Rotation Removed Stabilization of Omnidirectional Videos Using Optical Flow

Sarthak Pathak, Alessandro Moro, Atsushi Yamashita, Hajime Asama
Graduate School of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

Omnidirectional cameras are often used with robots to provide an immersive view of the surroundings. However, robots often have unstable motion and undergo rotation. In this work, we formulate a method to stabilize the viewpoint of Omnidirectional videos by removing the rotation using dense optical flow fields. The method works by first projecting each Omnidirectional video frame on a unit sphere, measuring the optical flow at every point on the sphere, and finding the direction that minimizes its rotational component in a frame by frame manner. The Omnidirectional video is de-rotated and a 'rotation-less, translation-only' viewpoint is generated. Such a technique is well suited to work in any environment, even with sparse texture or repeating patterns where feature correspondence based methods may fail.

1 Introduction

Omnidirectional cameras are becoming more and more popular for use with drones or other robots for surveying and other tasks as the videos produced can be viewed from any desired angle, thereby providing an immersive view of the surroundings. However, the robots that they are placed on, especially drones and vehicles, have inherently unstable motion and undergo rotation. Thus, there is a need to stabilize videos obtained from these cameras to be able to 'fix' a particular orientation at which the video can be viewed, or to change the viewpoint as desired.

A similar problem of Omnidirectional video stabilization has been solved by feature based approaches [1] or by using other sensors [3]. While feature based methods can easily fail in scenes with repeated elements (bridges, corridors, buildings, etc.), methods based on sensor fusion are difficult to calibrate and synchronize. Meanwhile, correspondence-less methods based on densely optimizing error functions across the entire image have been quite successful with Omnidirectional images [2]. Furthermore, for stabilization of videos taken from normal cameras, optical flow is often used [4].

Hence, for our problem of rotation stabilization, we try to follow a correspondence-less approach similar to [2]. In this work, we take a cue from video stabilization approaches [4] and attempt the use of optical flow along with a minimization that de-rotates the omnidirectional video to a 'rotation-less, translation-only' viewpoint.

2 Spherical Images and Optical Flow

There are many ways to obtain Omnidirectional videos. For this research, we have used the popular Ricoh Theta m15 camera, which provides a completely stitched Omnidirectional video in an equirectangular projection through two internally calibrated fisheye cameras. This equirectangular image is projected to a unit sphere to give us a spherical image (to equalize distortion everywhere). If S is the unit sphere,

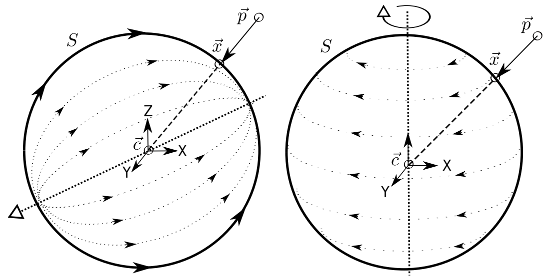

Fig. 1 Optical flow patterns on a Spherical Image for (a) Pure Translation (b) Pure Rotation.

the image $\vec{x}$ of any point $\vec{p}$ in space is given as the intersection of the ray from $\vec{p}$ towards $\vec{c}$ (the sphere center) with the surface of S, as shown in Figure 1.

Now, considering that our camera is moving within a scene, some patterns of optical flow will be generated on the spherical image. These patterns were discussed in great detail by Nelson and Aloimonos [5]. Specifically, they discussed optical flow patterns produced when the camera undergoes pure rotation and translation (Figure 1). For pure translational motion, the optical flow vectors emerge from a point and converge at a diametrically opposite point, while for pure rotational motion, they form loops around the rotation axis. Since our spherical image can be rotated in any direction without loss of information, there must exist an orientation to which it can be rotated such that the flow field resembles a pure translation-like pattern.

As mentioned before, for our problem, the aim is to obtain a precise frame-by-frame estimate of rotation, which is usually quite small. Therefore, this estimate can be obtained by optimizing the orientation to make the flow field resemble pure translational motion.

3 Optimizing the Flow Field and De-rotating

The first step is to calculate the flow between the current spherical image frame and the preceeding frame. We can then keep rotating the current frame and reprojecting the flow accordingly, till we reach the angle for which the flow field resembles a pure translation-like pattern.. Thus, we need to minimize the rotational component, which can be represented as the moment of Optical Flow vectors after Magnitude Normalization (to account for the 3D structure of the environment) around the center, as shown in Equation 1:

$$\vec{M} = \sum_{\forall \vec{x_i} \in S} (\vec{x_i} X \vec{F_i}) \qquad (1)$$

where $\vec{M}$ is the moment, $\vec{F_i}$ is the Normalized Optical Flow vector, and $\vec{x_i}$ ($\forall \vec{x_i} \in S$) is the radius vector of the point. This moment is largely due to rotation of the camera. Hence, our desired angle can be computed by minimizing the moment of the optical flow using the Levenberg-Marquardt optimization. Assuming a small rotation, the initial value can be provided as zero. Thus, our minimization problem can be posed as given in Equation 2:

$$\underset{\alpha,\beta,\gamma}{\text{minimize}}|\vec{M}| = \underset{\alpha,\beta,\gamma}{\text{minimize}}|\sum_{\forall \vec{x_i} \in S} (\vec{x_i} X \vec{F_i})| \tag{2}$$

where α, β, and γ are the rotation angles along the $z-y-z$ Euler axes. Because of the dense optimization, the choice of the optical flow algorithm or of textureless regions and occlusions do not affect the results, as long as there at least sparse, well distributed textures throughout the scene. In our case, we used the popular Farneback approach [6] to calculate dense flow fields in the equirectangular image and projected them to the sphere.

