HYPERLAPSE GENERATION OF OMNIDIRECTIONAL VIDEOS BY ADAPTIVE SAMPLING BASED ON 3D CAMERA POSITIONS

Masanori Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa

The University of Tokyo
{m_ogawa, yamasaki, aizawa}@hal.t.u-tokyo.ac.jp

ABSTRACT

Capturing a city with an omnidirectional camera can produce a continuous motion street view and provides the view of the moving person. However, watching such a video is not necessarily pleasant because the videos are often excessively lengthy. In this paper, we propose a method for shortening an omnidirectional video. Subsampling a video captured by a hand-held camera displays a significant amount of destabilization resulting from shaking. This prompted us to propose an adaptive subsampling scheme that selects optimal frames by minimizing our cost function based on 3D camera positions. This optimal selection suppresses only the translational camera instabilities. The rotational instabilities are initially ignored and later compensated for. This approach allowed us to successfully generate a subsampled and stable omnidirectional video. In addition, we propose an alternative measure of translational instability to evaluate our frame selection.

***Index Terms*—** Omnidirectional Video, Hyperlapse, Timelapse

1. INTRODUCTION

Omnidirectional imaging is designed to capture the $360^\circ \times 180^\circ$ world around the camera in one shot. Cameras and video recorders with this capability are becoming popular, and services making use of omnidirectional videos are increasing (e.g., YouTube and Facebook). In addition, an omnidirectional video can be utilized with virtual reality (VR) head-mounted displays, which are expanding in the market.

Omnidirectional video is highly beneficial for capturing motion street views because they provide us with the view of the person moving along the street. Contrary to Google street view, motion street view can reproduce continuous motion scenes of streets and are much more informative in terms of the status of the area and the directions to our destinations. However, such videos are excessively lengthy and are time consuming to watch. Obtaining an instant impression of the area and the directions would require us to generate a hyperlapse video. We also have to pay attention to stabilization because a video recorded with a hand-held camera is characterized by unintentional camera instability. In this paper, we propose a method to shorten an omnidirectional video by ensuring that destabilization resulting from movement is reduced to the minimum possible amount. The contributions of our paper are summarized below:

- We describe the novel system we built for generating omnidirectional hyperlapse videos. The system performs frame subsampling and rotational compensation without image warping, which may lead to distortion.
- We propose an adaptive subsampling scheme that selects optimal frames by minimizing our cost function based on an estimation of the 3D camera positions.
- We propose an alternative way to measure instability to evaluate the effectiveness of our frame selection.

2. RELATED WORK

Stabilization of conventional videos has been well studied. Karpenko et al. [1] used special sensors to measure camera motion. However, a hardware-based method such as theirs cannot be applied to archived video. Various other groups [2, 3, 4, 5, 6] employed software-based methods that rely on a cropping technique. Although these methods are effective for the stabilization of ordinary videos, the cropping process cannot be utilized for omnidirectional videos.

Regarding hyperlapse generation of ordinary video, Kopf et al. [7] first estimated a sequence of 3D camera positions and synthesized a scene by blending regions of different frames chosen based on an estimation of ideal camera location and orientation. Joshi et al. [8] selected optimal frames for stabilization and stabilized selected frames with cropping.

In terms of the stabilization of omnidirectional video, Kamali et al. [9] proposed omnidirectional structure-from-motion and synthesized scenes such that feature trajectories could be smoothed. Kopf [10] proposed a deformed-rotation motion model that warps frames to reduce small amounts of translational jitter, and also demonstrated hyperlapse generation. In contrast to the above two methods [9, 10] our method does not perform image warping. Instead, our method compensates for rotation by only rotating selected optimal frames. Rotation parameter estimation of omnidirectional video is

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

also discussed by [11] using spherical optical flow and by [12] using special sensors.

3. PROPOSED METHOD

We propose a method that generates hyperlapse of omnidirectional videos. Contrary to ordinary video, the view captured by the omnidirectional camera is $360^\circ \times 180^\circ$ and does not change when the direction of the camera is rotated. Camera instabilities are divided into those that are translational and those that are rotational, and we can compensate for the rotational instabilities in the omnidirectional video. Thus, the most important consideration for stabilizing omnidirectional video is to suppress translational instabilities.

