Robot Aided Stereo Panorama

Kushagr Gupta, Suleman Kazi

Department of Electrical Engineering
Stanford University
Stanford CA, 94305
Email: {kushagr , sbkazi}@stanford.edu

***Abstract*—This article describes the generation of a stereo panorama from images acquired by the rotation of a single camera. The camera is mounted on a robotic arm which ensures stability and a steady rotational velocity. Strips are taken from the right and left side of the collected images. These strips are then mosaiced together to form two images. The strips from the right side form the image for the left eye and vice versa. An automatic disparity control algorithm is then applied to the obtained stereo pair to control disparity. The stereo panorama obtained can be viewed on virtual reality displays or using other methods like anaglyphs.**

I. Introduction

With an increase in the number of commericaly available consumer virtual reality (VR) devices, techniques for VR compatible content generation and display has garnered interest. Images and videos acquired using a single camera cannot be directly displayed on these devices as they lack stereo depth information. In order to provide an immersive user experience stereo content needs to be developed. This content can be computer generated / animated or can be pictures and videos from the real world. Stereo panoramas provide the ability to view scenes from the real world on VR devices. Since they provide a complete $360°$ view they can allow the user to turn around and rotate their head to see around them. As opposed to traditional monocular panoramas which only have one single viewpoint, stereo panoramas provide different images to the left and right eye and hence create depth perception.

II. Data Acquistion

In order to acquire the data to generate the stereo panorama a KUKA IIWA robotic arm (Fig. 1) is used. A camera is mounted on the end-effector and the robot is rotated about one of it's joints to capture a video at a recording rate of 30 frames per second. The robot rotates at a rotational velocity between $2^{\circ} - 3^{\circ}$ per second. Due to the rotational limits of the robot a full 360° view cannot be achieved, instead, only 340° of rotation are possible which means that one video acquired consists of between 1800 and 2700 frames. After acquiring the video, frames are extracted from it using a freely available 3^{rd} party software (*"Free Studio"*). Preprocessing of the frames involves rotating them to account for slight misalignments of the camera. The images are then also corrected for lens distortion. In order to perform undistortion, camera parameters must be estimated. This is done by calibrating (Fig. 2) the camera using the MATLAB camera calibration toolbox. The vertrical and horizontal field of views of the camera are also calculated. In the intital cases of acquiring data, different QR code markers were added at locations throughout the scene being captured in order to have objects at known depths in the scene.

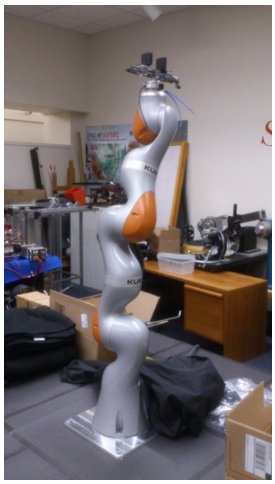

Fig. 1. The KUKA IIWA robotic arm

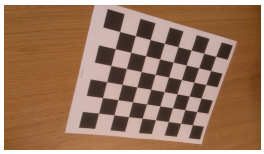

Fig. 2. Pattern used for camera callibration

III. Stereo Panorama Generation

In this project we adopted the philosophy as mentioned in [1], wherein the stereo (two views- one for each eye) panoramas (wide field of view image, upto $360°$) are generated using images from one single camera. In order to generate two views from one camera we use the concept of circular viewpoint projections wherein both the left ad the right eye images share the same cylindrical image surface. Hereafter, in order to enable stereo perception, left and the right viewpoints lie on a circle (known as the viewing circle) which is concentric and inside the image surface circle. The viewing direction is

on a line tangential to the viewing circle. Left eye projections use rays tangential to the viewing circle in a clockwise manner and the right eye projections use rays tangential to the viewing circle in anticlockwise manner. This concept is illustrated in Fig. 3.

