A PERCEPTUALLY-INSPIRED 2D VIDEO STABILITY ESTIMATOR

Biao Ma and Amy R. Reibman

School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA

ABSTRACT

Video stability is a significant problem for videos captured by handheld or body-worn cameras. An accurate video stability estimation that is consistent with human perception is the basis of effective video stabilization algorithms. It is also useful for comparing different video stabilization algorithms and constructing a benchmark. In this paper, we present a perception-inspired video stability estimator based on 2D image motions. It calculates the fraction of information in each frame that can be perceived by human eyes. Experimental results show that our stability estimator can accurately estimate subjective video stability scores. It requires less time to compute and is more accurate and robust under different scene structures than methods based on 3D motions.

***Index Terms*—** Video stability, eye movement, estimator, 2D image motion

1. INTRODUCTION

Mobile cameras play an important role in our daily living. They are installed on our cellphones or wearable devices, and allow us to capture videos nearly any time and any place. However, the resulting videos are often too shaky to watch comfortably. Thus, it is important both to estimate the stability of videos before releasing them and to evaluate the effectiveness of the applied video stabilization algorithms [1–7]. Our goal is find such a video stability estimator.

The criteria of what makes a good stability estimator depends on the application [8]. For scenarios which require fast computation, such as video compression, objective estimators like Peak Signal-to-Noise Ratio (PSNR) are preferred. The stability estimator we want here should be consistent with human perception because then (1) a video that is evaluated to be stable is comfortable for humans to watch, and (2) the evaluation criteria of the estimator is consistent across different scene structures so that different stabilization algorithms can be compared using different videos statistically.

Currently, there are many video stability estimators, but few of them are consistent with human perception. Some [1–7] are side-products implicitly generated by a video stabilization algorithm. Some [9–15] are independent studies based on video quality assessment. They all compute stability scores based on the camera motion and can be classified into two categories: the motion types they analyze and the motion properties they measure.

Different motion types: The video stability estimators are based on analyzing 2D image motion or 3D camera motions. 2D image motion refers to motion in the image plane, i.e., the motion of interest points from frame to frame. Estimators [1–4, 9–15] either track local feature points or apply optical-flow algorithms to calculate the 2D motion. 3D motion refers to the real camera movements in the 3D world. These methods [5–8] estimate camera movements from frame to frame using the epipolar geometry.

To estimate the degree of video stability, approaches based on 2D motions have two advantages. Firstly, the 2D image motion is faster and more robust to compute. The whole process of 3D motion estimation is fragile and time consuming. To accurately estimate the 3D camera motion, enough local feature points need to be observed across different frames, and computational power is needed for optimization. Secondly, theoretically, video stability estimation based on 2D image motions has a greater potential to be consistent with human perception. This is because what viewers actually perceive are 2D images with 2D motions. 2D motions include the information of both the camera and the objects in the video. However, the 3D camera motion only provides information about the camera itself; the scene structure information is missing. Therefore, to achieve these two advantages, in this paper we measure the stability based on 2D image motion.

Different motion characterizations: To analyze the stability or shakiness of camera motions, methods focus on two different characterizations of the motion: intensity and frequency. Studies that focus on the camera motion intensity assume that camera motion with higher intensity is shakier. Most of them use the camera motion amplitude as the measurement of the motion intensity. For example, [9, 10] uses the average motion amplitude of local feature points between frames as the measurement. The Inter-Frame Transformation Fidelity (ITF) [11] is a special case in this category, which does not estimate the actual motion but uses the PSNR between adjacent frames as the measurement of motion intensity. One problem of these methods is that they do not consider the frequency of the camera motion. For instance, back-and-forth motion and single-direction motion may generate the same score under these algorithms but viewers would perceive a different degree of stability.

