

A Panoramic Video System

Imran Achmed, James Connan
Department of Computer Science

University of the Western Cape, Private Bag X17 Bellville, 7535, South Africa

Telephone: +(27) 21 959-3010, Fax: +(27) 21 959-3006

Email: 2507311@uwc.ac.za, jconnan@uwc.ac.za

Abstract—Video systems are used to present visual data to users. This paper describes the development of a panoramic video system using off-the-shelf commodity hardware. This system captures multiple video sequences using multiple cameras, with each camera at a specified angle. Each video sequence has a limited field-of-view. The video sequences are transferred to a computer where the panoramic video system merges the video sequences to form a single wide field-of-view video. The system makes use of tools such as Barrel Transformation, Perspective Transformation, Gray-scaling, Edge Detection and Chamfer Distance Transformation.

Index Terms—Chamfer Distance Transformation, Panoramic Video, Perspective Transformation.

I. INTRODUCTION

Technology is continuously evolving. Panoramic Video has been around for a while with many applications that can be created and incorporated with it [1]. Existing systems are costly and require specialized hardware; however, we propose a more cost effective way in developing such systems using off-the-shelf hardware not intended for 3D capturing.

Panoramic Video is similar to a QuickTime Virtual Reality [2] scene where there is movement in all directions and there are many applications for it, such as video conferencing, class room lectures or even as virtual tours of hotels, museums or exotic travel destinations. Google StreetView is another example of a novel application. This is a feature of Google Maps and Google Earth that creates 360° panoramic views of the streets in several countries by using a specialized camera system.

Image Panoramas, in which multiple photos are taken in a sequence from a single point of view and then merged into one large image, date back to the mid-19th century [3]. Viewed interactively on a computer with software such as QuickTime Virtual Reality, image panoramas offer a much more immersive experience to the user than the usual photographs with constricted fields of view [3].

In addition to this, a range of image/video combination approaches, which include video elements playing inside an image panorama, have been attempted [3].

While still image panoramas are common [4], we are interested in developing a less common type of panorama that makes use of video instead of standalone images.

By using multiple cameras, we are able to simultaneously record video sequences. Each of the cameras should overlap the adjacent camera [5] [6], in order for a matching area to be identified [7].

The video sequences are transferred to the computer. Once transferred, the system maintains precise frame synchronization. The video sequences are merged and a single wide field-of-view video sequence is produced. The delivery is done with post-processed video, as doing it in real time is computationally expensive.

Our approach is thus to generate a new seamless panoramic video that retains and merges pixels from the original videos while respecting the dynamic appearance thereof.

II. BACKGROUND ON PANORAMIC VIDEO

Conventional visual systems portray a two-dimensional field-of-view (FOV). These systems fall short on providing a realistic sense of depth. Furthermore, normal video is restricted to a small part of the scene, based on the human eye's ability to capture a larger part of the scene [8]. The objective of this system is to address both of these issues.

Not everyone owns cameras such as the fish-eyed lens camera that can capture a wide-field of view. Most cameras can only capture a limited field-of-view, both horizontally and vertically. This limitation cannot be solved by using multiple cameras. Thus a system is needed to link the video sequences in a synchronized form. The system we present hereupon is a solution to this.

III. IMPLEMENTATION DETAILS OF THE SYSTEM

We now describe the system. The system has both hardware and software components, where the hardware component captures the video data and the software component process this data. In the subsections below, we describe each of these components in detail.

A. Hardware Component

1) Framework Design



Fig. 1. The special equipment used to capture 360° FOV [7].

Special equipment, which makes use of an omni-directional camera on a customized framework, as depicted in fig. 1 is used to capture a 360° field-of-view with just one

camera [9]. To capture video sequences in a 360° field-of-view, the system makes use of multiple cameras. The overall framework is designed to produce a wider field-of-view. We initially used two cameras to capture an approximately 180° field-of-view but the system is built in such a way that it can easily be extended to capture 360° field-of-view by adding more cameras. The cameras are statically calibrated by the user in such a way so as to take into account the orientations as well as lens distortions and mirror effects.

According to research done by [10], the cameras should be placed close together in such a way that their centre of projection coincides. Arranging the cameras in this way makes the merging process easier. It was found that between the views of the two adjacent cameras, an overlap range of 30% to 50% works well in aiding the merging process [3].

2) Multiple Cameras

Single cameras, such as the fish-eye lens camera, capture a wide area. An alternative to this is multiple commodity cameras which are less expensive. The fish-eye lens camera proves to be inefficient for use in panoramic related tasks. Planar projections from the lenses of the camera do not evenly or fully cover the pixel array, thus complicating the merging process. The disadvantage of using multiple cameras is that it requires calibration for each of the cameras and viewpoints, which often is the case with far-field scenes [4].

