Immersive Video Stitching of Dual Fisheye Videos

Xinyu Fu, *Author*
Hanqiu Sun, *Member, IEEE, Supervisor,* Zhensong Zhang, *Ph.D., Supervisor*

***Abstract*—Dual-fisheye cameras are popularly used nowadays to create 360-degree immersive videos. This paper proposes a comprehensive stitching pipeline from de-warping to blending, to seamlessly stitch dual-fisheye videos into 360-degree videos. The method introduced in the paper generates 360-degree videos from dual-fisheye videos recorded by Samsung Gear 360, with better quality compared to Samsung's official tool.**

***Index Terms*—Dual-fisheye, Stitching, 360-degree Videos, Virtual Reality.**

I. Introduction

Immersive videos, also known as 360-degree videos or spherical videos, are ones in which a view of all directions are recorded. Viewers can control viewing directions during playback of 360-degree videos, which enables a great potential for providing contents for immersive VR viewing experiences. 360-degree videos can be recorded using an omnidirectional camera or a special rig of multiple cameras. Dual-fisheye lenses are a popular construct of commercial 360-degree recording devices, since a single fisheye lens can capture hemispherical (or larger) images, and thereby dual-fisheye lenses can create a complete spherical view. This kind of devices includes Samsung Gear 360, Ricoh Theta S, Garmin VIRB 360, Insta360, and so on.

For dual-fisheye cameras, two separate footages are produced at the same time, by the two fisheyes respectively (Fig. 4a). The separate footages would then be merged into one by a process called video stitching. Nowadays, there are some existing commercial solutions for stitching videos, such as VideoStitch Studio [1] and Kolor Autopano Video [2]. They can roughly stitch and produce 360-degree videos. Although their interface is user-friendly, they are not designated to dual-fisheye stitching. Also, even for Gear 360 Action Director [3], the official tool of Samsung Gear 360, the result videos usually contain misalignments, artifacts, and distortions.

The techniques for dual-fisheye video processing has not been fully examined by far. So, our aim is to design and implement an effective pipeline to de-warp, stitch, calibrate and refine videos captured from dual-fisheye cameras, into seamless 360-degree videos with less disadvantages mentioned above.

The dedicated stitching pipeline is listed as follows,

Prof. Hanqiu Sun is with Department of Computer Science and Engineering, The Chinese University of Hong Kong, who supervised this summer research project.

Zhensong Zhang is a Ph.D. student of Prof. Sun, who helps me with guidance during this summer.

Manuscript lastly revised August 17th, 2017.

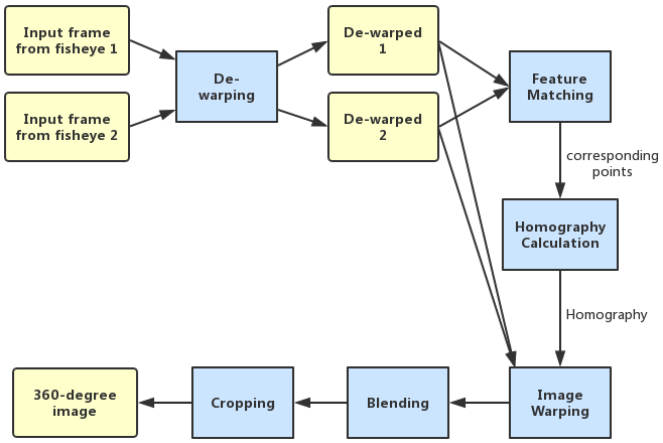
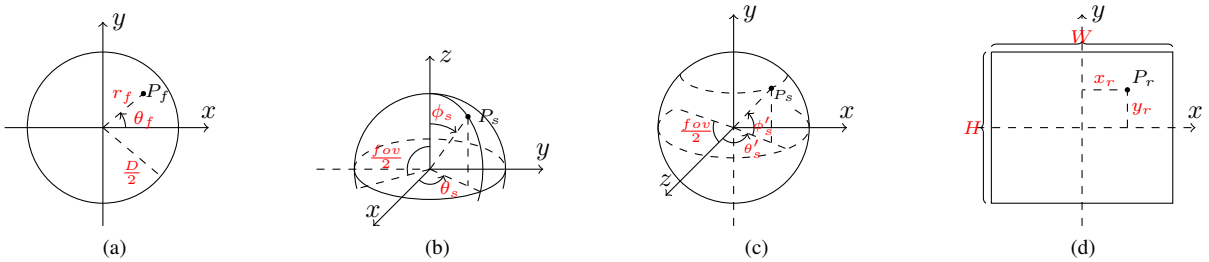


Fig. 1. An overview of 360-degree video generating pipeline.

