CAAI Transactions on Intelligence Technology

Research Article

Stable stitching method for stereoscopic panoramic video

ISSN 2468-2322
Received on 2nd January 2018
Revised on 21st January 2018
Accepted on 15th February 2018
doi: 10.1049/trit.2018.0002
www.ietdl.org

Zhuo Tan, Shenghao Zhang, Ronggang Wang ✉

School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, People's Republic of China
✉ *E-mail: rgwang@pkusz.edu.cn*

Abstract: This paper proposes a stable method of generating a stereoscopic panoramic video with the omnidirectional stereo (ODS) format. Different from the traditional image stitching method which can only be applied to generate a monocular panorama, we adopt an optical flow-based blending method to create two panoramas for binocular vision. In addition, traditional image stitching methods based on seam-finding tend to cause the problem of temporal flicker. We address this problem by restricting the optical flow field of the new frame with its previous frame's optical flow field. Thus, the generated video is stable and out of temporal flicker. There are four key operations in our approach. First, we adopt the ODS format which is the basis of stereoscopic panorama. Second, we do effective exposure compensation, making the brightness of the two eyes' panoramas consistent. Third, we employ an optical flow-based blending method to synthesis the final panorama effectively. Fourth, we take the previous frame's optical flow field as the restriction of the present frame's optical flow field to acquire a stable video. The final output videos can deliver a pleasant and impressive stereoscopic viewing experience to the audience when the audience watches the videos in the virtual reality headset.

1 Introduction

With the virtual reality (VR) headsets becoming more and more popular, the demand for immersive videos grows. However, the immersive video is hard to be directly captured with a traditional single camera. Thus, the efficient production of the immersive video content is becoming a new issue to be addressed [1]. This study focuses on how to stitch a stereoscopic panoramic video content captured with multiple cameras. A stereoscopic panorama consists of a pair of panoramic videos, where one panorama is for the left eye, and another panorama is for the right eye.

There are quite a lot of research studies on image stitching in the past ten years. Also, many tools and commercial software are available to produce panoramas. However, almost all these tools can only create a monocular panorama. By adopting a monocular image stitching method to separately produce the panorama for the left eye and right eye at the same time will not get the correct stereo video at some view angle.

To overcome the above limitation, this study proposes an improved method to generate a stereoscopic panoramic video with the omnidirectional stereo (ODS) format [2–6]. Fig. 1 shows a brief sketch of how to generate a panorama of the ODS format. The input images are captured with a normal camera rotating about an axis behind it.

In the ODS format, a stereoscopic panorama is generated by stacking specific strips extracted from the input images representing different viewpoints horizontally. Since each strip is a narrow part of a perspective input image, the panorama of the ODS format retains the parallax effect and naturally multi-perspective. Although the ODS format is a good solution to the stereoscopic panorama, there still exist two unsolved problems in creating ODS video content.

First, we need to design proper camera system to obtain input images for a panoramic video. The ODS can just stitch a panorama with a series of images capturing by rotating a camera on a rotating arm. If we want to generate stable and continuous panoramic video content, we need a camera system with no moving parts namely a set of fixed cameras. We choose a circular rig with 10 camera positions (see Fig. 2). Different from the ODS method, each panorama is stitched from several input images captured at the same time.

Second, we need to design an efficient algorithm to generate a stereoscopic panoramic video with video sequences of only 10 viewpoints. As mentioned above, the method to generate a panorama of the ODS format is extracting vertical narrow strips from the input images (if we rotate the camera to capture a video, the input images are all the frames of this video) and then stacking them horizontally together to generate the final panorama. This is not applicable to our condition.

In this study, we adopt the Facebook Surround360 framework [7], which warps the images with an optical flow field to simulate a rotating camera. We do two main improvements over Facebook Surround360. First, we add a step of effective exposure compensation, which allows the following computation of optical flow field more accurate. In addition, this step makes the brightness of the left-eye and right-eye panoramas consistent. Second, to overcome the temporal flicker of the output video sequence, we restrict the optical flow field of the new frame with its previous frame's optical flow field. Experimental results testify the effectiveness of our proposed method.

The structure of this paper is shown as follows. Section 2 gives an overview of the related work. Section 3 presents our proposed method of generating a stereoscopic panoramic video. Section 4 provides the experimental results and this study is concluded in Section 5.

