Enhancing Viewability for First-person Videos based on a Human Perception Model

Biao Ma, Amy R. Reibman
School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA

Abstract—**First-person videos (FPVs) captured by wearable cameras have undesired shakiness because of fast changing views. When existing video stabilization techniques are applied, FPVs are transformed into cinematographic videos, losing the First-person motion information (FPMI) such as the recorder's interests and actions. We propose a system that can enhance viewability of FPVs by stabilizing them while preserving their FPMI. The viewability is charaterized based on a human perception model. Objective tests show that our method has competitive stabilization performance relative to existing video stabilization techniques. And subjective tests show that spectators still experience the FPMI from the resulting videos while shakiness is reduced.**

I. Introduction

Wearable cameras, such as GoPro and Pivothead, are becoming popular recently. For entertainment purposes, people use them to record First-person videos (FPVs), which is a kind of egocentric video that differs from hand-held videos. However, when people play back these videos, they find that what they recorded looks quite different than what they actually experienced. The frames are shaky with uncomfortable viewing angles. The camera motions may also make spectators feel dizzy. All in all, it is often an unpleasant experience.

In this paper, we aim to enhance the viewability of a FPV by stabilizing it while preserving its First-person motion information (FPMI). By this we mean that spectators do not feel that the resulting video is too shaky, and it still can convey the recorder's interests and actions.

A direct solution would be using hardware-based video stabilization techniques such as the built-in function in GoPro Hero 5 and other hand-held stabilizers. The problem with this solution is that it has limited performance. The built-in function in GoPro Hero 5 can only remove small amounts of shakiness since the camera records in real-time and must avoid obvious stitching errors when applying the stabilization function. Hand-held stabilizers require users to hold the device, which limits the users' activities to riding or driving. Although wearable gimbals are available, they are large and heavy for users who want to have long-period activities.

Another straightforward idea would be to apply traditional video stabilization techniques to FPVs. Video stabilization can smooth the changes between adjacent frames in order to make the original jittery video watchable. Normally, three steps are necessary: motion estimation, motion smoothing and frame construction. There are two main approaches of video stabilization: 2D and 3D solutions. In 2D solutions [1]–[5], frame-based 2D linear motions are estimated by detecting and tracking local feature points. Then the homographies are computed to warp the current frame with respect to the previous one in order to smooth the trajectories of the tracked features. In 3D solutions [6]–[9], the 3D camera motion is first estimated, which includes the relative camera orientations and camera translations. Based on this information, a new 3D path is designed by smoothing the original jittery one. Then the new frames are synthesized by projecting the original frames onto the new path. By adding constraints to the path-smoothing process, some methods try to minimize the missing area caused by the projection.

Note that these video stabilization techniques are designed for hand-held or vehicle-mounted videos. These techniques try to stabilize the videos so that the results are like being shot from a smooth path (linear or parabolic path [7]). This general idea is further extended and defined as *Re-Cinematography* [11]: recovering a cinematographic video from a shaky one. Re-Cinematography is inspired by [10] and carefully developed in [11], [12]. Their goal is to improve apparent camera motions within videos so that the outputs look like they are shot by professionals with tripods. In [12], an input video is segmented into a series of shots. For each shot, small motions are considered to be shakiness and are removed by video stabilization techniques. Large motions are edited with a profiled velocity. The direction of these motions are obtained by identifying and tracking important objects, which actually is a subjective procedure and includes many challenging pattern recognition topics. Later in [7], Re-Cinematography was applied to 3D video stabilization techniques, where the subjective procedures were replaced with allowing users to choose the camera paths.

However, FPVs are different than hand-held videos. They usually come from cameras mounted on the human body, often on the head. The content of the scene is usually recorded passively since the recorder only treats the camera as a wearable kit. In contrast, the interests and actions (FPMI) are performed actively, which makes them as important as the scene itself. By applying video stabilization or Re-Cinematography techniques, this distinguishing part of FPVs will be removed with high probability (see [6], [7] for examples). The viewer may not recognize the human-like-motion and the recorder's motion intentions from the resulting video. Several works [6], [13]–[15] focused on creating a watchable egocentric video by reducing the unwatchable content using a fast-forwarding approach. Although [15] tried to only reduce the non-semantic parts of videos, the FPMI is also lost. Therefore, in this paper,