- Fisheye de-warping: A rigorous geometric model for general fisheye de-warping;
- Feature matching: A combination of Shi-Tomasi corner detector [4] and normalized template matching;
- Homography calculation: RANSAC (Random Sample Consensus) [5] to reject outliers and obtain reasonable homography matrix for stitching;
- Optimal seamline: Modified minimum error boundary cut algorithm [6] with temporal coherence;
- Blending: Multi-band blending (Laplacian pyramid blending) [7] to balance intensity differences between fisheye images.

II. Related Works

The study of fisheye lenses on creating panoramic images has been started since long before. In 1997, Yalin Xiong and K. Turkowski reported a method for creating VR views using self-calibrated fisheye lens [8]. About ten years later, Richard Szeliski (2006) in Microsoft Research provided a comprehensive tutorial for image alignment and stitching [9]. Also in 2006, a research about calibration of fisheye lenses was conducted by Frank Heuvel, Ruud Verwaal and Bart Beers [10]. Then in 2008, a study of panorama generation using two fisheye lenses, combining fisheye modeling and calibration, was reported by Xiaoming Deng, Fuchao Wu, Yihong Wu and Chongwei Wan [11]. Meanwhile, Shaoxing Hu and Xuyong Feng developed out a CCD-panoramic-line scanner with fisheye lens in 2009 [12]



(a) (b) (c) (d)

Fig. 2. De-warping model. (a) Original fisheye. (b) Azimuthal equidistant projection of the original fisheye. (c) Coordinate system after transformation. (d) Rectangular de-warped image.

Recently, consumer-level fisheye cameras emerged, along with several commercial image/video stitching software with varying performance. Just in this year (2017), Tuan Ho and Madhukar Budagavi proposed their method for panoramic imaging using commercial dual-fisheye cameras. Our paper would be an further exploration into 360-degree video generating.

III. Method

Fig. 1 is an overview of our method. For each frame from a dual-fisheye video, it is de-warped firstly to a equirectangular form. Features would then be extracted from two de-warped images and matched. Using paris of corresponding points to calculate homography, the two images can then be warped into a primitive pano, which will go through blending and cropping to become the final 360-degree image.

A. Fisheye De-warping

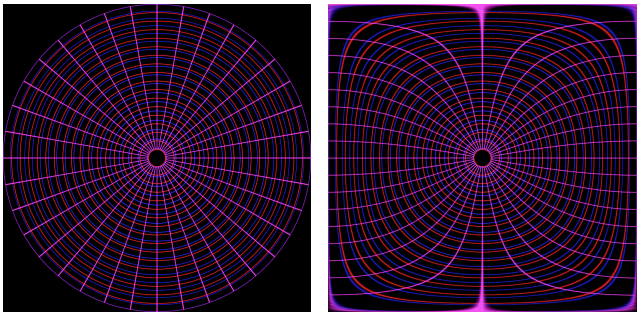
A fisheye lens generally has an ultra-wide FOV (field of view) by means of bending the incident light, which results in strong visual distortion intended to create a wide panoramic image, as shown in Fig. 4a. Fisheye de-warping is needed for the following reasons.

- Circular fisheye images are not suitable for image processing;
- The convex non-rectilinear appearance of fisheye needs to be corrected, especially for those in the periphery.
- Most importantly, the equirectangular form is supported as default by most 360-degree media viewers;

Therefore, a de-warping process of fisheye images is necessary to be conducted first to facilitate stitching process.

