Camera Motion Estimation by Feature Masking and Kalman Filter for Video Stabilization

Jay Neil Gapuz
Electrical and Electronics Engineering Institute
University of the Philippines - Diliman
Quezon City, Philippines
jayneil.gapuz@live.com/jay.neil.gapuz@eee.upd.edu.ph

Abstract— Handheld devices used for capturing videos are susceptible to shaky motion during the recording process. Various post processing techniques can correct and stabilize the unwanted movement the video had suffered. This paper introduces a simple and robust method to stabilize video by means of feature masking to compare frames and estimate the camera motion of the captured video. The calculated camera flow is smoothened using a Kalman filter, and the resulting motion provides compensation on the pixel shift relative to the obtained estimate. Effectiveness of this method was evaluated by comparing the result of simulated shaky video to the stabilized video. The system has provided significant reduction on the unwanted camera movement when applied on the actual sample video files.

Index Terms— Digital video, Feature extraction, camera motion, Kalman filter, dual axis stabilization

I. Introduction

Several handheld digital and drone cameras are being utilized nowadays to record scenes. These cameras capture image sequences and translate them to a video. However, not smooth camera movement may result in unsteady image sequences thus creating a shaky or jerky output. Video stabilization techniques have been introduced to compensate for the irregular movements which video experienced during recording.

The video stabilization can either be achieved by hardware or post image processing approach. Hardware approach can be further classified as mechanical or optical stabilization. Mechanical stabilizer uses gyroscopic sensor to stabilize the entire camera [1]. But most people use low-cost compact cameras which are not equipped with necessary hardware to compensate for the unsteadiness. This is the reason why post processing algorithms are highly preferred.

Post processing video stabilization is normally composed of three major phases: (1) motion estimation, (2) motion filtering and (3) motion compensation. Methods presented in this paper underlie these stages. Tracking the camera movement is necessary to attain a smooth flow of the video by compensating the instability of the motion during the recording process. Handheld camera devices, which are normally light, are susceptible to dual axis unsteadiness especially during panning. Computing the amount of shift at each point from the interpolated camera position will result to a smoother image sequence flow.

Simple IIR Low pass filter cannot provide an optimal smooth camera flow and introduces a delay during motion displacement calculation. This can give poorer displacement over frames, since this method will also require proper selection of filter coefficients. Kalman filtering is known to be an effective method to keep track and optimize the next measurement points given the series of measurements over time without creating a delay in the system.

This paper introduces a simple and efficient algorithm to track camera motion by masking the feature extracted from the frames and stabilizing the movement by using the Kalman filter which predicts and updates displacement at given points. The study focuses on providing a robust dual axis stabilization to establish a steady output given a shaky camera movement.

II. Proposed Digital Video Stabilization System and Algorithm

The proposed system is comprised of four main stages: segmentation and pre-processing, feature extraction and masking, camera motion estimation and filtering, and lastly frame transformation to create a stabilized image sequence.

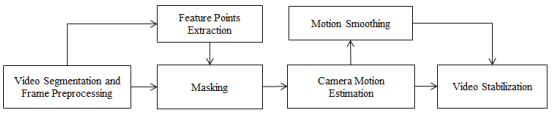

Figure 1. Block diagram of the system

Segmentation process splits the video into frames by analyzing the frame rate and separating image sequences into blocks. A mask proportional to the width and height of a single frame is then created. Each block passes through a pre-processing stage where the image is transformed into grayscale and then normalized. To emphasize and minimize the objects to be extracted with features, the block is passed through a Sobel edge detector. This process will also filter out additive image noise that may affect feature extraction. During this process, the image has been transformed into a one-bit image to reduce pixel density for a faster operation and provide an efficient extraction of feature points.

