International Journal of Civil Engineering and Technology (IJCIET)
Volume 9, Issue 12, December 2018, pp. 77–85, Article ID: IJCIET_09_12_011
Available online at http://www.iaeme.com/ijciet/issues.asp?JType=IJCIET&VType=9&IType=12
ISSN Print: 0976-6308 and ISSN Online: 0976-6316

METHODS AND ALGORITHMS FOR STITCHING 360-DEGREE VIDEO

Aleksandr Evgenjevich Petrov, Dmitrii Aleksandrovich Sytnik and Ivan Yur'evich Rubcov

Complex Systems LLC, Russia, 170021, Tver, Gollandskaya Street, 83, ofis 1

ABSTRACT

The rapid development of virtual reality technologies in recent years has led to an increase in interest in 360-degree video and, as a consequence, in the production of equipment for shooting. Shooting 360-degree video differs from regular video shooting by the need to use multiple cameras (lenses) to create panoramic video. Stitching the video from several video cameras (lenses) in order to form panoramic video comes to the fore in this case. Currently, there are a number of algorithms and software solutions available for implementing video stitching. The purpose of the paper is to analyze and search for optimal algorithms and tools for 360-degree video stitching. The analysis takes into account, first of all, the quality of the stitching algorithms, which involves the absence of visible seams in the resulting image. The performance of the stitching methods also plays an important role, since the speed of processing the video footage is critical and ideally should be done in the real-time mode, which allows broadcasting 360-degree video. The main result of the study is the selection of optimal methods and parameters of video stitching, depending on the purposes, determined by the balance of speed and quality of processing. Given the increased interest in this topic in recent years, this study allows systematizing existing methods and will help 360-degree video producers with a choice of methods and tools for stitching.

Keywords: 360-degree video, stitching, panoramic video, video alignment, image composition, video synchronization, pixel alignment, feature-based alignment.

Cite this Article: Aleksandr Evgenjevich Petrov, Dmitrii Aleksandrovich Sytnik and Ivan Yur'evich Rubcov, Methods and Algorithms for Stitching 360-Degree Video, International Journal of Civil Engineering and Technology, 9(12), 2018, pp. 77–85

http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=9&IType=12

1. INTRODUCTION

Often, when it is told about panoramic video, it refers to the so-called 360-degree video. The 360-degree panorama, in turn, can be divided into a cylindrical panorama [1] and a spherical panorama [2]. The cylindrical panorama allows a horizontal 360 degrees angle of view and does not cover the top and bottom of the scene, while the spherical panorama allows a full view, that is, 360 degrees horizontally and 180 degrees vertically. The spherical panorama is called Full 360.

Full-360 video usually allows greater immersion for the user when viewing the video using a special device, for example, virtual reality glasses.

The approach to creating 360-degree video involves using multiple cameras or lenses that make up the system, which covers the desired angle of view. The relative geometry of cameras (lenses) is fixed and known throughout the entire shooting.

An example of using 7 GoPro Hero 3+ cameras is given in [3]. Five cameras are used to capture the front and rear of the scene, as well as the sides, 2 cameras are used for the top and bottom. Next, the set of images from all cameras is taken; they are aligned relative to each other and then “stitched” into one image. The resulting image is projected onto the plane using equirectangular projection [4]. The sequence of such images in equirectangular projection forms the 360-degree video.

An example of an image in this projection is given in [3]. This paper includes the formal definition of 360-degree video through plenoptic functions.

If the Full-360 video is viewed in a “regular” video player, it will be displayed as a sequence of the stitched images in equirectangular projection. Therefore, specialized players are used to correctly display such video, for example, GoPro VR Player [5], etc. Rendering of such image involves the inverse transformation to spherical coordinates in order to get the original scene. Moreover, these players allow changing the angle of view to see different parts of the whole scene at a given point in time. It is possible to change the angle of view by the movements of a computer mouse or by turning the virtual reality helmet, equipped with a smartphone or a gyro sensor, if it is supported by the player.

The purpose of image stitching is to obtain a panoramic seamless image [6] using a sequence of overlapping images. Overlapping images means that there is a common area of the scene in them. The common areas in the images may not be identical, since they could have been taken either at a different point in time with one camera or with different cameras, but their settings were different.

The top of Figure 1 [7] shows a sequence of overlapping images, and the bottom shows a stitched panorama.

Figure 1. Input images to form a panorama (top) and a stitched panorama (bottom)

Although image stitching has received widespread attention in the field of computer vision, much less attention has been paid to the related area – panoramic video stitching. Even fewer papers consider the 360-degree spherical video.

