Automatic stitching method for spherical panoramic video

Lihong Luo, Jianqing Mo*, Jiazhen Li
Digital Media Department, Guangdong University of Technology, Guangzhou, 510006, China
**Corresponding author: qinggdut@163.com*

Abstract

Panoramic video has become a hot research topic in the field of virtual reality in recent years. Existing research suggests that panoramic video cannot automatically be stitched yet. In addition, stitching accuracy and time consumption are not satisfactory. After analyzing the relationship among the several coordinate systems involved in shooting and images, this research puts forward a manual method and steps for stitching a panoramic video. Then a matching point searching algorithm based on Harris corner detection, empirical position and HSV matching is posited. With this algorithm, matching points are searched and found automatically, and panoramic videos are automatically stitched. Comparison experiments were carried out. The test results proved that the described automatic stitching algorithm for panoramic videos performs better than the alternatives in both accuracy and time consumption.

Keywords: Automatic stitching; Harris corner detection; matching point searching; mosaic; Panoramic video

1. Introduction

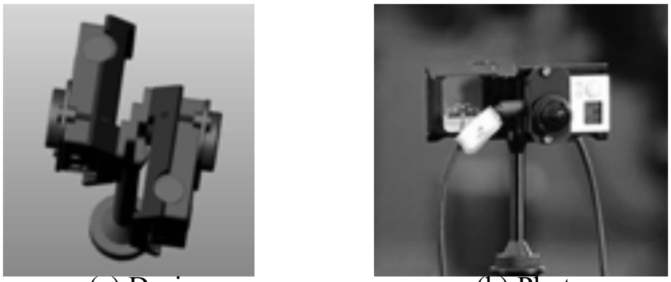
Panoramic video is a kind of special video, in which images can record information in all directions. Panoramic video can also support interaction. For example, viewing direction can be changed and viewing scene can be switched. Panoramic video is stitched together from several regular videos that are shot in different directions. It has many military and medical applications, and it is useful for computer vision, video conferencing, and security systems.

Panoramic video evolved from panoramic images. Methods for panoramic image mosaic (called stitching) have been studied widely. Three kinds of methods are often used: one based on frequency domain, another based on feature, and the other based on gray level and gradient. The multilayer fractional Fourier transformation approach (Pan *et al*., 2009; Yang & Cao, 2014) is a typical method based on the frequency domain. This method can solve the matching problem of translating, rotating and scaling, but it cannot solve the matching problem of perspective transformation. Scale-invariant Feature Transformation (SIFT) is a method that is based on features. This method can solve the matching problem of perspective transformation, but it is complex and very time-consuming, and it does not match texture well. The methods based on gray level and gradient calculate the transforming parameters by minimizing the differences in the gray level of the image. For example, Shum and Szeliski (2002) presented a mosaic method using global and local alignment, while Siavash and George (2005) put forward a method using log-polar mapping. These methods produce the transforming parameters using LM optimization. However, they are easily affected (and sometimes distorted) by variations in light. On the other hand, according to different forms of construction, panorama can be divided into three types: cylindrical, cubic and spherical. Cylindrical panorama has been the most widely studied (De Carufel & Laganiere, 2011; Xu *et al*., 2014). Many new technologies came out at first in the area of cylindrical panoramas. For example, an automatic stitching method was discussed in Yang and Cao (2014), and cylindrical panoramic video was researched in Xu *et al.* (2014). Research shows that cylindrical panorama has disadvantages. For example, its spatial information is not entire. It contains no top or bottom information. However, spherical panorama has entire information, but it is more complex than cylindrical panorama. In addition, there is less research on spherical panorama. The methods explained in Shum and Szeliski (2002) and Zhang *et al.* (2015) can be used in spherical panoramas. However, Shum and Szeliski's method uses many rows and columns of narrow angle photos and requires dozens of photos. It is inconvenient to shoot so many photos. Errors may accumulate and increase when using so many photos. Automatic stitching cannot often be carried out because of this. Since manual adjustment is needed when this occurs, the method is considered semi-automatic. The method described in Zhang *et al.* (2015) also uses a manual method, requiring the manual adjustment of stitching parameters in a template image. These parameters are used to stitch in the actual shots. Because stitching parameters cannot be modified during shooting, the image quivers. This occurs for various reasons (i.e., camera vibration), which cannot be eliminated. Hence, the image quality of this kind of stitched video is low. This study develops a method for creating a spherical panoramic video mosaic that is automatic, requires few photos and has a high quality. First, the shooting method is introduced, namely creating an image using two fisheye lens photos. Second, the relationship among the several coordinate systems in shooting and images is analyzed. Third, a manual method and the steps of stitching a panoramic video are introduced. Then a matching point searching algorithm based on Harris corner detection, empirical position and HSV matching is developed. With this algorithm, matching points are searched and found automatically, allowing the panoramic video to be stitched automatically. Thereafter, results are analyzed and discussed.

2. Image mosaic

2.1 Shooting

If a wide-angle or fisheye lens is used, few photos are needed. There is a difference between shooting panoramic video and shooting a single panoramic image. When shooting a single panoramic image, one photo is taken and then the camera is rotated to take the next one. Photos from all different directions can be taken by one camera. Each photo is taken at different times. In contrast, panoramic video should not be shot in that way. For the image to work, time differences cannot occur. Videos from different directions must be shot at the same time. For this study, two GoPro cameras (GoPro Hero4 black), two fisheye lenses (16MP 1/2.3), a self-made platform and a tripod were used. The maximum angle of the vision field of the 16MP lens was 220°. The design of the installation can be seen in figure 1



(a) Design (b) Photo

Fig. 1. Shooting equipment

2.2 Coordinates

We can imagine the forming of a panoramic image in this way. A person sits inside an enormous glass sphere. One of his eyes is just at the center. Rays of lights set out from objects to the eye through the glass sphere. In this hypothetical situation, every ray creates an image on the glass sphere when it passes through it. The outcome is the panoramic image of all the objects around it on the glass sphere (Figure 2). The coordinate system XYZ in Figure 2 is the world coordinate. To obtain a panorama: two photos (or videos) using a camera and a fisheye lens are taken. Then they are stitched together using the algorithm discussed below. The first shoot is shown in Figure 3. In Figure 3, the glass panoramic image described in section 2.2 is displayed on the inner surface of the sphere. A camera with a fisheye lens was placed at the origin *(O)*. Taking a photo of the real world directly and taking a photo for the image in the glass inner surface will result in the same picture, as shown in Image 1. When the camera photographs the left part, the area it can photograph is called the left hemisphere. The fisheye photo (Image 1) was placed on the left of the glass sphere, and then the image coordinate system *O'X'Y'* and the left shooting coordinate system $OX_LY_LZ_L$ were built. Notice that the directions of OX_L and *O'X*' are the same, and so are the directions of OY_L and *O'Y'*. The shooting coordinate system should be a right-handed system, just like the world system *XYZ*, so we take the opposite direction of camera shooting to be the direction of Z_L. With these coordinate systems, the pixels on the inner surface of the left glass hemisphere may be expressed by the coordinates X_L, Y_L and Z_L. They have computable relationships with their corresponding pixels in Image 1.

