Stabilizing Omnidirectional Videos Using 3D Structure and Spherical Image Warping

Mostafa Kamali*, Atsuhiko Banno*, Jean-Charles Bazin*, In So Kweon† and Katsushi Ikeuchi*
*The University of Tokyo, {mostafa, vanno, jcbazin, ki}@cvl.iis.u-tokyo.ac.jp
†Korea Advanced Institute of Science and Technology, iskweon@ee.kaist.ac.kr

Abstract

This paper addresses the problem of stabilizing spherical videos. Whereas various techniques have been proposed to stabilize conventional videos, this work is the first approach proposed for omnidirectional videos. We introduce a method for extracting the camera path and 3D information of the environment. A desired smooth stabilized path is obtained by modifying the original camera path. We propose a method for synthesizing a stabilized video with respect to the desired path. Each output frame is generated by warping a single frame from the input video. Experiments on real data show that the proposed approach removes shakes and jitters from omnidirectional videos successfully.

1 Introduction

A wide gap exists between the quality of amateur and professional videos. One of the most important reasons is the unwanted motions of camera which implies shakes and jitters in amateur videos, which implies shakes and jitters in amateur videos, while shooting amateur videos. Whereas, professional videos are shot under controlled conditions with a smooth camera motion. The aim of *video stabilization* is to remove shakes from videos.

Existing methods are developed for *conventional* videos, which are obtained by perspective (pinhole) cameras. These cameras have a limited view angle. Cameras able to capture the entire environment are becoming more popular and being applied to various tasks [10, 18]. Frames acquired by these cameras are called *omnidirectional* because they can observe in all the directions.. Since omnidirectional images capture data from the entire environment, they contain more information. Hence, it is preferable to use these videos instead of conventional ones in numerous applications like autonomous vehicle driving, localization, and digital mapping [2].

Omnidirectional cameras usually consist of a cluster of conventional cameras. Each camera takes a picture of a portion of the space, then all these pictures are stitched to form a single image. Displaying omnidirectional images is not as straightforward as conventional ones. One method is to expand them on a 2D plane which is called the *panoramic representation* (Fig. 1-left). Another method to represent omnidirectional images is the *spherical representation* where the image is mapped on a unit sphere (Fig. 1-right). Since epipolar geometry properties for spherical images are similar to those of conventional images, we consider spherical images in this work.

Since pinhole camera properties are not satisfied in panoramic representation of omnidirectional images, we cannot apply existing methods for conventional images on them. Therefore, new approaches should be developed for omnidirectional images. Our work is the first method which can perform video stabilization for omnidirectional videos; specifically, for spherical images. We also propose a structure from motion technique for spherical images in this paper. One of the advantages of using spherical images is that we can apply this method on all central omnidirectional videos (e.g. aligned camera cluster, central catadioptric sensors, fisheyes, etc) since all of them can be represented in the spherical representation [1, 15]. Furthermore, in conventional image stabilization, some empty regions appear in the margin of the output image because of the deformation. To remove these regions, most methods crop images so that the resolution of the output video becomes lower than the input video. In spherical images we can naturally overcome this problem. Also, our method synthesizes each output frame using a single input frame; therefore, it prevents ghosting regions in the presence of dynamic objects.

Figure 1. Panoramic (left) and spherical (right) representation of omnidirectional images.

2 Related Work

Most video stabilization methods consist of three main steps: motion estimation, motion compensation, and image composition [16]. The task of motion estimation aims to compute the camera trajectory, given a motion model (e.g. homography, or essential matrix). The smoother these motions are, the less shakes the video has. Original motions are smoothed in the compensation step, and a stabilized video is synthesized in the third step. Stabilization methods can be divided into two main groups based on considering 2D or 3D motions.