978–1–5090–3649–3/17/$31.00 © 2017 IEEE

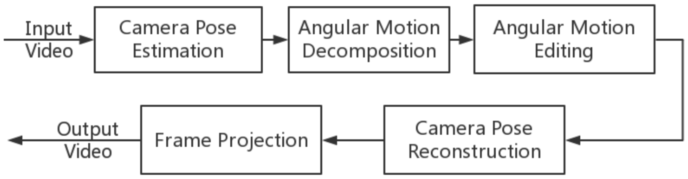

Fig. 1: Framework of First-person Video Enhancement

we develop a system that can stabilize original FPVs and also preserve their FPMI as much as possible without reducing video length.

Our approach follows the pipeline of traditional 3D video stabilization techniques. However, the 3D motion estimation algorithm is modified to only estimate the 3D rotations. This is motivated by our human perception model of FPVs, which is also used to design the new camera path instead of using the Re-Cinematography method. In the next section, our entire system and innovations are discussed. In section III, we provide the details of our work including the camera pose estimation, angular motion decomposition and editing. Viewability is defined and then enhanced based on our human perception model. In section IV, both the objective and subjective tests are discussed. Finally, in section V, we conclude with our contributions and future plans.

II. System overview

Fig. 1 shows the framework of our system. Our work includes three novel aspects: a human perception model for angular motion editing, a human rotation motion model for angular motion decomposition, and a modified 3D motion estimation algorithm for camera pose estimation. Based on our hypothesis, the translations, especially the vertical one, are necessary to convey the First-person feeling. So we only estimate the rotation and only edit the angular motions.

We propose a human perception model of First-person motion conveyed by FPVs. It is based on human eye movements and viewing distance. Given the camera poses of each frame, it can localize the undesired part of the First-person motion in a video. Using this model to edit the undesired motion, we can design a camera path whose goal is to provide more stable view while preserving the FPMI. The model and path selection are described in section III-C.

We build a human rotation motion model to explain the relationship between motions. The rotation motions are decomposed based on the motion importance and freedom. A given rotation is decomposed into first yaw, then pitch, and then roll. The order of the motion indicates their importance and freedom. Motion with high importance and freedom may cause motion with lower importance and freedom.

We modify the traditional 3D motion estimation algorithm used in video stabilization. Since we only estimate the rotations, our algorithm is not restricted by Structure From Motion (SFM) [16] whose core is long-term feature tracking, 3D reprojection and local feature-based bundle adjustment. In our case, only the feature matching between two views is needed, which relaxes the constraint that enough features must always be seen across time. Instead, we propose a inexpensive way to estimate the rotations, and we apply graph optimization techniques so the computational process is independent of local features.

Based on these three aspects, we design the camera path that enhances the viewability of FPVs.

III. Algorithm Details

In this section, our camera pose estimation and angular motion decomposition algorithms are discussed. Then we introduce the core of our system: the angular motion editing algorithm that is based on our human perception model.

A. Camera Pose Estimation

Camera pose estimation is a well-known problem in the robotics (defined as vSLAM [17]) and computer vision (defined as SFM [16]) communities. It is used to estimate both the rotation and the 3D translation of the camera. Most previous 3D video stabilization approaches use these results to remove the translations in x and y directions.

However, in contrast, as in [18], we do not remove and thus do not need to estimate the translations. This is because translations include important FPMI. For example, the frequency and amplitude of translations can reflect the moving speed of the recorder.

The importance of translations is evident from First-person video gaming, since watching a First-person video is similar to playing a First-person video game, except the viewer cannot control the viewing angle. The translation is called “head bobbing” in video games. Recent popular video games such as “Call of Duty 4” and “Grand Theft Auto V” use it to make the game more realistic. [19] showed that most players prefer games that have head bobbing.

Since we do not estimate translations, it is unnecessary to observe a local feature across more than two frames. The rotation between each two frames can be estimated independently during a first pass. Then we use the graph optimization techniques introduced in [20] to further reduce the error. Assume the estimated rotation from frame m to frame n is $R_{n,m}$. The estimated camera pose of a single frame i with respect to the first frame is R_i:

$$R_i = R_{i,i-1} R_{i-1,i-2} \cdots R_{2,1}. \tag{1}$$

When estimating the rotation between two frames, we use SURF [21] features with RANSAC [22]. We first assume the two frames share a large enough baseline that we can use triangulation. In this situation, we use the epipolar geometry to find the relative rotation. When the triangulation fails (not enough inliers after RANSAC), we consider it is a pure rotation between two frames and the fundamental matrix is degenerated to be a homography.