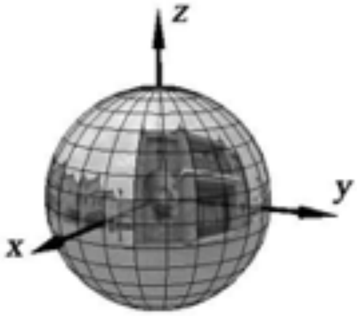


Fig. 2. Spherical panorama and observation

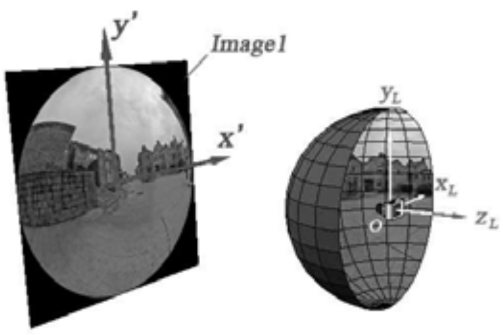


Fig. 3. First fisheye photo

The second fisheye photo was taken by rotating the camera 180 degrees to photograph the right part (Figure 4). The result is the fisheye photo Image 2.

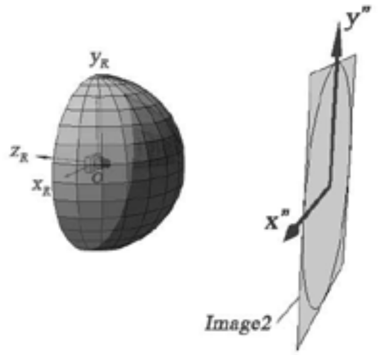


Fig. 4. Second fisheye photo

In Figure 4, the back of Image 2 can be seen. Its frontal image is just like the right image in Figure 14. The coordinate system *O"X"Y"* of Image 2 and the right shooting coordinate system $OX_RY_RZ_R$ are built into the figure. Just like above, the shooting coordinate system should be a right-handed system, so the opposite direction of camera shooting is taken as the direction of Z_R. The pixels in the inner surface of the right glass hemisphere may be expressed by the coordinates X_R, Y_R and Z_R, and they have computable relationship with their corresponding pixels in Image 2.

Figure 5 shows all coordinate systems drawn together.

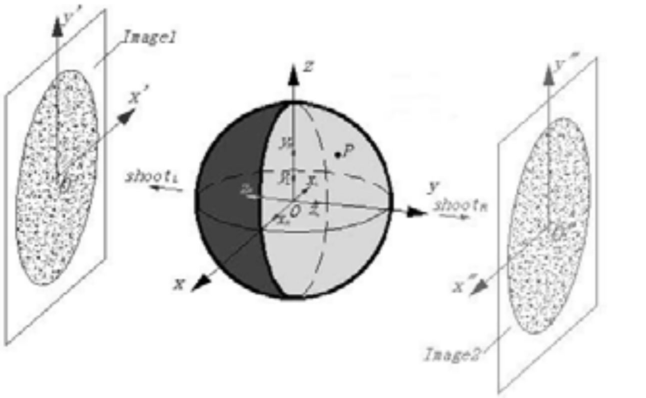


Fig. 5. Shooting directions and coordinates

In Figure 5, the arrow shootL is the direction from which Image 1 was taken, while the arrow shootR relates to Image 2. There are five coordinate systems in the figure: (1) the world or global system *(OXYZ)*. It is the coordinate system of the 3D world. Longitude and latitude in the panorama are calculated by this system. (2) The shooting coordinate system OX_LY_LZL for the left hemisphere. Shoot$_L$—the shooting direction of Image 1—is contrary to the axis Z_L. (3) The image coordinate system *O'X'Y'* for the left photo (Image 1) is a plane coordinate system. We use it to observe Image 1. (4) The shooting coordinate system $OX_RY_RZ_R$ for the right hemisphere is shot in the direction shootR, which is contrary to its axis Z_R. (5) The image coordinate system *O"X"Y"* for the right photo (Image 2) is a plane coordinate system. It is used to observe Image 2. Besides these five coordinate systems, there is another longitude-latitude coordinate system for the unfolded image of the global sphere (see Figure 8). A fisheye photo can be described as a type of uniform angle image, while a panorama is a longitude-latitude one. In order to change fisheye images into a panoramic image, the method of coordinate system transformation is used.

2.3 Relationship between two shooting coordinate systems

Though most of the right hemisphere does not appear in Image 1, it can be expressed in the coordinate system $OX_LY_LZ_L$. We can get the system $OX_LY_LZ_L$ by rotating the system $OX_LY_LZ_L$. The coordinates in system $OX_RY_RZ_R$ of every point on the right hemisphere can be read from Image 2. Every point can be transformed into $OX_LY_LZ_L$, that is, if it is multiplied by a matrix, as shown by the following formula:

$$\begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} = A \begin{bmatrix} x_R \\ y_R \\ z_R \\ 1 \end{bmatrix} \qquad (1)$$

In this formula, (x_L,y_L,z_L) refer to the coordinates in system $OX_LY_LZ_L$, and (x_R,y_R,z_R) refer to the system $OX_RY_RZ_R$.

Consider the transformation between a spherical coordinate system and a rectangular coordinate system (Figure 6).

The sphere radius in Figures 5 and 6 can be of any value without affecting the imaging. If it is 1, then the relationship between the longitude and latitude (θ, φ) and the rectangular coordinates (x,y,z) are as follows:

$$\begin{aligned} x &= \cos\phi\cos\theta \\ y &= \cos\phi\sin\theta \\ z &= \sin\phi \end{aligned} \qquad (2)$$

And conversely,

$$\theta = \arctan(\frac{y}{x}) \qquad (3)$$

$$\phi = \arctan(\frac{z}{\sqrt{x^2+y^2}}) \qquad (4)$$

According to Formula (2), Formula (1) can be rewritten as Formula (5).

$$\begin{bmatrix} \cos\phi_L\cos\theta_L \\ \cos\phi_L\sin\theta_L \\ \sin\phi_L \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix} \begin{bmatrix} \cos\phi_R\cos\theta_R \\ \cos\phi_R\sin\theta_R \\ \sin\phi_R \\ 1 \end{bmatrix} \qquad (5)$$

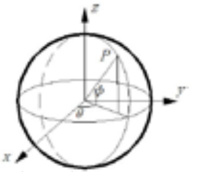


Fig. 6. Transformation from rectangular coordinates to longitude and latitude

Formula (5) is a system of linear equations. It has 15 panorama transforming parameters *$a_{11}, a_{12}...a_{43}$*. Five pairs of matching points are required to solve these equations. We use the Gaussian elimination method to solve it.

