



An Effective Dual-Fisheye Lens Stitching Method Based on Feature Points

Li Yao^{1,2(✉)}, Ya Lin¹, Chunbo Zhu³, and Zuolong Wang³

¹ School of Computer Science and Engineering, Southeast University, Nanjing 211189, People's Republic of China

Yao.li@seu.edu.cn

² Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing 211189, People's Republic of China

³ Samsung Electronics, Suwon, South Korea

Abstract. Fisheye lens is a super-wide-angle lens which is very light. Usually two cameras can shoot 360-degree panoramic images. However, the limited overlapping field of views make it hard to stitch in the boundaries. This paper introduces a novel method for dual-fisheye camera stitching based on feature points. And we also put forward the idea of expanding to video. Results show that this method can be used to produce high-quality panoramic images by stitching the original images of the dual-fisheye camera Samsung Gear 360.

Keywords: Dual-fisheye · Stitching · Panorama-video · Virtual reality

1 Introduction

Dual-fisheye lens cameras are becoming popular for 360-degree video capture. The focal length is very short and a single lens's viewing angle can approach even more than 180°. Compared to the traditional and professional 360-degree capturing systems such as [1] and [2], their portability and affordability make them available for live streaming. It has been widely used in safety monitoring, video conference, panoramic parking because of its large viewing angle and small size.

However, the limited overlapping filed of views and misalignment between the two lenses increase the difficulty of stitching. For stitching of images from the multiple cameras, a classic method is autostitch [3], which extract features from the images being stitched and calculate the homography matrix to transform them to the same plane. This method relies on accurate feature points and cannot be directly applied to the dual-fisheye camera. Gao et al. [4] use two homographies per image to produce a more seamless image. Lin et al. [5] use more affine transformations which have stronger alignment capabilities. Although these two methods improve the stitching results, they are heavily dependent on feature points having high computational complexity and cannot be used in real-time image processing. In video stitching, He et al. [6] present a parallax-robust video stitching technique for timely synchronized surveillance video. But this algorithm requires that the camera position and background remain unchanged. Lin et al. [7] presented a algorithm that can stitch videos captured

by hand-held cameras and can get good results, but the efficiency is too low. Ho [8] et al. proposed a two-step alignment method for dual-fisheye lens using fast template matching as a substitute for feature points, but fast template matching is considered to be computationally expensive [9]. There are many problems with these methods directly applied to the dual-fisheye lens.

In this paper, we propose a feature point-based stitching method whose efficiency can meet the requirements of real-time performance. This algorithm contains four steps: color correction, unwarping, alignment and blending. Our contributions are:

- (1) A simple and effective color correction is used to correct the color inconsistency between two lenses which can easily meet the requirement of real-time.
- (2) In the spherical model, we map the image outside the 180° view to the other hemisphere of the sphere and expand the entire sphere. We can easily find overlapping areas which help calculate color differences and detect feature points.
- (3) By matching feature points in sliding window, we make it possible to match the feature points in the dual-fisheye image.
- (4) By grading the homography, we can align the left and right sides of the fish-eye image separately using different rotation matrices.
- (5) We optimized the method of multi-band blending [10] to make it more suitable for fisheye image, which is faster but never reduce the image quality.

2 Dual-Fisheye Stitching

Figure 1 shows the processing flow of our approach. There are 4 steps in total, where the overlapping area mapping matrix and the affine warping matrix could be pre-computed and remain unchanged. We will generate a new warping matrix according to the rotation angle in the process of alignment. If we need, the new one could also be pre-computed because the range of the rotation angle is small, so the speed of our algorithm could be very fast.

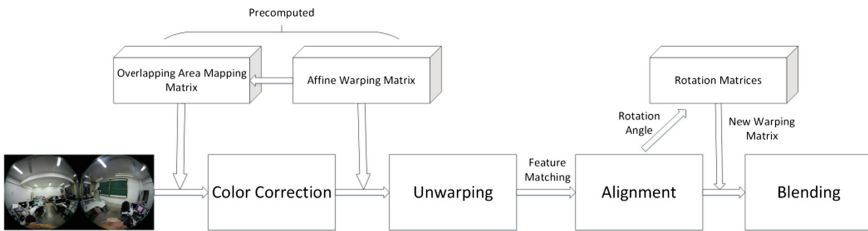


Fig. 1. The processing flow of this paper.