The method of feature extraction that was used in this paper is Shi-Tomasi detection method which is an improvement of Harris Corner detection. The goal of this process is to find relevant corners and edges of the processed frame. Significant points are then obtained and added into the original masked

image. Each feature mask is correlated with the adjacent frame, computing the amount of shift on the x and y axis and interpreted as the new camera position. The shifting is tracked and translated into a camera motion for the total duration of the video. Each motion point is presumed to be affected by unstable movement during the recording process. This will be compensated by passing the noisy camera motion to a Kalman filter with a smoother to further provide an even camera flow. Both the original and filtered motion flow is compared and the difference of the two is computed to acquire the final frame shift. An affine transformation on the x and y axis is applied for each frame depending on the computed values. The image sequences are then synthesized back into a stabilized video.

A. Video Segmentation and Pre-processing

The video is split into frames which match the video height and width. Each frame is necessary to analyze the overall camera motion for the total length of a given footage or clip. The number of frames to be generated is equal to the frame per second used by the camera and the total length of the video wherein the result should be a positive integer.

Frame pre-processing is composed of making the frame into grayscale and normalizing it into a one-bit image. This process also eliminates the problem of frame ambiguity and motion blur when each succeeding frame suddenly differs in brightness level. It also reduces the additive noise that is present in the frame and further emphasizes the objects with the highest interest.

B. Feature Extraction and Masking

A frame mask is created with a proportional dimension as that of the generated frames and passed through an edge detector. A Sobel operator was used due to its simplicity compared to the canny edge method. Edge detection is necessary to simplify and speed up the feature extraction.

The Sobel operator performs a 2-D spatial gradient measurement on images. It uses a pair of horizontal and vertical gradient matrices (described in Figure 2) whose dimensions are 3×3 for edge detection operations [2]. One kernel is simply the other rotated by 90°. It responds maximally to edges running vertically and horizontally relative to the pixel grid.

-1	**0**	**1**
-2	**0**	**2**
-1	**0**	**1**

Sx

1	**2**	**1**
0	**0**	**0**
-1	**-2**	**-1**

Sy

Figure 2. Sobel operators for edge detection

The preprocessed frame is convolved with the Sobel operators and applied with thresholding to enhance the edges due to the gradient nature of the Sobel method. The main disadvantage of this method is noise sensitivity. This is the reason why the frame undergoes a preprocessing to suppress some pixel noise present.

After all edge elements have been identified, the filtered frame will be extracted with feature points. The system utilizes an improved Harris Corner Detection method which is the "*Shi-Tomasi Good Features to Track*" Method. The difference of this method comparing to the latter is the scoring function that is described by Eqn 1.c. The method finds the difference in intensity for a displacement *(u,v)* in all directions expressed by Eq 1.

$$E(u,v) = [u\ v]\ M \begin{bmatrix} u \\ v \end{bmatrix} \quad (1a)$$

$$\text{where } M = \sum_{x,y} w(x,y) \begin{bmatrix} IxIx & IxIy \\ IxIy & IyIy \end{bmatrix} \quad (1b)$$

where *Ix* and *Iy* are image derivatives in x and y directions respectively

$$R = \min(\lambda_1, \lambda_2) \quad (1c)$$

The presence of a corner in a window is decided by setting a threshold value. After the points are determined, it will be added to the mask created. This process replaces the original frame by feature points masked in a new empty frame.

C. Camera Motion Estimation and Motion Filtering

An important consideration for video stabilization is that for every frame transition, it is affected by random movement. This is described by Eq. 2 where s_n and w_n are the wanted and unwanted motion points. If we can estimate the camera motion, the amount of random translation on x and y can be effectively corrected.

$$\boldsymbol{m(x_n,y_n) = s(x_n,y_n) + w(x_n,y_n)} \quad (2)$$

The masked features are compared with the succeeding frames which gets the 2D cross correlation and acquires the pixel position with the maximum correlation. A convolution method with the other frame flipped was used to speed up the process. The displacement is then given by Eq 3. by adding the previous position to the shift difference of the current and previous frame. It is presumed that initial position is at the origin since the first frame is convolved by itself thus pixel position is located at its center.

$$\boldsymbol{p(x_n,y_n) = [f(n) * f(n-k)^*]} \quad \textbf{(3)}$$
$$\boldsymbol{m(x_n,y_n) = p(x_n,y_n) + p(x_{n-1},y_{n-1})}$$