3) Camera Calibration

Initially the user will need to set up the cameras to capture the scene. All cameras are adjusted in such a way that each camera viewpoint overlaps the other. In each overlapped area there is a virtual chessboard for the camera calibration to take place as illustrated in fig. 2 [11].

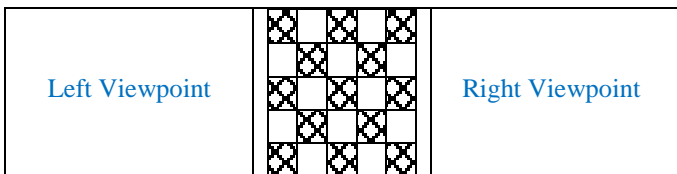


Fig. 2. The center area is the overlapped area for matching.

Each of the cameras will have several blocks overlapping each other and thus providing a matching area for the videos. The scene used will be a 2D model and the only way or approach to follow to get the 2D model to work correctly, is if the video sequences are at the same depth. In other-words all objects of the scene are supposed to be at almost the same distance from the cameras [11]. If there is a difference in distance from the cameras to the objects, it would cause distortion on the panoramic video and we would have to build 3D models instead of 2D [11].

The distortion can be decomposed into three major components as explained in [10], a shift of the optical centre of the cameras, radial distortion and de-centring distortion. It might not be visible when viewing a single image but when stitching multiple images in the video sequences the distortion may cause a noticeable mismatch. The term images are used as videos are made up of frames and these frames can be represented as images.

4) Video Capturing

The cameras are connected to the PC. Video sequences from the camera are recorded so that it can be processed by

the system. Since multiple video sequences are to be recorded, it is important to synchronize the video sequences accurately. If this is not achieved the merging and rendering process may incorrectly use the frames of different time instances to build the panoramic video and thus cause discontinuity in the final panoramic video [10].

In order to properly synchronize the videos, we ensure that each channel receiving a video feed is captured at the same frame rate. The cameras used, have a maximum frame rate of 25 frames per second. Due to latency produced, a frame rate of 8 frames per second is achieved. At other times, 10 frames per second is achieved, but this change usually occurs when the lighting in a particular scene is reduced.

5) Image Acquisition

In capturing a video sequence, the camera only has a limited field-of-view. By using multiple cameras a wide field-of-view can be obtained. The problem with using multiple cameras is that a common optical centre is required. This can be overcome by mounting the cameras on a specially designed framework that it is portable as shown in fig. 3.



Fig. 3. The specially designed framework on which the cameras rest.

6) Image Registration

Transformation recovery makes use of the fact that two spatially neighbouring images are related to each other by a homography transformation [12]. Homography is the relationship between two frames such that any given point in one frame corresponds to one and only one point in the other. By recovering the homography transformation, images can be merged together and ultimately a video can be created. Methods in [12] explain that to recover the homography transformation, the methods rely on the detection of image features, with which the camera motion can be computed. Automatic feature extraction methods will fail when distinctive features are absent [10].

B. Software Component

In this section we discuss high end processing such as Barrel Transformation, Perspective Transformation, Gray-scaling, Edge Detection and Chamfer Distance Transformation. A brief explanation to illustrate why these techniques are needed follows.

1) Distortion

Lenses found in cameras reproduce an object or view in such a way that its proportions are preserved correctly over the entire image, and remain the same without any

distortions. In other words a straight line should remain a straight line and a square should remain a square. It is often the case that when capturing video footage or even still images, straight lines on the edges of the picture are slightly bent or curved. This is known as distortion on the video or image. In our case, the equipment used caused barrel distortion in the video.

2) Barrel Transformation

Barrel distortion is a lens effect which causes images to be spherical in shape or perceived to be inflated [13]. We used a Colour Compact Mini Camera with Pinhole Lens that produced this kind of distortion. Converters in the software used to capture the videos amplified the effect. Thus, in order to eliminate this distortion we needed to apply the transformation.

The distortion is implemented based on a set of 4 coefficient values. The correcting function is a third order polynomial. This function relates the distance of a pixel from the center of the source image (r_{src}) to the corresponding distance in the corrected image (r_{dest}) [13]:

$$r_{src} = (A * (r_{dest} * 3) + B * (r_{dest} * 2) + C * r_{dest} + D) * r_{dest} \quad (1)$$

3) Perspective Transformation

Perspective transformation generates a perspective by viewing the 3-D space of an image from an arbitrary view point. The capturing of the videos is done in 2-D space and we find that when placing two cameras alongside each other at an angle the view point changes as depicted in figure 4 [14].