The purpose of stitching panoramic video is to create a panoramic seamless video using several overlapping video streams that capture different parts of the scene at the same time. The cameras are installed so that they capture all angles of view, but their relative geometry remains unchanged throughout the shooting. Overlapping videos taken with different cameras, which are together called multiview video [8], are input to the stitching program. The output is a panoramic video obtained as a result of stitching frames from the original video.

It should be noted that stitching a panoramic video is not just stitching individual frames taken from different cameras videos using well-known techniques for stitching images. This process has the same stages as image stitching, such as: frame alignment, color correction, blending, etc. However, there are unique challenges: video synchronization, video stabilization, etc., which should be taken into account by any commercial system for video stitching.

Synchronization refers to the synchronization of shooting with individual cameras. Stabilization is aimed at removing the effect of “motion jitter” in the frame. These stages are performed during the preprocessing of the video prior to stitching.

This paper considers only the stitching stage, which includes the alignment and composition of the multiview video. The composition does not have a clear term and in the papers on image stitching is called image merging, image composition, image blending, etc. In this paper, the term image composition is used.

2. METHODS

Approaches applicable to stitching individual images are also applicable to stitching video, since video is a sequence of frames, that is, images.

2.1. Image alignment

Two overlapping images are given. The purpose of image alignment is to find a geometric transformation of the common areas of the image. One of the simple transformations is the translation in the plane. That is, one image is shifted relative to the second so that the common areas “coincide” most fully. Of course, there are other transformations such as: rotate, similarity transform, affine transform, perspective transform, or just homography. There is a similar transformation hierarchy for 3D space.

A formal description of these and other transformations is given in [9].

Let us return to alignment. It is clear that in order to “overlay” the common areas of two different images on each other, more complex transformation than just translation or rotation may be needed, for example, the combination of translation + rotation, etc.

As shown by [9], there are two main methods for image alignment: 1) direct, pixel-based, 2) image feature-based.

2.2. Pixel-based alignment

This method is based on minimizing the divergence of pixels. In general, the pixel-based method is to shift or wrap the images relative to each other and see how many pixels coincide. A brute force approach, even in case of a simple shift, is very time-consuming. Indeed, one takes one image A and starts looking for it in image B. To do this, it is necessary to search for all the shifts in B. If A is N × N pixels, and the search area is M × M pixels, then the total search time will be $O(N^2 \times M^2)$. Therefore, various optimizations are used to reduce the search area: for example, the pyramid method [10], which is also called hierarchical motion estimation, based on constructing a pyramid of images, where the base of the pyramid contains the original image, and as it approaches the top, this image is scaled to lower resolution. Thus, after finding the area of images' intersection in low resolution, it is possible to move towards the base of the pyramid, thereby limiting the search area and achieve the result by gradually increasing the resolution. This category also includes algorithms based on Fourier transforms, incremental methods based on decomposition into a Taylor series, image functions, and others [9].

In this category, Lucas and Kanade [11] developed a patch-based translational alignment method. In the paper [12], Zernike moments are utilized to search for differences in the rotation and scaling of images.

2.3. Feature-based alignment

This approach is to extract the distinctive features from the images and to match these features to establish a global correspondence between them and then find a geometric transformation (homography) between the images. This approach has been gaining popularity recently [9].

There is no general definition of what an image feature is. It depends on the problem. Informally, the feature is an image area of interest, which is distinguishable from other areas of the image, and is constant and stable. The stability means resistance to noise and errors, and constancy is resistance to affine transformations. A popular feature in computer vision algorithms is the Harris corner. An example of the detected Harris corners is shown in Figure 2.

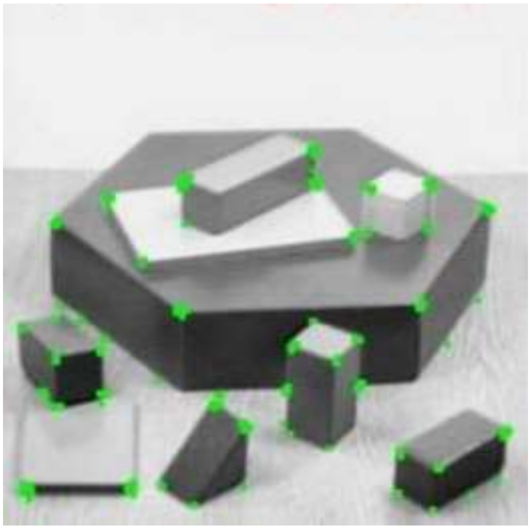

Figure 2 Harris corners

A detailed description of the features is beyond the scope of this paper and is given in [13].