2.1 Color Correction

Due to the uneven brightness of the ambient light, the camera will inevitably have inconsistent hue and brightness when imaging. Ho et al. [11] solved the problem of vignetting through intensity compensation. Because there are also nuances in different cameras, it is difficult to accurately quantify the difference in color. In the process of stitching, a simple and efficient method is to correct the color of the image in different color spaces. For two images with large color difference, we assume that the overlap area after registration is A , then the two images to be stitched must have the same number of pixels in A . In general, the two images in the overlapping area are under the same scene, so we can quantify the color difference with the statistics of this area.

Take the Samsung Gear 360 as an example, we calculate the sum of the two images on the RGB three channels respectively. On different channels, the greater the difference between the sum, the greater the error. Figure 2(a) shows the original image with a large color difference, from which we can see that the fisheye image on the left is yellowed compared to the right one. The stitching result showed in Fig. 1(b) also proved this. From the results showed in Table 1, we can see that the gap between the channels is not very significant. But when converting to the HSV model [12], we can clearly see the difference between the two images in the S channel. So we only need to scale all the pixels in the S channel. and the result is shown in Fig. 2(e).

Such a color correction method only needs to perform a calculation operation on a specific area as a whole, and can meet the requirements of real-time performance (Table 2).

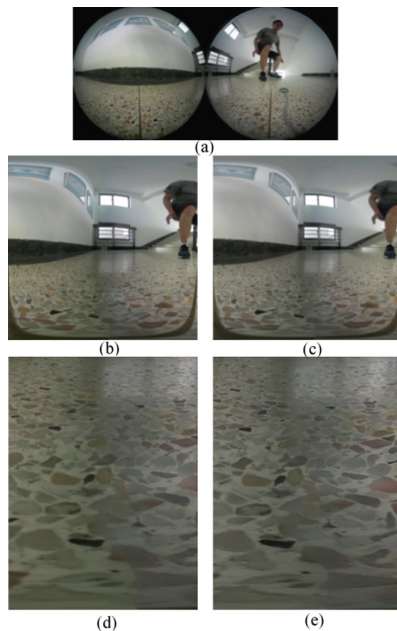


Fig. 2. (a) Original image taken by Samsung Gear 360. (b) Stitch without color correction. (c) Stitch using the color correction method we proposed. (d), (e) are the enlarged parts of (b), (c).

Table 1. Cumulative sums of RGB channels in overlapping regions.

| | R | G | B |
|-------------|---------------|---------------|---------------|
| Left image | 8.01446e + 07 | 8.45898e + 07 | 8.3173e + 07 |
| Right image | 7.91371e + 07 | 8.47454e + 07 | 7.79583e + 07 |
| SumL/SumR | 0.886 | 0.998 | 1.067 |

Table 2. Cumulative sums of HSV channels in overlapping regions.

| | H | S | V |
|-------------|---------------|---------------|---------------|
| Left image | 5.8555e + 07 | 2.65566e + 07 | 8.65651e + 07 |
| Right image | 5.44479e + 07 | 3.91667e + 07 | 8.53828e + 07 |
| SumL/SumR | 1.075 | 0.678 | 1.014 |

2.2 Fisheye Unwarping

The ability of a fisheye lens to capture large viewing angles is at the expense of the intuitiveness of the image, the most serious being barrel distortion [13]. Most of the algorithms cannot perform well on a distorted image. In addition, the original fisheye image cannot be stitched directly. Spherical perspective model [14] is commonly used to describe the imaging process of a fisheye lens. This model can be used not only to correct distortion but also to convert the shape of fisheye images.

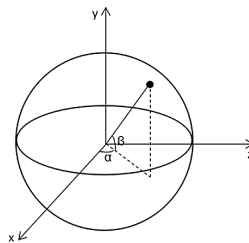


Fig. 3. Fisheye unwarping.