2 Related work

A stereoscopic panoramic namely 3D-360° video contains three key points. First, we need to stitch a circular 360° panorama from a series of input images. Second, we need to generate panoramas for both eyes at the same time to provide the feeling of immersion and stereopsis for video audience. Finally, we need to make the video

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

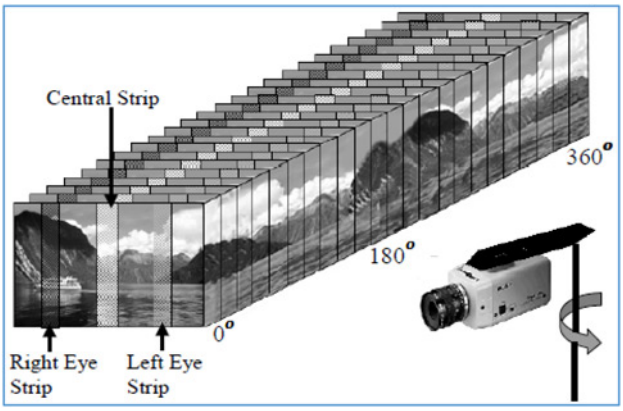

Fig. 1 *Process to generate a stereoscopic panorama of the ODS format. A stereoscopic panorama can be generated with the images captured with a normal camera rotating about an axis behind it. The final panorama consists of a series of strips taken from each image [6]*

Fig. 2 *Our camera rig. There are ten camera positions on the side ring. To get the top and bottom of a panorama, we can choose to place another four cameras*

stable and continuous with a proper stitching method applied to every frame.

2.1 Image stitching

The first key point of the stereoscopic panoramic video is panorama stitching. Panorama stitching is a special case of image stitching with a view field of 360°. Image stitching is the process of combining multiple photographic images where overlapping regions exist to create a high-resolution image with a wide field of view.

An early image stitching method called AutoStitch with good performance under most conditions was proposed by Brown. The AutoStitch finds global homographies to bring images into alignment and blend them [8–12]. Nowadays, most commercial software is based on AutoStitch, such as the stitcher module of OpenCV, ICE by Microsoft, Photoshop, Autopano, and so on. However, the approach of AutoStitch is not robust sometimes. The approach of AutoStitch makes use of the homography which is a 2D projective transformation to align input images. Since the camera image is a 2D projection of the 3D real world scene, the homography can just align some part of the image on the same plane in the real world well. The usage of the 2D homography in alignment tends to cause artefacts or 'ghosting' in the final results.

In recent years, there is great progress in the field of image stitching. To improve the alignment quality which is limited by the usage of a global homography, the as-projective-as-possible (APAP) warp was proposed [13]. The main idea of APAP is to mesh cutting and deformation. Instead of a global homography, APAP computes homographies for each mesh grid for more accurate alignment of input images.

Following APAP, there were many excellent works. Chang *et al.* proposed the method of shape-preserving half-projective (SPHP) [14]. The main idea of SPHP is the combination of the projective transformation and similarity transformation. In addition to the usage of mesh cutting, SPHP does space division of the overlapping region and the non-overlapping region to process them separately. With mesh cutting and homography, SPHP obtains a good alignment in the overlapping region. With the usage of the extra similarity transformation in the non-overlapping region SPHP preserves the objects' shape of the initial perspective projection, thus reduces the distortion caused by mesh cutting.

Zhang and Liu presented a local warping stitching method combining global homography and content-preserving warping (CPW) [15, 16]. This method can deal with large displacements due to the parallax effect between adjacent images well. After an optimised global alignment with improved homography, local mesh grid deformation is done with several constraint conditions of practical physical significance. In their paper, they defined three constraint conditions including the key point matching constraint, the non-overlapping region constraint, and the smoothness constraint. Later works concentrated on the strict and robust constraint conditions.

2.2 Stereoscopic panorama

The second key point of the stereoscopic panoramic video is a stereoscopic panorama. We need to generate both the left-eye and the right-eye panorama. Unfortunately, these methods mentioned above are only suitable for monocular panorama stitching. Such panoramas lack parallax and cannot provide a stereoscopic and immersive experience. As mentioned above in Section 1, adoption of a monocular image stitching method to separately produce the panorama for the left eye and right eye at the same time will bring not convincing results.

To overcome this limitation, Zhang and Liu presented a method for stitching stereoscopic panoramas from stereo images casually taken using a stereo camera [17]. The advantage of using a stereo camera is that the input images are stereoscopic inherently. Then we just need to stich the left-eye panorama with the left-eye input images and stich the right-eye panorama with the right-eye input images. The only difficult is the consistency of the two panoramas since their panorama stitching method is based on seam-finding. They used the optical flow field to restrict the generation of the right-eye panorama to keep the consistency. Even though the results of this work look perfect, the method only stitches one 3D panorama with a series of images captured by moving a stereo camera.

Not using a stereo camera, a panorama of the ODS format is generated by stacking specific strips extracted from the input images representing different viewpoints horizontally. Since each strip is a narrow part of a perspective input image, the panorama of the ODS format retains the parallax effect and naturally multi-perspective (see Fig. 3).

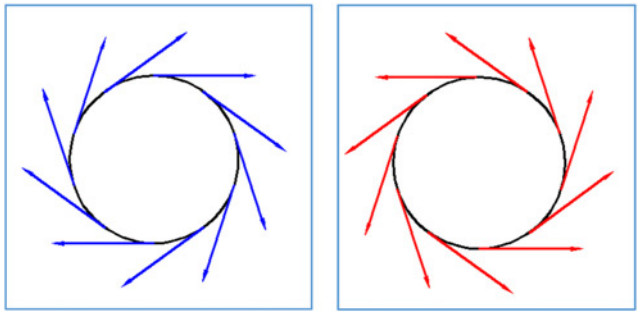

Fig. 3 *Projection of the ODS. Every ray tangent to the viewing circle indicates looking out from a specific viewpoint. Blue and red rays represent the view direction of the left and right eyes, respectively*

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

2.3 Video stitching

The third key to the stereoscopic panoramic video is video stitching. In general, joining all frames stitched together directly will cause the problem of video jitter and flicker. There have not been many valuable research results in video stitching up to now.