After extracting the features in two images, they should be matched. This process is called feature matching. Based on this matching, it is possible to find geometric transformation (homography) between the images. Often, the RANSAC (RANdom SAmple Consensus) algorithm is used to find the homography. A detailed example of image stitching using the feature-based method can be found in [14].

Note that there are interesting features that have useful properties. For example, SIFT features are invariant to rotate and scale. This means that it is possible to search for scaled and rotated images in some original image. There are also other signs: SURF, ORB [15].

2.4. Direct-based methods versus Feature-based methods

As noted by Szeliski in [9], each of the methods has its advantages and disadvantages. The direct-based method is more accurate because it has information about all pixels. However, it is more resource intensive. Optimization with the pyramidal representation of the image may not provide the desired accuracy, because information is lost in the image at some levels of the pyramid.

Feature-based methods at the early stage did not give the desired accuracy, since they did not work well for areas of the image that were either too textured or not textured enough. Today, feature-based methods solve these problems and show good accuracy. Also, such methods are invariant to scale and rotate, which allows them to be used for automatic stitching.

Most of the stitching systems in the market are feature-based.

2.5. Image composition

After setting up the correspondence between the input images, it is necessary to select the surface on which all images will be projected to obtain the final panorama. Such surfaces can be: plane, cylinder, sphere, cube, etc. The plane is usually suitable when the panorama field-of-view does not exceed 90 degrees. Otherwise, the obtained panorama will be distorted. A cylinder or sphere shall be selected for a larger angle of view.

In the case of a full view panorama, the sphere shall be selected. In this case, the images are displayed on the sphere and stitched on it. Then blending techniques are applied to remove the seams and compensate the difference in color in the final panorama. A detailed description of these techniques is given in [9].

This paper briefly describes the main stages of image stitching. Then a review of existing approaches to video stitching is given.

2.6. Video

Unlike image stitching, 360-degree panoramic video is not widely represented in the literature.

Algorithms for creating 360-degree video in a cylindrical panorama are proposed in [16]. The proposed system is capable of producing stitched video in the real-time mode with the following characteristics: 30fps (frames per second) on Intel i7 3930K CPU 2.3GHz with 8GB DDR3 RAM in the operating system Linux Ubuntu 12.10. In the prototype, the authors of the paper use 4 cameras, which are located at an angle of 90 degrees relative to each other.

Initially, pre-processing of the input frame is performed, namely, image correction due to distortions caused by fish-eye lenses. Then the images are aligned in a cylindrical projection using the feature-based method. The authors use the SIFT algorithm for searching features.

The next stage is the search for the “seam” area, where the images will be stitched. In stitching video, there is a problem of moving objects at the seam boundary. If the object moves from one side of the image to the other through a fixed seam, then this may cause distortion of the object when stitching. Consequently, the position of the seam should change dynamically depending on current conditions. The paper proposes a dynamic seam adjustment scheme. A comparison of the average brightness in a 3 by 3 pixel window on both sides of the seam and dynamic programming to find the best seam are used.

2.7. Stitching and normalization of image tone

The stitching is performed along the seam using linear interpolation. It may happen that, due to different camera settings or ambient light, the same objects on different cameras may have different color tone. Therefore, it is necessary to normalize the tone. The paper proposes a method based on SIFT features for normalizing the tone of images before stitching. First, the matched points between the images are found using SIFT. Then RGB data of these points are used when solving equations of the form (1).

$$\begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix} = \begin{pmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{pmatrix} \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} \beta_r \\ \beta_g \\ \beta_b \end{pmatrix}, \tag{1}$$

where:

(r_1,g_1,b_1) is the color of the matched point pixel of the first image,

(r_2,g_2,b_2) is the color of the matched point pixel of the second image, parameter

α is the scaling factor,

β is the fine tuning coefficient.

Using the SIFT algorithm, the authors managed to reduce the complexity since there was no need to calculate histograms for the whole image.

The papers [17] and [18] propose a novel method for stitching images and videos taken using the Samsung Gear 360 Dual-fisheye lens camera. Field-of-view of this camera lenses is close to 195 degrees for each lens. Images generated by both lenses have a very limited common area. The feature-based method with SIFT features is used for image stitching. When using the traditional algorithm, an incorrect homography matrix can be produced, since the overlapping region is small [17].