The first step is to map the original fisheye image to a three-dimensional unit sphere. Create a unit spherical model as shown in Fig. 3. In order to reduce the calculation for filling in the blank pixel points and facilitate the expansion, a reverse mapping method is used. Assume that the size of the image after expansion from the sphere is $h \times w$, let x-axis positive direction be the starting longitude and establish w warps at intervals from the angle $-\pi$ to $+\pi$. Similarly, from $-\pi/2$ to $+\pi/2$, we establish h wefts. We can get a total of $h \times w$ intersections. For one point on the sphere whose longitude is α and latitude β , we can calculate its three-dimensional coordinates:

$$\begin{aligned}
 x &= \cos \alpha \times \cos \beta \\
 y &= \sin \beta \\
 z &= \sin \alpha \times \cos \beta
 \end{aligned} \tag{1}$$

Each intersection needs to be mapped to a point on the fisheye image. Let f be the camera's field of view (FOV), and we assume the camera's FOV is uniform. For a fisheye camera with a 180-degree FOV, it maps perfectly to a hemisphere. Then the projection of the original image on the sphere will exceed the hemisphere when the FOV exceeds 180°, so for the part beyond 180°, it should be mapped to the other side of the sphere.

For a point on the sphere with coordinates (x, y, z) , we can calculate its deviation from the x axis:

$$\theta = \arccos x \tag{2}$$

Then we can get the scale factor from the center in the original fisheye image:

$$\varphi = \frac{\theta}{\pi} \times \frac{180}{f} \times r \tag{3}$$

Where r is the radius of the original fisheye image. Finally the corresponding point on the fish-eye image is:

$$(z \times \varphi, y \times \varphi) \tag{4}$$

if we assume that the center coordinates of the fisheye image are $(0, 0)$.

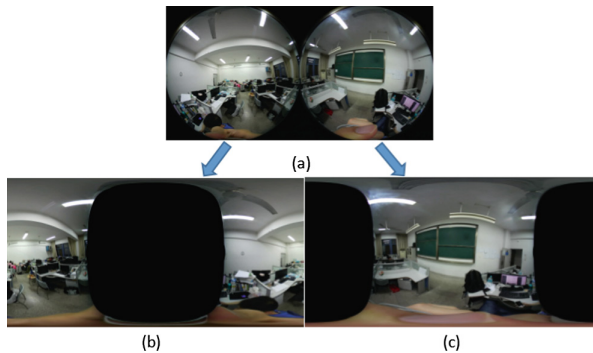


Fig. 4. Fisheye unwarping results.

Now we can map any point on the sphere to the original fisheye image, we need to map the points on the sphere to a plane that is easy to stitch. We have chosen a plane of size $h \times w$, the number of points on the sphere is also $h \times w$, although their

distribution on the sphere is not uniform. Points at the same latitude should be on the same line of the expanded image, the same is true for longitude. Knowing this, when expansion, it can be segmented from any one of the longitude lines, and the pixels can be arranged in the expansion view in order. Figure 4 (b), (c) show the expanded images of original image (a) Photographed by Gear 360.

In general, the spherical model is only a rough description of the fisheye imaging process. There may be various types of distortion in the imaging process, and the FOV of the lens may not be uniform. So we need more accurate alignment.

2.3 Alignment

By mapping the fisheye image of the circular area to the image shown in Fig. 4, we can clearly see the overlapping region of the two images, its shape is roughly as shown in Fig. 5. Before blending them together, we adopted a alignment process to make the same objects as close as possible. The method of computing homography matrix based on feature points is very mature, but a lot of adjustments are needed when performing on the fisheye images.

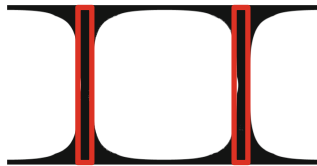


Fig. 5. Overlapping area (Marked with black).

One of the differences between a fisheye camera and an ordinary camera is that we can measure the FOV in advance and the value will remain, we can reduce calculations and make the result more accurate by taking use of this information. The overlapping area of the fisheye lens is generally small which is approximate band shape. We only search and match feature points in the overlapping area. In order to improve the accuracy of matching, we can set some fixed window areas and match them within the window pairs [15]. The wrong point pairs will undoubtedly have a negative impact on the RANSAC [16] algorithm. The matching points on fisheye images usually do not differ much in horizontal direction, so we can manually remove some of the points where the angles are very different before performing RANSAC algorithm (Fig. 6).