2.4 Relationship between shooting system and global system

Consider the transformation from system *$OX_LY_LZ_L$* to system *OXYZ*. The matrix A, which is solved using Formula (1) or Formula (2), is the matrix used to transform system *$OX_RY_RZ_R$* to system *$OX_LY_LZ_L$*. The global coordinate system is different from these two coordinate systems (Fig. 5). System *$OX_LY_LZ_L$* also can be transformed into system *OXYZ,* if the following transformation is applied to it: rotate 180° around the *Y_L* axis and then rotate 90° around the *X_L* axis (see Formula (6)).

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\frac{\pi}{2}) & -\sin(\frac{\pi}{2}) & 0 \\ 0 & \sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\pi) & 0 & -\sin(\pi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\pi) & 0 & \cos(\pi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} = B \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} \qquad (6)$$

2.5 Relationship between image coordinate system and shooting coordinate system

The relationship between system *O'X'Y'* and system *$OX_LY_LZ_L$,* and between system *O"X"Y"* and system *$OX_RY_RZ_R$* should also be considered. In these relationships, the longitude and latitude of the image coordinate system and the shooting coordinate system are the same. In order to calculate the longitude and latitude, Formulae (3) and (4) are applied to the coordinate systems *OXYZ, $OX_LY_LZ_L$* and *$OX_RY_RZ_R$*. In system *O'X'Y'* and *O"X"Y*" (in the fisheye images), longitude can be calculated using a trigonometric function as in Formula (7).

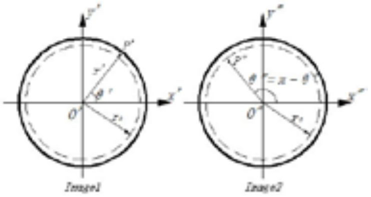


Fig. 7. Calculation of longitude and latitude in fisheye images

$$\theta' = \arccos(\frac{x'}{r'}) = \arcsin(\frac{y'}{r'}) = \arctan(\frac{y'}{x'}) \qquad (7)$$

In this formula, *r'* is the distance from the center to the calculating pixel *P'*. It equals $\sqrt{x'^2 + y'^2}$.

However, the calculation of the latitude requires a comparison with the radius of a circle having a latitude of 0° (the dashed circle):

$$\phi' = \frac{\pi}{2}(\frac{r'}{r_0} - 1) \qquad (8)$$

The latitude *φ'* in Image 2 can be calculated in a similar way. The turning radius is denoted as r0. This is the distance from the center to the pixels where the camera field angle equals *180°*. In coordinate systems *$OX_LY_LZ_L$* and *$OX_RY_RZ_R$,* the latitudes of those pixels equal 0. If two images are shot by the mode of *2×180°,* they have the following relationship:

(1) The turning radius equals the average of the distances in Image 1 and Image 2 from the center to the two matching points, as in

$$r_0 = \frac{r'_p + r''_p}{2} = \frac{\sqrt{x'^2_p + y'^2_p}}{2} + \frac{\sqrt{x''^2_p + y''^2_p}}{2} \qquad (9)$$

(2) The relationship between the longitudinal angle in Image 1 and that in Image 2 is:

$$\theta'' = \pi - \theta' \qquad (10)$$

2.6 Relationship between unfolding image coordinate system and global coordinate system

The stitched panorama actually is an image unfolded from the sphere in the global coordinate system according to longitude and latitude. Longitude and latitude can be calculated from the pixel coordinates in the unfolding image. Longitude and latitude can also be transformed into global coordinates. For example, the longitude and latitude of the pixel in row *i* and column *j* can be calculated as follows:

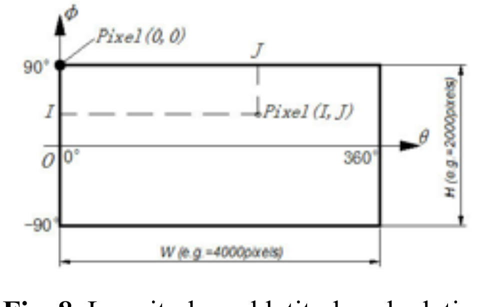


Fig. 8. Longitude and latitude calculation in a panoramic image

$$\theta_{ij} = \frac{2\pi J}{W} \tag{11}$$

$$\phi_{ij} = \frac{H-2I}{2H}\pi \tag{12}$$

Then longitude and latitude can be transformed into rectangular coordinates using Formula (2).

2.7 Operating steps

After understanding the relationship between different coordinate systems, the steps to generate the panoramic unfolding image can be obtained. For example, in order to generate a panoramic image of a size of 4000*2000 pixels, these steps must be taken:

(1) Find five pairs of matching points in Image 1 and Image 2, and substitute in the equations of Formula (5). Solve to find the transformation matrix *A*. *θ* and *φ* in Formula (5) can be calculated using Formulae (7) and (8). Then all of the pixels in the unfolding image are traversed (Figure 8). After this, fill in the colors for them. To determine what color each pixel should be, we execute steps (2)-(4) in a loop.

(2) Calculate the longitude *θ* and latitude *φ* of the traversing pixels using Formulae (11) and (12). Then the rectangular coordinates *(x,y,z)* are coordinated in the global system *OXYZ* using Formula (2).

(3) According to Formula (6), this pixel corresponds to the position in the left shooting coordinate system:

$$\begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} = B^{-1} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{13}$$

According to Formula (1), it corresponds to the position in the right shooting coordinate system:

$$\begin{bmatrix} x_R \\ y_R \\ z_R \\ 1 \end{bmatrix} = A^{-1}B^{-1} \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} \tag{14}$$

-1 is the operation of the matrix inversion. The values for longitude and latitude in the shooting coordinate system are the same as those in the corresponding image coordinate system. Hence, using Formulae (7) and (8) conversely, we can calculate the pixel coordinates (x',y') and (x",y") in the two photo images.