The paper [18] is a continuation of [17]. In both papers, the algorithm consists of several common steps: (1) contrast compensation, (2) equirectangular projection of the original images, (3) two-phase alignment, (4) stitching and blending.

Contrast compensation is needed in order to even out the contrast of the entire image. The equirectangular projection represents a spherical image (and the image from this camera represents a hemisphere) on the plane. The first part of the two-phase alignment is finding the alignment matrix. For this, the authors use control points presented in the form of a chessboard. These points fall on both images. Then the common points are manually selected to find the affine transformation matrix between the points. The second step is the refined alignment, which is aimed at minimizing horizontal gaps. Thus, the authors were able to achieve high-quality stitching.

2.8. Tools and performance

The author of the paper [19] proposes an interesting approach – to reuse the homography matrix, calculated using previous frames, in the following frames. Also, the matrix shall be recalculated every X frames to avoid accumulating errors. The feature-based method is used to construct homography. The program uses two threads. The first thread stitches images based on the last homography obtained. The second thread calculates the homography as quickly as it can. In Table 1, the author shows that the method of reusing the homography significantly speeds up the work.

Table 1 Comparison of the algorithm with one and two threads and reuse of the homography matrix

	Time to calculate 100 frames	Time to calculate 28 homographies	FPS	HPS (Homography per second)
Single threading	68.95 secs	19.31 secs	1.450	1.450
Multithreading (2 threads)	**15.42 secs**	**14.70 secs**	**6.485**	**1.904**

The above-described papers use the CPU for all stages of stitching. The use of the GPU for speeding up some stages of stitching is of interest. Unfortunately, only a few review articles on this topic were found in open sources.

In the paper [20], the authors use the OpenCL [21] technology for computing on a graphics card. OpenCL is used directly for stitching images and displaying them on a sphere to get the final panorama. The authors state that they managed to achieve an increase in speed (Table 2), in comparison with the CPU-based approach in the same environment.

Table 2 Comparison of image stitching speed on CPU and GPU with OpenCL

Configuration	Time (ms)	Speedup
CPU-based	386	1x
OpenCL w/ 1 buffer	69	5x
OpenCL w/ 1 buffer + 1 PBO	39	9x
OpenCL w/ 2 buffers + 2 PBOs	30	12x

PBO means Pixel-Buffer-Object here. Table 2 shows that the use of OpenCL allowed acceleration by more than 12 times compared to the CPU.

Another paper [22] uses CUDA technology [23] by NVidia in order to speed up finding and matching points between images. The ORB algorithm is used [24].

The results of this paper are presented in Figure 3.

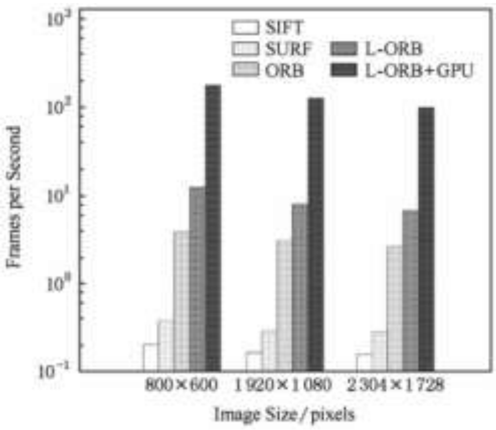

Figure 3. Comparison of the effectiveness of the algorithms for determining interesting points in images [22]

This diagram shows that ORB + GPU version at any image resolution works almost 2 times faster than the CPU version.

3. RESULTS

As a result of the analysis of methods and tools for stitching 360-degree video, it is possible to establish that, currently, the feature-based alignment methods are the most common, because, unlike pixel-based methods, they provide better performance and are insensitive to images scaling and rotation. As a rule, the SIFT, SURF and ORB algorithms are used to search for and compare features in different images.

The best stitching performance can be achieved by reusing the homography matrix, as well as by utilizing GPU resources. While the quality of stitching video, which was taken using 360 mobile cameras or which includes moving objects, can be improved by dynamic seam adjustment.

4. DISCUSSION

This paper covers algorithms and tools for stitching 360-degree video. A logical continuation of the study will be the analysis of existing solutions present in the market. When comparing such solutions, it will be necessary to take into account the methods and tools used for stitching, quality, performance, and cost.

There is specialized Gear 360 ActionDirector software used for stitching an amateur video shot on Samsung Gear 360 Dual-fisheye lens camera. Since these cameras are entry-level cameras, the high quality should not be expected. Autopano or VideoStich software is used stitching video shot on semi-professional or professional cameras. Thus, the comparison of these products and their analogs should form the basis of the subsequent analysis.