This paper adopts a rigorous geometric de-warping model that projects fisheye images into rectangular images. The model divided into two phases: circular-to-spherical projection phase and spherical-to-rectangular projection phase.

1) Circular-to-spherical projection phase: As its name says, this phase projects a circular area—i.e. the input fisheye image (Fig. 2a), into a spherical area (Fig. 2b). Azimuthal equidistant projection [13] is used here because of its useful propertiy that all points on the map are at proportionately correct distances from the center point. This property ensures that marginal objects would not be compressed (cartesian-to-spherical projection [14]) or amplified (stereographic projection [15]), so as to keep the same scale as objects near center.



(a) (b)

Fig. 3. (a) Fisheye mesh. (b) De-warped mesh with FOV set to 180.

Let $P_f(r_f, \theta_f)$ be a pixel on a fisheye image whose center is located at the origin. Then the corresponding point $P_s(r_s, \theta_s, \phi_s)$ over the unit sphere is constructed as

$$\begin{aligned} r_s &= 1 \quad \theta_s = \theta_f \\ \phi_s &= \frac{fov}{D} r_f \end{aligned} \tag{1}$$

, where fov is the lens' field of view and D is the diameter of this fisheye image. Naturally its cartesian coordinate $P_s(x_s, y_s, z_s)$ is given by

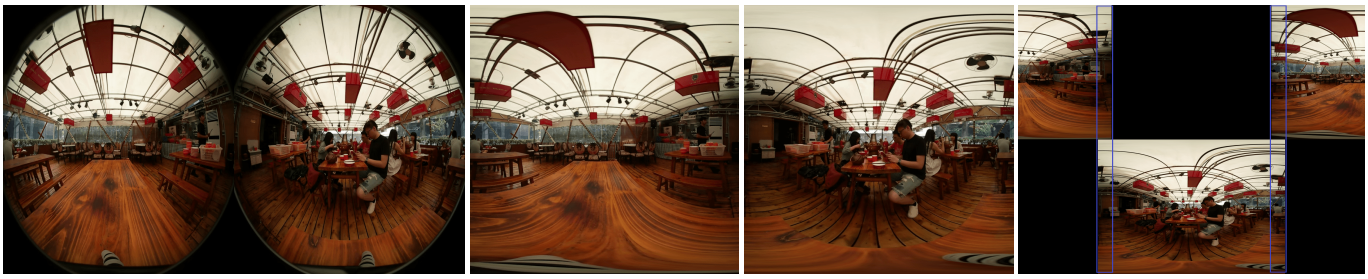
$$\begin{aligned} x_s &= \sin\phi_s \cos\theta_s \\ y_s &= \sin\phi_s \sin\theta_s \\ z_s &= \cos\phi_s \end{aligned} \tag{2}$$

For the next projection phase, a transformation over the coordinate system needs to be performed here to solve longitude $\theta_s{}'$ and latitude $\phi_s{}'$ in Fig. 2c.

$$\begin{aligned} \theta_s{}' &= \arctan\frac{x_s}{z_x} \\ \phi_s{}' &= \arctan\frac{y_s}{\sqrt{x^2+z^2}} \end{aligned} \tag{3}$$

, where $\theta_s{}'$ ranges in $[-\frac{fov}{2}, +\frac{fov}{2}]$ while $\phi_s{}'$ ranges in $[-\frac{\pi}{2}, +\frac{\pi}{2}]$.

2) Spherical-to-rectangular projection phase: This phase projects a spherical area (Fig. 2c), into a rectangular area which is the de-warped image (Fig. 2d). Equirectangular projection [16] is used here.



(a) (b) (c) (d)

Fig. 4. (a) A dual-fisheye image captured by Samsung Gear 360 (2016) [17]. (b) Dewarped left fisheye. (c) De-warped right fisheye. (d) Placement of de-warped fisheye images

So the corresponding pixel $P_r(x_r, y_r)$ on the de-warped image is constructed as

$$x_r = \frac{W}{fov}{\theta_s}' \\ y_r = \frac{H}{\pi}{\phi_s}' \tag{4}$$

, where W and H are width and height of the de-warped image. Note that a small region of the sphere would be lost here, because of the range of ϕ_s.