(4) For *(x',y')* in Image 1 and *(x",y")* in Image 2, there is at least one color value. In the unfolding image, some of the points come from Image 1, some of them come from Image 2, and some of them appear in both images. We determine the appropriate color using the Formula (15).

$$c = \begin{cases} c_1 & \text{Imaging only in } \textit{image} \\ c_2 & \text{Imaging only in } \textit{image 2} \\ k_1c_1 + k_2c_2 & \text{Imaging in both image1 and image2} \end{cases} \tag{15}$$

where *c* is the color value that we want to determine in the unfolding image. *k1* and *k2* are weight coefficients, and *k1+k2=1*. Figure 9 shows how we can set their values.

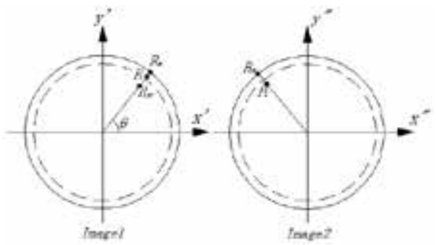


Fig. 9. Calculation of color mixing weight

In Figure 9, P_1 and P_2 are a pair of matching points. The radius crossing P_2 meets the edge of Image 2 at point P_{2e}. P_{2e1} is the matching point for P_{2e} in Image 1 (calculated by multiplying by the transformation matrix). Similar to P_{2e}, P_{1e} is the intersection point for which the radius that crosses P_1 meets at the edge of Image 1. k_1 and k_2 can be calculated according to the distances from P_1 to P_{1e} and from P_1 to $P2_{e1}$ (Formula (16)).

$$k_1 = \frac{|P_1P_{1e}|}{|P_{1e}P_{2e1}|} \qquad k_2 = \frac{|P_1P_{2e1}|}{|P_{1e}P_{2e1}|} = 1 - k_1 \tag{16}$$

3. Method of automatic stitching

The method shown in section 2 is manual. It requires five pairs of matching points that are found manually to calculate the turning radius r0 in Formula (9) and solve equations (5). If there is a way to find the five pairs of matching points automatically, the stitching method would become automatic.

In regard to methods for automatically searching matching points, SIFT is the most understood. Since its inception by Lowe (2004), the SIFT algorithm has been improved and developed. Kupfer *et al.* (2015) posited a SIFT-based mode-seeking algorithm. Zhao *et al.* (2016)

improved the algorithm using color and exposure. Xie *et al.* (2015) presented a method that combined SIFT and wavelet transformation. SIFT has a positive effect on panoramic image matching point searching, but it is unsuited to panoramic video because it is time consuming. When stitching a single panoramic image, searching for matching points only needs to be done once, so the amount of time need to complete this part of the operation is acceptable. However, when stitching panoramic video, not all of the transformation matrices for each frame are the same. If we use only one matrix, for example the matrix for the first frame, quivering will often take place in the final stitched video. In order to remove the image quivering, we have to search again or adjust the matching points in many frames. Using SIFT every time—for example, for dozens or even hundreds of times—would require a completely unacceptable amount of time.

There are several reasons why panoramic video needs to adjust matching points in its frames. First, the cameras may vibrate when shooting, which cause the relative position between the two cameras to change slightly. Second, the optical center (Node) must not be changed in panoramic shooting. If this demand cannot be guaranteed, numerous errors on the edges of two images during the stitching will occur. When shooting panoramic images by rotating one camera, it is easy to guarantee it. However, when shooting panoramic video, two or more cameras shooting at the same time are required, and they must be installed together, occupying each other's place, which means that their Nodes cannot be the same. In this case, if the distance of certain objects varies during shooting, the position of the matching points will change. Therefore, algorithms for panoramic video automatic stitching must be more understood. We outline a new method for finding and adjusting the matching points based on empirical position and Harris corner detection. The idea of this method is as follow:

(1) Finding corners where the local colors vary greatly in Image1. See section 3.1.

(2) Select five of these then preliminarily determine the theoretical position in Image 2 through the empirical relationship. See Section 3.2.

(3) Search for the accurate matching position in the small area around the theoretical position according to color difference. See Section 3.3.

3.1 Harris corner detection

Usually, when we pick matching points manually, we often pick the corner points of an object in which color varies obviously. The Harris corner detecting algorithm is a sophisticated method for accomplishing this task. It is widely used in image processing. Indeed, the Harris corner detecting method can detect and judge corners, edges and flat areas in an image (Mahesh & Subramanyam, 2012). It studies the differences in intensity for a displacement of *(u,v)* in all directions as in the following formulae:

$$E(u,v) = \sum_{x,y} w(x,y)\,[I(x+u,y+v) - I(x,y)]^2 \approx [u \quad v]M\begin{bmatrix} u \\ v \end{bmatrix} \tag{17}$$

In the formula above, *M* is matrix:

$$M = \sum_{x,y} w(x,y)\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{18}$$

In these two formulae, I refers to the intensity of fisheye photo Image 1. I_x and I_y are the gradient images generated by filtering with horizontal and vertical difference operators. *w(x,y)* is the two dimensional Gaussian function.

We then calculate the corner response function *R*:

$$R = \det(M) - k(trace(M))^2 \tag{19}$$

In this formula, *det* is the determinant of matrix *M*. trace is the trace of matrix. The determinant and the trace have a relationship with the Eigen values of *M:*

$$\det(M) = \lambda_1\lambda_2 \tag{20}$$

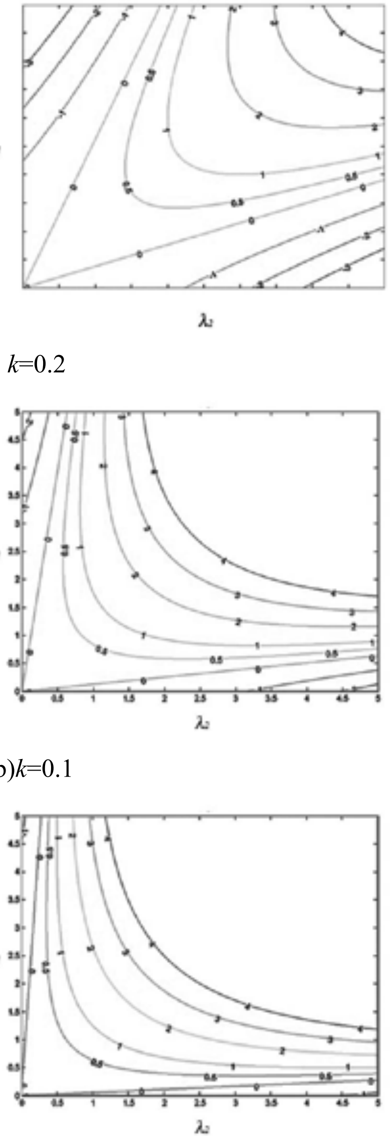
$$trace(M) = \lambda_1 + \lambda_2 \tag{21}$$

λ_1 and λ_2 are the maximal and minimal Eigen values of M. They mean the two directions along which gradient changes the most rapidly or the most gently.