It is also useful to consider different approaches to stitching and compare the algorithms used in image alignment (ORB, SURF, SIFT) in different conditions (natural and urban landscapes, camera mobility when shooting, moving objects).

5. CONCLUSION

As a result of the study, the main stages and methods of stitching 360-degree video were determined. In addition, the algorithms used in the alignment are considered, as well as methods' performance estimates using different computational tools are given. This paper can be a starting point for developers of 360-degree video devices and video stitching software.

ACKNOWLEDGMENT

The work was financially supported by the Ministry of science and higher education of the Russian Federation (Grant Agreement 14.576.21.0102, the unique identifier for applied scientific research RFMEFI57617X0102).

REFERENCES

[1] Tsilindricheskaya panorama [Cylindrical Panorama]. https://ru.wikipedia.org/wiki/Цилиндрическая панорама

[2] Sfericheskaya panorama [Spherical Panorama]. https://ru.wikipedia.org/wiki/Сферическая_панорама

[3] Budagavi, M., Furton, J., Jin, G., Saxena, A., Wilkinson, J., & Dickerson, A. 360 Degrees Video Coding Using Region Adaptive Smoothing. In: 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015 (pp. 750-754). IEEE. (2015).

[4] Equirectangular Projection. https://en.wikipedia.org/wiki/Equirectangular_projection

[5] GoPro VR Player. Kolor GoPro. http://www.kolor.com/gopro-vr-player/download/

[6] Panoramic Photography. https://en.wikipedia.org/wiki/Panoramic_photography

[7] Create Panoramas. https://helpx.adobe.com/camera-raw/using/create-panoramas.html

[8] Xu, W. Panoramic Video Stitching. Boulder: University of Colorado. (2012).

[9] Szeliski, R. Image Alignment and Stitching: A Tutorial. Technical Report MSR-TR-2004-92. Redmond, WA: Microsoft. (2004).

[10] Pyramid (Image Processing). https://en.wikipedia.org/wiki/Pyramid_(image_processing)

[11] Lucas, B.D., & Kanade, T. An Iterative Image Registration Technique with an Application in Stereo Vision. In: Proc. Seventh International Joint Conference on Artificial Intelligence (IJCAI-81) (pp. 674-679). (1981).

[12] Badra, F., Qumsieh, A., & Dudek, G. Rotation and Zooming in Image Mosaicing. In Proc. IEEE Workshop on Applications of Computer Vision (pp. 50-55). IEEE. (1998).

[13] Understanding Features. OpenCV. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html#features-meaning

[14] Automatic Image Stitching with Accord.NET. Code Project. https://www.codeproject.com/Articles/95453/Automatic-Image-Stitching-with-Accord-NET

[15] Feature Detection and Description. OpenCV. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html

[16] Huang, K., Chien, P., Chien, C., Chang, H., & Guo, J. A 360-Degree Panoramic Video System Design. In *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*, Hsinchu, 2014 (pp. 1-4). (2014).

[17] Ho, T., & Budagavi, M. Dual-Fisheye Lens Stitching for 360-degree Imaging. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017 (pp. 2172-2176). IEEE.

[18] Ho, T., Schizas, I., & Rao, K.R., & Budagavi, M. 360-Degree Video Stitching for Dual-fisheye Lens Cameras Based on Rigid Moving Least Squares. In 2017 IEEE International Conference on Image Processing (ICIP), 17-20 Sept. 2017. IEEE. (2017).

[19] Blair, K., & Wu, S. Real Time Video Stitching. Muncie, Indiana: Ball State University, 2016. (2016).

[20] Liao, W.S., Hsieh, T.J., & Chang, Y.L. (2012). GPU Parallel Computing of Spherical Panorama Video Stitching. In 2012 IEEE 18th International Conference on Parallel and Distributed Systems (pp. 890-895). IEEE. DOI: 10.1109/ICPADS.2012.146.

[21] OpenCL. https://ru.wikipedia.org/wiki/OpenCL

[22] Du, C., Yuan, J., Chen, M., & Li, T. Real-Time Panoramic Video Stitching Based on GPU Acceleration Using Local ORB Feature Extraction. Journal of Computer Research and Development, 54(6), 1316-1325. (2017).

[23] CUDA. https://ru.wikipedia.org/wiki/CUD

[24] ORB (Oriented FAST and Rotated BRIEF). OpenCV. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html