A DECOUPLED VIRTUAL CAMERA USING SPHERICAL OPTICAL FLOW

Sarthak Pathak, Alessandro Moro, Atsushi Yamashita, Hajime Asama

The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

ABSTRACT

In camera-equipped teleoperated robots, it is often tedious for the operator to manage both the viewpoint and the shaky/unstable navigation, leading to disorientation. Our proposal is to create a virtual, freely rotatable camera that is decoupled from the robot's rotation. It is implemented using a complete spherical camera and removing its rotation in-image with a novel algorithm based on aligning the dense spherical optical flow field along the epipolar direction. Finally, any area on the rotation-less image sequence can be undistorted, resulting in the desired decoupled camera. We illustrate the concept by showing the effect on some videos taken from a spherical camera under different robot motions.

***Index Terms*—** Robot Teleoperation, Spherical Camera, Video Stabilization

1. INTRODUCTION

Teleoperated robots with cameras are often used for surveying in dangerous environments. Usually, the camera viewpoint is tied to the robot motion. [8] and [7] talk about how a fixed viewpoint can affect cognitive judgment and lead to disorientation. They explain how a decoupled, independently rotatable camera can increase search performance and reduce disorientation. The operator can fix the camera for navigation, or decouple it for surveying, avoiding repetitive position adjustments of the robot. However, even with a Pan-Tilt-Zoom (PTZ) apparatus, the operator would still need to adjust for the unstable robot motion. Meanwhile, recently developed spherical cameras (e.g. Ricoh Theta [15]) can provide an all-round view. Since they capture information from all directions, any camera rotation can be completely reversed or 'derotated' and adjusted as desired in-image. Any area on the sphere can be easily unwarped to a perspective view in that direction (Figure 1). Thus, they form good candidates for implementing such a decoupled camera. Hence, we aim to stabilize spherical image sequences by removing their rotation, allowing for a virtual camera that can be independently oriented.

This work was in part supported by the 'Cross-ministerial Strategic Innovation Promotion Program (SIP), Infrastructure Maintenance, Renovation, and Management', Council for Science Technology and Innovation (funding agency: NEDO). Contact: `<last-name>@robot.t.u-tokyo.ac.jp`

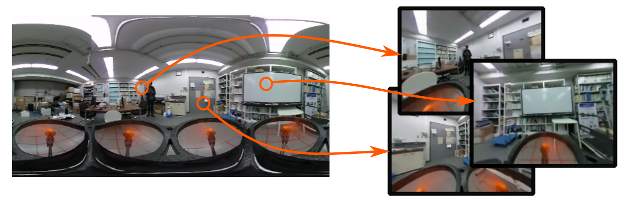


Fig. 1. Spherical images (in Equirectangular Projection) can be unwarped at any direction. Such images can be rotated without information loss.

2. RELATED RESEARCH AND OUR APPROACH

Along similar lines, many have used spherical cameras to stabilize/decouple camera rotation. [1] and [17] implement a virtual decoupled camera using additional sensors to estimate rotation and continuously derotate each spherical video frame. [10], [19], and [11] use feature point matching for the same. [12] also uses feature-point based methods to stabilize spherical images to provide an immersive view on a wearable headset. However, methods with additional sensors need difficult synchronization and data fusion and it is well known that point-matching based can be unstable towards outliers/repeated textures.

Meanwhile, dense optical flow methods like [16] are very popular for perspective video stabilization and can work in all kinds of situations. [20] also explained how regularization in dense optical flow prevents outliers and results in accurate epipolar geometry. Similarly, we attempt the same on spherical videos. Long ago, the authors of [18] discuss the patterns of flow formed on a spherical camera undergoing pure rotation and translation (Figure 2). They explain that since any spherical camera motion is a combination of the two, any optical flow field on the sphere can be separated as such. Thus, they pose rotation estimation as a derotation-based pattern recognition problem and theoretically suggest multiple searches for the 5-DoF motion parameters along 3 separate axes. This coupling of translation and rotation parameters along 3 different searches could be quite slow. Several others ([9, 3]) attempted similar approaches to estimate the full 5-DoF motion. [21] also follows a similar technique for stabilizing the roll and pitch motions of a mobile robot. However, they assumed the robot motion to be non-holonomic.