He and Yu present a parallax-robust video stitching technique [18]. However, their approach based on seam-finding is specifically aimed at the surveillance video stitching. Their method cannot generate the stereoscopic panorama degrees and remove the problem of flicker.

Huawei Media Lab contributes a novel method of video stitching with spatial-temporal CPW [19]. Their method extends the step of seam-finding from 2D to 3D. Thus, they can generate a stable video out of flicker. However, their method using mesh cutting and seam-finding cannot generate the stereoscopic panorama and is very time-consuming.

Some other works of video stitching concentrate on the shakiness removing [20, 21] caused by the moving of cameras when capturing videos. That is irrelevant to the problem of flicker. Since we adopt an optical flow-based blending method, we perform restriction on the computation of the optical flow field within time domain to generate a stable video out of flicker.

2.4 Simulation of the ODS with viewpoint interpolation

To allow generating a panoramic video, we use ten cameras placed uniformly in a rig and perform interpolation of viewpoints between adjacent cameras to simulate a rotating camera. The part will introduce the strategy of simulating the ODS with viewpoint interpolation and the sketch map is shown as Fig. 4.

We use the spherical or equirectangular projection as the format of our panorama. In the spherical or equirectangular projection, each column of the projection image corresponds to a particular longitude of a sphere; each column of the projection image corresponds to a particular latitude of a sphere. It is convenient to recover the 3D effect in the VR head set with equirectangular projection. In addition, a specific longitude represents a specific orientation of the audience's head.

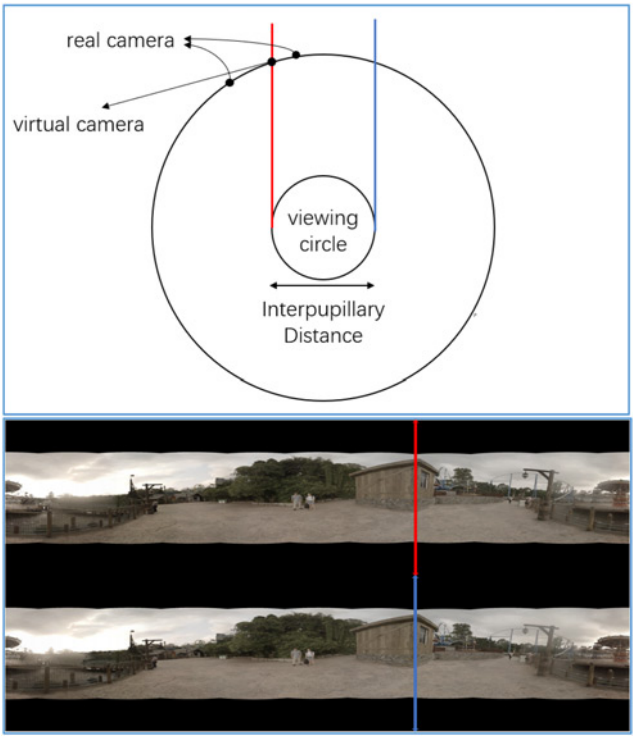

Fig. 4 *ODS is multi-perspective. If a ray to generate the panorama is virtual, we need to perform interpolation of viewpoint*

Since the interpupillary distance of people is generally about 6 cm. We can construct the viewing circle whose diameter is the interpupillary distance. Then the point in the viewing circle is a viewpoint. Looking out at a viewpoint, the pixels of the ray originates the viewpoint and tangents the view circle is what we need.

Imaging that we look out is around full to 360° in the viewing circle. At a specific viewpoint, the pixels of the two rays' tangent the viewing circle are what we need to create the stereoscopic panorama. If a ray passes through a real camera, we can directly get pixel information from the captured image.

While in most cases, the ray intersects the camera circle at a position of no camera. Thus, we perform interpolation of viewpoint to assume a virtual camera between the two actual cameras and get the pixels of the ray with the interpolation method.

3 Our approach

In this section, we describe our proposed approach in detail. The first four parts (A, B, C, and D) give the pipeline of generating a panorama based on the optical flow field. The part E gives our proposed strategy to make the video stable and out of temporal flicker.

3.1 Calibration

We make use of the OCamCalib Toolbox contributed by Davide Scaramuzza [22–24] to calibrate the intrinsic parameters and the relative pose of the individual cameras in the rig. This calibration toolbox is easy to use and more accurate than regular tools such as OpenCV. It can be used for catadioptric and fish-eye cameras up to 195°.