2D approaches consider motions of interest points on the image plane. A common method is to study pixel motions between two successive frames by estimating a homography and smoothing all homographies in the whole sequence. In [6], authors use optical flow to detect a large planar region and fit a homography to that region. Although homography-based approaches perform well for planar scenes or when objects and

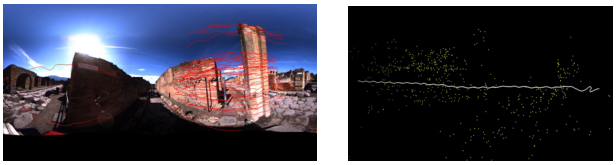

Figure 2. Trajectories extracted from the input video (left). SfM result: the camera path and 3D points corresponding to the trajectories (right).

buildings are far enough from the camera, they fail in highly non-planar scenes with different depths. To solve this problem, a 2.5D motion model is proposed in [7]. An approach for smoothing using Kalman filter is proposed in [11] and an image alignment method for motion estimation in [19]. Most of these methods damp shakes and jitters between two successive frames. In [9], authors propose a method to consider more than two frames. They track some interest points in the video and extract their trajectories. Then, these 2D trajectories are smoothed and the original frames are deformed with respect to them.

One of the drawbacks of considering 2D motions is that they do not contain all information of the motion because motions are happened in 3D. Hence, some people believe considering 3D motions instead of 2D motions leads to better results. 3D motion of a camera corresponds to its location and orientation while shooting the video. One may consider replacing the original camera path with a smoother one along which if the camera had moved, the video would have had less shakes. We propose a 3D approach for omnidirectional videos in this paper.

Different approaches have been proposed to smooth the original camera path and synthesize a new video for it. A method using motion inpainting for full-frame stabilization and deblurring is proposed in [14]. In [3], a method based on non-metric scene reconstruction for un-calibrated video sequences is proposed. It renders each output frame using multiple input frames. The drawback of this method is that dynamic objects are ghosting in output frames. To avoid this problem, authors in [12] propose a method which uses a single frame. They consider a grid on the input frame and find its corresponding grid in the output frame and use texture mapping to synthesize the output frame.

3 Obtaining a Desired Camera Path

In this section, we explain how the original camera path is estimated and a desired smooth path is obtained.

3.1 Extracting the Original Camera Path

We build a number of trajectories in the input video (Fig. 2-left) by detecting and tracking interest points in successive frames. Interest points can be detected and described using SIFT [13] on panoramic images or SIFTS [4] on spherical images.

These feature trajectories are used to obtain the camera path using *Structure from Motion* (SfM) for spherical images. The aim of SfM is to estimate a set of points cloud and camera poses. The points cloud $\{\mathbf{X}\}$ contains 3D points whose projections in the images are referred as trajectories. The camera poses consist of two components: location and orientation.

We propose an SfM method for spherical images in this paper. The camera poses of the first two frames of the sequence are estimated by decomposing the essential matrix using the method proposed in [17]. The points cloud is initialized by back projecting points observed from the first two frames and finding their intersections. The camera pose, location and orientation, for the third spherical frame is obtained by minimizing the reprojection error between projection of the points cloud and feature points obtained by SIFT. The points cloud is updated by adding new points which were not back projected previously. This process is performed for each frame in the sequence, frame by frame. After a certain number of frames, we refine the estimations of all camera poses and the points cloud obtained so far by minimizing the reprojection error.

Once camera poses of all frames in the sequences are estimated, all camera poses and the points cloud are refined by using a bundle adjustment method that we have developed for spherical images. Let $\mathbf{X}_p \in \{\mathbf{X}\}$ be corresponding to trajectory p. We project this point in a frame f where it is observed using equation

$$\mathbf{x}_{fp} = \mathbf{R}_f^\top (\mathbf{X}_p - \mathbf{T}_f) \tag{1}$$

where $\mathbf{R}_f$ and $\mathbf{T}_f$ are the rotation and translation, respectively, for frame f, and $\mathbf{x}_{fp}$ is on the unit sphere. Let $\mathbf{x'}_{fp}$ be the corresponding point to $\mathbf{X}_p$ extracted by SIFT in frame f. We minimize the angle $\Delta\theta_{fp}$ between all corresponding points $\mathbf{x}_{fp}$ and $\mathbf{x'}_{fp}$ as follows

$$\min_{\{\mathbf{X}_p, \mathbf{R}_f, \mathbf{T}_f\}} \sum_p \sum_f w_{fp} (\Delta\theta_{fp})^2 \tag{2}$$

where w_{fp} is 1 if $\mathbf{X}_p$ is observed at frame f and is zero otherwise. A result of SfM is shown in (Fig. 2-right).