For our concerned application, only rotation estimation is necessary. Hence, the main contribution of this paper is a simple but novel dense minimization of the rotational component of the optical flow field without solving the complete 5-DoF epipolar estimation problem. Estimating the rotation independently is computationally lighter, results in lower errors, and makes it applicable for the full range of 3D motions and any kind of translation. The use of dense optical flow results in a particularly smooth stabilization. The popular Ricoh Theta spherical camera has been used in this research. In the remainder of this paper, we first explain the theoretical basis behind our method, then explain the minimization used for derotation, followed by error evaluations and a few examples.

3. SPHERICAL OPTICAL FLOW

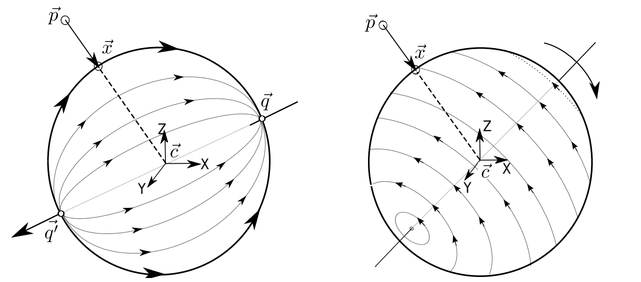
As mentioned earlier, spherical motion fields were discussed in great detail in [18]. For pure translational motion, the optical flow vectors are seen diverging away from a pole $\vec{q'}$ and converging towards a diametrically opposite pole $\vec{q}$ i.e. aligned along the epipolar lines from $\vec{q}$ - $\vec{q'}$. As for pure rotational motion, the optical flow vectors form parallel loops around the rotation axis. Both are shown in Figure 2.

Any arbitrary motion of the camera forms a combination of these two fields. Since we can obtain the flow from any direction (unlike a perspective camera), we can theoretically distinguish the translational and rotational components of the motion field [18]. Moreover, rotations in spherical images are completely recoverable without any information loss. Hence, it should be possible to 'derotate' the frame and re-compute the optical flow to reach an orientation at which the motion field is perfectly aligned according to the epipolar direction, i.e. a translational state. Based on this, we formulate our derotation technique.

Figure 3 shows the real estimated optical flow vectors for a purely translational movement and a purely rotational movement (both generated by accordingly moving the camera on a tripod) respectively, in the spherical projection. They were estimated using a simple Farneback algorithm [4] on the equirectangular projection followed by an appropriate transformation to the spherical projection [5]. Patterns similar to Figure 2 can be observed.

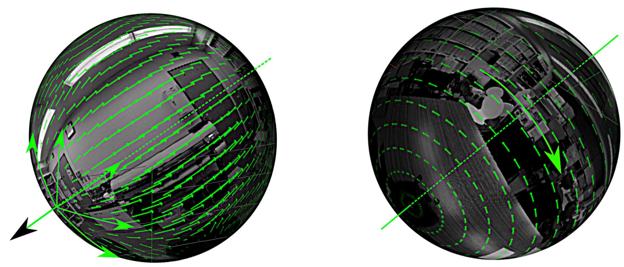
4. EPIPOLAR ALIGNMENT AND DEROTATION

Thus, we can conclude that will always be an orientation to which a frame can be derotated such that the motion field aligns along the epipolar vector i.e. a pure translational state. Without knowing this vector, it can be said that in this symmetrical state, the *moment* of the optical flow vectors about the center of the sphere would be zero, if their magnitudes were exactly the same (Figure 2). To remove the depth dependence of the optical flow vectors, they can be divided by their magnitudes and reduced to unit length. Essentially, we



(a) Translation (b) Rotation

Fig. 2. Motion fields on the spherical image for the camera undergoing (a) Pure Translation (b) Pure Rotation.



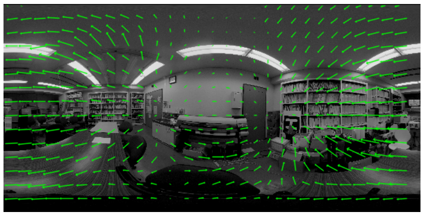
(a) Translation (b) Rotation

Fig. 3. Computed real motion fields on the spherical image for (a) Translation (b) Rotation. Patterns similar to those in Figure 2 above can be noticed.