Our aim is to produce a hyperlapse version of the given omnidirectional video under the condition of a certain subsampling rate (speed-up rate) ν. We define a path as a subsequence of input frames, with the path corresponding to the output video. Further, we define a cost function on the transition between frames. This function depends on the camera positions of the two frames. The function finds a path minimizing the total cost to enable the suppression of translational instabilities by observing the specified speed-up rate to some extent, after which we reduce the rotational instabilities.

3.1. Estimation of Camera Position

First, we estimate the camera position in each frame of the omnidirectional video, A few methods for the estimation of the camera position and orientation have been reported [13, 14, 9, 15], and we use the method [15] in our experiments to estimate the camera position. Let $\boldsymbol{X}_t$ be the camera position of the t-th frame.

Next, we apply filtering to the sequence of the camera positions $\boldsymbol{X}$ to generate smooth camera positions $\boldsymbol{X}'$, which are considered the ideal camera positions without translational instabilities. Similarly to Kamali et al. [9], we use Gaussian smoothing to filter $\boldsymbol{X}$ and define the camera motion cost function using $\boldsymbol{X}$ and $\boldsymbol{X}'$.

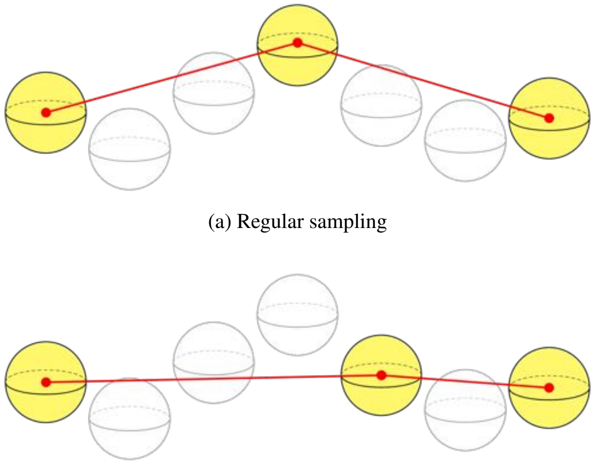
3.2. Optimal Frame Selection

Fig. 1 shows the difference between regular and the proposed sampling. We define the transition cost between frames to enable us to suppress the translational instabilities by observing the specified speed-up rate ν.

We define the inter-frame transition cost C from the i-th frame to the j-th frame by using three cost components.

$$C(h,i,j,\nu) = C_m(i,j) + \lambda_s C_s(i,j,\nu) + \lambda_a C_a(h,i,j), \quad (1)$$

where h is the index of the frame that is already selected before the i-th frame, C_m is the cost for penalizing unintentional camera motion, C_s is the cost for a frame sampling interval violation of the specified speed-up rate, C_a is the cost to maintain the frame sampling intervals at a constant level, and λ_s and λ_a are the weights of the respective costs. Since C_s and C_a do not depend on the form of videos, they are the same as those used previously [8].



(a) Regular sampling

(b) Optimal frame selection

Fig. 1: Difference between regular sampling (a) and optimal frame selection (b). The center of each sphere corresponds to the camera position of each frame and the yellow spheres represent selected frames. In (a), frames are selected at regular intervals. In (b), the frames are irregularly selected while suppressing the camera translational instabilities and a stable trajectory is obtained.

$$C_s(i,j,\nu) = \min(|(j-i)-\nu|^2, \tau_s), \quad (2)$$

$$C_a(h,i,j) = \min(|(j-i)-(i-h)|^2, \tau_a). \quad (3)$$

C_m is specially defined in our proposal. Previously [8], for ordinary video, C_m was defined as the extent of movement of the image center by the homography transformation between frames, which reflects both the translational and rotational camera instabilities. On the other hand, the rotational instability of the camera does not matter in the case of omnidirectional video used to capture views of $360^\circ \times 180^\circ$. The rotational instability in the omnidirectional frame can be removed by an inverse rotation at a later stage. Therefore, the cost C_m specially defined for omnidirectional video penalizes only the translational camera motion. We propose determining the cost C_m by using both the estimated and smooth ideal camera positions:

$$C_m(i,j) = \left\| (\boldsymbol{X}_j - \boldsymbol{X}_i) \times \frac{\boldsymbol{X}'_j - \boldsymbol{X}'_i}{\left\| \boldsymbol{X}'_j - \boldsymbol{X}'_i \right\|_2} \right\|_2 , \quad (4)$$

where $(\boldsymbol{X}_j - \boldsymbol{X}_i)$ is the actual motion vector of the camera between frames, and $(\boldsymbol{X}'_j - \boldsymbol{X}'_i)$ is the smooth motion vector. Our C_m represents the amount of camera movement in the direction orthogonal to the smooth motion. Therefore, the larger this value, the more translational movement is present in the path.

Based on the inter-frame transition costs, we find a path to minimize the total cost. A dynamic programming method is employed to search for the optimal path to determine the frames that minimize the cost.

3.3. Rotation Compensation

In the hyperlapse version of an ordinary video, stabilization is applied as a post-processing step to selected frames in which cropping is performed to smooth the motion. However, the cropping process is inadequate in the case of omnidirectional video, which captures a $360^\circ \times 180^\circ$ view around the camera.

In our proposed method, we perform rotation compensation instead of cropping. We obtain the camera rotation parameters and estimate the camera positions simultaneously. Then, we generate hyperlapse omnidirectional videos by rotating the selected frames with the appropriate parameters.

4. EXTENSIONS

In the previous section, we explained omnidirectional hyperlapse video generation. Here we describe some extensions that provide meaningful effects.

4.1. Constant Motion

When generating a hyperlapse, [7, 10] maintained the camera motion speed at a constant level. By not only stabilizing the camera motion but also keeping it constant, they were able to enhance the resultant view considerably. We generate constant motion hyperlapse by changing the argument ν of $C_s(i, j, \nu)$ to $\tilde{\nu}(i)$ determined by the camera motion speed $v(i)$ at the i-th frame.

Although a similar extension was implemented in [8], the definition of $v(i)$ would have to be different in our case. In our ominidirectional video, $v(i)$ and $\tilde{\nu}(i)$ is defined using $\boldsymbol{X}$:

$$v(i) = \left\| \boldsymbol{X}_{i+1} - \boldsymbol{X}_i \right\|_2 , \quad (5)$$

$$\tilde{\nu}(i) = \alpha \frac{\overline{v}}{v(i)} \nu + (1 - \alpha)\nu, \quad (6)$$

where $\overline{v}$ is the average of $v(i)$ and α is the degree of constant motion.

4.2. Time-Limit Constraint

The target omnidirectional video and the desired speed-up rate form the input to our system. Our method does not control the length of the output video accurately. The same situation exists in [8] for ordinary video. Our irregular subsampling tends to be longer than regular subsampling. There is also a need for video shortening within a specified time constraint.

We additionally propose a method to ensure the output video remains within the time-limit constraint. The method iterates the frame selection by adjusting the subsampling rate until the video length becomes shorter than or equal to the time-limit constraint. The initial value of the speed-up rate ν is given by dividing the length of the input video by the time-limit constraint. As ν increases, the number of selected frames decreases. Starting from the initial value, and gradually increasing the subsampling rate, we terminate the iteration when the length condition meets the constraint. The computational cost of frame selection is not large. Therefore, repeating the optimal selection of frames is not problematic.

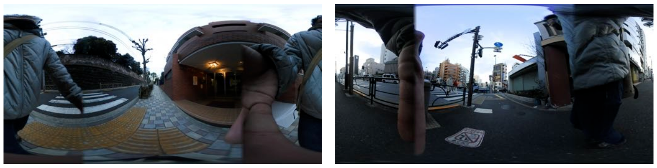

(a) Straight line motion video (b) Curved motion video

Fig. 2: Two target videos recorded with the hand-held omnidirectional camera.

5. RESULTS

5.1. Experimental Condition

Fig. 2 shows two videos recorded with a hand-held omnidirectional camera (Nikon KeyMission360) for the experiments. The video in (a) was recorded when moving in a straight line, and that in (b) when turning a corner.