Therefore, many authors [12–15] consider the problem in the frequency domain. After obtaining the camera motion, some use the first, second and even the third derivative of the motion amplitude to measure the motion shakiness. Many video stabilization algorithms adopt these methods since they are easy to calculate. In [12–14], the original motion is first filtered to generate a smooth motion. Then the difference between the original motion and the smoothed motion is assumed to be the high frequency component, which is used as the shakiness measurement. However, these methods also have a drawback in that they do not have a reliable human perception model. The threshold between shaky motion and smooth motion in the frequency domain is ambiguous. A recent work [15] tries to solve this problem by applying a machine learning process to analyze how different frequency bands of camera motion influence human perception.

We approach this problem using a human perception model learned from psychophysics. In our previous works [8, 16], we proposed an algorithm based on the 3D camera motion, which is called 3D-based Viewing Experience score. It is fragile because of an unstable motion estimation process, and it relies on saliency models. In this paper, based on the same human perception model, we propose an estimator that measures the video stability using 2D image motion. **The resulting estimator is more robust and faster to compute. Moreover, it does not depend on different scene structures and its scores are consistent with human perception even without incorporating saliency models**. In section 2, we introduce our new stability estimator: **2D-based Viewing Experience score**. In section 3, experimental results are presented. We show that our stability estimator can accurately measure the subjective video stability. Under different scene structures, it is more robust than compared methods and our previous work [8], and has higher accuracy in practical situations. Finally, we conclude our work in section 4.

2. 2D-BASED VIEWING EXPERIENCE SCORE

Our new video stability estimator, 2D-based Viewing Experience Score (2D-VE score), measures the fraction of visual information that can be perceived by viewers across frames on a block-by-block basis. In this section, we introduce the inspiration of 2D-VE score: a human perception model and its 2D mathematical model.

2.1. Smooth pursuit eye movement

To evaluate the video stability, we incorporate a human perception model. This differs from existing works which evaluate the smoothness of the motion simply using derivatives of the motion. The basis of this human perception model is a human eye movement called Smooth Pursuit Eye Movement (SPEM). Its mathematical model gives us an accurate and clear answer of whether a specific motion is stable or shaky.

Watching videos can be considered as a visual tracking task. SPEM is the main eye movements that viewers apply to track visual targets. It is applied after the visual target has started to be tracked by viewers' eyes. And during the SPEM period, visual information can be perceived by human eyes. The more information viewers can perceive, the more stable the video is. However, to track the visual target, viewers need to perform another eye movement called the **catch-up saccade**, which takes around 0.15 to 0.2 seconds [17]. During the catch-up saccade period, nearly no visual information is perceived, which is the reason for the feeling of shakiness. To evaluate the stability, we apply a mathematical model on 2D image motions, which helps us to judge when the SPEM or the catch-up saccade is performed. The mathematical model is proposed in our previous work [16] and based on results from psychophysics research [18].

2.2. Our previous work

Our proposed mathematical model of the eye movement status is based on the target motion and eye motion in three adjacent frames. We assume the viewer is tracking a visual target from frame n to frame $(n+1)$ using SPEM. If a target suddenly moves and its position in frame $(n+2)$ is out of the region predicted by the human visual system, the viewer may fail to track the target and may need to perform a catch-up saccade to re-catch the same target or to choose and lock on another visual target. The proposed mathematical model is shown in condition (1). It is used to judge if the target moves out of the predicted region. The derivation of the model can be found in [16, 18].

$$0.04 \leq \frac{|PE(\beta_n; n+2)| + b}{|\omega(\beta_n; n+2)|} \leq 0.18 \; or \; |PE(\beta_n; n+2)| < MAR. \tag{1}$$

Condition (1) is based on angular values. PE is the position error between the target and the eye-gaze point at frame $(n+2)$. ω is the retinal slip at frame $(n+2)$. MAR is the minimum angular resolution of human eyes. b is the bias of position error estimation. β_n is the target angular position with respect to the camera at the n^{th} frame.