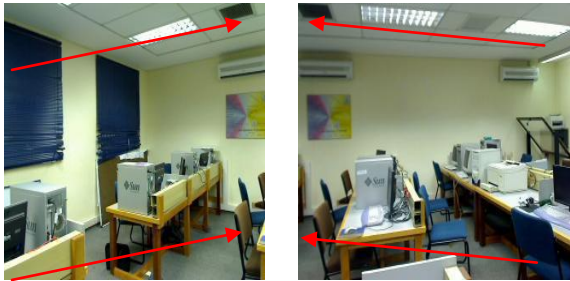


Fig. 4. The left image and right image alongside each other.

On completing the Barrel Transformation on the images, we eliminate the barrel distortion found and we are left with a perspective on the image as seen in fig. 4. If we assume the cameras have a planar image surface and that there are no lens distortions, then the mathematical form of the transformation is rather straightforward [15]:

$$x = \frac{X \cdot f}{Z}, y = \frac{Y \cdot f}{Z} \quad (2)$$

These are the equations of the perspective transformation. The constant f is defined as the focal length of each of the cameras, and thus represents the distance from the optical centre of the lens to the image plane. X , Y and Z specify the position of a point in space; where x and y specify the position of its image in the image plane. These equations collectively allow us to calculate x and y if we already know the values for X , Y and Z . This process is not reversible,

since knowing x and y is not sufficient to find the centre position of the view point in 3-D [15].

Thus by choosing the appropriate coordinates and applying this perspective transformation to an image we are able to straighten the image so that it can be matched as seen in fig. 5.

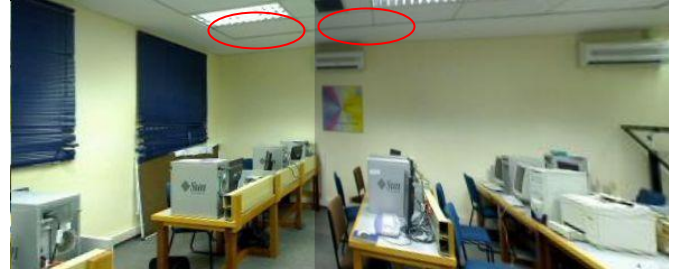


Fig. 5. Perspective Distortion corrected.

4) Grayscale

A grayscale image is an image in which each pixel has a single sample value. The reason for converting the image from colour to grayscale is because less information needs to be provided for each pixel and it serves as a preparation for the edge detection. Thus, in order to convert a colour image to grayscale we will need to apply certain ratios to the pixels provided in the image, [18] i.e.:

$$(R * 30) + (G * 59) + (B * 11) = \text{Grayscale Pixel} \quad (3)$$

where R is the value for every red pixel, G is the value for every green pixel and B is the value for every blue pixel.

5) Edge Detection

Edge detection is an important task in the process of finding the matching areas within the images. These edge detection techniques are a combination of filtering, thresholding and logic [16]. There are several edge detection techniques that can be followed but in this system we make use of the Marr-Hildreth Algorithm. The advantage to using this algorithm is the simplicity and the fact that all edges are closed loops, thus convenient in segmenting the objects. This algorithm for edge detection is based on the zero-crossings of the Laplacian of the Gaussian operator as seen in the equation below:

$$\nabla^2 g(x, y) = \frac{1}{\sigma^2} \left(\frac{x^2 + y^2}{\sigma^2} - 2 \right) e^{-\left(\frac{x^2 + y^2}{2\sigma^2} \right)} \quad (4)$$

This method yields fine results since the Gaussian operator smoothes the image and the Laplacian operator computes the second derivative. Edges are then detected by applying a matrix mask and a threshold to the grayscale image [16]. Thus we achieve a transformation as seen in fig. 6.

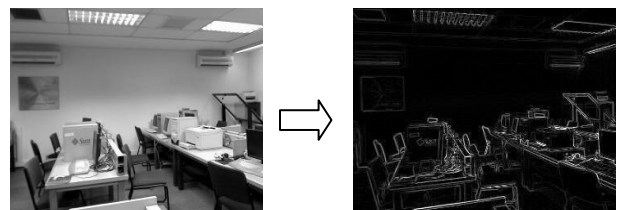


Fig. 6. Converting grayscale to edge detection.