GoPro Hero4 Black is used as the camera. The videos captured by the GoPro camera are severely affected by the fish-eye lens. Applying to each frame of the videos with the OCamCalib Toolbox, we can get the calibrated frames. The calibration example is shown in Fig. 5.

3.2 Projection

As mentioned in Section 2.3, we need to build equirectangular projections of each input image out of fish-eye influence. By performing key point matching between all adjacent camera images, we can compute the extrinsic parameter matrixes of all cameras in the rig. Then with the extrinsic parameter matrixes, we can project the input images out of fish-eye effect.

After the equirectangular projection, we can easily cut the overlapping region between adjacent input images according to the horizontal field angle of the camera and the number of the camera.

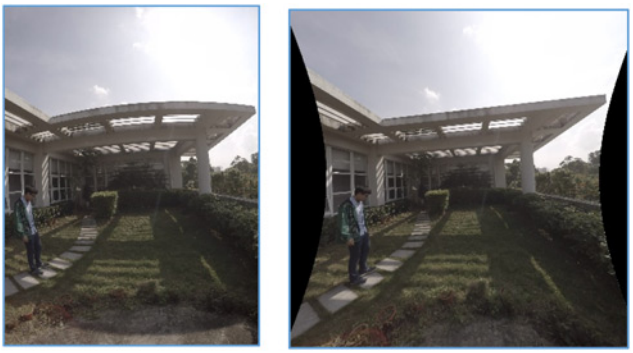

Fig. 5 *Left image is one frame of a video captured by a GoPro camera. The fish-eye influence is obvious that the straight lines of buildings are curving. Applying the OCamCalib Toolbox, we get the right image out of fish-eye influence. Apparently, all the straight lines are put back to normal, the scene is nature and confirms the reality*

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

The horizontal width of each overlapping region is

$$\text{overlapping width} = \frac{(\text{fov} \times n - 2\pi)}{n \times 2\pi} \times \text{cols}, \quad (1)$$

where fov is the horizontal field angle of a side camera, n is the number of side cameras, cols is the width of an input image. For our rig and camera model, $n = 10$, fov $= 106.26°$. The horizontal field angle involved in computing is smaller than the real horizontal field angle of the GoPro Hero4 Black camera because we cut much marginal region lacking texture (the residual black region shown in Fig. 5) during the calibration. The overlapping region of two adjacent images is shown in Fig. 6.

3.3 Exposure compensation

Since the exposure time of all cameras in the rig is independent. The adjacent camera images may have a very different brightness over a wide range. The regions in the same position of the left-eye and right-eye panoramas may have different brightness, which makes it difficult for the audience to fuse the imagery when viewing the stereoscopic panorama in the VR headset. It is necessary to perform exposure compensation before the blending step [12, 25].

Concretely, we estimate a gain parameter for each image by minimising an error function. The definition of the error function to be minimised is

$$\text{error} = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(g_i I_i(u_i) - g_j I_j\left(u_j\right)\right)^2 + \delta(1 - g_i)^2, \quad (2)$$

where g_i is the gain parameter of the image i, $I_i(\text{Region})$ is the average brightness of region in image i, u_i is the overlapping region between images i and j, $\delta = 0.001$ is a small value which makes the result stable. It makes all gain parameters close to 1.

3.4 Optical flow-based blending

3.4.1 Geometric theory: In Section 1, we introduce the ODS format. The ODS panorama consists of a series of strips with different viewpoints. We extract specific strips from each image and combine them into a left and right output view. The omnidirectional effect is achieved by collecting rays that are all tangent to a common viewing circle.

If we rotate a camera about an axis behind it, we can make the width of strip even small to one pixel. In fact, if the width of the strip is not enough small, there will be obvious seams between adjacent strips in the resulting panorama. However, under our condition, we only have 10 cameras in the ring. So, we cannot extract the strips with a width of 100 pixels, just following the operation of ODS. Here we describe the geometric theory of our blending method.

Since we have a small number of cameras on a ring (we denote the number of cameras by n), we can fix n viewpoints on the view circle. For each eye, there are only n real rays which originate from viewpoints and go through the corresponding cameras (see Fig. 7*a*). We can obtain only n real rays for the final panorama directly. The rest virtual rays between adjacent real rays need to be synthesised.

As shown in Fig. 7*b*, we let A and C be the intersections of the spherical plane with two real rays through the cameras and tangential to the view circle. All the rays of arc section AC need to be synthesised. Considering an object at any distance in the real world, P^1 and P^2 are, respectively, the projective positions of the two adjacent images. We can interpolate the intermediate point P between points P^1 and P^2 to synthesise the virtual ray crossing the virtual camera.

Since an object's projection in two adjacent camera images is, respectively, P^1 and P^2. In general, the difference between P^1 and P^2 is related to the depth of the object in the scene. We choose to use optical flow-based blending to synthesise these virtual rays. We denote the optical flow field from the left image to the right image by F. Since the optical flow field represents the point correspondences of two images, the point P^2 can be expressed as $P^2 = P^1 + F(P^1)$. We need to map P^1 and P^2 to P at the same time. Therefore, we require the paired optical flow fields namely the optical flow field from the left image to the right image and the optical flow field in the opposite direction [2] at the same time.