Note that both the position error PE and retinal slip ω are functions of the target motion and the eye motion at frame n to frame $(n+2)$. In our previous work, we use the 3D camera motion to approximate these under the assumption that the angular positions of visual targets are uniformly distributed. This leads to the consequence that the linear relationship between measure scores and subjective scores vary across different scene structures. This means, without incorporating that has an accurate saliency model, the resulting measurement depends on different scene structures. Meanwhile, if the 3D camera motion estimation fails or has errors, the measurement result is also inaccurate. Therefore, in this work, we propose a new estimator that computes the video stability using 2D image motions, so it is more robust under motion estimation errors and does not depend on different scene structures.

2.3. 2D-Mathematical model of SPEM

To create our 2D-VE stability estimator, we recalculate condition (1) based on target position on the screen with respect to human eyes using the 2D image motion of interest points instead of target angular position β_n with respect the camera.

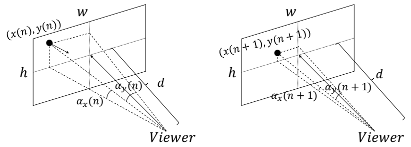


Fig. 1: 2D image motion of a visual target from frame n to frame $(n+1)$.

Assume a viewer tracks a target from frame n to $(n+1)$ and its 2D image motion is estimated as shown in Fig. 1. The target moves from $\big(x(n), y(n)\big)$ to $\big(x(n+1), y(n+1)\big)$ while this event is observed by the viewer with viewing distance d on an image with width w and height h. $\alpha_x(n)$ and $\alpha_y(n)$ are target angular positions in the viewer's eyes in the horizontal and vertical directions respectively. $\alpha_x(n)$ can be calculated as:

$$\alpha_x(n) = \arctan \frac{\big|x(n) - w/2\big|}{\sqrt{\big(y(n) - h/2)^2 + d^2}}. \tag{2}$$

$\alpha_y(n)$ and similar results can be derived for target angular positions in frame $(n+1)$ and $(n+2)$. We assume the viewer's eye movement has a constant speed after the target is tracked at frame n. Then the position error PE and retinal slip ω at frame $(n+2)$ in the horizontal or vertical direction can be calculated:

$$PE_k(n+2) = \alpha_k(n+2) - 2\alpha_k(n+1) + \alpha_k(n), \tag{3}$$

$$\omega_k(n+2) = \frac{PE_k(n+2)}{FPS}, \tag{4}$$

where $k = x$ or y indicating the horizontal and vertical components. FPS is the video frame rate. After substituting Equation (3) and (4) into condition (1), if the condition is satisfied for both vertical and horizontal motion, we can conclude that the current tracked target can be continually tracked at frame $(n+2)$. We use $C(\cdot)$ to denote the processing of checking condition (1) based on the input motion. The output of $C(\cdot)$ is binary: "1" indicates that the condition is satisfied for both vertical and horizontal motion. "0" indicates a catch-up saccade will be triggered. During the required 200ms (about 5 frames at 30fps) nearly no information can be perceived. In the next section, we introduce how the video stability is computed using this 2D mathematical model.

2.4. 2D-VE of a video

Given a video, we first identify the interest points in each frame. Then we calculate the possible tracking length of each interest points using the model we propose above. The stability of a video is measured using the averaged possible tracking length of all interest points.

For a N-frame length video segment, we split its frames into I by J regions (vertically and horizontally). For each region, we choose its center as the interest point. Then we calculate the optical-flow of each frame. Suppose we start to track the target from frame k and focus on the $(i,j)^{th}$ interest point. Its pixel location at frame $n (k < n \leq N)$ is interpolated using the computed optical-flow of frame n, denoted as $m_{i,j,k}(n)$. We define the stability S of the $(i,j,k)^{th}$ interest point at frame $(n+2)$ to be:

$$S_{i,j,k}(n+2) = C\Big(m_{i,j,k}(n); m_{i,j,k}(n+1)\Big) \tag{5}$$

$S_{i,j,k}(n+2)$ represents whether the $(i,j,k)^{th}$ target can be tracked using SPEM at frame $(n+2)$. This is calculated only based on the motion of frame n and $(n+1)$. However, to know about the stability of this target across the time, we need also consider the catch-up saccade periods. For example, if $S_{i,j,k}$ is "1011111", then it needs to be converted to "1000001" to account for the fact that once a viewer can no longer track at the second frame, and that a catch-up saccade lasting 5 frames is required. For the converted resulting vector, we use the notation $S^*_{i,j,k}$.