Using the graph optimization tools provided by [20], we refine the estimated rotations using the objective function:

$$\min_{R_i, R_j \in SO(3)} \sum_{(i,j)\in L} \left\| Log(R_{i,j}^T R_i R_j^T) \right\|^2, \qquad (2)$$
$$L = \{(i,j) : i - j = l \; and \; i, j \in \mathbb{Z}^+\}.$$

Note that this approach relaxes the constraint of feature tracking but requires the camera to be calibrated. We find $l = 5$ is large enough to have a large baseline with a frame rate of 30.

B. Angular Motion Decomposition

A rotation matrix R_i describes the relative pose of frame i with respect to frame 1. We decompose R_i into rotations around the x, y and z axes, which are pitch, yaw and roll respectively. However, there is no unique solution. In order to make the results have realistic meaning, the decomposition order should coincide with the importance of human motions. This means the most important or the main motion should be extracted first. We believe that the yaw is the primary and most important motion since it is performed to look around. Then pitch has intermediate importance since it is performed to look up and down. Roll is believed to be an inessential motion, as it is rarely performed on purpose. So R_i is decomposed as:

$$R_i = R_z(\theta_z) R_x(\theta_x) R_y(\theta_y). \qquad (3)$$

Then

$$\theta_y = \tan^{-1}\left(\frac{-R_i(3,1)}{R_i(3,3)}\right), \qquad (4)$$

$$\theta_z = \tan^{-1}\left(\frac{R_{zx}(2,1)}{R_{zx}(1,1)}\right), \qquad (5)$$

$$\theta_x = \tan^{-1}\left(\frac{-R_{zx}(2,3)}{R_{zx}(2,2)}\right), \qquad (6)$$

where

$$R_{zx} = R_i R_y(\theta_y)^{-1}. \qquad (7)$$

$R(k,l)$ is the (k,l) entry of matrix R.

C. Angular Motion Editing

We first introduce our human perception model: the basis of the angular motion editing algorithm. The recorded video is not identical to what the recorder experienced, because humans perform different eye movements in real life and watching FPVs. In real life, the human eye movement related to our stabilization topic is *vestibulo-ocular movement* [23]. In this situation, human motion can be classified into intentional motion and unintentional motion, and only the unintentional motion needs to be compensated. Given the information from *semicircular ducts*, the *vestibulo-ocular reflex* will be triggered to compensate for the unintentional rotation of our head in order to keep the image fixed on our retina.

However, when we watch a FPV, this reflex is disabled since our head is not moving. Meanwhile, both the intentional and unintentional motion need to be compensated. In this situation, all the motion is compensated by an eye motion called *smooth pursuit movement* [23]. It is used to follow a target using only visual clues. As long as the motion of *smooth pursuit* aligns with the camera motion as accurately as the *vestibulo-ocular movement*, we can watch the FPV comfortably. However, this eye movement is not as efficient as the *vestibulo-ocular movement*, which is the reason that we may feel that the recorded video is not the same as what we experienced.

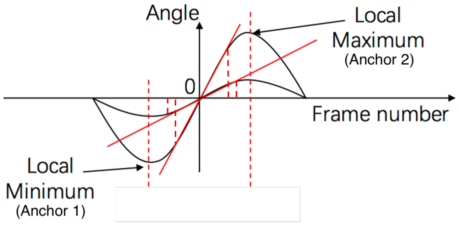

Fig. 2: Model of Angular Motions

Before we use *smooth pursuit movement* to track a target, our eyes need 125 ms to start to catch the target, which is called a *catch-up saccade* [24]. At 30 frames per second, this corresponds to 3.7 frames. Within these 4 frames, we cannot follow the recorder's motion, which is a reason for motion sickness.