3.2 Estimating a Desired Smooth Path

A desired path is one along which if the camera had moved, we would have had a less shaky movie. To obtain a desired path, we modify the pose of each camera so that the whole path becomes smoother. Camera locations can be represented as 3D translation vectors and camera orientations as rotation matrices. We smooth translation component by Gaussian smoothing applied element-wisely. Since the space of rotation matrices is not linear, theoretically rotation matrices cannot be smoothed element-wisely using Gaussian filtering. In [8], a method is proposed for smoothing a sequence of rotation matrices by transferring them into a linear space. However, experimental results show that if we consider quaternion representation of rotation and smooth them using Gaussian filter, we can obtain good results. Also, in specific scenarios we can fit a certain model to our data. For example, if we know that the camera moved on a straight line without rotation, we can assign a constant rotation to all frames. This can be performed by considering the average value of quaternions.

4 Synthesizing a New Video

After obtaining a desired path, we generate a stabilized video which would have been shot if the camera

had moved along this path. For each frame I from the original video, we synthesize a corresponding stabilized frame I'. 3D points from the point cloud obtained by SfM which are observed from I are likely observed from I' too. We project these points into both frames to obtain some corresponding points. These corresponding points guide us to warp I to obtain I'. However, these points are distributed sparsely on the image, and some regions may not contain any points. This causes an inaccurate warping. To generate an appropriate output frame, we consider a structure for I and preserve this structure in I'. Our approach is inspired from [12]. Their method is developed for conventional images but we propose a triangular grid on the surface of a unit sphere as the structure for omnidirectional images (Fig. 3-left). Corresponding points are utilized to find a similar corresponding triangular grid in I'.

Assume we have a triangular grid on spherical image I and we want to find its corresponding grid on I'. Let $\mathbf{P}_j$ and $\mathbf{P}'_j$ be the projections of a point from the point cloud in images I and I', respectively. Let $\mathbf{P}_j$ be inside triangle $\mathbf{V}_i$ whose vertices are $\mathbf{V}_i^1$, $\mathbf{V}_i^2$ and $\mathbf{V}_i^3$, and also assume $\mathbf{P}'_j$ is inside triangle $\mathbf{V}'_i$ whose vertices are $\mathbf{V'}_i^1$, $\mathbf{V'}_i^2$ and $\mathbf{V'}_i^3$. Vertices of triangles $\mathbf{V}_i$ and $\mathbf{V}'_i$ should represent corresponding points. We expect the relative position of $\mathbf{P}'_j$ to its surrounding triangle in I' to be similar to the relative position of $\mathbf{P}_j$ to its surrounding triangle in I because the structure of the grid is preserved. This relative position for $\mathbf{P}_j$ can be represented by linear combination

$$\mathbf{P}_j = \sum_{k=1}^{3} w_k \mathbf{V}_i^k. \tag{3}$$

Since the relative position of $\mathbf{P}_j$ and $\mathbf{P}'_j$ to their surrounding triangles are similar, the following equation should be satisfied for $\mathbf{P}'_j$

$$\mathbf{P'}_j = \sum_{k=1}^{3} w_k \mathbf{V'}_i^k \tag{4}$$

where w_k are computed in Eq. (3), and $\mathbf{V'}_i^k (k = 1, 2, 3)$ are unknown. We can estimate these unknown variables by solving the minimization problem

$$E_{data} = \sum_j \|\mathbf{P'}_j - \sum_{i=1}^{3} w_k \mathbf{V'}_i^k\|^2 \tag{5}$$

for all points $\mathbf{P}'_j$ projected from the points cloud.

Since we have just a few corresponding points between I and I', many triangles may not contain any points; hence, their vertices do not appear in Eq. (5). To solve this problem, we added a term which considers all vertices of the grid, similar to [12]. Since the new camera pose and the original one are close to each other, the relative positions of vertices in two corresponding triangles should be similar. A relative position proposed in [5] is utilized in this work.