After R is calculated, we determine by these rules: (1) if R is large (both λ_1 and λ_2 are large and $\lambda_1 \sim \lambda_2$), the position is a corner; (2) if $R<0$ ($\lambda_1 >> \lambda_2$ *or* $\lambda_1 << \lambda_2$), the position is on an edge (similar to the performing line detection to the image (Rangasamy & Subramaniam, 2017) in this condition); (3) if $|R|$ is small (both λ_1 and λ_2 are small), the position is in a flat area. There may be many positions that are judged to be corners. We use the n positions (e.g. $n=30$), which R are the largest for further screening.

k is a constant. In order to decide what value *k* should be, we draw some contour charts which show the relationship of *R* and λ_1, λ_2:

Figure 10 contour charts show (1) The area that both λ_1 and λ_2 are small is at the bottom left, close to origin. It corresponds to the flat regions in the image. (2) Two areas that $\lambda_1 >> \lambda_2$ or $\lambda_1 << \lambda_2$ are at the bottom right or top left. They correspond to the edge regions in the image. (3) The area where both λ_1 and λ_2 are large is at the top right. It corresponds to the corner regions in the image. (4) The area that the contour lines are dense is transitional area. We hope that the



(a) *k*=0.2

(b)*k*=0.1

(c) *k*=0.05

Fig. 10. Selection of the value of k in formula (19)

transitional area is small, so that we can judge it more accurately. According to the contour charts, we can see that the smaller the *k* value, the smaller the transitional area. However, we can also see that if *k* is too small, the areas of *R<0* at the bottom right and top left will become too small. It is difficult to judge edges at that time. So, after weighing the advantages and disadvantages, *k* should be between 0.04-0.06.

Only five positions are needed because equations (5) only needs five pairs of matching points. As for these five points, as well as corners where colors might obviously vary, we hope that the distances between the points are far. If they are close, it is likely that there will be errors in the solved matrix (*A* in Formula (1)). This is accomplished by selecting four corners whose distance from the other close corners are the largest from the four quadrants, respectively. Then the fifth is selected, whose distance is the largest in the remainder corners for the whole image.

Sometimes, if there is a lot of noises in the image, some noise points will be recognized as corner points, causing errors. At this time, the anisotropic diffusion method can be used to denoise the image before performing corner detection (Gharsallah & Braiek, 2017).

3.2 Searching area based on empirical position

When shooting panorama, if the Node is not changed and the camera field angle is known accurately, we can precisely calculate where a point in Image 1 is in Image 2. When the shooting mode is *2×180°*, we can calculate those using Formulae (9) and (10). To clarify, a point in Image 1 with polar coordinates $(r_p^{'},\theta)$ has a matching position in Image 2 that should be $(2r_0 - r_p^{'}, \pi - \theta^{'})$. However, because the camera Node cannot remain in the same position during shooting, there may be errors, and the position of matching points may not be accurate. There are often deviations. Furthermore, because cameras may vibrate, and the distances of the objects being shot may vary, the positions of matching points should be often adjusted. Therefore, before this method is employed, we use the theoretical position of the matching point to be the center, specify an area, and then search in this area to find the real matching point. For example, an *l×l* rectangular area (*l* can be 0.1-0.3 times *r₀*) should be specified and then sought. When stitching a new frame of video, if deviation is found, this method is used to search again. This searching method, which is based on empirical positions, allows the searching area to decrease resulting in a less time-consuming process.

3.3 Match detecting base on HSV

In the search area, we must find exactly which pixel is the matching point. To do this, first a small window in each fisheye image must be specified. The corner in Image 1 is used as the center of the window. Next, all pixels in the searching area in Image 2 is traversed by using the traversing pixel as the center of the window. After that, the weighting sum of color differences between the two windows in the two images is calculated and compared.

Human eyes mainly recognize objects according to hue and saturation. It is better to use the HSV model than to use the RGB model to detect the matching points because the HSV model is closer to human eye function. Therefore, we must change the images into HSV color space, and then we can use the Formula (22) to calculate the weighting sum of color differences.

$$E_{x,y} = \sum_{u,v\in[0,l]} w(u,v)[k_h(H^{''}_{x+u,y+v} - H^{'}_{s+u,t+v})^2 + k_s(S^{''}_{x+u,y+v} - S^{'}_{s+u,t+v})^2] \qquad (22)$$

In this formula, $E_{x,y}$ is the sum of the color differences at pixel *(x,y)* in Image 2. $H'_{s+u,\ t+v}$ and $S'_{s+u,\ t=v}$ are the hue and saturation at *(s+u,t+v)* in Image 1, while and are the hue and saturation in Image 2. k_h and k_s are two coefficients. They are used to control the importance of hue and saturation. We set k_h as 0.8 and k_s at 0.2. That means that hue is four times more important than saturation. *w(u,v)* is the weighing function for the different window pixels. The two-dimensional Gaussian function was used for this process.

In Image 2, we traverse all the pixels in the search area to calculate $E_{x,y}$. We determine the pixels where $E_{x,y}$ is minimal. That is the matching point of the corner in Image 1.

With one pair of matching points, we can find the turning radius *r0* (Formula (9)). With five pairs of matching points, we can find the transformation matrix (Formula (5)), and then automatic stitching can be done.

3.4 Adjusting the matching points and transformation matrix

As related above, the transformation matrix in Formula (2) may change because of slight vibrations in the cameras and/or a variation in the distances of the objects. We must find matching points and calculate the transformation matrix again at that time. We determine when to adjust the matrix by checking the overlap area between the two images. As shown in Fig. 11, when shooting in the mode of 2×180°, the shaded parts are overlapped areas. The overlapped area in the unfolding image is a narrow rectangle in the middle. If it is not stitched well, there may be many connecting errors in this rectangle. The connecting errors become corners, and the Harris corner method can detect them. Therefore, when stitching one frame, we can check if the number of corners in the overlapped area of the fisheye image (e.g. Image 1) is approximately equal to that in the unfolding image. If true, the stitching effect has been successful. However, if the number of corners in the folding image is much greater than that in the fisheye image, then the stitching effect has been diminished, and five pairs of matching points must be found and the transformation matrix must be calculated again.