Fig. 2 shows two motions that have different amplitudes. Each of them starts at one local extreme and ends at the next one, defining two *motion anchors*. A new *smooth pursuit* starts at each anchor. When the motion velocity changes rapidly, our eyes must perform a *catch-up saccade* near both *anchors*. The constant speed part, modeled by the slope, is where our eyes can perform *smooth pursuit*. As a result, unless a motion lasts more than 8 frames, we cannot follow it. In contrast, the efficiency of *vestibulo-ocular movement* is 100 Hz (0.3 frames) [25]. So in real life, we can follow such a motion in real-time.

Our human perception model consists of this eye motion characteristic and the motion model in Fig. 2. The constant speed part is defined as $\{\theta(k) : |\hat{\theta}(k)| \leq \theta_M, \theta(k) \in \theta(n)\}$. $\theta(n)$ is a sequence of angles we estimate in section III-B. $\hat{\theta}(k)$ is calculated as:

$$\hat{\theta}(k) = \theta(k) - 2\theta(k-1) + \theta(k-2). \qquad (8)$$

θ_M (around 0.02 degree) is the minimum angular resolution of human eyes [26]. Based on this model, we define the *stability* of a FPV to be the fraction of frames within the constant speed part.

However, viewing distance must also be accounted for. The motion spectators perceive is not as large as what the recorder performed. Assume the motion we estimate is $\theta(n)$, the equivalent focal length is f and the viewing distance is d. Then the actual perceived motion is $\omega(n) = f\theta(n)/d$. If our motion editing algorithm modifies ω to be $\tilde{\omega}$, the frame index of motion anchors are $A(i)$, and the video length of the i^{th} motion is L_{total}^i, then we have stability of the i^{th} motion:

$$F(i) = \frac{\sum_{n=A(i)}^{A(i+1)-1} f(n)}{L_{total}^i}, \qquad (9)$$

$$f(n) = \mathbb{1}_{\{x \leq \theta_M\}} |\ddot{\tilde{\omega}}(n)| \cdot \left[f(n-1) + \mathbb{1}_{\{x=0\}} \sum_{k=n-3}^{n-1} f(k)\right]. \qquad (10)$$

Note that $f(n)$ is the stability of frame n. Frame n can be observed only if it is one of the frames in the constant speed part and also outside the *catch-up saccade* period.

The idea of our motion editing algorithm is based on the perception model above. Note that in Fig. 2, the smaller slope has longer constant speed duration, which can increase $F(i)$. So we create a new path that has longer constant speed duration by decreasing the amplitude of each single motion anchor. The decreasing rate of the i^{th} motion anchor is:

$$D(i) = \frac{\Big|\tilde{\omega}\big(A(i)\big) - \omega\big(A(i)\big)\Big|}{\Big|\omega\big(A(i)\big) - \omega\Big(A\Big(\arg\min_{i \neq j} \big|A(j) - A(i)\big|\Big)\Big)\Big|}. \tag{11}$$

Given two edited motion anchors, the value between them is interpolated as:

$$\tilde{\omega}(n) = \tilde{\omega}\left(A(i)\right) + s \cdot \left[\omega(n) - \omega\left(A(i)\right)\right], \tag{12}$$

$$s = \frac{\tilde{\omega}\left(A(i+1)\right) - \tilde{\omega}\left(A(i)\right)}{\omega\left(A(i+1)\right) - \omega\left(A(i)\right)}. \tag{13}$$

As a result, $\tilde{\omega}$ is a function of D. Our target is to find D to get the actual editing strategy. Thus, we perform an optimization problem using particle swarm on the objective function (14), which defines the *viewability* of a FPV.

$$\min_{D} \sum_{i=1}^{|\{A\}|} \left(1 - F(i)\right)^2 + \alpha \sum_{n=1}^{|\{\tilde{\omega}\}|} \left|\tilde{\omega}(n) - \omega(n)\right|^2. \tag{14}$$

The idea here is to enlarge the stability and at the same time keep as much FPMI as possible. The second term expresses the difference between the new motion and the original motion, which is the FPMI and also the size of black area. α is the weight of the FPMI. Larger α preserves more FPMI. Its valid value region is from 0 to 0.2. Finally, the modified motion of the camera should be $\omega_{new}(n) = d\tilde{\omega}(n)/f$. All frames are projected based on the new camera positions.

Note that only yaw and pitch are retained by this algorithm. The roll motions are all removed since humans rarely perceive this motion in real life.

