# EXPONENTIAL COORDINATES BASED ROTATION STABILIZATION FOR PANORAMIC VIDEOS

*Hoang-Phong Nguyen* *Tien-Thong Nguyen Do* *Jinwook Kim*
*hpnguyen@imrc.kist.re.kr* *dn.tienthong@gmail.com* *jwkim@imrc.kist.re.kr*

Imaging Media Research Center, Korea Institute of Science and Technology

**ABSTRACT**

We propose a practical algorithm for stabilizing panoramic video. We represent a stabilizing rotation by using exponential coordinates and compute the stabilizing rotation for all video frames by smoothing the shaky trajectories of visual features on the panoramic video images. Analytic derivatives of the optimization objective function involving stabilizing rotation matrices are formulated and thus enable efficient solvers for the optimization process. We also suggest a practical strategy to stabilize panoramic video interactively considering a user's viewing direction and further application scenario.

***Index Terms—*** panoramic video stabilization, exponential coordinates

## 1. INTRODUCTION

In last two years, 360 action cameras are getting more popular. Omni-directional videos captured from these cameras are extremely useful when watching freely in any direction with a virtual reality (VR) headset, or cropping in a regular narrow field-of-view later for some other purposes. However, the videos from the hand-held or wearable cameras are usually shaky; therefore, users viewing with head-mounted displays often become dizzy or nauseous [1]. It is called cyber-sickness [2].

Image stabilization is a fundamental technology to reduce cyber-sickness [3] as well as to improve the quality of the video. The best stabilization techniques use the mechanical tools, or optical or electronic devices; however, these methods are expensive and based on sophisticated camera sensors. In the paper, we consider digital video stabilization techniques using only image contents, especially the visual point features, for an interactive video player application in VR.

In [4], the authors computed camera path using structure-from-motion which is precise but slow, then, they estimated the desired smooth path by applying a Gaussian smoothing filter for translation and rotation separately. Finally, they used the spherical mesh-based warping technique similar to [5] to get a new image that compensates the camera shakes. However, due to the warping, image quality becomes worse, and this method is not really necessary for a VR application which requires fast computing and as low as possible camera acceleration.

Kasahara et al. [6] estimated 3D relative affine transforms between adjacent frames, then eliminated the rotational parts by multiplying with their inverted rotation matrices to make the shortest trajectories, but they did not consider the smoothness. In [7], Kopf improved the Kasahara's method by introducing one more term, a discrete Laplacian operator, to encourage the smoothness. However, his algorithm represents a stabilizing rotation by using 3-dimensional axis-angle parametrization which is more complicated than ours.

In our algorithm, the stabilizing rotation is represented by using exponential coordinates. It leads our proposed optimization objective function to compute the stabilizing rotations for all video frames more compact. Moreover, analytic derivatives of the objective function are formulated and enable efficient solvers for the optimization process. Our stabilization method (a) uses visual features directly instead of estimating the relative transform, (b) considers 3 degree-of-freedom rotation in the optimization process, and (c) ensures the smoothness of the trajectories.