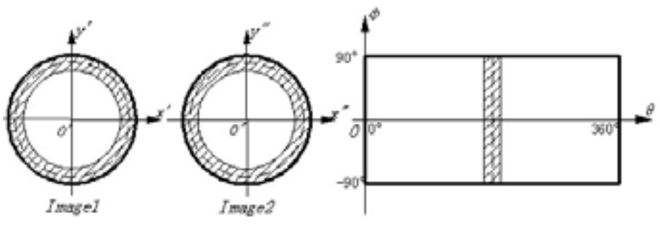


Fig. 11. Overlap area when stitching

Not every frame of the video needs to be checked. We can check it at intervals (e.g., every 10 frames). In addition, not all the overlapped area must be checked, if it is large. In the fisheye image, we can expand and shrink *n* pixels from the turning radius (e.g. *n=5*), and then take the region of the ring between the expanding and shrinking radii. In the unfolding image, we move to *n* pixels from the central line to the left and move n pixels from the central line to the right. Thereafter, the rectangular area between them can be taken. Almost all the corners created by connecting errors are on the central line.

3.5 Comparison experiment of matching point searching

In order to compare the performance of the algorithm described above, we take some matching point searching experiments using various methods.

(1) The complete Harris method: i.e., using the Harris method in both corner detecting and matching point searching;

(2) The SIFT method;

(3) Detecting corners using the Harris method. Thereafter matching points are searched based on the empirical position and RGB (Later, this is referred to as the Harris-Emp-RGB method; and,

(4) Detecting corners using the Harris method. Then matching points are searched based on the empirical position and HSV (Harris-Emp-HSV method).

The first experiment uses the Harris method to detect corners in two images. These are compared to the intensity accumulation of the corners and the surrounding points between the two images. Figure 12 is a representative result. The result shows that although corner detecting is accurate using the Harris method, there are many errors in its matching points result. Thus, its matching result shouldn't be used in the subsequent stitching computation.

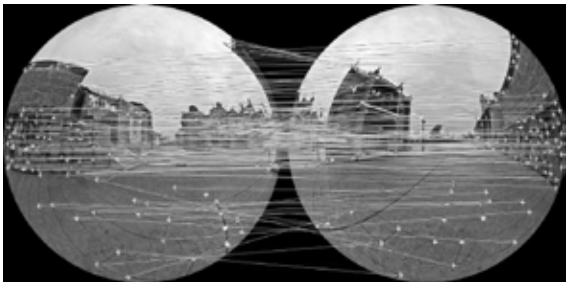


Fig. 12. The matching result of the Harris method

As stated SIFT is a widely-used matching point detecting method. The fisheye images should be corrected before a matching point search (Zhang *et al*., 2015). The matching result for the same image is shown in Figure 13.

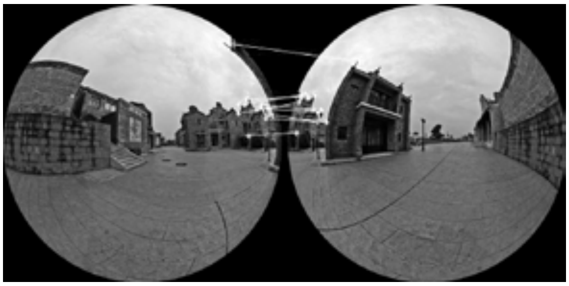


Fig. 13. The matching result of the method SIFT

The matching result shows the accuracy of the SIFT. However, it has disadvantages. First, it is time consuming. Second, sometimes the matching points are gathered in one particular local area of the images (Figure 13). Using such matching points to achieve the stitching computation may cause some error or deviation. In the shooting mode of 2×180°, all the overlapped objects are far from the center, and often, they are only at the left and right. When there is no obvious corner at one side, this phenomenon occurs often.

The Harris-Emp-HSV (discussed in Sections 3.1-3.4) matching result is shown in Figure 14.

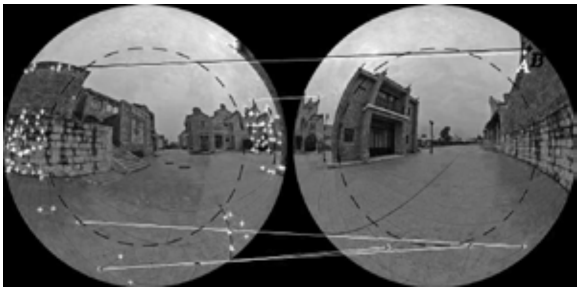


Fig. 14. The matching result of Harris-Emp-HSV

In Figure 14, the dashed circle is the boundary of overlapped image. Its radius and the corresponding field angle can be calculated with the image radius and lens parameters. For example, if the largest field angle of the fisheye lens is 220°, the radius of the dashed circle will be 7/11 of the image radius, and its corresponding field angle is 140°. There is no overlapped content inside the dashed circle, so corners need to be detected and match points need to be searched only at the outside of the circle. In Fig. 14, five pairs of matching points are achieved. The matching points are accurate and scattered, meaning they are very good for a stitching computation.

In order to determine whether the Harris-Emp-HSV or Harris-Emp-RGB is better, we also ran tests using the Harris-Emp-RGB method. Methods described in Section 3.1-3.4 were used, but to accumulate and compare, Formula (22) used the components of red, green, and blue instead of hue and saturation. Figure 14 also shows the matching result (the black point and the black line. One of the deviated five matching points is shown. The correct position is the white point A, but the black point B was found. Such errors often occur when some points have different hue but similar intensity.

Beside accuracy, computational time consumption of these methods differs. In addition to the above tests, 50 groups of fisheye photos were taken. Matching tests were carried out using the four methods. Results are shown in Table 1.

Table 1 shows that SIFT and Harris-Emp-HSV have very good accuracy. However, if considering the runtime and scattering, we can see the Harris-Emp-HSV method is better.