IV. Experiments

A. Objective tests

Our system is designed to stabilize a FPV while preserving its FPMI. Before testing its overall performance on enhancing viewability of FPVs, we first evaluate it as a video stabilizer.

Our test is based on 5 video sets, which are recorded in 5 different scenes (available at [27]). Each set includes 6 different versions of the same video: an original video, an output of our system, an output result from Microsoft Hyperlapse (HL) [6], an output from Deshaker (DS) [28], an output from Youtube stabilizer [2] and an output from [3]. The original videos are 10 seconds and recorded by a GoPro Hero Session 4 with 1080p. To minimize the black area of all results, each output is cropped to 1280×720.

The objective measurement of video stability is based on inter-frame transformation fidelity (ITF) [29]. The test results are shown in Table I where a larger value indicates higher video stability. According to Table I, all stabilization methods successfully stabilize the videos. Although our method does not have the highest value among all methods, the difference between all 5 methods are significantly smaller than the improvements.

TABLE I: ITF scores of different video versions

	Orig	Ours	HL	DS	[2]	[3]
Yard	29.2	33.5 (0.6%)	33.7	33.7 (17.2%)	32.9	**34.9** (37.0%)
Cave	29.3	33.6 (0.03%)	33.8	33.8 (7.3%)	33.5	**34.1** (7.8%)
Beach	26.6	30.3 (0.74%)	**30.9**	**30.9** (15.5%)	30.5	30.8 (10.0%)
Climb1	28.0	32.7 (0.5%)	32.6	32.6 (19.0%)	32.3	**33.2** (22.6%)
Climb2	28.4	32.4 (0.76%)	**33.6**	**33.6** (13.0%)	32.3	33.0 (13.4%)
Average	28.3	32.5 (0.5%)	32.9	32.9 (14.4%)	32.3	**33.2** (18.2%)

TABLE II: Revised ITF scores

	Orig	Ours	HL	DS	[2]	[3]
Yard	29.2	33.5	**33.7**	33.0	32.9	32.3
Cave	29.3	33.6	**33.8**	33.4	33.5	33.7
Beach	26.6	30.3	**30.9**	30.4	30.5	30.3
Climb1	28.0	**32.7**	32.6	32.2	32.3	32.0
Climb2	28.4	32.4	**33.6**	32.5	32.3	32.3
Average	28.3	32.5	**32.9**	32.3	32.3	32.1

Note that ITF only measures the ability of a stabilization method to smooth the camera motion. However, ITF cannot measure the amount of black pixels at the edges of the image. Table I shows, in parentheses, the percentage of black area for each method. Hyperlapse and Youtube stabilizer do not have black area while [28] and [3] have significant amount of black area. Note that when we obtain the results from [3], [28], we disable the option to remove the black area. When this option is enabled, it either scales the frames or introduces significant stitching errors. However, this black area issue is not reflected by the ITF scores, which will significantly degrade the stability in practical situation. For example, the Youtube stabilizer has the highest ITF score while its resulting videos are obviously less stable than our method, which can be verified in the database [27].

As a result, we modified the ITF by taking the black area issue into account. The revised ITF is calculated as:

$$ITF = \frac{1}{N-1} \sum_{k=1}^{N-1} PSNR(k), \tag{15}$$

where N is the number of frames, and:

$$PSNR(k) = 10 \log_{10} \Big(\frac{255^2}{MSE(k)} \Big). \tag{16}$$

To deal with the black area issue, we compute the mean square error (MSE) based on the average of non-black area S of adjacent frames:

$$MSE(k) = \frac{1}{S} \sum_{i} \sum_{j} \Big(I_k(i,j) - I_{k-1}(i,j)) \Big)^2. \tag{17}$$

The revised ITF scores are shown in Table II, where Hyperlapse has the highest score and our method is in the second place.

B. Subjective tests

Our proposed method is not just a video stabilizer. It is designed to improve the viewability of FPVs. We aim to stabilize FPVs while preserving its FPMI. Therefore, it is critical to evaluate our method subjectively. Thus, we conducted a subjective test with 25 participants. If our human perception model holds true, participants should evaluate our resulting videos to have higher stability than the original videos and higher FPMI than videos, produced by other video enhancing methods.