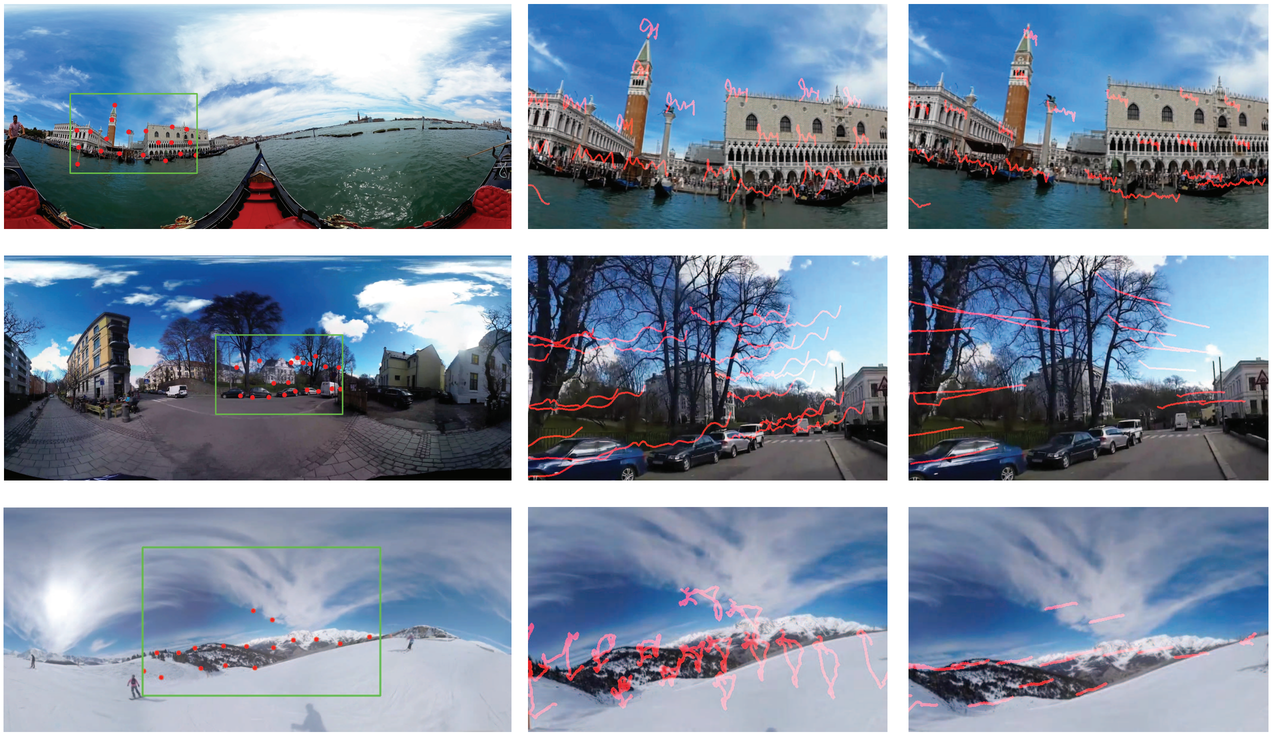
The rest of the paper is organized as follows. In section 2, our stabilization algorithm is explained including tracking visual features, selecting key frames, and defining and solving the objective function. Some experimental results are provided and one interactive application is described in section 3. Section 4 summarizes the paper conclusion.

## 2. PANORAMIC VIDEO STABILIZATION

### 2.1. Tracking visual features and key frame selection

Like other video stabilization algorithms, we start finding the trajectories of 2D visual features (sparse optical flows) of all video frames. We use Shi-Tomasi detector [8] and pyramidal Lucas-Kanade tracking algorithm [9] implemented in the OpenCV library [10]; however, any other visual feature detecting and tracking algorithms may work well.

Our algorithm has a notion of key frame. At key frames, we detect a new set of features for subsequent tracking. A track or a motion track is defined as a flow of one feature over



(a) Full spherical $360^\circ \times 180^\circ$ video with region of interest | (b) Original motion tracks | (c) Stabilized motion tracks

**Fig. 1**: Omnidirectional videos from internet: (a) Features detection in a region of interest (green rectangle), (b) Motion tracks in the region of interest of original video, and (c) Motion tracks in the region of interest of stabilized video with our propose method.

a sequence of continuous frames. Our optimization needs at least three tracks between two consecutive key frames. Nonetheless, by setting the minimum Euclidean distance between two features, ten or more tracks spreading all over the image are preferred for better stabilization. One track will be dropped if (1) it runs out of a region of interest, or (2) its confident value is less than a threshold (we use 0.2). If there is not a sufficient number of tracks surviving, the previous frame turns into a new key frame.