Thus, with equation systems (1) (2) (3) (4) combined, a fisheye image can be rigorously de-warped based on its FOV. Fig. 3 illustrates the effect of our de-warping model.

B. Feature Matching

The core part of image stitching is to find pairs of corresponding points between two images. This is generally achieved by extract feature points on both images, and then match them by calculating distances or similarities.

Performing feature matching directly over the de-warped images generally do not produce a good result, there would be plenty of outliers and mismatches as shown in Fig. 5. Characteristics of dual-fisheye cameras are supposed to be involved into consieration.

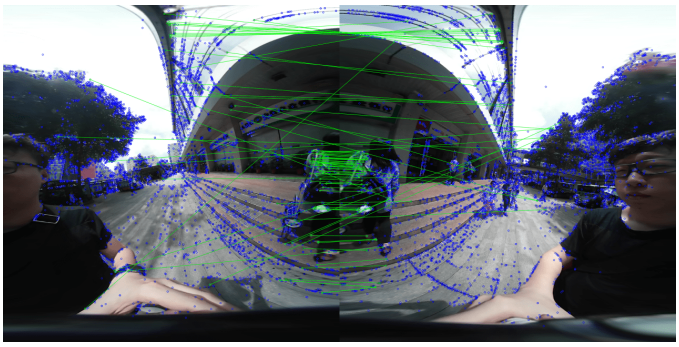


Fig. 5. Feature matching using SIFT, Lowe's ratio test [18] and FLANN based matcher [19].

Because of the extra FOV of dual-fisheye cameras, there is a fixed overlapped region between two lenses. When it comes to de-warped rectangular images, the left most and right most regions of both images are overlaps, from which we can gain advantages for feature matching. So we shifted and placed the de-warped images as shown in Fig. 4d, and performed feature matching in overlaps using a combination of template matching and Shi-Tomasi corner detector [4].

The template matching method is inspired by Tuan Ho and Madhukar Budagavi's work [20]. The method scans through source image to search for the most similar area to a template patch. The comparison is based on similarity calculation, which is normalized cross-correlation as shown below.

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y') \cdot I'(x+x', y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x', y+y')^2}}$$

with

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'',y'') \\ I'(x+x', y+y') = I(x+x', y+y') \\ - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x+x'', y+y'')$$

, where I is source image, T is template image with resolution $w \times h$ [21]. Template matching well suits dual-fisheye images, since the overlapped regions are nearly identical except for minor shift and/or perspective difference.

However, a complete template matching between two images is highly time-consuming and less reliable because of outliers. To solve these problems, we designed a preselection of interest points using Shi-Tomasi corner detector [4], an improved version of Harris corner detector [22]. The image patches around strong corners determined by the algorithm would be the templates for template matching mentioned above.

C. Homography Calculation

For image stitching or panorama creating, a common practice is to warp images using a matrix H called homography. The equation below maps the pixel (x_s, y_s) on the source image to the pixel (x_d, y_d) on the destination image.

$$[x_d \ y_d \ 1] = \frac{[x_s \ y_s \ 1] \times H}{H_{13}x_s + H_{23}y_s + 1} \\ \text{where } H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{21} & H_{22} & 1 \end{bmatrix}$$

Theoretically, homography can be calculated from four pairs of points, since H has 8 degrees of freedom. If arbitrarily picking out four pairs of points from the result of last section, H may be incorrect due to potential outliers. Therefore, Random Sample Consensus (RANSAC) is used here to detect and remove outliers [5]. Then the calculation result would be most likely to be correct.

After calculation of homography, the lower image (Fig. 4d) is warped using the equation above, with bilinear interpolation to determine intensity of each pixel. The result image is shown in Fig. 7a.

D. Blending and Cropping

After image warping, overlapped pixels are to be blended together to achieve a smooth transition between images. Direct averaging or feathering of overlapped pixels is not a good practice. They both result in ghosting artifacts, which happens due to moving objects, parallax, etc.