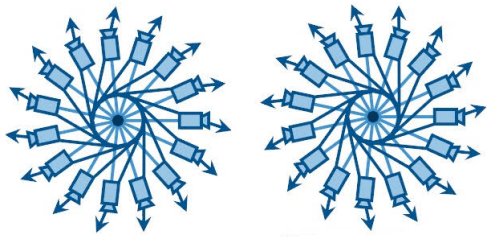

Fig. 3. Right and Left Eye Panoramas (Source: [2])

A. The Setup

The camera is rotated about an axis behind it i.e. about an axis of rotation which does not lie at camera's optical centre. This distance between axis of rotation and optical centre (arm length) is a function of the disparity and thus helps in stereo vision and depth perception. A video is captured using this setup and frames extracted which are used to produce stereo panorama. Now, in order to generate the panoramas two strips are extracted from all the frames at a constant distance from the center. The right strip corresponds to the ray which falls tangentially on the viewing circle for the left eye and left strip corresponds to the ray which falls tangentially on the viewing circle for the right eye.

B. Strip Location Calculation

Once we have the frames it is important to choose a slit location which will provide decent disparity variation such that our eyes can fuse the images together and get depth perception due to stereo views. Thus, we fix the baseline (inter-pupiliary distance) to be $6.5cm$ and calculate the strip location from the centre of frame from it. Fig. 4 illustrates the concept, which is explained in detail hereafter.

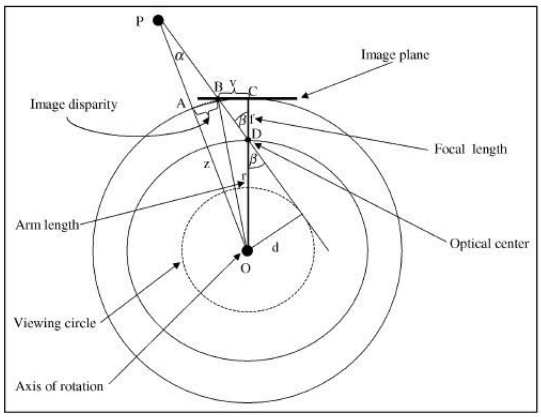

Fig. 4. Strip location calculation (Source: [1])

Distance d in the figure represents radius of the viewing circle on which the eyes are located. As the baseline is $6.5cm$ and our eyes don't lie diametrically opposite about the head centre but on a chord of this circle, the diameter of this circle is more than the baseline. We therefore, choose the diameter to be $8cm$ and thus $d = 4cm$ (radius). For one of our experiments we chose arm length to be $r = 32cm$. The focal length of the camera used without the crop factor that is usually present in mobile cameras was $4mm$. Crop factor for the phone chosen (Xperia Ion) is 7x since focal length = $28mm$ with it, as per the website www.dpreview.com. Sensor size of the phone camera is $\frac{1}{2.1}$ and using these values we calculated the strip gap (v) from centre. The frame size in pixels is 1920×1080 (vertical x horizontal). The calculations are shown as follows:

$$\beta = sin^{-1}(\frac{d}{r})$$

$$v = f \times tan(\beta)$$

$$v_{pixel} = \frac{v}{sensorsize} \times 1080$$

From these calculations we get the location of strips, and then the panoramas are generated for right and left eye which when viewed on a VR headset give the depth perception.

C. Improving depth perception -Automatic Disparity Control

Fixing the strip gap from centre also fixes the disparity in all directions which isn't a very good approximation of the real world view, as our eyes can tolerate a range of disparity upto 0.5ž. Thus in order to improve perception of depth we use a technique known as automatic disparity control as described in [1] which helps in accomplishing the said task. The procedure is explained hereafter:

1) After obtaining the stereo panoramas with constant strip gap, we align the two images such that we have vergence at infinity, which means both the eyes view the farthest point in the scene with no disparity.

2) For each column in the image, we take a window of size 50 pixels in three regions of the image namely top, middle and bottom and calculate the mean squared error between this image and the other image.

3) Maximum of the three is taken which tells us the range of depth variation in the scene i.e. the variation between closest and farthest object.