To calculate the stability of an N-frame video segment, we need to calculate equation (5) for all IJ interest points and for all possible starting frame index k $(1 \leq k < N)$. The overall video stability is estimated using our 2D-VE score, which is calculated as:

$$VE_{2D} = \sqrt{\frac{1}{K-N+1} \sum_{k=1}^{K-N+1} \Big(mean_{i,j} \frac{|S^*_{i,j,k}|}{N}\Big)^2}. \tag{6}$$

K is the total number of frames while $|\cdot|$ computes the L1 norm. $\frac{|S^*_{i,j,k}|}{N}$ calculates the fraction of frames over which the $(i,j,k)^{th}$ interest point can be tracked using SPEM. And the overall VE score is the average over all interest points. In the experiments, we set N to 30 frames, I and J to 10 and 20.

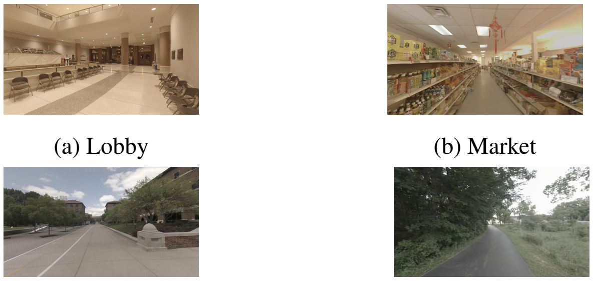
3. EXPERIMENTAL RESULTS

In this section, we present the performance and usage of our stability estimator 2D-VE. We apply the results of our previous subjective test [8] to help us learn about the bias parameter in the human eye movement model (1) when using 2D motion. Meanwhile, we explore predictive models that can use the 2D-VE scores to predict subjective video stability scores. Compared with our previous work and two other estimators [11, 14], our proposed method has good predictive accuracy and is more robust across different scenes. And we demonstrate how we use our method to effectively compare different video stabilization algorithms.

3.1. Performance of 2D-VE score

The bias parameter in the mathematical model (1) must be learned using subjective tests. It characterizes human visual tracking ability including the position error estimation and the feedback control of eye movements.

In [8], we built a dataset which has videos from 4 different scenes. Fig. 2 shows example frames of the dataset. Each scene has 9 different versions which have different amount of synthesized motions. For each scene, we select 4 versions to be the training set and 5 versions to be the test set. We performed a subjective test to acquire their subjective stability scores and used the scores to learn about the bias parameter for the 3D motion model. All test details can be found in [8]. Using the test result, we can obtain the optimal bias parameter which maximizes the sum value of the Pearson Linear Correlation Coefficient (PLCC) between the subjective scores and the corresponding VE scores. Meanwhile, we get the fitting models between the subjective scores and VE scores.



(a) Lobby

(b) Market

(c) University Street (US)

(d) Park Walkway (PW)

Fig. 2: Example frames of source videos

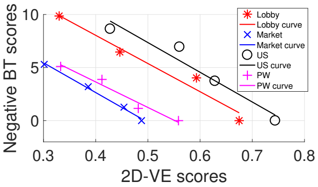


Fig. 3: Fitting models of 2D-VE scores and subjective video stability scores

In this paper, we apply the same dataset and experiment to help us learn about the 2D motion model. The final bias parameter is 0.038. The learned fitting models corresponding to the optimal bias parameter are shown in Fig.3.

We use the models to predict the subjective scores of the videos in the test set. The PLCCs and the average of mean square errors (MSE) are shown in Table 1. We also fit the subjective scores with the estimator scores of [14] and [11]. The fitting and predicting results are also shown in Table 1.