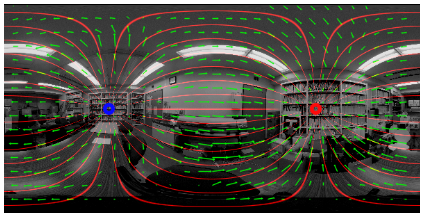
are not interested in their magnitudes or the epipolar vector in itself, but only in aligning the *direction* of all optical flow vectors along it, similar to the translational patterns in Figures 2 and 3. Any deviation from this alignment will cause a moment around the center. For every point i on the sphere S, if $\vec{F_i}$ is the optical flow vector (relative to point i) and $\vec{x_i}$ is the radius vector, the total moment is (Equation (1)):

$$\vec{M} = \sum_{\forall i \in S} (\vec{x_i} \times \frac{\vec{F_i}}{|\vec{F_i}|}) \tag{1}$$

Thus, our basic idea is to search for the angle at which the optical flow vectors align according to an epipolar vector, i.e. with zero or minimal moment $\vec{M}$ (Equation 1). The frame can be derotated and flow vectors re-computed till we reach such an angle - which would give us the rotation between two frames. We pose this as a Levenberg-Marquardt minimization [14] of the total moment. Instead of re-computing the optical flow field in every iteration, we simply re-project it based on the initial state resulting in a much higher speed. Assuming a small frame-to-frame rotation (to maintain validity of optical flow vectors as well), we choose an initial value of zero rotation and the optimization converges quickly to the closest minima. If the camera motion is too large, an initial value



(a) Before Optimization: Translation + Rotation



(b) After Optimization: Translation flow with epipoles and epipolar lines

(c) After Optimization: (a)-(b), Rotation Only

Fig. 4. Example: Derotating a single frame: Optical flow in equirectangular projection (a) before and (b) after optimization. In (a), Rotation + Translation patterns are superimposed and unsymmetric. In (b) and (c), separated patterns similar to Figure 3 can be noticed. The red and blue points indicate the epipoles, and red lines are epipolar lines (drawn manually). It can be seen that the flow vectors are aligned along them.

can also be obtained with the usual 8-point RANSAC [6]. In accordance, we chose the optimization space to be that of the Euler angles α, β, and γ with fixed $x-y-z$ axes to maintain a small search space. Thus:

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

where $\vec{F_i}$ has been calculated after derotation with the current iteration's estimates of α, β, and γ. Figure 4 shows an example of this minimization in two frames. The distribution of optical flow vectors before optimization is a superimposition of rotation and translation. After optimization, the optical flow vectors appear to be similar to that of pure translation.

We accumulate the rotation and derotate every frame in this manner to obtain the 'translation-only' viewpoint for the entire sequence. We can then choose to undistort it to a perspective view in any desired direction to give us the corresponding virtual camera viewpoint.

5. EXPERIMENTS AND PERFORMANCE

The algorithm runs at 5 FPS with equirectangular images of 250×500 pixels without any parallel processing on a Core i7 laptop. Since our application is about keeping a fixed viewpoint, drift error is an inevitable concern. Hence, the accuracy was evaluated by flying an AR Drone 2.0 in a room equipped with a motion capture system for groundtruth. The proposed optical flow method was used as a refinement over an 8-point RANSAC implementation (with 200 A-KAZE features [2] per frame) and the results were compared against the estimation using feature points alone. Figure 5 shows the absolute errors. The average drift errors in 40 seconds, given below in Table 1 are stable enough for our purpose.

Table 1. Average Drift Errors in 600 frames (40 seconds)

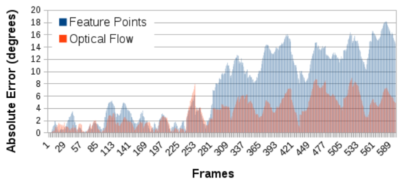
Average Drift Rate	**Roll**	**Pitch**	**Yaw**
Feature Points (deg.)	7.52	3.86	10.80
Optical Flow (deg.)	3.41	1.59	3.99

Further, spherical videos collected from a moving AR Parrot Drone 2.0, a Pioneer P3-DX robot, and random hand movement were also processed. Figures 6 and 7 show some sample frames along with the virtual decoupled perspective views[1]. For lack of space, only the front view has been shown. The same can be done for any direction on the sphere to obtain the corresponding decoupled view. It can be seen that in the decoupled frames, the orientation remains fixed for all kinds of rolling, pitching, and yawing motions, as opposed to the haywire motion in the unstabilized case.