Consider a triangle cell in the original frame, which lies on the surface of the unit sphere. Each vertex of this cell can be represented with respect to a local coordinate system whose axes are the direction of the line connecting two other vertices and its 90 degrees rotation (Fig. 3-right). This can be written as follows

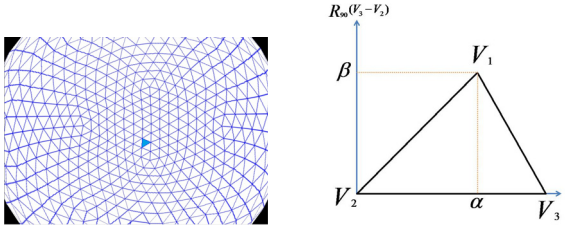

Figure 3. A uniform triangular grid on the unit sphere (left) which is used to preserve the structure of the input image, and a cell of this grid (right) to explain the relative location of one of its vertices with respect to other two vertices.

$$\mathbf{V}_1 = \mathbf{V}_2 + \alpha(\mathbf{V}_3 - \mathbf{V}_2) + \beta\mathbf{R}_{90}(\mathbf{V}_3 - \mathbf{V}_2) \tag{6}$$

where $\mathbf{R}_{90}$ is a 90 degrees rotation matrix in the plane on which the triangle lies, and α and β are the coordinates of $\mathbf{V}_1$ with respect to $\mathbf{V}_2$ and $\mathbf{V}_3$. Since we preserve the structure of the grid, the same relation should be satisfied for the corresponding cell in the new frame. We consider this for all vertices of all triangles. We will have the minimization term

$$E_{struct} = \sum \|\mathbf{V}'_2 + \alpha(\mathbf{V'}_3 - \mathbf{V}'_2) + \beta\mathbf{R}_{90}(\mathbf{V}'_3 - \mathbf{V}'_2) - \mathbf{V}'_1\|^2 \tag{7}$$

for all triangles in the grid.

To obtain the vertices of the grid in the output frame, we solve a constrained minimization problem whose objective function is

$$E = E_{data} + \lambda E_{struct} \tag{8}$$

and constraints are forcing each vertex to have unit length because they should lie on the unit sphere. We applied a simple Levenberg-Marquardt minimization, which provided satisfying results.

After obtaining the grid in the output image, we perform texture mapping to fill inside each triangle on the surface of the unit sphere to synthesize the final output.

5 Experimental Results

We applied our method on various videos shot by an omnidirectional camera, LadyBug2[1]. This system consists of 6 cameras and is able to capture data from more than 70 percent of the full sphere around it. Videos are very shaky because the system was mounted on a helmet and carried by a pedestrian. Results show that our approach removes shakes successfully.

First, we extract some trajectories from the input video. We use SIFT for detecting and describing interest points in the panoramic representation. Then, interest points are transferred on the equivalent unit sphere and next steps are performed considering the unit sphere. Trajectories on the sky are removed because they are not useful in SfM. Our proposed SfM method for spherical images is used to estimate the

[1] http://www.ptgrey.com/products/ladybug2/

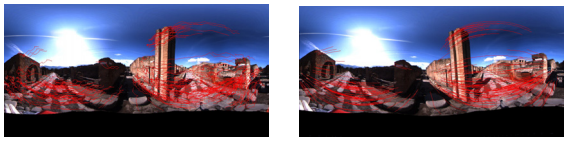

Figure 4. Feature trajectories before stabilization on the original frame (left) and after on the synthesized frame (right).

camera path and 3D structure. The original camera path is smoothed to obtain a desired path. We apply the Gaussian filter element wisely to the location component and quaternion representation of the orientation component. If the camera had moved along this desired path, the video would have not had shakes. We synthesize a new video with respect to the desired path.

To synthesize an output frame, we consider a uniform triangular grid on its corresponding spherical input frame. This grid has 642 vertices and partitions the surface of the unit sphere to 1280 triangles. The corresponding grid for the output frame is estimated using the method explained in section 4. Then, the content of the input frame is transferred to the output frame by mapping texture of each triangle from the input frame to its corresponding triangle in the output frame.