6) Distance Transformation

There are several types of Distance Transformation algorithms and the one used in the Panoramic Video System is based on the Chamfer Distance Transformation algorithm. This Distance Transformation technique operates by assigning values to each pixel based on the distance from the pixel to the nearest edge. This is a crucial method and necessary to match the left image with the right. This concept resembles the one in figure 7 below:

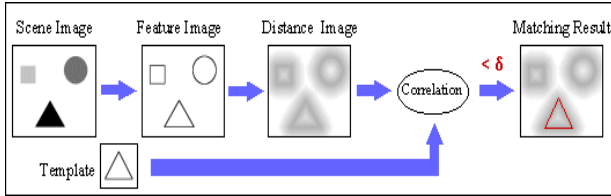


Fig. 7. Finding a matching area using the Chamfer Distance Transformation [17].

The Chamfer Distance Transformations are produced by using a mask and scanning twice over an image. First the forward scan is executed, by starting in the upper left corner of the image, then moving from left to right and from top to bottom. On completing the forward scan, the backward scan is executed. This scan begins in the lower right corner, then moving from right to left and from bottom to top as depicted in fig. 8. From this we obtain local distances, for example d_1 and d_2 . These local distances represent the pixel values. After each scan these pixel values are added and the minimum of the five sums is assigned to the central pixel as the mask moves along. On completion of the distance transformation, pixels near to an edge will have a lower pixel value and those further from an edge, a higher pixel value [17].

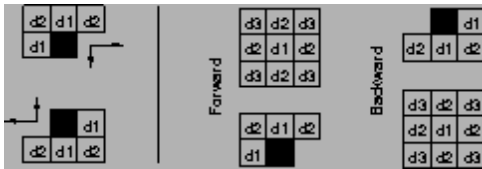


Fig. 8. The procedure in which values are assigned to pixels [17].

We have only provided a basic procedure to the way the Chamfer Distance Transformation functions as this is all that was needed. For further information regarding the Chamfer Distance Transformation, see “A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching” by Dariu M. Gavrilă [17].

IV. TESTING

A. Testing the System

Two different tests were done on the system itself. The performance tests were not done as the frame rate is dependent on the computer used; however, the tests were visually analyzed.

1. Testing the system for continuity in the joint frames.
2. Testing the system for motion sensitivity.

1) Testing the System for Continuity in the Joint Frames

Three different tests were performed.

- The first was done to test the continuity at the point where the frames are joined.
- The second was done to test the continuity while rotating the framework of the camera system.
- The third was done to test the continuity while rotating the framework of the camera system on an unstable and unsteady surface.

a) Purpose of Test

One of the objectives in creating this system is to deploy the system on a moving vehicle, as the inspiration for this came from Google StreetView. These tests were performed as a preparation for this particular objective and were thoroughly run to ensure this objective would be a success.

b) Result of Test

Successive tries were carried out to test different situations in testing the continuity between the points where the frames are joined. In fig. 9, we show the results illustrating the test case. It is shown that the videos are seamlessly joined.



Fig. 9. Testing the continuity in the panorama where the frames are joined.

Successive attempts were carried out to test the different situations in testing whether movement would cause any discontinuity. We tested this by rotating the camera system. In fig. 10, we show the results. The results show that the videos are seamlessly joined.





Fig. 10. Testing the continuity of the panorama when the camera framework is rotated.

The third test case was not successful but we found the system to handle the instability at certain times. At times the discontinuity is masked by the movement and therefore not perceptible to the viewer. Assumptions were then made that this discontinuity could be compensated for by the movement of the vehicle.

2) Testing the System for Motion Sensitivity

Finally in this test case, we deployed the camera system on the rooftop of a vehicle.

a) Purpose of Test

This test was done on one of the objectives as stated earlier, i.e. deploying the system on a moving vehicle, so as to demonstrate the camera system as well as the Panoramic Video System holds for this case. Being able to show this suggests we were successful in obtaining this particular objective.

b) Result of Test

We found the movement compensated for the instability. On achieving this success, we conclude our testing as we achieved what we set out to achieve. Below, fig. 11, we find images of the final panoramic video created while driving on University of the Western Cape premises. The left and right viewpoints, however, are distinguishable but this is due to the fact that lighting affects the cameras in different ways even though the same cameras were used. It becomes increasingly identifiable when the angle at which the camera is placed is receiving a larger fraction of sunlight. Furthermore, as the vehicle moves forward, the lighting continuously adjusts automatically in both cameras; nevertheless this does not affect the matching process.



Fig. 11. Testing the continuity of the panorama for motion sensitivity.