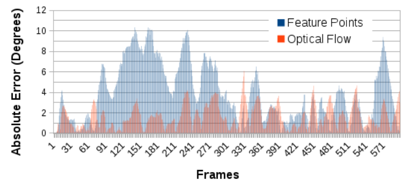
6. DISCUSSION AND CONCLUSIONS

Aiming to implement a freely rotatable, decoupled camera, this method estimates and removes the rotation in a single optimization without solving the complete 5-DoF epipolar problem. The use of dense optical flow gives a smooth decoupled view, as seen in the resultant videos[1]. The main motivation was to ease the dual-cognitive problem of surveying and operating robots [8][7]. Since it handles all kinds of roll, pitch, and yaw motions with any kind of translation, it can also be used on drones to provide an aerial decoupled camera (much easier than rotating the drone itself). Similar to [13] and [12], it can be complemented with VR headsets. Moreover, it only utilizes an inexpensive, commercially available spherical camera (Ricoh Theta). The spherical camera essentially serves two purposes: providing the image information, as well as optical flow information from all directions in order to distinguish and separate the translational and rotational flow patterns. Future work involves increasing the speed, using some global features (vanishing points, etc.) to eliminate drift, and a Kalman filter for smoothing.

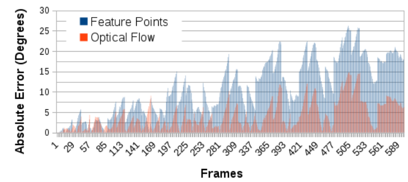
[1] Videos at: `http://tiny.cc/stabilization`



Absolute Roll Errors

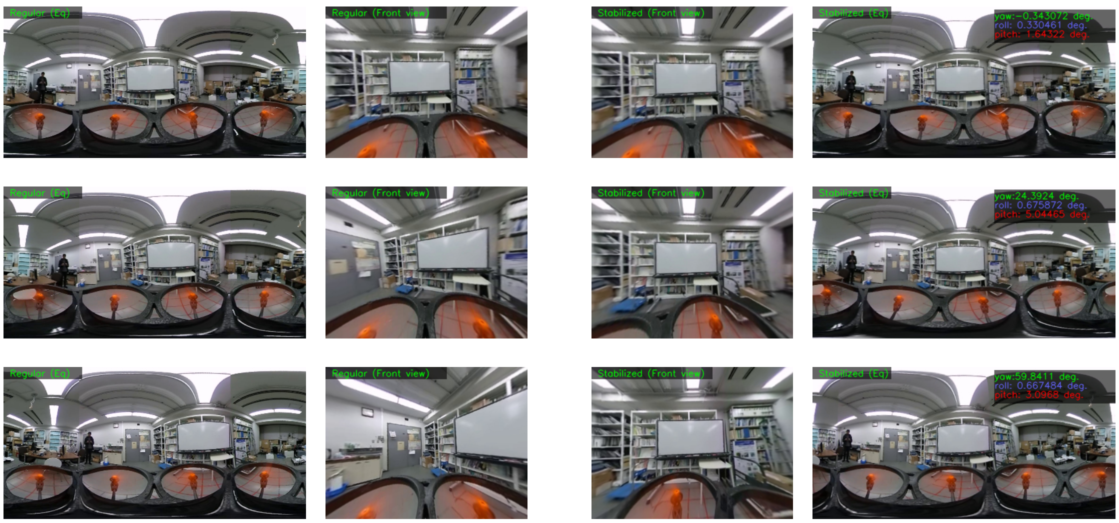


Absolute Pitch Errors



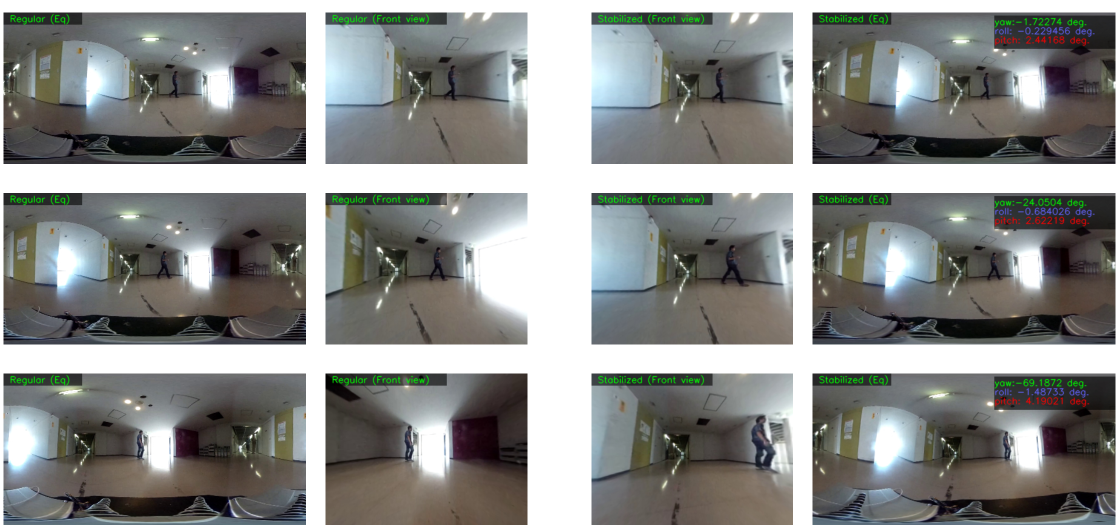
Absolute Yaw Errors

Fig. 5. Absolute Errors (Motion Ranges: Yaw: ±180 degrees, Pitch/Roll: ±30 degrees)