There are two overlapping areas in the expanded view of the fisheye lens. Since the two overlapping regions differ by exactly 180° in space, their parallax is likely to be different. In order to get a panoramic image with size $h \times w$, and leave no blanks on the border, we stitch the two overlapping areas separately, and handle conflicting parallax conflicts properly.

For a set of matching point pairs (x_1, y_1) and (x_2, y_2) , the pixel difference in the vertical direction between them is $y_2 - y_1$. Return to the spherical model and the angle difference between them is:

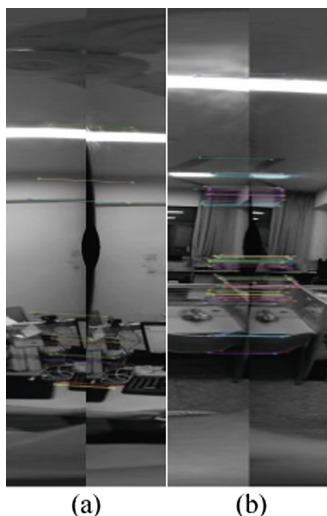


Fig. 6. Feature points matching results. (a) Matching results on the left side. (b) Matching results on the right side.

$$X = \arcsin(y_2 - y_1) \tag{5}$$

In order to get more accurate angle, we take the average value of the angle difference of n pairs of matched points.

With the angle difference, we just need to rotate one image on the sphere by (X, Y, Z) . Here we don't consider Y and Z for the time being). Convert it to a normalized quaternion (a, b, c, d) , and then create rotation matrix R from the quaternion [17]:

$$R = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{pmatrix} \tag{6}$$

We would only align one side if we rotate the entire image. However, the calculated rotation angles on both sides may be inconsistent, so we make a smooth process for the rotation matrix in order to make the two sides do not affect each other. Assume that the original rotation matrix is R' , we build a series of evenly changing matrices $(R_0, R_1, R_2, \dots, R_k, \dots, R_n)$ from R to R' , where the number of matrices can be equal to $w/4$ (Roughly half of a single image). From edge to center, multiply each column of pixels by the corresponding rotation matrix(the k th column pixels multiply by R_k). In this way, we only stitch one side without affecting the other, and this uniform gradient matrix does not have an adverse effect on the visual. The same method can be used for horizontal correction which affects angle Z .

2.4 Blending

Blending is the last step of the stitching, which can make smoother transition in overlapping area. A common practice is to find the best seam [18], and then perform the multi-band blending method on the images on both sides of the seam.

The multi-band blending method can eliminate the seam well, but it reduces the image quality [19]. Here we use the method proposed by Xiao [19] et al. In this way we perform multi-band blending only on the overlapping area which is very narrow in a fisheye image. When we get our best seam showed in Fig. 7(a), then we take a small piece of each image on the left and right side of the image for blending. After that, we get Fig. 7(b). Calculate the weighted average pixel value between the original left and right image we used last step and Fig. 7(b) according to the distance from the seam. Let (r, c) be the pixel at row r and column c in the overlapping region and we assume that one point where the seam passes is $S(r', c')$. Then the blended pixel $B(r, c)$ on the left side of S can be calculated as follows:

$$B(r, c) = \frac{c' - c}{d} \times L(r, c) + \left(1 - \frac{c' - c}{d}\right) \times O(r, c) \tag{7}$$

whereas d represents the distance from the point of furthest to $S(r', c')$ to $S(r', c')$, and $O(r, c)$ represents the point in the temporary blending region such as Fig. 7(b). Finally, we get our final result showed in Fig. 7(c).

This approach accelerates the speed of blending without degrading image quality. In the example of Fig. 7, the size of the panorama we eventually get is 2048×4096 . On the side we show in Fig. 7, the size of our blending area is 2048×600 . So the total

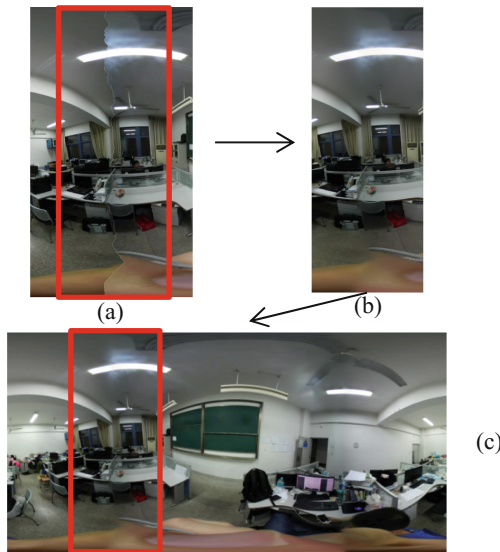


Fig. 7. Blending only on the overlapping region.

size of blending area is 2048×1200 , which is about one quarter of the whole image. This means saving three-quarters of the computing time in the blending stage.