### 2.2. Representation of stabilizing rotation and its derivatives

We cast visual features of one frame in equirectangular projection onto an unit sphere by $\pi^{-1} : (\theta, \phi) \rightarrow S^2$ where

$$\pi^{-1}(\theta, \phi) = [\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi]^T.$$

Also a 3D unit vector on a sphere where panoramic image is projected is mapped onto the equirectangular image space by

$$\pi(p) = [\tan^{-1}(p_y/p_x), \sin^{-1}(p_z)]^T.$$

Then we transform a coordinate of a pixel by applying the stabilizing rotation $R$ on the whole panoramic image as

$$y = \pi(R\pi^{-1}(x)).$$

To express rotation compactly, we parametrize the rotation matrix by using exponential coordinates in the Euler-Rogrigues formula [11, 12] as

$$e^{[s]} = I + \alpha[s] + \beta[s]^2$$

where $\theta = \|s\|, \alpha = \frac{sin\theta}{\theta}, \beta = \frac{1-cos\theta}{\theta^2}$ and $[\cdot]$ denotes $3 \times 3$ skew-symmetric representation. Note that near to the identity, i.e., $\theta \approx 0$, these coefficients are approximated as $\alpha \approx 1 - \frac{1}{6}\theta^2, \beta \approx \frac{1}{2} - \frac{1}{24}\theta^2$, and thus the exponential coordinates based parametrization does not become singular.

Now we differentiate a 3D vector $q = e^{[s]}p$ with respect to the exponential coordinates as

$$\frac{\partial}{\partial s}(e^{[s]}p) = \beta(s^Tp)I - \alpha[p] - \alpha ps^T + \beta sp^T + (\gamma(s^Tp)I - \delta[p])ss^T,$$

where $\gamma = \frac{\alpha-2\beta}{\theta^2}, \delta = \frac{\cos\theta-\alpha}{\theta^2}$. Also these coefficients are approximated near to the identity as $\gamma \approx -\frac{1}{12} + \frac{1}{180}\theta^2, \delta \approx -\frac{1}{3} + \frac{1}{30}\theta^2$.

Finally, we have

$$\frac{\partial y}{\partial s} = \frac{\partial \pi}{\partial q}\frac{\partial y}{\partial s}$$

where

$$\begin{aligned} y &= \pi(q), \\ q &= e^{[s]}p, \\ \frac{\partial \pi}{\partial q} &= \begin{bmatrix} -\frac{q_y}{1-q_z^2} & \frac{q_x}{1-q_z^2} & 0 \\ 0 & 0 & \frac{1}{\sqrt{1-q_z^2}} \end{bmatrix}. \end{aligned}$$

Near to the identity, the derivatives can be simplified further as

$$\frac{\partial}{\partial s}(e^{[s]}p) \approx \frac{1}{2}[p][s] - [s][p] - [p]$$

and therefore

$$\frac{\partial y}{\partial s} \approx \begin{bmatrix} -\frac{q_y}{1-q_z^2} & \frac{q_x}{1-q_z^2} & 0 \\ 0 & 0 & \frac{1}{\sqrt{q_x^2+q_y^2}} \end{bmatrix} (\frac{1}{2}[p][s] - [s][p] - [p]),$$

where $q = e^{[s]}p \approx p + s \times p$.

More clarifications and proofs are referred in [13].

### 2.3. Optimizing panoramic video sequences

Given rotation matrices for all video sequences, visual features from trajectories are on equirectangular image space. Assume that there are $M$ visual features and we handle a video sequence comprising $(N+2)$ frames, we find the optimal rotation matrices parametrized by exponential coordinates such that these visual feature trajectories become as smooth as possible.

Let $x_j^i$ denote the coordinates of the $j$th visual feature on equirectangular image of the $i$th frame, and $y_j^i$ denote the transformed coordinates of $x_j^i$ due to stabilizing rotation matrix $R^i = e^{[s^i]}$.

Now we define the objective function inspired by [7] as

$$\begin{aligned} E(s^1, \ldots, s^N) = \sum_{i=1}^{N+1}\sum_{j=0}^{M} ||y_j^i - y_j^{i-1}||^2 \\ + \sum_{i=1}^{N}\sum_{j=0}^{M} \lambda^2 || - y^{i-1} + 2y^i - y^{i+1}||^2, \end{aligned} \quad (1)$$

where $\lambda$ is a smoothness term (we choose $\lambda = 1$). The first term is to encourage trajectories to be as short as possible while the second term is to encourage smoothness using a discrete Laplacian operator. The exact weight for each term is not critical, so it is set to be equal to one.

Note that $x_j^0$ and $x_j^{N+1}$ are fixed boundary conditions, hence $y_j^0 = x_j^0$ and $y_j^{N+1} = x_j^{N+1}$.