Since the videos in this database are using synthesized motions, we can access the ground truth of the camera motion. Table 1 shows that our 3D-VE with saliency models computed over ground truth motion has the lowest MSE of the prediction. However, in practice, we rarely have access to the ground truth motions. In this case, we estimate them using the-state-of-art visual odometer [19], the performance of 3D-VE is not as good as before. However, even when using estimated motion, our 2D-VE still has a similar performance to the 3D-VE (using ground truth motion) **though it does not incorporate any saliency model at all**.

The last column of Table 1 is another important criteria for the performance of estimators, which is the coefficient variance (cv%) of slopes of the fitting lines across different scenes. The smaller the value is, the more similar the slopes of the fitting lines are. We desire to have fitting lines with similar slopes, because it means the estimator does not depend on different scene structures. Otherwise, a similar amount of score improvement between a stabilized video and the corresponding shaky video would have different subjective meanings under different scene structures. In [8], we showed that incorporating a saliency model into the 3D-VE significantly reduced the cv%. However, our 2D-VE has a lower cv%, even without a saliency model. The reason is that the 2D motion model not only includes the information of the camera motion, but also includes the information of the scene structure.

Table 1: Results of fitting and testing

Measurement	PLCC	MSE	Average MSE	cv% of slopes
2D-VE	0.9814	0.6271	0.5662	**7.0**
	0.9972	0.2259		
	0.9711	0.6959		
	0.9816	0.7161		
3D-VE [8] (saliency models + ground truth motion)	0.9744	0.4886	**0.4927**	7.9
	0.923	0.7962		
	0.9687	0.3291		
	0.9513	0.3570		
3D-VE [8] (saliency models + estimated motion)	0.9854	0.6116	0.7392	16.97
	0.934	0.7163		
	0.9684	0.8228		
	0.9191	0.8086		
MV-MSE [14]	0.9657	1.2538	1.0571	28.6
	0.9644	0.864		
	0.9374	1.3048		
	0.9496	0.8056		
ITF [11]	0.9960	0.7539	0.9086	12.3
	0.9526	1.1421		
	0.9985	0.9		
	0.9940	0.8383		

Table 2: Compare video stabilization methods using the difference between the 2D-VE of stabilized video and the original video

	Ma et al. [20]	Microsoft [21]	Deshaker [22]	Youtube [1]	Liu et al. [3]
Yard	0.272	**0.46**	0.03	0.213	0.005
Cave	0.266	**0.344**	0.121	0.243	0.192
Beach	0.258	**0.396**	0.008	0.221	0.197
Climb1	**0.265**	0.227	0.063	0.057	0.044
Climb2	**0.284**	0.271	0.016	0.157	0.035
Average	0.269	**0.34**	0.047	0.178	0.094
Std	0.01	0.094	0.046	0.075	0.092

In addition, the computational speed of our 2D-VE is much faster than 3D-VE. In our implementation, accurate 3D motion estimation needs at least 5 seconds for each frame while to compute 2D-VE, we only need 400 ms per frame.

All these desirable features of our 2D-VE score enable us to use it to effectively and statistically evaluate video stabilization algorithms using many videos. Table 2 shows an example. We apply 5 stabilization algorithms [1,3,20–22] on our video set [23] and compute the relative scores between the stabilized videos and original videos. As can be seen, the most effective stabilization algorithm is [21], and [20] is in the the second place. However, because [20] is more carefully designed based on stability measurement, it is more robust than [21] according to the standard deviation of their 2D-VE scores. Methods [22] and [3] do not have good performance on this dataset. All these are consistent with our previous observations in [20] and can be verified visually using [23]. Note that these analyses using average scores and standard deviation are only meaningful when the estimator is accurate and does not depend on different scene structures.

4. CONLCUSION

In this paper, we propose a 2D-based video stability estimator: 2D-based Viewing Experience (2D-VE) score. It can accurately measure the subjective video stability and is more accurate than other methods including our previous work [8]. Although it is modified from our previous 3D-VE [8], since our 2D-VE score is based on 2D image motions, it is faster computationally and more robust under practical situations. Unlike [8], its scores are more consistent with human perception even without incorporating a saliency model. Using our 2D-VE score, video stabilization algorithms can be effectively and systematically evaluated.