In this experiment, we used an existing method [15] for camera location and orientation estimation because it is very fast. Since the method assumes a rectangular frame of an ordinary video, we applied it to the front face of a cube map created to represent the omnidirectional image. We use the location and orientation parameters for frame selection and rotation compensation, respectively.

For rotation compensation we tested two methods: (i) inversely rotate and align the front face of all frames to the standard direction, and (ii) align the front of each frame to the smoothed moving direction in the horizontal plane using $\boldsymbol{X}'$. In the experiments, we applied (i) and (ii) to (a) and (b), respectively.

Our system only has a few parameters. We set the weights of the costs as $\lambda_s = 0.2 \times 10^{-3}, \lambda_a = 0.8 \times 10^{-4}$ and also specified the degree of constant motion as $\alpha = 0.7$.

5.2. Evaluation of Translational Instabilities of Selected Frames

We evaluated our frame selection in both a subjective and objective manner.

First, we visualize comparisons between the spatial camera trajectories of regular sampling and our optimal frame selection. Fig. 3 shows an example of trajectories in which we can clearly see the effect of translational instability suppression. In the figure, each dot represents the estimated camera

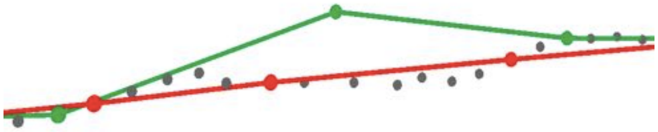


Fig. 3: Comparison between camera positions in regular and proposed sampling. Each dot represents the estimated camera position of each frame. Green and red dots denote the results of regular sampling and our frame selection, respectively.

Table 1: Comparison of results: regular sampling vs. our optimal frame selection

(a) Straight line motion

ν	S_{reg}	S_{opt}	S_{opt} / S_{reg}	$\tilde{\nu}$
6x	3.85	2.59	0.67	5.13x
8x	3.05	0.96	0.32	6.81x
10x	1.86	1.07	0.57	8.10x
15x	1.05	0.66	0.63	13.77x
20x	1.65	0.76	0.46	18.57x

(b) Curved motion

ν	S_{reg}	S_{opt}	S_{opt} / S_{reg}	$\tilde{\nu}$
6x	0.84	0.39	0.47	5.40x
8x	0.61	0.26	0.42	6.72x
10x	0.31	0.25	0.80	9.40x
15x	0.29	0.21	0.71	13.45x
20x	0.36	0.24	0.66	18.21x

S_{reg}: S by regular sampling
S_{opt}: S by our optimal frame selectoin
$\tilde{\nu}$: the actual subsampling rate

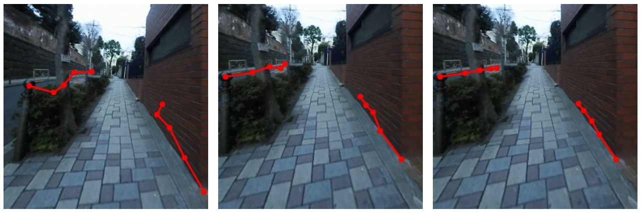
position of each frame, and the green and red dots correspond to frames selected by regular sampling and our optimal selection, respectively. Although the regular sampling trajectory (in green) exhibits considerable translational instability, our optimal selection can avoid selecting such positions.

Next, we quantitatively evaluate the translational instabilities of paths. We introduce an additional numerical measure for the entire path. The amount of instability S of a path is defined as follows:

$$S = \sum_{n=2}^{N-1} \left\| \tilde{\boldsymbol{X}}_{n+1} - \tilde{\boldsymbol{X}}_n \right\|_2 \sin \frac{\theta_n}{2}, \tag{7}$$

$$\theta_n = \arccos \left(\frac{(\tilde{\boldsymbol{X}}_{n+1} - \tilde{\boldsymbol{X}}_n) \cdot (\tilde{\boldsymbol{X}}_n - \tilde{\boldsymbol{X}}_{n-1})}{\left\| \tilde{\boldsymbol{X}}_{n+1} - \tilde{\boldsymbol{X}}_n \right\|_2 \left\| \tilde{\boldsymbol{X}}_n - \tilde{\boldsymbol{X}}_{n-1} \right\|_2} \right), \tag{8}$$

where $\tilde{\boldsymbol{X}}_n$ corresponds to the camera position of the n-th frame in the selected N frames ($n = 1, ..., N$) and S represents the sum of differences between the actual camera position and the position predicted from the last position when maintaining the same direction of motion. Table 1 shows S of the path by regular sampling and our optimal frame selection under several different subsampling rates for the two different videos. The absolute value of S is influenced by the length of the sequence. We focus on the ratios of S of the two methods as an indication of the degree of translational instability suppression. The effect of our optimal frame selection method is clearly apparent for both videos.