To be exempted from the black area issue, the Hyperlapse is chosen to compare with our system. It is one of the available systems that has a similar goal to ours: enhancing viewability of FPVs. However, unlike other similar ones [13]–[15], it works well when the playing speed is 1, which ensures no frames are discarded. Also, it is representative since its has similar ITF scores to other video stabilization methods [2], [3], [28].

Test videos are played on a 27-inch screen with 82 PPI. The camera is calibrated and the focal length is 830 pixels, so the equivalent focal length is about 10 inches. The viewing distance of participants is set to 40 inches. As a result, the scale of motion perceived is 0.25 as discussed in section III-C. Also the α in our system is set to 0.02 for yaw motion and 0.001 for pitch motion. To run the particle swarm for equation (14), 800 particles are used. The overall speed of our system is around 6 seconds per frame with nearly 80% of the time is spent on 3D motion estimation. In this paper, we do not focus on optimizing the motion estimation algorithm, which can be achieved by cooperating with the work in [30].

Our test was done using paired comparison. In each scene, each pair of videos are shown to participants who are asked the following questions: (1) **Which video is more stable**; (2) **Whether it is stable or not, in which video you can recognize more First-person motion**; (3) **If your friend tries to share his/her First-person experience with you, which one do you prefer**. The participants have to choose one of the two videos as an answer. The subjective scores are computed using Bradley-Terry model [31] and shown in Fig. 3. Higher subjective score indicates higher stability, more FPMI or higher preference.

In Fig. 3, we can see that the original videos have the best FPMI, the Hyperlapse videos have the best stability, and ours are in second place on both FPMI and stability. In addition, the FPMI of our resulting videos is very close to that of the original videos, while their stability is midway between the other two. This demonstrates that we achieved the intended goal of stabilizing FPVs while preserving the FPMI as much as possible. Moreover, our resulting videos have the highest preference while that of Hyperlapse videos' is lower and close to the original videos. This may be explained using the feedback of several participants: although the result of Hyperlapse is stable, the video style is more like flying rather than a First-person style such as running or jumping. This also indicates that the advantage of preserving FPMI is that the resulting videos are more interesting and more realistic.

Fig. 4 shows the estimated yaw motion of all three versions of a running video [27]. At the beginning, there are about 100 frames of head bobbing in the original video, which causes a shaky video. Our method reduces the amplitude of those sine-waves, stabilizing the video while preserving enough FPMI to convey the running experience. However, all the First-person motions are removed in Hyperlapse video. At around 150 frames into the video, the recorder turns around and looks at houses. Compared with our result and the original video, the result of [6] loses the information of the recorder's interests.

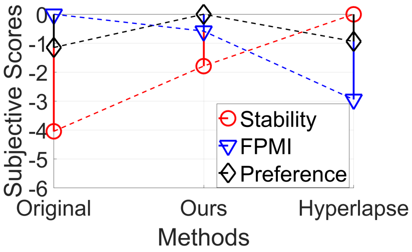

Fig. 3: Subjective Scores of 3 versions of videos

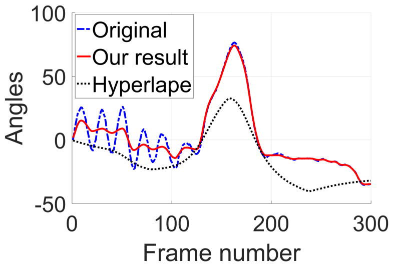

Fig. 4: Example of Yaw motions of 3 versions of videos

V. Conlcusion

In this paper, we propose a system that can improve the viewability of FPVs by stabilizing them while preserving their FPMI. Based on our human perception model, the stability is described using the fraction of a FPV that a human can follow. The objective test shows that our method has similar performance on smoothing camera motion relative to the Hyperlapse method or ordinary video stabilization methods [2], [3], [28]. The subjective test shows that, compared with original videos and results from [6], our results have a middle level of stability and a high level of FPMI that is close to that of the original videos. Moreover, our results also have higher preference scores. In this work, we do not concentrate on removing rolling shutter. We plan to include it in our future work. Rolling shutter can be removed by extending our motion estimation results to incorporate the approaches in [32]. We also plan to refine our resulting videos by applying image stitching algorithms to further remove the black areas.

References

[1] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, "Video stabilization using robust feature trajectories," in *IEEE International Conference on Computer Vision*, 2009, pp. 1397–1404.

