. Foundations and Trends® in Computer  
812 Graphics and Vision 2 (1), 1–104.
- 813 Tian, G. Y., Gledhill, D., Taylor, D., Clarke, D., 2002. Colour correction for panoramic imaging. In:  
814 International Conference on Information Visualisation. Vol. 215. pp. 483–488.
- 815 Uyttendaele, M., Eden, A., Skeliski, R., 2001. Eliminating ghosting and exposure artifacts in image mo-  
816 saics. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).  
817 Vol. 2. pp. II–509.

- 818 Wan, Y., Wang, D., Xiao, J., Lai, X., Xu, J., 2013. Automatic determination of seamlines for aerial image  
819 mosaicking based on vector roads alone. *ISPRS Journal of Photogrammetry and Remote Sensing* 76,  
820 1–10.
- 821 Xiong, Y., Pulli, K., 2010. Fast panorama stitching for high-quality panoramic images on mobile phones.  
822 *IEEE Transactions on Consumer Electronics* 56 (2), 298–306.
- 823 Xu, W., Mulligan, J., 2010. Performance evaluation of color correction approaches for automatic multi-  
824 view image and video stitching. In: *IEEE Conference on Computer Vision and Pattern Recognition*  
825 (CVPR). pp. 263–270.
- 826 Yamamoto, K., Oi, R., 2008. Color correction for multi-view video using energy minimization of view  
827 networks. *International Journal of Automation and Computing* 5 (3), 234–245.
- 828 Yang, Y., Gao, Y., Li, H., Han, Y., 2011. An algorithm for remote sensing image mosaic based on valid  
829 area. In: *International Symposium on Image and Data Fusion (ISIDF)*.
- 830 Yao, J., 2006. Modeling and rendering from multiple views. Ph.D. thesis, The Chinse University of Hong  
831 Kong.
- 832 Yao, J., Cham, W.-K., 2006. 3D modeling and rendering from multiple wide-baseline images by match  
833 propagation. *Signal Processing: Image Communication* 21 (6), 506–518.
- 834 Yao, J., Cham, W.-K., 2007. Robust multi-view feature matching from multiple unordered views. *Pattern  
835 Recognition* 40 (11), 3081–3099.
- 836 Yu, L., Holden, E.-J., Dentith, M. C., Zhang, H., 2012. Towards the automatic selection of optimal seam  
837 line locations when merging optical remote-sensing images. *International Journal of Remote Sensing*  
838 33 (4), 1000–1014.
- 839 Zaragoza, J., Chin, T.-J., Brown, M., Suter, D., 2013. As-projective-as-possible image stitching with  
840 moving DLT. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2339–  
841 2346.