1) Optimal seamline: Our solution to it is to find the optimal seamline between warped images. This can be done by using graph cuts, a computer vision technique that sets up weights between pixels (nodes) to build a graph and finds a min-cut [23]. Graph cut is extremely time consuming for high resolution videos. This paper rather employs a fast, simple and yet effective dynamic programming method called minimum error boundary cut algorithm [6]. The details are illustrated by Fig. 6.

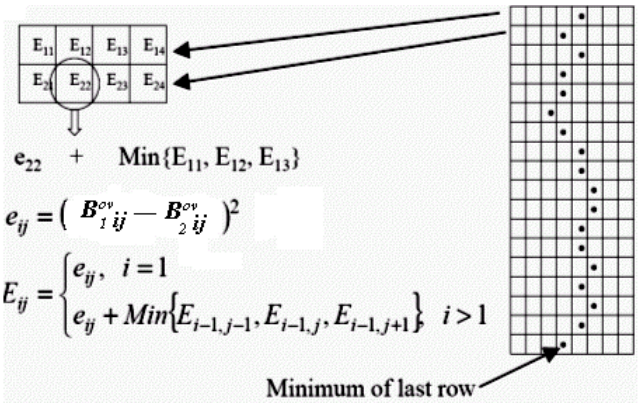


Fig. 6. Minimum error boundary cut algorithm [6]. B_1^{ov} and B_2^{ov} are the overlapped regions on the two images; e is squared error of pixel intensity; E is the energy computed in a dynamic programming manner. The pixel with minimal E of each row is determined as the boundary pixel of that row.

The optimal seamline computed by this algorithm is local optimum of each frame. The seamline may vibrate during video playback. To let the seamline coherent from frame to frame, we modified minimum error boundary cut algorithm, so that the output becomes a weighted combination of the original result and the results from previous frames.

2) Multi-band blending: There might exist inconsistency of pixel intensities across the seamline between warped images. There are multiple possible reasons for that. For example, differences between fisheye cameras in terms of incident light direction, exposure/white balance settings, or even cameras themselves. To smooth the abrupt change of intensities and at the same time avoid blurring, we applied multi-band blending [7] to the two images along with their seamline. This blending method decomposes an image into a low frequency band and a high frequency band. Effectiveness of this method is shown in Fig. 7b.

(a)