After the optimization, the frame is de-rotated to the orientation of the previous frame. Thus, in a frame-by-frame manner, we can obtain a rotation-less sequence.

4 Experiments

First, the algorithm was evaluated for drift using an angular graded tripod. It was found to drift around $5°$ in $240°$ of rotation in one direction, which is acceptable given that a robot's motion will not be in one direction continuously and that the system can always be reset at will. It works at a speed of 5 frames per second on Omnidirectional video sequences of 500×250 pixels on an Intel Core i7 1.8 GHz processor and could be made realtime on a more powerful processor.

Further, experiments were conducted using an AR Drone 2.0 and a Pioneer P3-DX robot undergoing arbitrary motion. Some equirectangular frames from both sequences and their de-rotated versions are shown in Figures 2 and 3. In each sequence, the red rectangle indicates a $120°$ FOV (that of wide angle cameras and human vision) and the green rectangle indicates a relevant object of interest. In each de-rotated frame, the object of interest remains inside the red rectangle, demonstrating the rotation-less, translation-only effect.

5 Conclusion

This method can be used to obtain rotation-less video frames from an Omnidirectional camera placed on a robot to provide a stable immersive viewpoint of the surroundings. It requires no extra sensors or calibration apart from the Omnidirectional camera. The effectiveness of the technique can be observed from the resultant frames shown in Figures 2 and 3. Further work can be done to implement a global rotation estimation method to remove drift.

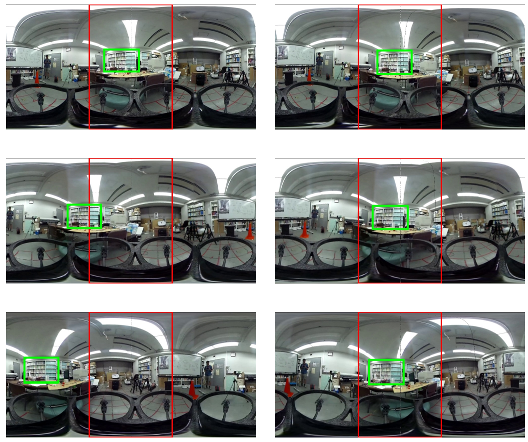

Fig. 2 AR Drone Sequence Frames 34, 70, and 91. De-rotated frame on the right. (Equirectangular Projection). Red: 120 degree FOV. Green: object of interest

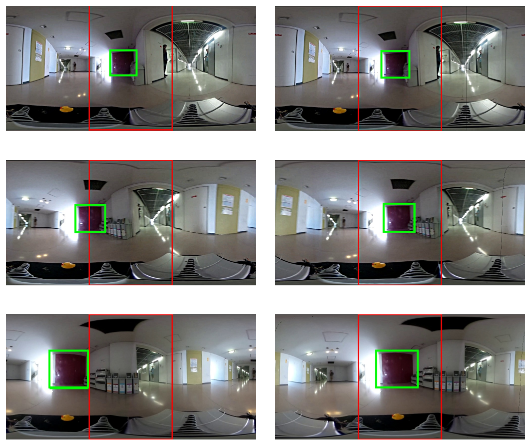

Fig. 3 Pioneer Sequence Frames 97, 130, and 170. De-rotated frame on the right. (Equirectangular Projection). Red: 120 degree FOV. Green: object of interest

Acknowledgements

This work was in part supported by the SIP Program (Cross-ministerial Strategic Innovation Promotion Program).

References

[1] M. Kamali, A. Banno, J. C. Bazin, I. S. Kweon, and K. Ikeuchi, "Stabilizing omnidirectional videos using 3D structure and spherical image warping", *Proceedings of the 2011 IAPR Conference on Machine Vision Applications*, pp.177-180.

[2] A. Makadia, L. Sorgi, and K. Daniilidis, "Rotation estimation from spherical images", *Proceedings of the 2004 International Conference on Pattern Recognition*, Vol.3, pp.590-593

[3] T. Albrecht, T. Tan, G. West, and T. Ly, "Omnidirectional video stabilization on a virtual camera using sensor fusion," *Proceedings of the 2010 International Conference on Control Automation, Robotics, and Vision*, pp.920-925.

[4] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H. Y. Shum, "Full-Frame Video Stabilization with Motion Inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.28, No.7, pp.1150-1163, 2006.

[5] R. C. Nelson and J. Aloimonos, "Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head)," *Biological Cybernetics*, Vol.58, no.4, pp.261-273, 1988.

[6] G. Farneback, "Two-frame motion estimation based on polynomial expansion", *Lecture Notes in Computer Science*, pp.363-370, 2003.