3.4.2 Computing optical flow and blending: In this part, we introduce the specific operation to synthesise the virtual rays. We have obtained the overlapping region of the two adjacent images shown in Fig. 6. Denoting a pair of overlapping region images by Image_l and Image_r, we compute the paired optical fields of the two overlapping region images. We denote the paired optical flow fields between any pair of adjacent overlapping region images l and r by F_{l_r} and F_{r_l}. How to use the two optical flow fields to warp the original images and combine the warped images is a difficult problem.

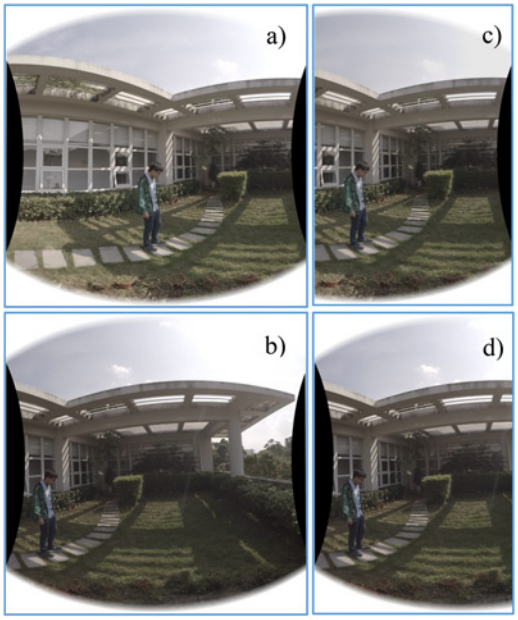

Fig. 6 *By computing the width of the overlapping region, we can extract the overlapping regions of the two adjacent equirectangular projective images*
a Left image of *b*
b Equirectangular projection of Fig. 5
c, *d* Extracted results

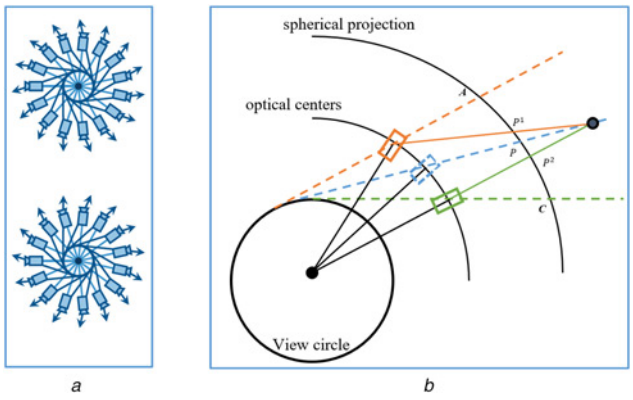

Fig. 7 *The virtual ray corresponded to the virtual viewpoint needs to be synthesised*
a Real rays of the ODS format [2]. In the ODS format, every ray represents the observation in a certain viewpoint
b Chunk namely arc section AC in the ODS format. Points A and C are the projections of the real rays, point P needs to be interpolated by some means

CAAI Trans. Intell. Technol., 2018, Vol. 3, Iss. 1, pp. 1–7
This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

Mentioned earlier in previous part and shown in Fig. 7*b*, we need to synthesise all the rays in the arc section AC. We define the arc section AC as a chunk of the final panorama. The number of chunks of the final panorama is equal to the number of cameras. So, the width of arc section AC is smaller than the width of the overlapping region and a chunk is only part of a pair of overlapping region images (see Fig. 8).

To keep the continuity at the boundaries of the chunks in the final panorama, we can set the distance from every pixel point to the chunk boundary as the weight of the pixel point's optical flow. Then, we use the weighted optical flow fields to warp the chunk part of original overlapping region images. Denoting the horizontal distance between the point x and the left boundary of the chunk by d, the two warped chunks are

$$wC_l(x) = C_l\big(x + d \times F_{l_r}(x)\big), \tag{3}$$

$$wC_r(x) = C_r\big(x + (1-d) \times F_{l_r}(x)\big). \tag{4}$$

With this, linearly blending of a chunk between two adjacent images gives our final optical flow-based blending results as

$$\text{Chunk}(x) = (1-d) \times wC_l(x) + d \times wC_r(x). \tag{5}$$

Since we use the distance as the weight of the warped chunk, the final chunk's left and right boundaries are equal to the real rays, keeping the continuity in the final panorama.

3.5 Stable video without temporal flicker

Now, we can successfully generate a frame of the stereoscopic panorama. However, if we directly produce a video just by joining all frames together, the video may be unstable. The audience will feel the video jitter when watching it in the VR headset.

Traditional image stitching methods based on seam-finding also lead to the temporal flicker because the seam of every frame varies much. Fortunately, our blending method is based on the optical flow field. Therefore, we can constrain the optical flow field of the present frame with the previous frame's optical flow field.