4) This signal is filtered using a median filter in order to obtain a smoothened output which tells us about the regions where there is significant variation in depth. If there is significant variation in depth, we donot want to increase the disparity as it will end up in a situation where the eyes won't be able to fuse the whole region appropraitely. In the regions where there isn't much variation in depth, i.e. the range is less, we can increase disparity which will help improving the depth perception in that region. Thus by following this method we can modify the disparity of the scene

locally which gives improved depth perception.

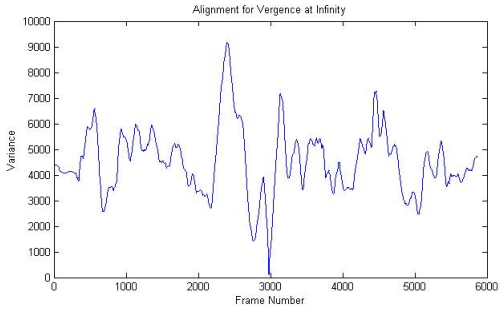

Fig. 5. Alignment for vergence at infinity

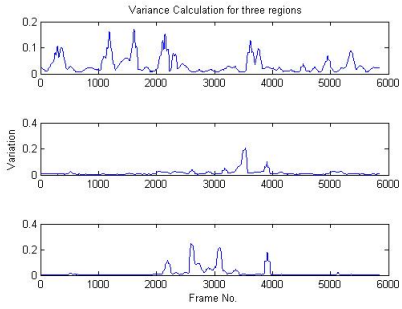

Fig. 6. Variance at three locations in the image (top, middle, bottom)

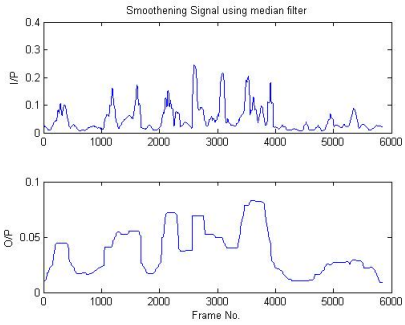

Fig. 7. Filtered maximum variance

IV. Head Translation

An improvement over the generated stereo panorama was to incorporate the effects of head translation while viewing stereo content with the suggestion of Dr. Hari. This is illustrated in Fig. 9. The concept is explained as follows. Head translation to the left corresponds to extracting slits from the original frames that are translated to the right by a certain number of pixels. Similarly head translation to the right corresponds to extracting slits from the original frames that are translated to the left by a certain number of pixels. The calculation of number of pixels to

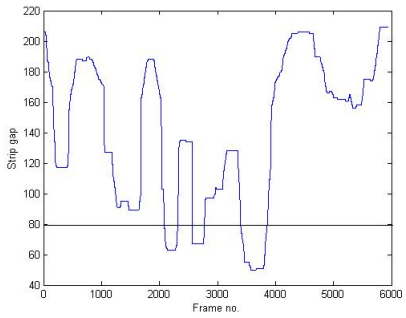

Fig. 8. Strip gap updated after ADC

translate in a direction in order to get new views is as follows:

$$SHC = SceneHorizontalCoverage$$
$$SC = SceneDepth$$
$$HFOV = HorizontalFOV$$
$$HT = HeadTranslation$$
$$PT = PixelTranslation$$
$$SHC = SD \times HFOV$$
$$PT = \frac{HT}{SHC} \times ImageWidth$$

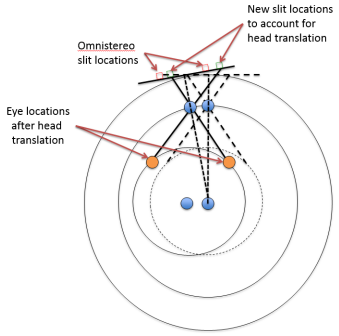

Fig. 9. Head Translation (Courtesy Dr Haricharan Lakshman)

V. Post processsing and display

In order to view the right and left eye panoramas in stereo, two methods were employed

A. Anaglyphs

Anaglyphs are images that enable viewers to perceive 3D depth. They contain two different images, each of which is displayed to the left and right eye resepectively. This is done by using colored filters (typically red an blue) so that only one image is viewable by each eye. MATLAB's stereo vision toolbox provides a convenient way to generate an anaglyph from a stereo pair. This was used to generate a stereo panorama as shown in Fig. 10.