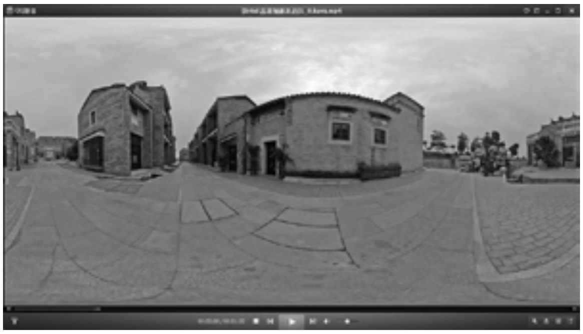


Fig. 15. Screenshot of a stitched video 1

Table 1. Comparison of four matching methods

Method	Average Matching Accuracy	Average Runtime (ms)	Scatter
Harris	26%	143	Unstable
SIFT	96%	930	Unstable
Harris-Emp-RGB	82%	87	Yes
Harris-Emp- HSV	96%	94	Yes

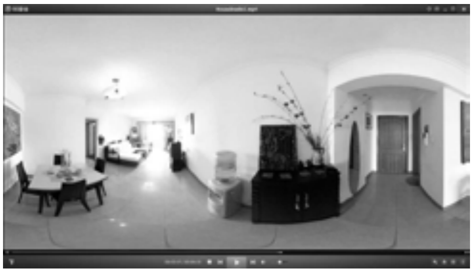


Fig. 16. Screenshot of a stitched video 2

4. Instances and test

Using the aforementioned methods, more shooting and stitching tests were conducted. The photographs in Figures 15 and 16 are two screenshots of two stitched videos. These photos show that the stitching is accurate at the central line, and there are smooth transitions. There was no quivering when the video was played. We tested many videos of different sizes and lengths. The data are shown in Table 2

Table 2. Stitching testing data for videos of different sizes and durations

Group	Video size and duration	Amount of testing video	Longest stitching time	Shortest stitching time	Average stitching time
1	2000×1000,15s	20	24s	13s	18s
2	4000×2000,30s	20	82s	66s	73s
3	8000×4000,1min	15	443s	375s	402s
4	16000×8000,2min	10	2113s	1876s	1983s

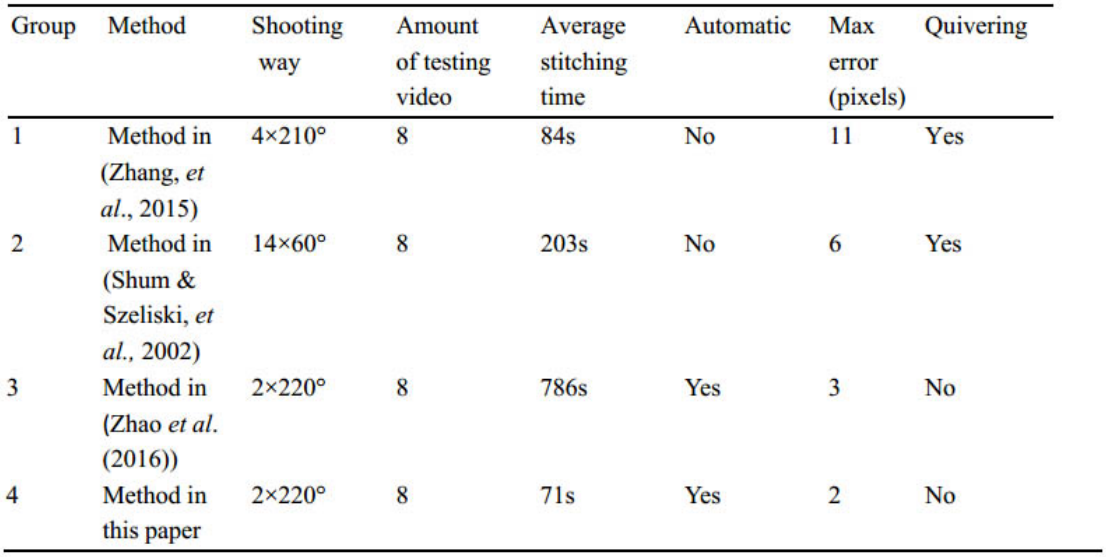
The experiment was carried out on a computer with a dual-core 3.2G CPU, 8G memory, and 4G video memory. The stitching program was developed with C++ and OpenCV. In every group in Table. 2, multiple videos were tested. Outdoor and indoor scenes were used. The stitching time increased when the dimension and duration of a video increased but not proportionally. The most time-consuming work was the matching point searching, but the stitching program does not need to do this work for every frame. This work is only needed when the transforming matrix is invalid (see Section 3.4). If the camera vibrates when shooting or the distances of shot objects change rapidly, more adjustment of the transforming matrix is needed. This means that the stitching time will increase.

In order to compare the performance of the video stitching, we also tested using other typical methods. Table 3 shows the perform

In Table 3, four methods for panoramic video stitching are compared. All the videos were 4000×2000 pixels in size and 30s in time. Spherical. The same below We chose eight different places. In the third method, stitching algorithm of this paper and the matching point searching SIFT algorithm (Zhao *et al.*,2016) were used. In the fourth method, both stitching and matching point searching were performed using algorithms in this paper (i.e. Harris-Emp-HSV). We shot and stitched panoramic videos using each method for each place. Finally, there were eight videos made by each method. The stitching time of these eight videos was averaged (values in the column "Average stitching time" in the table). We carefully checked each frame of all the videos, found the error places, recorded the pixel deviation, and picked out the maximal error for each method (see the column "Max error"). For example, the photograph in Figure 17 shows the errors marked out on a screenshot of a video in group 1.

Several error places are magnified, and the pixel deviations are shown beside them so they are more visible. We can see in this picture that the maximal error is 11 pixels.

Table 3. Performance of several methods



Group	Method	Shooting way	Amount of testing video	Average stitching time	Automatic	Max error (pixels)	Quivering
1	Method in (Zhang, *et al.*, 2015)	4×210°	8	84s	No	11	Yes
2	Method in (Shum & Szeliski, *et al.*, 2002)	14×60°	8	203s	No	6	Yes
3	Method in (Zhao *et al.* (2016))	2×220°	8	786s	Yes	3	No
4	Method in this paper	2×220°	8	71s	Yes	2	No

The photograph in Figure 18 shows the maximal error of group 2.