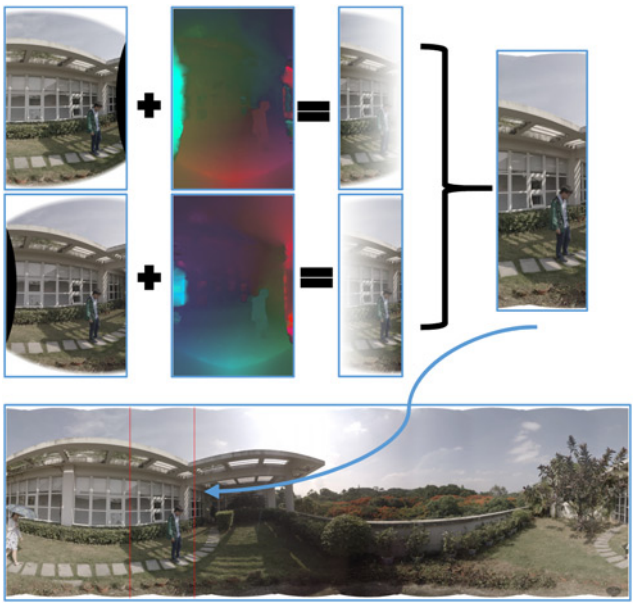

Fig. 8 *Process of generating a chunk of the final panorama. We first calculate the pair of optical flow field of the two overlapping region images. Then, we use the optical flow field to warp the original image to get the warped chunks (both the left and the right). Finally, we get the optical flow-based blending chunk. The chunk is part of the final panorama. Every pair of overlapping region images can extract a chunk. We stack all chunks horizontally together to get the final panorama*

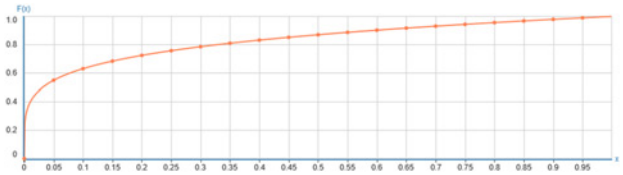

Fig. 9 *How the weight of the present frame's optical flow field changes with the difference between two frames. When two frames are similar, the present optical flow field is also like the previous. When the present frame changes a lot against the previous frame, the weight of previous optical flow field need to be very small*

We choose to compute a weighted summation of the previous frame's optical flow field the present frame's optical flow field as a new optical flow field providing for the present frame. The weights of the two frames' optical flow field need to be carefully chosen. To reduce or remove temporal flicker, the present frame's optical flow field should be as much as possible as the previous frame's optical flow field if the two frames are similar. However, if there is a great difference between two frames, the restriction on the present frames' optical flow field should be weak. Otherwise, the quality of the final panorama is not good enough.

Thus, we use a power function to compute the weight (see Fig. 9). The horizontal axis represents the difference between two frames. The vertical axis represents the weight of the present frame's optical flow field. When two frames are similar, the present optical flow field is also like the previous. When the present frame changes a lot against the previous frame, the weight of previous optical flow field is set to be very small.

We compute the difference at each pixel. The weighed summation is also computed at every pixel. We first add the subtractive results of three channels together then perform normalisation by dividing 255 (8 bits for every channel of a pixel point). Then we calculate the weight using the power function. The index of the power function $\gamma = 0.3$ in our experiment. Finally, we get the weighted summation

$$\text{Diff}(x) = \frac{\left(\sum_{\text{all chanel}} I_1(x) - I_2(x)\right)}{3 \times 255}, \tag{6}$$

$$w(x) = \text{Diff}(x)^{\gamma}, \tag{7}$$

$$F_{\text{new}}(x) = (1 - w(x)) \times F_1(x) + w(x) \times F_2(x). \tag{8}$$

4 Experimental results and analysis

We tested our approach of generating a stereoscopic panoramic video on various scenes with a widely divergent background. We capture videos using our rig with 10 side cameras (the camera type is GoPro Hero4 Black). In most cases, the performance of our approach is excellent.

The resolution of the captured video depends on the setting. In general, we set the resolution to 2704×2028 or 1920×1440. Then the resolution of input images after calibration for the panorama stitching is proper to set to 1920×1920. Thus, the resolution of the final panorama can be set to 4, 6 or 8 K.

Fig. 10 shows the striking effect of our exposure compensation. Fig. 10*a* is the result of Surround360 without our exposure compensation and Fig. 10*b* is the result of our method. Since every camera' internal exposure time is different and cannot be easily controlled and adjusted to the same artificially (in agreement with the GoPro Hero4 Black), the overall brightness of the stitched panorama is not uniform. In addition, the brightness of the left eye's panorama differs greatly from the other one in the same region, causing hard fusion when viewing it in the VR headset. Applying our exposure compensation, we address the two

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

problems of brightness and the subjective experience of the stereoscopic panorama is much better.