5. REFERENCES

[1] Matthias Grundmann, Vivek Kwatra, and Irfan Essa, "Auto-directed video stabilization with robust L-1 optimal camera paths," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 225–232.

[2] Ken-Yi Lee, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung, "Video stabilization using robust feature trajectories," in *IEEE International Conference on Computer Vision*, 2009, pp. 1397–1404.

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

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

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

[6] Guofeng Zhang, Wei Hua, Xueying Qin, Yuanlong Shao, and Hujun Bao, "Video stabilization based on a 3D perspective camera model," *The Visual Computer*, vol. 25, no. 11, pp. 997–1008, 2009.

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

[8] Biao Ma and Amy R. Reibman, "Estimating the subjective video stability of first-person videos," *Human Vision and Electronic Imaging*, 2018, (available at: https://engineering.purdue.edu/VADL/publications/BiaoMa_HVEI18.pdf).

[9] Tao Mei, Xian-Sheng Hua, Cai-Zhi Zhu, He-Qin Zhou, and Shipeng Li, "Home video visual quality assessment with spatiotemporal factors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 6, pp. 699–706, 2007.

[10] Yedid Hoshen, Gil Ben-Artzi, and Shmuel Peleg, "Wisdom of the crowd in egocentric video curation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 573–579.

[11] Lucio Marcenaro, Gianni Vernazza, and Carlo S. Regazzoni, "Image stabilization algorithms for video-surveillance applications," in *International Conference on Image Processing*. IEEE, 2001, vol. 1, pp. 349–352.

[12] Kazi Masudul Alam, Mukesh Saini, Dewan T. Ahmed, and Abdulmotaleb El Saddik, "VeDi: A vehicular crowd-sourced video social network for VANETs," in *IEEE 39th Conference on Local Computer Networks Workshops (LCN Workshops)*, 2014, pp. 738–745.

[13] Mukesh Kumar Saini, Raghudeep Gadde, Shuicheng Yan, and Wei Tsang Ooi, "Movimash: online mobile video mashup," in *Proceedings of the 20th International Conference on Multimedia*. ACM, 2012, pp. 139–148.

[14] M.J. Tanakian, M. Rezaei, and F. Mohanna, "Camera motion modeling for video stabilization performance assessment," in *Machine Vision and Image Processing (MVIP)*. IEEE, 2011, pp. 1–4.

[15] Zhaoxiong Cui and Tingting Jiang, "No-reference video shakiness quality assessment," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 396–411.

[16] Biao Ma and Amy R. Reibman, "Measuring and Improving the Viewing Experience of First-person Videos," in *ACM Multimedia Thematic Workshops 2017*. ACM, 2017.

[17] Sophie de Brouwer, Marcus Missal, Graham Barnes, and Philippe Lefèvre, "Quantitative analysis of catch-up saccades during sustained pursuit," *Journal of Neurophysiology*, vol. 87, no. 4, pp. 1772–1780, 2002.

[18] Sophie De Brouwer, Demet Yuksel, Gunnar Blohm, Marcus Missal, and Philippe Lefèvre, "What triggers catch-up saccades during visual tracking?," *Journal of Neurophysiology*, vol. 87, no. 3, pp. 1646–1650, 2002.

[19] Jakob Engel, Vladlen Koltun, and Daniel Cremers, "Direct sparse odometry," *arXiv preprint arXiv:1607.02565*, 2016.

[20] Biao Ma and Amy R. Reibman, "Enhancing Viewability for First-person Videos based on a Human Perception Model," in *IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, 2017.

[21] Neel Joshi, Wolf Kienzle, Mike Toelle, Matt Uyttendaele, and Michael F. Cohen, "Real-time Hyperlapse creation via optimal frame selection," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 63, 2015.

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

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