Regular Equirectangular   Regular Perspective   Decoupled Perspective   Decoupled Equirectangular

Fig. 6. Frames 77, 85, and 111 (top to bottom) from the AR Drone sequence: Equirectangular view and perspective undistorted front view, regular and decoupled. The fixed orientation in the decoupled camera can be noticed.



Regular Equirectangular   Regular Perspective   Decoupled Perspective   Decoupled Equirectangular

Fig. 7. Frames 100, 106, and 120 (top to bottom) from the Pioneer Robot sequence: Equirectangular view and perspective undistorted front view, regular and decoupled. The fixed orientation in the decoupled camera can be noticed.

7. REFERENCES

[1] T. Albrecht, T. Tan, G.A.W. West, and T. Ly. Omnidirectional video stabilisation on a virtual camera using sensor fusion. In *Proceedings of the International Conference on Control Automation Robotics Vision*, pages 2067–2072, December 2010.

[2] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proceedings of the British Machine Vision Conference*, September 2013.

[3] A. Briod, J.C. Zufferey, and D Floreano. A method for ego-motion estimation in micro-hovering platforms flying in very cluttered environments. *Autonomous Robots*, 40:1–15, September 2015.

[4] G. Farneback. Two-frame motion estimation based on polynomial expansion. *Lecture Notes in Computer Science*, 2749:363–370, June 2003.

[5] J. Gluckman and S. K. Nayar. Ego-motion and omnidirectional cameras. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 999–1005, January 1998.

[6] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:580–593, June 1997.

[7] S. Hughes and M. Lewis. Robotic camera control for remote exploration. In *Proceedings of the SIGCHI conference on Human factors in Computing Systems*, pages 511–517, July 2004.

[8] S. Hughes, J. Manojlovich, M. Lewis, and J. Gennari. Camera control and decoupled motion for teleoperation. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 1339–1344, October 2003.

[9] T.W. Hui and R. Chung. Determining motion directly from normal flows upon the use of a spherical eye platform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2267–2274, June 2013.

[10] M. Kamali, A. Banno, J. C. Bazin, I. S. Kweon, and K. Ikeuchi. Stabilizing omnidirectional videos using 3d structure and spherical image warping. In *Proceedings of the IAPR Conference on Machine Vision Applications*, pages 177–180, June 2011.

[11] M. Kamali, S. Ono, and K. Ikeuchi. An efficient method for detecting and stabilizing shaky parts of videos in vehicle-mounted cameras. *SEISAN KENKYU*, 66:87–94, January 2015.

[12] S. Kasahara, , and J. Rekimoto. Jackin head: An immersive human-human telepresence system. In *SIGGRAPH Asia 2015 Emerging Technologies*, pages 14:1–14:3, November 2015.

[13] K. Kruckel, F. Nolden, A. Ferrein, and I. Scholl. Intuitive visual teleoperation for ugvs using free-look augmented reality displays. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4412–4417, May 2015.

[14] M.I.A. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in c/c++. `http://www.ics.forth.gr/~lourakis/levmar/`, July 2004.

[15] Ricoh Company Ltd. Ricoh theta. `https://theta360.com/en/`.

[16] 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*, 28(7):1150–1163, May 2006.

[17] R. Miyauchi, N. Shiroma, and F. Matsuno. Development of omni-directional image stabilization system using camera posture information. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 920–925, December 2007.

[18] 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*, 58:261–273, March 1988.

[19] A. Torii, M. Havlena, and T. Pajdla. Omnidirectional image stabilization by computing camera trajectory. *Advances in Image and Video Technology, Springer Berlin Heidelberg*, 5414:71–82, January 2009.

[20] L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert. Dense versus sparse approaches for estimating the fundamental matrix. *International Journal of Computer Vision*, 96:212–234, January 2012.

[21] Y. Yagi, W. Nishii, K. Yamazawa, and M. Yachida. Stabilization for mobile robot by using omnidirectional optical flow. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 618–625, November 1996.