Fig. 11 shows the video quality comparison between Surround360 and our method. Fig. 11*a* is the right-eye panorama of one frame stitched by Surround360. Since the man and the roller coaster is fast moving in the scene. There are artefacts in the frame (see Fig. 11*b*). With our method, the frame is well stitched out of any artefacts (see Figs. 11*c* and *d*).

Fig. 12 shows the left-eye panoramas of some scenes. We have watched our results in the VR headset, there is no temporal flicker and we can get a sense of immersion and stereo perception.

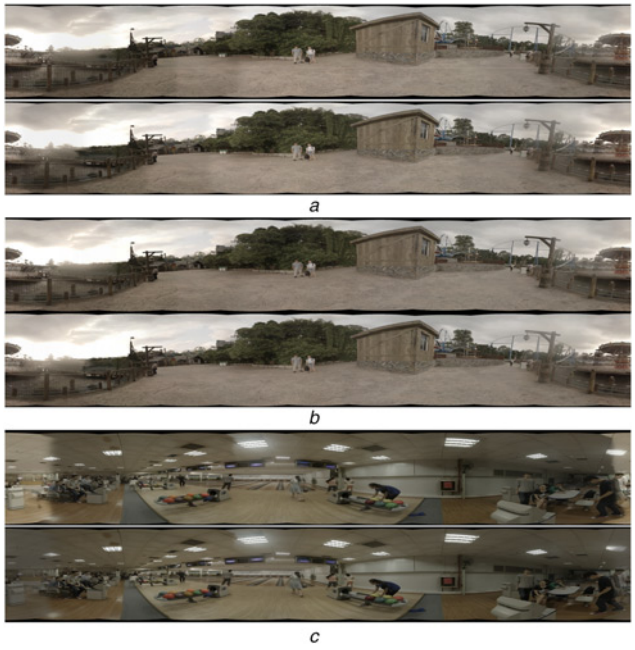

Fig. 10 *Striking effect of our exposure compensation*
a Result of Surround360
b Result of our method
c Contrast on another scene. *Our exposure compensation greatly improves the subjective*

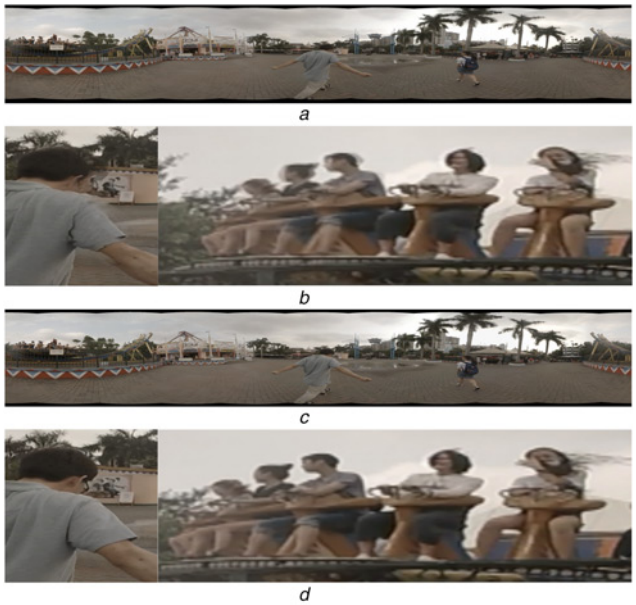

Fig. 11 *Video quality comparison between Surround360 and our method*
a, b Stitched by Surround360. The man and the roller coaster are fast-moving in the scene, causing the phenomenon of ghosting
c, d Stitched by our method. We can zoom in to see details. There does not exist any hosting artefacts

Fig. 12 *Left-eye panorama taken from four different scenes. Quality of the panorama is nearly perfect. In addition, we can get the feeling of immersion and stereo perception when watching the panoramic video in the VR headset*

In some special cases, the quality of the stereoscopic panoramic video is still not good enough.

In the case when a large-scale object in the foreground and this object are far away to those background objects, the occlusion occurs. The parts of the background objects which are blocked by the foreground object are different. Then the overlapping regions of the adjacent images are not consistent. The following steps including computation of optical flow field and image blending are not accurate as well. Finally, the final panorama is blurry in some regions. The occlusion problem also occurs when the object is too closer to the rig.

In the second case, there is an object with very fast movement on the scene. The definition of fast movement on the scene indicates that the position of the object changes significantly in two adjacent frame images. In fact, an object with a high speed in the real world may be slow in the video if it is far away from the camera. Furthermore, an object with a low speed in the real world may be fast in the video if it is very close to the camera. In this case, there is a notable discrepancy between adjacent two frames. Thus, the restriction of the optical flow field may fail to remove temporal flicker and keep the video stable.

5 Conclusion

This study demonstrated a method of creating a stereoscopic panoramic video of the ODS format. We adopt an optical flow-right-eye panorama at the same time. Exposure compensation is proposed to make the brightness of the final panorama uniform and the brightness of the two eyes' panoramas consistent. In addition, we take the previous frame's optical flow field as the restriction of the present frame's optical flow field to create a stable video. Experimental results show that our approach is

quality of the stitched stereoscopic panorama

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

effective. Also, we analyse the failure cases of our approach. In the future work, we will proceed to optimise the optical flow algorithm and blending method to deal with the occlusion problem.