(a) Regular sampling only (b) Regular sampling and rotation compensation (c) Our optimal frame selection and rotation compensation

Fig. 4: Comparison of the three results in Fig. 2a using three types of processing: (a) unpleasant to view, (b) certain degree of instability due to rattling, and (c) very small amount of instability produced by our system.

5.3. Visual Evaluation

We compared the following three results shown in Fig. 4: (a) regular sampling only, (b) regular sampling and rotation compensation, and (c) our optimal frame selection and rotation compensation. (a) exhibits a significant amount of instability. (b) appears to be stabilized, but continues to display some instability due to rattling. (c) appears the most stabilized and has little instability. We can also confirm the effectiveness of our frame selection method to suppress translational instability by the output of the omnidirectional videos. We include the outputs of Fig. 2b and Fig. 2a as supplemental material.

5.4. Limitations

The proposed method is effective as long as it is possible to track local features successfully because a sequence of camera positions is required to determine the motion cost.

6. CONCLUSION

In this paper, we proposed a method for generating a shortened and stabilized omnidirectional video. We introduced a inter-frame transition cost function and proposed the selection of optimal frames by minimizing the total cost to suppress the translational instability while observing the specified subsampling rate to some extent. We performed rotation compensation for selected frames as a post-processing step. As a result, we successfully generated a hyperlapse omnidirectional video. The suppression of instabilities was both quantitatively and qualitatively evaluated.

7. REFERENCES

[1] Alexandre Karpenko, David Jacobs, Jongmin Baek, and Marc Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes," *CSTR*, vol. 1, pp. 2, 2011.

[2] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala, "Content-preserving warps for 3D video stabilization," *ACM TOG*, vol. 28, no. 3, pp. 44, 2009.

[3] Matthias Grundmann, Vivek Kwatra, and Irfan Essa, "Auto-directed video stabilization with robust L1 optimal camera paths," in *CVPR*. IEEE, 2011, pp. 225–232.

[4] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala, "Subspace video stabilization," *ACM TOG*, vol. 30, no. 1, pp. 4, 2011.

[5] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun, "Bundled camera paths for video stabilization," *ACM TOG*, vol. 32, no. 4, pp. 78, 2013.

[6] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun, "Steadyflow: Spatially smooth optical flow for video stabilization," in *CVPR*, 2014, pp. 4209–4216.

[7] Johannes Kopf, Michael F Cohen, and Richard Szeliski, "First-person hyper-lapse videos," *ACM TOG*, vol. 33, no. 4, pp. 78, 2014.

[8] Neel Joshi, Wolf Kienzle, Mike Toelle, Matt Uyttendaele, and Michael F Cohen, "Real-time hyperlapse creation via optimal frame selection," *ACM TOG*, vol. 34, no. 4, pp. 63, 2015.

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

[10] Johannes Kopf, "360° video stabilization," *ACM TOG*, vol. 35, no. 6, pp. 195, 2016.

[11] Sarthak Pathak, Alessandro Moro, Atsushi Yamashita, and Hajime Asama, "A decoupled virtual camera using spherical optical flow," in *ICIP*. IEEE, 2016, pp. 4488–4492.

[12] Thomas Albrecht, Tele Tan, Geoff AW West, and Thanh Ly, "Omnidirectional video stabilisation on a virtual camera using sensor fusion," in *ICARCV*. IEEE, 2010, pp. 2067–2072.

[13] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[14] Georg Klein and David Murray, "Parallel tracking and mapping for small AR workspaces," in *ISMAR*. IEEE, 2007, pp. 225–234.

[15] Raul Mur-Artal, JMM Montiel, and Juan D Tardós, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.