[2] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust l1 optimal camera paths," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 225–232.

[3] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Transactions on Graphics*, vol. 30, no. 1, p. 4, 2011.

[4] H. Qu and L. Song, "Video stabilization with L1–L2 optimization," in *IEEE International Conference on Image Processing*, 2013, pp. 29–33.

[5] G. Puglisi and S. Battiato, "A robust image alignment algorithm for video stabilization purposes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1390–1400, 2011.

[6] N. Joshi, W. Kienzle, M. Toelle, M. Uyttendaele, and M. F. Cohen, "Real-time hyperlapse creation via optimal frame selection," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 63, 2015.

[7] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 44, 2009.

[8] G. Zhang, W. Hua, X. Qin, Y. Shao, and H. Bao, "Video stabilization based on a 3D perspective camera model," *The Visual Computer*, vol. 25, no. 11, pp. 997–1008, 2009.

[9] E. Ringaby and P.-E. Forssén, "Efficient video rectification and stabilisation for cell-phones," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 335–352, 2012.

[10] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin, "Multiperspective panoramas for Cel animation," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 243–250.

[11] M. L. Gleicher and F. Liu, "Re-cinematography: improving the camera dynamics of casual video," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 27–36.

[12] ——, "Re-cinematography: Improving the camerawork of casual video," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 5, no. 1, p. 2, 2008.

[13] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *European Conference on Computer Vision*. Springer, 2014, pp. 505–520.

[14] Y. Poleg, T. Halperin, C. Arora, and S. Peleg, "Egosampling: Fast-forward and stereo for egocentric videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4768–4776.

[15] M. M. Silva, W. L. S. Ramos, J. P. K. Ferreira, M. F. M. Campos, and E. R. Nascimento, "Towards semantic fast-forward and stabilized egocentric videos," in *European Conference on Computer Vision*. Springer, 2016, pp. 557–571.

[16] J. L. Carrivick, M. W. Smith, and D. J. Quincey, "Background to Structure from Motion," *Structure from Motion in the Geosciences*, pp. 37–59.

[17] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[18] S. Kasahara, S. Nagai, and J. 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.

[19] S. Foley, "Camera Movement in First Person Games," Ph.D. dissertation, Worcester Polytechnic Institute, 2010.

[20] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4597–4604.

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] D. Purves, G. J. Augustine, D. Fitzpatrick, L. C. Katz, A.-S. LaMantia, J. O. McNamara, and S. Williams, "Types of eye movements and their functions," *Neuroscience*, pp. 361–390, 2001.

[24] S. De Brouwer, D. Yuksel, G. Blohm, M. Missal, and P. Lefèvre, "What triggers catch-up saccades during visual tracking?" *Journal of Neurophysiology*, vol. 87, no. 3, pp. 1646–1650, 2002.

[25] S. Aw, G. Halmagyi, T. Haslwanter, I. Curthoys, R. Yavor, and M. Todd, "Three-dimensional vector analysis of the human vestibuloocular reflex in response to high-acceleration head rotations. II. Responses in subjects with unilateral vestibular loss and selective semicircular canal occlusion," *Journal of Neurophysiology*, vol. 76, no. 6, pp. 4021–4030, 1996.

[26] M. Yanoff, J. Duker, and J. Augsburger, *Ophthalmology*. Mosby Elsevier, 2009.

[27] "Test-set of enhancing viewability of FPVs," https://engineering.purdue.edu/VADL/resources/Enhancing_Viewability/testset_for_enhancingFPV.zip.

[28] G. Thalin, "Deshaker–video stabilizer," *Online at: http://guthspot.se/video/deshaker.htm*, 2014.

[29] L. Marcenaro, G. Vernazza, and C. S. Regazzoni, "Image stabilization algorithms for video-surveillance applications," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1. IEEE, 2001, pp. 349–352.

[30] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *arXiv preprint arXiv:1607.02565*, 2016.

[31] J. C. Handley, "Comparative analysis of bradley-terry and thurstone-mosteller paired comparison models for image quality assessment," in *PICS*, vol. 1, 2001, pp. 108–112.

[32] P.-E. Forssén and E. Ringaby, "Rectifying rolling shutter video from hand-held devices," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 507–514.