B. VR Display

A samsung gear VR device was used also used to display the images. Since out images were not a complete spherical 360°view, zero padding was done, this produced the effect of having black regions in the image displayed on the VR headset where the image regions were not in the field of view of the camera while acquiring the images. We created a function in MATLAB to carry out this process, with input as two panoramas. The zero padding to be used was determined-using the following equations:

$$Zeros_{Hor} = \frac{NumPixel_{HOR}}{FOV_{HOR}} \times 360 - NumPixel_{HOR}$$

$$Zeros_{Vert} = \frac{NumPixel_{HOR}}{FOV_{HOR}} \times 360 - (2 \times NumPixel_{VER})$$

A processed image for VR display is shown in Fig. 11.

VI. Future Work

As the demand for virtual reality gears and panoramic views increases, the need of stereo image content will increase. This tells us about the practical scope of the project and the improvements which need to be incorporated in order to obtain better results.

1) The technique adopted is prohibitively computationally taxing and takes a long time to generate stereo panoramic views which can be viewed. Thus one of the improvements over this would be to extract lesser number of images from the video and stitch the slits using some kind of blending technique which doesn't add more artifacts to the scene. One of promising techniques we encountered was the optical flow blending technique as described in [2]. It extracts the optical flow information from the video using Lucas-Kanade gradient based displacement estimation. Intermediate views are generated using the input images, optical flow information and then interpolating between them. This would help in reducing computation time and also remove the artifacts which creep into views when stitching is done.

2) For this project we used robotic arm to capture data so that hand jitter and motion would not be a factor to account for. But for it to be applicable in real world, we need to be able to robustly remove such effects as majority of the videos will be captured by hand.

3) The techniques can be used to generate 360° fully immersive spherical views rather than cylindrical views which were generated in this project.

4) By using multiple cameras and modification of these techniques it is possible to generate 360° stereo videos which would give the ultimate immersive real-life experience.

VII. Experimental Results

The final stereo panoramas generated were displayed on the Sasmusng Gear VR headset and also as anaglyphs. An anaglyph of a stereo panorama generated during the project is shown in Fig 10. and an image ready to be viewed on the VR headset is shown in Fig. 11.

VIII. Appendix

The raw videos collected for this project are available online at: https://goo.gl/axTkIx

The project workload was divided between the group members. **Kushagr Gupta:** Data acquistion, Omnistereo algorithm implementation - automatic strip gap calculation disparity control. **Suleman Kazi:** Data acquistion, pre and post processing and camera calibration, manual depth map disparity control.

ACKNOWLEDGMENT

We would like to thank Dr. Haricharan Lakshman and Matt Yu for their support and advice during the project, Professor Bernd Girod and Professor Gordon Wetzstein for teaching this informative class and the Stanford robotics group for access to the group's robotic arms during the project.

References

[1] Peleg, S.; Ben-Ezra, M.; Pritch, Y., "Omnistereo: panoramic stereo imaging," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.23, no.3, pp.279,290, Mar 2001

[2] Richardt, C.; Pritch, Y.; Zimmer, H.; Sorkine-Hornung, A., "Megastereo: Constructing High-Resolution Stereo Panoramas," Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on , vol., no., pp.1256,1263, 23-28 June 2013

[3] Peleg, S.; Ben-Ezra, M., "Stereo panorama with a single camera," Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. , vol.1, no., pp.,401 Vol. 1, 1999

Fig. 10. Stereo Panorama

Fig. 11. Panorama Processed for VR Display