3 Extend to Video

The method described above is for images, and it is time-consuming if we directly perform it on a video, there will also be discontinuities between frames. For the problem of discontinuities, we only recalibrate when objects are moving in the overlapping area. Algorithm 1 illustrates our method of maintaining the temporal coherence for the sequence. And for the improvement of time performance, we use some special techniques. We use ORB [20] for feature matching, which is proved to be faster than SIFT [21] and SURF [22]. The alignment process is the most time-consuming, which requires a lot of matrix operations to correct the offset angle. During the tests, we found that the offset angle has a fixed range and this range is not so wide because the position of our lenses is fixed. Therefore, the converted mapping matrix can be calculated in advance and corresponds to the angle. In the alignment process, we only need to find the best-fit mapping matrix according to the rotation angle calculated by the matching feature points.

Algorithm 1 Video Stitching

```

i ← 0
xAngle, yAngle, zAngle ← 0
isRotating ← False
rotMat ← defaultRotMat
scale ← defaultRotScale
for frame in real-time video stream do
  if i%N = 0 or isRotating or SceneChanged() then
    // We recalculate the rotation matrix in three cases:
    // 1. When this is the initial frame, where i = 0,
    // And every fixed frame N. Maybe the scene
    // changes slowly but the parallax changes.
    // 2. When it is rotating now.
    // 3. When scene changes.
    xAngleCur, yAngleCur, zAngleCur ← Recompute the x, y, z angle
    // We don't adjust to the final state instantly.
    xAngle ← adjust min(scale, xAngleCur - xAngle) along gradient
    yAngle ← adjust min(scale, yAngleCur - yAngle) along gradient
    zAngle ← adjust min(scale, zAngleCur - zAngle) along gradient
    if xAngle = xAngleCur and yAngle = yAngleCur
    and zAngle = zAngleCur then
      isRotating ← True
    else
      isRotating ← False
    end if
    rotMat ← calculate the rotation matrix with x, y, z angle
  end if
  warp with rotMat
  i ← i + 1
end for

```

4 Experiments and Analysis

First, we show the comparison result of color correction between Samsung Gear 360 software (Fig. 8(a)) and our algorithm (Fig. 8(b)). We use the black line to mark the stitching line in the result. It can be clearly seen from the left and right sides of the line

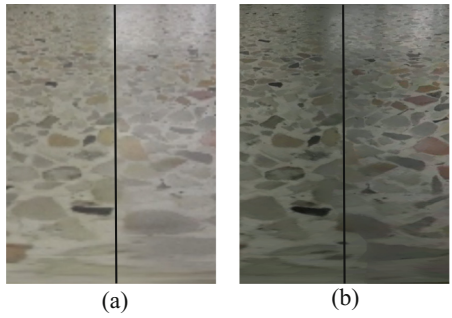


Fig. 8. Color correction result. (a) Enlarged result of Gear 360 software. (b) Enlarged result of our method.