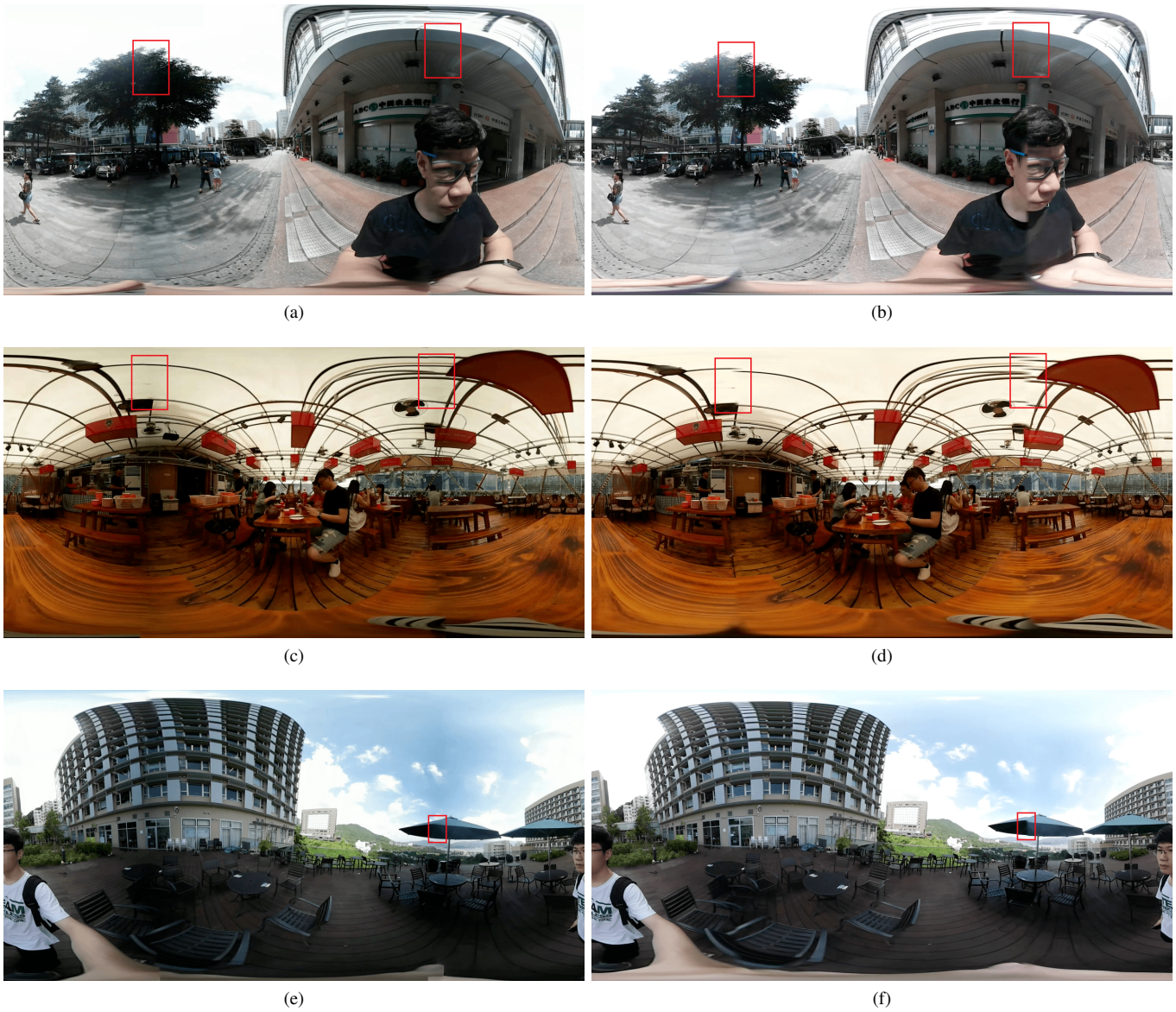
(b)

(c)

Fig. 7. (a) Warped. (b) Blended. (c) Cropped.

Note that before blending, the invalid (empty) area of each image is filled up with pixels from another image, so as to avoid darkening effect (intensity lost) around the seamline.

3) Cropping: Warping using the homography calculated in III-C may result in some invalid area where no pixel from any image resides, as shown in Fig. 7b. The invalid area usually falls at the top or bottom of the warped image in the case H is correct. The final result is supposed to be the largest valid rectangular image cropped from the picture. It is achieved by recalculating upper and lower boundary coordinates after warping using the homography H, and cropping the picture accordingly. After that, the picture is resized to a desired resolution. The final result is shown in Fig. 7c.



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

Fig. 8. Left column: results generated by our method. Right column: results generated by the official tool Samsung Gear 360 Action Director [3].

IV. Implementation and Results

An implementation of our method proposed above was written in Python 2 using OpenCV 3.3 library [24], and well tested on Ubuntu 16.04 64-bit version. We used Samsung Gear 360 (2016) [17] to record 2560×1280 dual-fisheye videos, and created 360-degree videos of the same resolution with FOV set to 194 degrees, which could be viewed on YouTube or other video players that support 360-degree videos. Fig. 8 demonstrates the performance of our method, compared to the stitching results generated by Samsung's official tool Gear 360 Action Director [3].

V. Discussion

Because of the nature of image stitching problem, the performance must be evaluated by manual checking of human eyes.

As shown in Fig. 8 and Fig. 9, our method produced better results than Samsung's official tool, in terms of alignment and consistency of pixel intensities around seamlines.

We speculate that Samsung's official tool does not apply feature matching to images, rather than that, the tool probably employ a fixed handcrafted mapping function plus some simple blending method around seamlines. Samsung's researchers' approach is understandable because it is computationally efficient, and a prior knowledge of their cameras' hardware parameters might help them build a delicate mapping function that suits most cases. However, this approach is not adaptive to scenes and therefore may cause misalignment or inconsistency of pixel intensities around seamlines, as shown in Fig. 8 and Fig. 9.