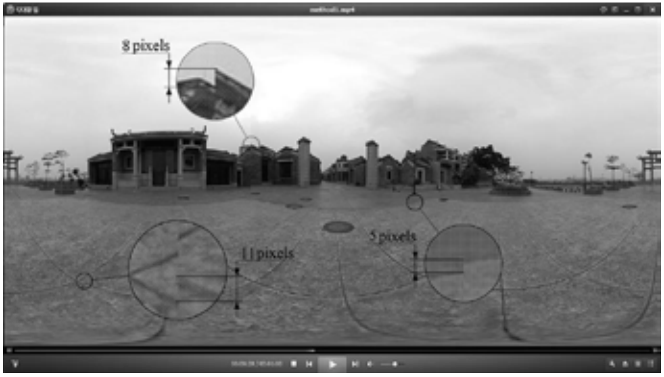


Fig. 17. The stitching result of the first method and its error

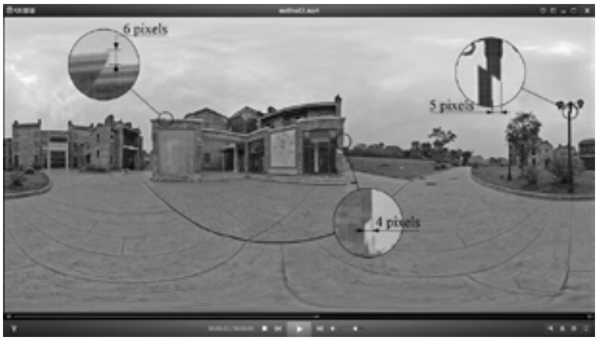


Fig. 18. The stitching result of the second method and its error

Similarly, the error places are magnified, and their pixel deviations are shown. The image in Figure 18 shows that the maximal error of Group 2 is 6 pixels. Table 3 shows that the errors of Group 3 (SIFT method) and Group 4 (Harris-Emp-HSV method) are fewer (3 pixels and 2 pixels). There is little difference accuracy between SIFT and Harris-Emp-HSV, but SIFT is more time consuming than the Harris-Emp-HSV. Therefore, the Harris-Emp-HSV method postulated in this paper is better than the other typical aforementioned methods.

5. Conclusion

In this paper, the production of panoramic video was researched. After studying several coordinate systems and the transformations among them, we put forward a method and steps for manually stitching panoramic video. An automatic matching-point search algorithm was developed based on the Harris corner detection, empirical positioning, and HSV matching methods. Comparing experiments proved that our methods perform better than the other typical methods in both accuracy and time consumption. We suggest more research based on our Harris-Emp-HSV methods, including the stitching and matching-point search algorithms for three or more cameras. Multi-camera shooting can improve the definition of panoramic video. The transformation matrix and the calculation of the empirical position in feature point search for multi-camera shooting are different from two cameras. These issues should be further studied.

ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of Guangdong Province of China. (Grant No.2015A030310112) and the Science and Technology Project of Guangdong Province of China (Grant No. 2016A040403110 and No. 2017B010110008).

References

De Carufel, J.L. & Laganiere, R. (2011). Matching cylindrical panorama sequences using planar reprojections. IEEE International Conference on Computer Vision. Barcelona, Spain.

Gharsallah, M.B. & Braiek, E.B. (2017). New anisotropic diffusion method to improve radiographic image quality. Kuwait Journal of Science, **44**(3): 56-64.

Kupfer, B., Netanyahu, N.S. & Shimshoni, I. (2015). An efficient SIFT-based mode-seeking algorithm for sub-pixel registration of remotely sensed images. IEEE Geoscience and Remote Sensing Letters. **12**(2): 379-383.

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, **60**(2): 91-110.

Mahesh, S.M.V. (2012). Automatic image mosaic system using steerable Harris corner detector. 2012 International Conference on Machine Vision and Image Processing. Taipei, Taiwan. **12**: 87-91.

Pan, W., Qin, K. & Chen, Y. (2009). An adaptable multilayer fractional fourier transform approach for image registration. IEEE Transactions on Pattern Analysis nd Machine Intelligence, **31**(3): 400-414.

Rangasamy, V. & Subramaniam, S. (2017). Framelet transform based edge detection for straight line detection from remote sensing images. Kuwait Journal of Science, **44**(1): 78-85.

Shum, H.Y. & Szeliski, R. (2002). Construction of panoramic image mosaics with global and local alignment. International Journal of Computer Vision, **48**(2): 151-152.

Zokai, S. & Wolberg, G (2005). Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. IEEE Transactions on Image Processing, **14**(10): 1422-1434.

Xie, X., Xu, Y. & Liu, Q. (2015). A study on fast SIFT image mosaic algorithm based on compressed sensing and wavelet transform. Journal of Ambient Intelligence and Humanized Computing. **6**(6): 835-843.

Xu, Y. Zhou, Q. & Gong, L. (2014). High-speed simultaneous image distortion correction transformations for a multicamera cylindrical panorama real-time video system using FPGA. IEEE Transactions on Circuits and Systems for Video Technology, **24**(6): 1061-1069.

Yang, L. *et al.* (2014). Optimized design of automatic panoramic images mosaic. Infrared and Laser Engineering. **43**(3): 985-990.

Zhang, B., Jia, Y. & Roning, J. (2015). Feature matching method study for uncorrected fish-eye lens image. Conference on Intelligent Robots and Computer Vision XXXII-Algorithms and Techniques. San Francisco, USA.

Zhang, B., Li, H., Li, D. & Dai, F. (2015). A simulation of image mosaic algorithm based on vehicle panorama system. Journal of Electronics and Information Technology, **37**(5): 1149-1153.

Zhao, Y., Zhai, Y. & Dubois, E. (2016). Image matching algorithm based on SIFT using color and exposure information. Journal of Systems Engineering and Electronics, **27**(3): 691-699.

Submission	: 10/05/2017
Revision	: 03/06/2018
Acceptance	: 04/06/2018

طريقة دمج تلقائية لفيديو بانورامي كروي

ليهونغ لوه، جيانكينغ مو، جيازين لي

قسم الوسائط الرقمية، جامعة قوانغدونغ للتكنولوجيا، قوانغتشو، 510006، الصين

الملخص

أصبح الفيديو البانورامي موضوع بحث مثير للاهتمام في مجال الواقع الافتراضي في السنوات الأخيرة. تشير الأبحاث الحالية إلى أنه لا يمكن دمج الفيديو البانورامي تلقائياً حتى الآن؛ بالإضافة إلى ذلك، فإن دقة الدمج والوقت المُستقطع غير مُرضيين. بعد تحليل العلاقة بين الأنظمة الاحداثية المتعددة في التقاط الصور، يقدم هذا البحث طريقة يدوية وخطوات لتصميم فيديو بانورامي. ثم يطرح خوارزمية بحث لنقطة التطابق مرتكزة على هاريس للكشف عن الزوايا والموضع التجريبي ومطابقة HSV. تم البحث عن نقاط المطابقة وتم العثور عليها تلقائياً باستخدام هذه الخوارزمية، وتم كذلك دمج مقاطع فيديو بانورامية تلقائياً، وأُجريت مقارنة بين التجارب. وأثبتت نتائج الاختبار أن خوارزمية الدمج التلقائي الخاصة بمقاطع الفيديو البانورامية تقدم أداءً أفضل من الخوارزميات الأخرى من حيث الدقة والوقت المُستقطع.