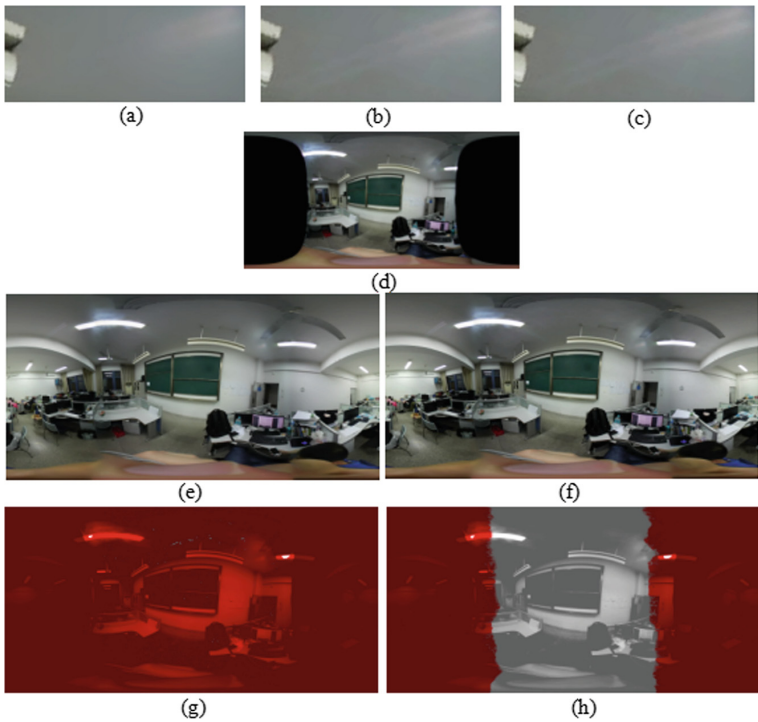


Fig. 9. Comparison of blending results. (d) Left lens original expanded image. (e) Stitching result of multi-band blending. (f) Stitching result of our method. (a–c) are enlarged parts of (e–f). (g), (h) are results of comparing (e, f) with (d) using Beyond Compare. (Color figure online)

that Gear360 has a poor correction effect on the color, and our method basically makes the color consistent.

In order to verify the advantages of the blending method used in this paper in image quality, we enlarge the projecting part of the light on the wall in Fig. 9(d). Figure 9(a), (b), and (c) are correspond to the region of (e), (d) and (f), respectively. It can be found that (c) has remained almost the same and (a) has become blurred. Besides, we use software Beyond Compare [23] to analyze pixel differences. We use the right expanded image as the standard for comparison because the right image of the original fisheye remains unchanged before and after alignment. We use the results of multi-band blending and our results to compare with Fig. 9(a) respectively. The comparison results are showed in Fig. 9(d), (e). Gray color means that the pixel value is the same here, and red means different. From the results we see that the multi-band blending algorithm changed the value of some pixels, but our method only changes the pixel value of the stitching area, which keeps the details of the image.

In Fig. 10, we showed the stitching results of two sets of videos using our stitching method. Each row in the figure is consecutive frames in the video, where the first row and the third row are the results of Gear 360 software, and the second and the last one are ours. From the results, we can see that the alignment ability of our method is better than that of Gear 360 software in both indoor and outdoor scenarios.

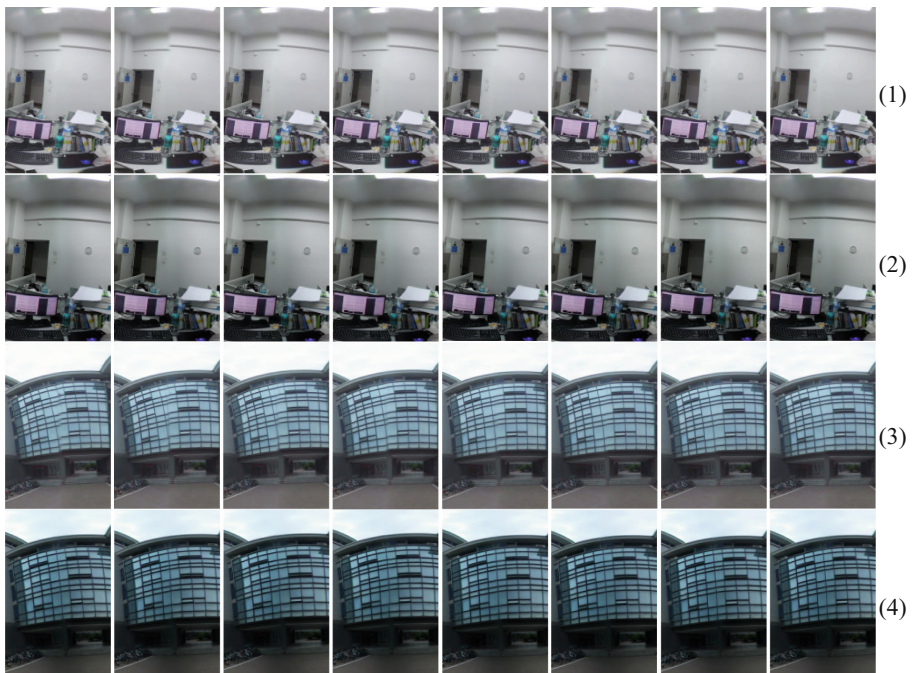


Fig. 10. Stitching boundary in consecutive frames. Row (1) and (3) are results of Gear 360. Row (2) and (4) are results this paper.