V. CONCLUSION

In this paper we discussed the development of a panoramic video system that may be used for various applications. We have illustrated the portability of the framework designed and shown that it was possible to mount the system on the rooftop of a vehicle. This framework was designed to provide for the necessary overlap between the cameras. The video sequences that are transferred to the computer contains barrel distortion but is resolved by doing a barrel transformation. Perspective transformation is performed, followed by gray-scaling and edge detection. Finally distance transformation is used to provide the matching area for the panoramic video to be created. The system was thoroughly tested and found to perform well under most circumstances.

Although systems exist that are able to create panoramic videos, these come at a huge expense and require specialized hardware. We have shown a novel implementation that is more cost effective using off-the-shelf hardware and simple ordinary algorithms. Furthermore, the system is portable by simply tweaking the camera calibration to the extent that it was possible to place the system on a moving vehicle.

We have only implemented this system using 2 cameras and was therefore unable to achieve a 360° field-of-view. As future work, we suggest the Panoramic Video System be expanded by using more cameras to achieve this.

ACKNOWLEDGMENT

We wish to thank all those that contributed to this paper for their much appreciated suggestions and criticism.

REFERENCES

- [1] J.E. Coleshill and A. Ferworn. 2004. Spherical panoramic video for micro-gravity applications. 55th International Astronautical Congress, Oct 4-8, 2004.
- [2] QuickTimeVR
<http://www.apple.com/quicktime/technologies/qtvr/>,
retrieved on 11th August 2008.

- [3] A. Agarwala, K. Zhang, C. Pal, M. Agarwala, M. Cohen, B. Curless, D. Salesin and R. Szeliski. 2005. Panoramic video textures. ACM SIGGRAPH 2005 Conference Proceedings, July 25, 2005.
- [4] T. Pintaric, U. Neumann and A. Rizzo. 2000. Immersive panoramic video. Proceedings of the 8th ACM International Conference on Multimedia, pp. 493-494, October 2000.
- [5] Panoram Technologies Inc. www.panoramtech.com, retrieved on 22nd September 2008.
- [6] Fullview.com inc. www.fullview.com, retrieved on 25th September 2008.
- [7] Panoramic viewing system with offset virtual optical centers. US Patent 6111702.
- [8] S. Tzavidas and A.K. Katsaggelos. 2005. A multi-camera setup for generating stereo panoramic video. NorthWestern University, USA, Volume 7, Issue 5, pp. 880 - 890, October 2005.
- [9] J. Baldwin, A. Basin and H. Zhang. 1999. Panoramic video with predictive windows for tele-presence applications. In Proceedings 1999 IEEE International Conference on Robotics and Automation, Detroit, USA, May 1999.
- [10] W. Tang, T. Wang and P. Heng. 2005. A system for real-time panorama generation and display in tele-immersive applications. Multimedia, IEEE Transactions on, Vol. 7, No. 2. pp. 280-292
- [11] A. Klechenov, A.K. Gupta, W. F. Wong, T. K. Ng and W.K. Leow. 2002. Real-time mosaic for multi-camera videoconferencing. November 1, 2002.
- [12] D. Browning, C. Neira, C. Sundin and T. DeFonti. 1993. The cave automatic virtual environment: projection-based virtual environments and disability. In Proceedings of the First Annual International Conference, Virtual Reality and People with Disabilities, January 1993.
- [13] H. Dersch. 1999. Correcting barrel distortion. Technical University Furtwangen, July 14, 1999.
- [14] Y. Izawa, H. Kadoya and H. Yamamoto. 1998. Image processing system and method including perspective transformation of three-dimensional objects utilizing clipping plane positions. June, 1998.
- [15] D. Young. 1993. Stereoscopic vision and perspective projection. February 1993.
- [16] H. Rhody. 2005. Image feature detection. pp. 2 – 10, November 3, 2005.
- [17] D. M. Gavrilu. 2007. A bayesian, exemplar-based approach to hierarchical shape matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 8, August 2007.
- [18] D. Nie, Q. Ma, L. Ma, and S. Xiao. 2007. Optimization based grayscale image colorization. International Association for Pattern Recognition, Pattern Recognition Letters, Vol. 28, Issue 12, 1 September 2007, Pages 1445-1451.

James Connan heads up the South African Sign Language (SASL) research group. He has a wide range of interests that include: databases, computer vision and machine learning.

Imran Achmed is currently a Telkom Centre of Excellence M.Sc. student at the University of the Western Cape. He is currently doing research on sign language synthesis and novel communication applications for the Deaf and hard of hearing.