Our method still has a problem of unstable seamlines over time. Even though the modified version of optimal seamline algorithm introduced in III-D1 alleviates the situation, the

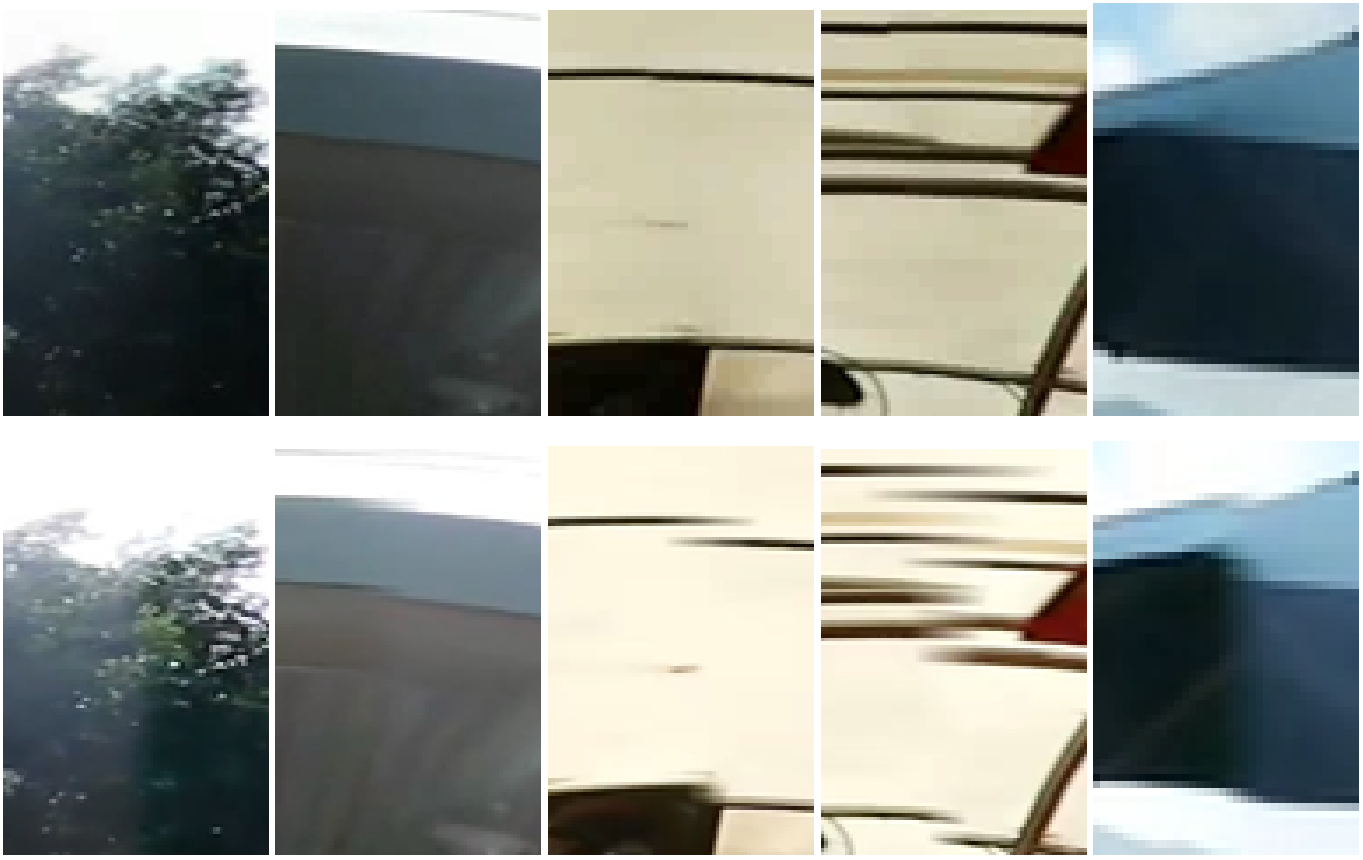


Fig. 9. First row: image patches from Fig. 8's left column (generated by our method). Second row: image patches from Fig. 8's right column (generated by Samsung's official tool).

vibration of seamlines still exists, especially for dynamic scenes.