To solve the above optimization problem as a form of nonlinear least squares problem, we derive a residual function corresponding to the objective function (Eq. 1) such that $E(s) = ||f(s)||^2$ as follows:

$$f(s) = \begin{bmatrix} Y^1 - Y^0 \\ \vdots \\ Y^{N+1} - Y^N \\ \lambda(-Y^0 + 2Y^1 - Y^2) \\ \vdots \\ \lambda(-Y^{N-1} + 2Y^N - Y^{N+1}) \end{bmatrix}$$

where

$$\begin{aligned} s &= [s^1, \ldots, s^N]^T \in \mathbb{R}^{3N\times 1} \\ Y^i &= \left[y_1^i, \ldots, y_M^i\right]^T \in \mathbb{R}^{2M\times 1}. \end{aligned}$$

Then we have its derivatives as

$$\frac{\partial f}{\partial s} = \begin{bmatrix} A \\ \lambda B \end{bmatrix},$$

where

$$\begin{aligned} A &= \begin{bmatrix} Y_1^1 & 0 & \cdots & 0 \\ -Y_1^1 & Y_2^2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -Y_{N-1}^{N-1} & Y_N^N \\ 0 & \cdots & 0 & -Y_N^N \end{bmatrix}, \\ B &= \begin{bmatrix} 2Y_1^1 & -Y_2^2 & 0 & \cdots & 0 \\ -Y_1^1 & 2Y_2^2 & -Y_3^3 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -Y_{N-2}^{N-2} & 2Y_{N-1}^{N-1} & -Y_N^N \\ 0 & \cdots & 0 & -Y_{N-1}^{N-1} & 2Y_N^N \end{bmatrix}, \\ Y_j^i &= \frac{\partial Y^i}{\partial s^j} \end{aligned}$$

We observe that standard Gauss-Newton algorithm converges to find the optimal rotation in a few number of iterations.

## 3. RESULTS

### 3.1. Evaluation

We tested our algorithms with several hand-held panoramic videos with different amounts of camera shakes. Source videos are captured by Samsung Gear 360 and downloaded from youtube.

Fig. 2 shows the comparison of the relative rotations in each component of quaternion representation between original video and stabilized video. Relative rotation is computed as a rigid transform between the tracks of two adjacent

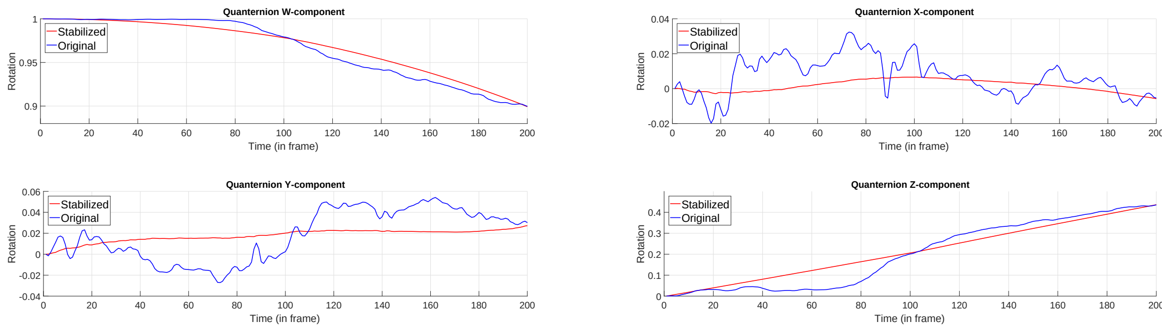



**Fig. 2**: Comparison of the relative rotations between original and stabilized video

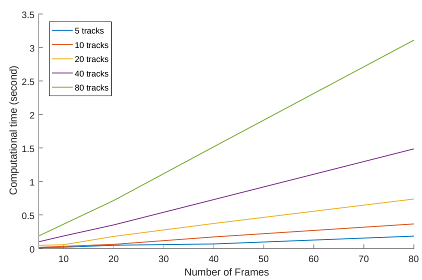



**Fig. 3**: Computational time (in seconds) of our proposed optimization with respect to the number of frames and the number of tracks

frames. The computation of relative rotation repeats over a subset of original and stabilized frames. We can see that the rotation changes smoothly over the time (200 frames) even there are 11 key frames which break the video sequence into 10 unequal lengths of segments.