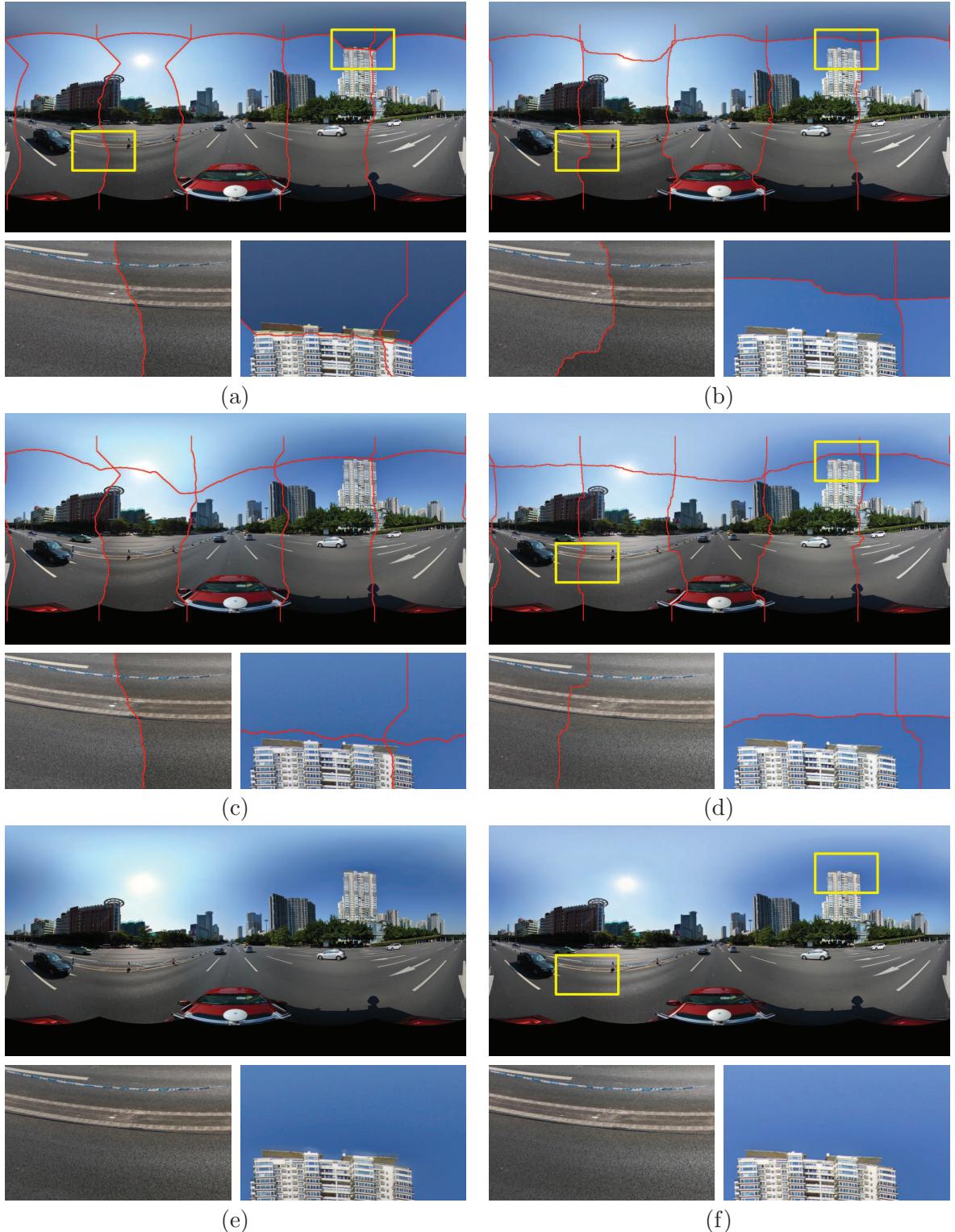


Figure 14: Visual comparison between our approach in the left column and the Xiong and Pulli's approach in the right column on the first group of images with relatively small color differences: (a)-(b) the results without the use of color correction; (c)-(d) the results with the use of color correction;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

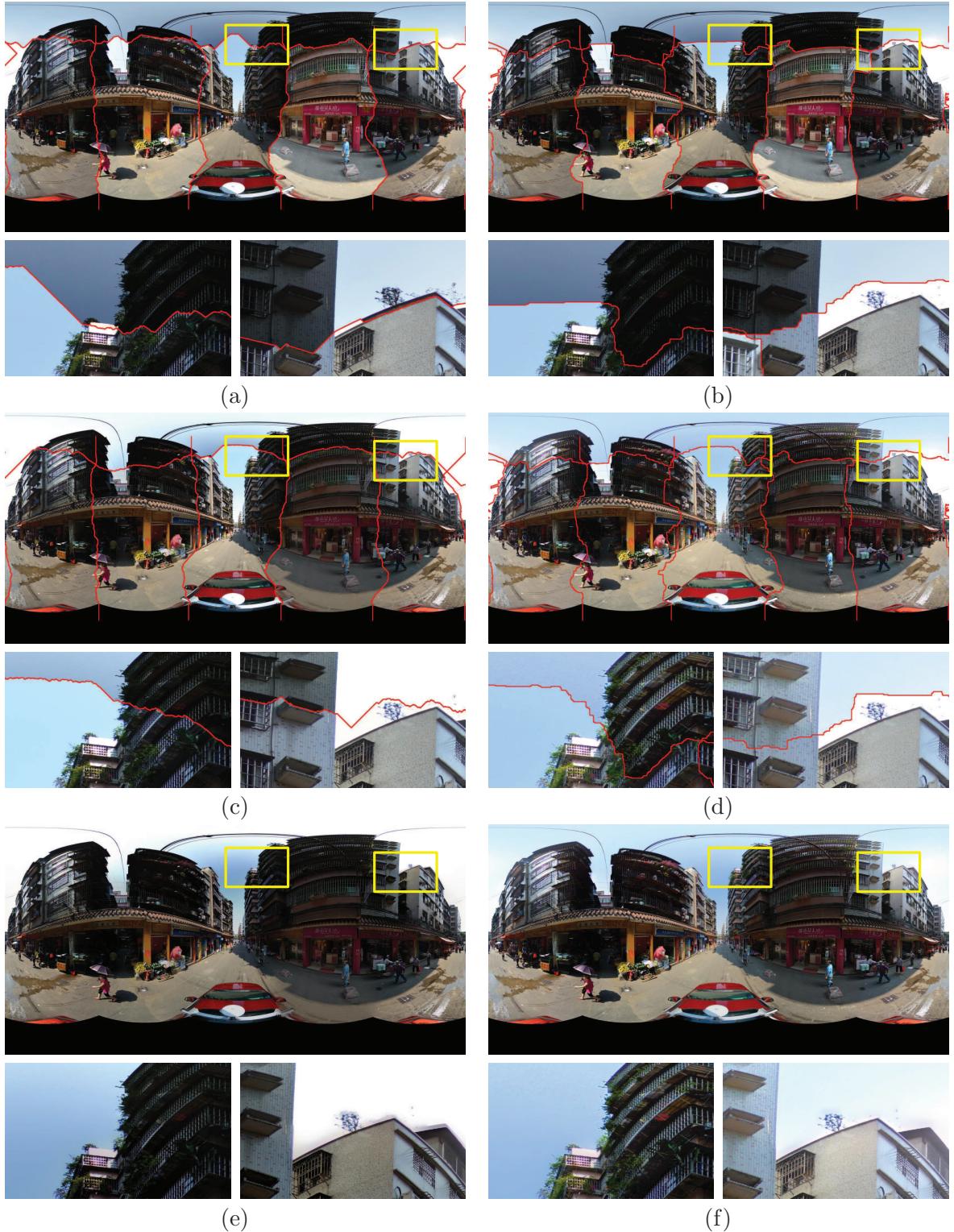


Figure 15: Visual comparison between our approach in the left column and the Xiong and Pulli's approach in the right column on the second group of images with large color differences: (a)-(b) the results without the use of color correction; (c)-(d) the results with the use of color correction; (e)-(f)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