So in general, compared to Samsung's official tool, our method consumes a bit more time, but in return gets adaptive to scenes and generates better 360-degree videos without knowing cameras' hardware parameters beforehand.

VI. Conclusion

This paper proposes a general method for dual-fisheye video stitching, consisting of de-warping, feature matching, homography calculation, optimal seamline, blending and cropping. Our method is a balance between computation complexity and performance, and produces better results than Samsung's official tool.

There are several potential improvement directions of our method in the future.

- Restricted homography calculation dedicated to fisheye stitching, to prevent some incorrect image warping results;
- A better algorithm to determine optimal seamline of each frame and reduce noticeable seamline vibration at the same time;
- 360-degree video stablization, using optical flow [25] or other techniques.

Acknowledgment

The author would like to thank Prof. Hanqiu Sun, Ph.D. Zhensong Zhang for their guidance and Faculty of Engineering, CUHK for providing this research opportunity and fundings.

References

[1] Orah. Videostitch studio. [Online]. Available: https://www.orah.co/software/videostitch-studio/

[2] Kolor. Autopano video. [Online]. Available: http://www.kolor.com/autopano-video/

[3] Samsung. Gear 360 action director. [Online]. Available: https://resources.samsungdevelopers.com/Gear_VR_and_Gear_360/Gear_360/02_Download_Gear_360_Action_Director

[4] J. Shi and C. Tomasi, "Good features to track," in *9th CVPR*. Springer, June 1994.

[5] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," SRI International, Tech. Rep. 213, March 1980.

[6] M. F. Abdul-Halim and N. A. Ibraheem, "Utilizing genetic algorithms for 2d texture synthesis," 2015.

[7] P. J. Burt and E. H. Adelson., "A multiresolution spline with application toimage mosaics," *ACM Tran. on Graphics*, no. 2(4), 1983.

[8] Y. Xiong and K. Turkowski, "Creating image-based vr using a self-calibrating fisheye lens," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, June 1997.

[9] R. Szeliski, *Image Alignment and Stitching: A Tutorial*. Now Publishers Inc, 2006.

[10] R. V. Frank A. van den Heuvel and B. Beers, "Calibration of fisheye camera systems and the reduction of chromatic aberration."

[11] Y. W. Xiaoming Deng, Fuchao Wu and C. Wan, "Automatic spherical panorama generation with two fisheye images," in *7th World Congress on Intelligent Control and Automation*. IEEE, June 2008.
[12] S. Hu and X. Feng, "Calibration of panorama based on linear ccd and fisheye lens," in *2nd International Congress on Image and Signal Processing*. IEEE, October 2009.
[13] Wikipedia. Azimuthal equidistant projection. [Online]. Available: https://en.wikipedia.org/wiki/Azimuthal_equidistant_projection
[14] ——. Spherical coordinate system. [Online]. Available: https://en.wikipedia.org/wiki/Spherical_coordinate_system
[15] ——. Stereographic projection. [Online]. Available: https://en.wikipedia.org/wiki/Stereographic_projection
[16] ——. Equirectangular projection. [Online]. Available: https://en.wikipedia.org/wiki/Equirectangular_projection
[17] Samsung. Gear 360 (2016) - samsung. [Online]. Available: http://www.samsung.com/us/support/owners/product/gear-360-2016
[18] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.
[19] M. Muja and D. G. Lowe. Flann - fast library for approximate nearest neighbors. [Online]. Available: http://www.cs.ubc.ca/research/flann/
[20] T. Ho and M. Budagavi, "Dual-fisheye lens stitching for 360-degree imaging," in *ICASSP*, 2017.
[21] OpenCV. Template matching. [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html
[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988.
[23] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *Proceedings of ICCV*, July 2001.
[24] OpenCV. Opencv library. [Online]. Available: http://opencv.org/
[25] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of Imaging Understanding Workshop*, 1981.