A comparison between point trajectories of the original video and stabilized video can show how our method works. If a video is shaky, its feature trajectories will have many jitters but if it is smooth, these trajectories will be smooth too. Trajectories in (Fig. 4-left) are extracted from the original video, and in (Fig. 4-right) from the stabilized video obtained by the proposed approach. It is easy to see that the trajectories in the output video are smooth while trajectories in the input video have many jitters. This indicates that our method succeeds to remove shakes.

6 Conclusion

This paper addresses the stabilization of omnidirectional videos. One of the contributions of this paper is proposing an SfM method for spherical images. The original camera motion is estimated using this SfM method and a desired trajectory is obtained by smoothing the original motion. Then, a new video is synthesized with respect to this desired path. This work is the first method for stabilizing omnidirectional videos. One of the advantages of using spherical images is that all representations of central omnidirectional systems can be converted to the spherical representation. Furthermore, in conventional videos the resolution of the output video is lower than the input video because frames are cropped to remove blank margins. In spherical images we can naturally overcome this problem. Also, our method can work in presence of dynamic objects because it uses a single image from the input video to synthesize its corresponding output frame. Experimental results on challenging real data show that our method works successfully.

Acknowledgments

This research is sponsored, in part, by Ministry of Education, Culture, Sports, Science and Technology under Digital Museum Project. It is also supported, in part, by Global Research Network Program (No. D00096(I00363)) of National Research Foundation of Korea.

References

[1] J.P. Barreto: "A Unifying Geometric Representation for Central Projection Systems," *CVIU*, 2006.

[2] J.C. Bazin, I.S. Kweon, C. Demonceaux and P. Vasseur: "A Robust Top-Down Approach for Rotation Estimation and Vanishing Points Extraction by Catadioptric Vision in Urban Environment," *IROS*, 2008.

[3] C. Buehler, M. Bosse and L. McMillan: "Non-metric image-based rendering for video stabilization," *CVPR*, 2001.

[4] J. Cruz, I. Bogdanova, B. Paquier, M. Bierlaire and J.P. Thiran: "Scale invariant feature transform on the sphere: theory and applications," *Technical Report, EPFL*, 2009.

[5] T. Igarashi, T. Moscovich and J. F. Hughes: "As-rigid-as possible shape manipulation," *ACM Transactions on Graphics*, 2005.

[6] M. Irani, B. Rousso and S. Peleg: "Recovery of ego-motion using image stabilization," *CVPR*, 1994.

[7] S.J. Jin, Z. G. Zhu and G. Y. Xu: "Digital video sequence stabilization based on 2.5D motion estimation and inertial motion filtering," *Real-Time Imaging*, 2001.

[8] J. Lee and S. J. Shin: "General construction of time-domain filter for orientation data," *IEEE Transactions on Visualization and Computer Graphics*, 2002.

[9] K.Y. Lee, Y.Y. Chuang, B.Y. Chen and M. Ouhyoung: "Video Stabilization using Robust Feature Trajectories," *ICCV*, 2009.

[10] S. Li and K. Fujiwara: "Full-view car navigator," *Conference on Automation Science and Engineering*, 2008.

[11] A. Litvin, J. Konrad and W. C. Karl: "Probabilistic video stabilization using Kalman filtering and mosaicking," *SPIE*, 2003.

[12] F. Liu, M. Gleicher and A. Agarwala: "Content-preserving warps for 3D video stabilization," *SIGGRAPH*, 2009.

[13] D.G. Lowe: "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.

[14] Y. Matsushita, E. Ofek, W. Ge, X. Tang and H.Y. Shum: "Full-frame video stabilization with motion inpainting," *PAMI*, 2006.

[15] C. Mei and P. Rives: "Single View Point Omnidirectional Camera Calibration from Planar Grids," *ICRA*, 2007.

[16] C. Morimoto and R. Chellappa: "Evaluation of image stabilization algorithms," *DARPA*, 1997.

[17] D. Nister: "An efficient solution to the five-point relative pose problem," *PAMI*, 2004.

[18] T. Pylvanainen, K. Roimela, R. Vedantham, J. Itaranta, R. Wang and R. Grzeszczuk: "Automatic alignment and multi-view segmentation of street view data using 3D shape priors," *3DPVT*, 2010.

[19] R. Szeliski: "Image alignment and stitching: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, 2004.