5 Discussion and Future Work

This paper has introduced a novel method for stitching the images generated by the dual-fisheye lens cameras. This method overcomes the shortcomings of small and severe distortion in the overlapping area of dual-fisheye images, enables feature points to be found and matched correctly, and the stitching of left and right side will not affect each other by making the rotation matrix gradual. Meanwhile, Based on the color correction of Gear 360, a new idea of quickly solving the color difference of stitching images is put forward. Our method can be applied to video through pre-calculation, and have the ability to adapt to the scenes changing slowly. But for fast-changing scenes, there are still no simple and effective strategies to meet real-time requirements. More work will be carried out about video in the future.

Acknowledgement. This work is supported by natural science foundation of Jiangsu Province under Grant No. BK20181267.

References

1. GoPro Odyssey. <https://gopro.com/odyssey>. Accessed 27 April 2018
2. Facebook Surround360. <https://facebook360.fb.com/facebook-surround-360>. Accessed 27 April 2018
3. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.* **74**(1), 59–73 (2007)
4. Gao, J., Kim, S.J., Brown, M.S.: Constructing image panoramas using dual-homography warping. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–56. IEEE Computer Society (2011)
5. Matsushita, Y.: Smoothly varying affine stitching. In: *Computer Vision and Pattern Recognition*, pp. 345–352. IEEE (2011)
6. He, B., Yu, S.: Parallax-robust surveillance video stitching. *Sensors* **16**(1), 7 (2015)
7. Lin, K., Liu, S., Cheong, L.F.: Seamless video stitching from hand-held camera inputs. In: *Computer Graphics Forum*, pp. 479–487 (2016)
8. Ho, T., et al.: 360-degree video stitching for dual-fisheye lens cameras based on rigid moving least squares. In: *IEEE International Conference on Image Processing*, pp. 51–55. IEEE (2017)
9. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_32
10. Burt, P.J.: A multiresolution spline with applications to image mosaics. *ACM Trans. Comput. Graph.* **2**(4), 217–236 (1983)
11. Ho, T., Budagavi, M.: Dual-fisheye lens stitching for 360-degree imaging. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2172–2176. IEEE (2017)
12. Stricker, A.M.A., Orengo, M.: Similarity of color images. In: *Proceedings of SPIE Storage & Retrieval for Image & Video Databases*, vol. 2420, pp. 381–392 (1995)
13. Ngo, H.T., Asari, V.K.: A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images. *IEEE Trans. Circuits Syst. Video Technol.* **15**(3), 436–444 (2005)

14. Ying, X.H.: Fisheye lense distortion correction using spherical perspective projection constraint. *Chin. J. Comput.* (2003)
15. Sharghi, S.D., Kamangar, F.A.: Geometric feature-based matching in stereo images. In: 1999 Proceedings of IEEE Information, Decision and Control, IDC 1999, pp. 65–70 (1999)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus. *Commun. ACM* **24**(6), 381–395 (1981)
17. comp.graphics.algorithms Frequently Asked Questions. www.faqs.org/faqs/graphics/algorithms-faq2. Accessed 27 April 2018
18. Gao, J., Li, Y., Chin, T.J., Brown, M.S.: Seam-driven image stitching. In: Eurographics (2013)
19. Xiao, J.S., Rao, T.Y.: An image fusion algorithm of Laplacian pyramid based on graph cutting. *J. Optoelectron. Laser* **25**(7), 1416–1424 (2014)
20. Rublee, E., et al.: ORB: an efficient alternative to SIFT or SURF. In: International Conference on Computer Vision, Barcelona, pp. 2564–2571 (2011)
21. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
22. Li, Y., et al.: A fast rotated template matching based on point feature. In: MIPPR 2005: SAR and Multispectral Image Processing, 60431P–60431P-7 (2005)
23. Beyond Compare. <http://www.beyondcompare.cc>. Accessed 6 May 2018