6 Acknowledgments

We thank the National Natural Science Foundation of China 61672063, Shenzhen Peacock Plan, Shenzhen Research Projects of JCYJ20160506172227337 and GGFW2017041215130858 for funding.

7 References

[1] Anderson, R., Gallup, D., Barron, J.T., *et al*.: 'Jump: virtual reality video', *ACM Trans. Graph.*, 2016, **35**, (6), p. 198
[2] Richardt, C., Pritch, Y., Zimmer, H., *et al*.: 'Megastereo: constructing high-resolution stereo panoramas'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 1256–1263
[3] Zheng, E., Raguram, R., Fite-Georgel, P., *et al*.: 'Efficient generation of multi-perspective panoramas'. IEEE 2011 Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), Hangzhou, China, 2011, pp. 86–92
[4] Peleg, S., Ben-Ezra, M.: 'Stereo panorama with a single camera'. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 1999, vol. 1, pp. 395–401
[5] Ishiguro, H., Yamamoto, M., Tsuji, S.: 'Omni-directional stereo for making global map'. IEEE Proc. Third Int. Conf. on Computer Vision, Osaka, Japan, 1990, pp. 540–547
[6] Peleg, S., Ben-Ezra, M., Pritch, Y.: 'Omnistereo: panoramic stereo imaging', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001, **23**, (3), pp. 279–290
[7] The Facebook Surround360 Project. Available at https://github.com/ facebook/ Surround360
[8] Faridul, H.S., Stauder, J., Trémeau, A.: 'Illumination and device invariant image stitching'. 2014 IEEE Int. Conf. on Image Processing (ICIP), Paris, France, 2014, pp. 56–60
[9] Thatte, J., Boin, J.B., Lakshman, H., *et al*.: 'Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax'. 2016 IEEE Int. Conf. on Multimedia and Expo (ICME), Seattle, WA, USA, 2016, pp. 1–6
[10] Chen, Y.S., Chuang, Y.Y.: 'Natural image stitching with the global similarity prior'. European Conf. on Computer Vision. Springer Int. Publishing, Amsterdam, The Netherlands, 2016, pp. 186–201
[11] Szeliski, R.: 'Image alignment and stitching: a tutorial', *Found. Trends® Comput. Graph. Vis.*, 2006, **2**, (1), pp. 1–104
[12] Brown, M., Lowe, D.G.: 'Automatic panoramic image stitching using invariant features', *Int. J. Comput. Vis.*, 2007, **74**, (1), pp. 59–73
[13] Zaragoza, J., Chin, T.J., Brown, M.S., *et al*.: 'As-projective-as-possible image stitching with moving DLT'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 2339–2346
[14] Chang, C.H., Sato, Y., Chuang, Y.Y.: 'Shape-preserving half-projective warps for image stitching'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 3254–3261
[15] Zhang, F., Liu, F.: 'Parallax-tolerant image stitching'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, 2014, pp. 3262–3269
[16] Liu, F., Gleicher, M., Jin, H., *et al*.: 'Content-preserving warps for 3D video stabilization', *ACM Trans. Graph. (TOG)*, 2009, **28**, (3), p. 44
[17] Zhang, F., Liu, F.: 'Casual stereoscopic panorama stitching'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 2002–2010
[18] He, B., Yu, S.: 'Parallax-Robust surveillance video stitching', *Sensors*, 2015, **16**, (1), p. 7
[19] Jiang, W., Gu, J.: 'Video stitching with spatial-temporal content-preserving warping'. IEEE Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 2015, pp. 42–48
[20] Nie, Y., Su, T., Zhang, Z., *et al*.: 'Dynamic video stitching via shakiness removing', *IEEE Trans. Image Process.*, 2018, **27**, (1), pp. 164–178
[21] Guo, H., Liu, S., He, T., *et al*.: 'Joint video stitching and stabilization from moving cameras', *IEEE Trans. Image Process.*, 2016, **25**, (11), pp. 5491–5503
[22] Scaramuzza, D., Martinelli, A., Siegwart, R.: 'A flexible technique for accurate omnidirectional camera calibration and structure from motion'. IEEE Int. Conf. on Computer Vision Systems, New York, NY, USA, 2006, pp. 45–45
[23] Scaramuzza, D., Martinelli, A., Siegwart, R.: 'A toolbox for easily calibrating omnidirectional cameras'. 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, China, 2006, pp. 5695–5701
[24] Scaramuzza, D.: 'Omnidirectional vision: from calibration to robot motion estimation'. PhD Thesis, ETH Zurich, 2008, p. 17635
[25] Xia, M., Yao, J., Xie, R., *et al*.: 'Color consistency correction based on remapping optimization for image stitching'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Venice, Italy, 2017, pp. 2977–2984

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)