Motion tracks of one specific region of interest in three different original videos and stabilized videos are computed and illustrated in Fig. 1. It is clear that motion tracks in stabilized videos change less wildly in terms of magnitude and frequency than those in original videos. Note that the computed stabilizing rotation is not the exact inverse relative rotation but it produces the smoothest result.

Supplemental video showing two sample videos can be watched here: `https://youtu.be/cl5hBRE0aCw`.

### 3.2. Performance

We implemented our optimization in MatLab without multi-thread support or GPU computing. The optimization computational time is nearly linear to the number of tracks and the number of frames (See Fig. 3). In the test videos, it usually takes less than 500ms, tens times faster than without providing the analytic derivatives of the objective function. The computational time seems to be fast enough that our algorithm can be used in an interactive application.

### 3.3. Limitations

Like other features-based stabilization, our method does not work well with static overlays such as text, logo, or a part of camera tripods, and moving big objects. In this case, our stabilizer might produce weird results even more shaky than input video.

### 3.4. Application

We introduce one possible application using our stabilization algorithm: an interactive panoramic video player for VR. Field-of-view of VR headset and user's viewing direction will determine a region of interest (ROI) where motion tracks are computed. ROI is updated when setting a new key frame. Therefore, key frames are triggered not only by a sufficient number of tracks but also by a number of tracked frames such that the stabilized ROI chases to the user's view. As a result, the ROI or the active viewing region has better stabilization than applying the algorithm for the full frame.

## 4. CONCLUSION

We have presented the method for panoramic video stabilization. Our optimization uses exponential coordinates to represent the stabilizing rotation, and the objective function to compute the stabilizing rotations for every video frame is compact. Moreover, the provided analytic derivatives of the objective function make the solver more efficient. Our proposed algorithm is very effective in the interactive video VR player to reduce the cyber-sickness.

## ACKNOWLEDGEMENTS

This work was supported by the Korea Institute of Science and Technology (KIST) Institutional Program (Project No. 2E27190).

## 5. REFERENCES


[1] PA Howarth and PJ Costello, "The occurrence of virtual simulation sickness symptoms when an hmd was used as a personal viewing system," *Displays*, vol. 18, no. 2, pp. 107–116, 1997.

[2] Kay M Stanney, Robert S Kennedy, and Julie M Drexler, "Cybersickness is not simulator sickness," in *Proceedings of the Human Factors and Ergonomics Society annual meeting*. SAGE Publications Sage CA: Los Angeles, CA, 1997, vol. 41, pp. 1138–1142.

[3] Robert S Kennedy, Julie Drexler, and Robert C Kennedy, "Research in visually induced motion sickness," *Applied ergonomics*, vol. 41, no. 4, pp. 494–503, 2010.

[4] Mostafa Kamali, Atsuhiko Banno, Jean-Charles Bazin, In So Kweon, and Katsushi Ikeuchi, "Stabilizing omnidirectional videos using 3d structure and spherical image warping," in *Proc. IAPR Conf. MVA*, 2011, pp. 177–180.

[5] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala, "Content-preserving warps for 3d video stabilization," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 44, 2009.

[6] Shunichi Kasahara, Shohei Nagai, and Jun Rekimoto, "First person omnidirectional video: System design and implications for immersive experience," in *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*. ACM, 2015, pp. 33–42.

[7] Johannes Kopf, "360° video stabilization," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 195, 2016.

[8] Jianbo Shi et al., "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[9] Jean-Yves Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, pp. 4, 2001.

[10] Itseez, "Open source computer vision library," `https://github.com/itseez/opencv`, 2015.

[11] KC Gupta, "An historical note on finite rotations," *Journal of Applied Mechanics*, vol. 56, pp. 139, 1989.

[12] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry, *A mathematical introduction to robotic manipulation*, CRC press, 1994.

[13] Guillermo Gallego and Anthony Yezzi, "A compact formula for the derivative of a 3-d rotation in exponential coordinates," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 3, pp. 378–384, 2015.