Camera motion is defined by the displacement of the correlated pixel position over definite frame intervals. Since the motion is presumed to have been corrupted by unwanted translation, the optimal camera flow is calculated by using the Kalman filter which doesn't introduce delay, thus providing an easier calculation at any motion point. The Kalman filtering method gives the optimal estimate of the displacement (x_n,y_n) and is described by a state representation shown in Eq. 4.

$$m(k) = A\,m(k-1) + Bu(k) \qquad (4)$$
$$z(k) = H\,m(k) + w(k)$$

The $m(k)$ represents the motion points sequence and is denoted as the sum control signal and process noise. The second equation is a combination of the signal value and the measurement noise which are both considered to be gaussian in nature at any measurement point. Process matrix A should have the same dimension while H and $w(k)$ are both vectors. Motion point $x(k)$ is already defined from the unfiltered motion estimate. Two important methods during Kalman filtering include time update and measurement update. Time update consists of projecting both the state and error covariance ahead, while measurement update computes the Kalman gain, updating the motion estimate by means of the previous measurement value. It is also necessary to define an iteration value in computing the Kalman gain.

The system used a Kalman model where velocity is assumed to be constant during each frame translation on the x and y axis. The method describes the transition matrix and observation matrix in Eq. 5 which both contain the x_n, x_n velocity, y_n and y_n velocity.

transition matrix:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

observation matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (5)$$

It is assumed that all state transitions and observations are linear with Gaussian distributed noise. The system also employed the usage of a Kalman smoother to further improve the steadiness of the motion estimate from the output of the filter. It computes the likelihood of the system and gives a better approximation of sequence states of the camera flow.

D. Motion Compensation

The final stage of video stabilization is compensating for the instability of the camera motion. After knowing the optimal camera motion, each frame undergoes affine transformation to correct the motion. It is compared to the original motion estimate and the difference on the two axes is interpreted as the amount of pixel shift of each frame . The transformed frames are synthesized back to a stabilized video

III. Results and Discussion

The proposed algorithm is evaluated by simulating 10-second videos with minimally moving objects mimicking a panning motion of a handheld camera. This was achieved by plotting points into a frame using Python matplotlib creating an image sequence. A noise is incorporated to simulate the shaky movement of the camera.

For the experiment, Python was used together with the following available libraries: *Scipy, Numpy* and *OpenCV*. The video is transformed first into a series of frames and then read as a grayscale image. Each frame was normalized and passed through an edge detector to further minimize image noise and highlight important objects in the images. A frame mask is generated based on the video dimension.

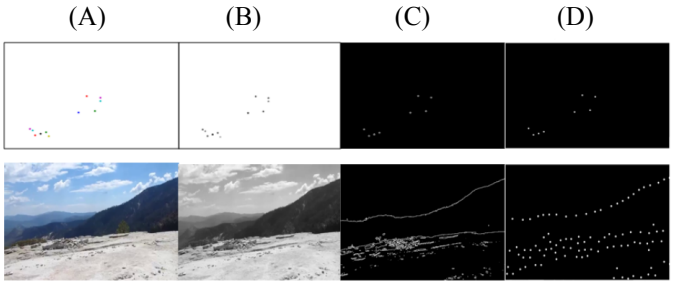

Figure 3. Frames from simulation video *(upper)* and actual video *(lower)*. (A) Original frame (B) Grayscale (C) Edge detection (D) Feature Masking

The minimized frame then passes through the feature selection scheme and extracted features are added to the empty mask. Each frame is correlated to a given frame interval and the maximum correlation point is selected. The method converts this point as the pixel shift for the compared frames.

The estimated camera motion is given by the amount of pixel shift that is added to the previous state. When there is no motion on an axis, there should be no shift incurred, thus it will be the same as the previous state.

Since the motion is presumed to have been affected by a shaky motion, the estimate suffers from erratic translation along the frames. The calculated motion estimate is filtered using a Kalman filter and a smoother was introduced to further smoothen the output which can be observed in Figure 4.

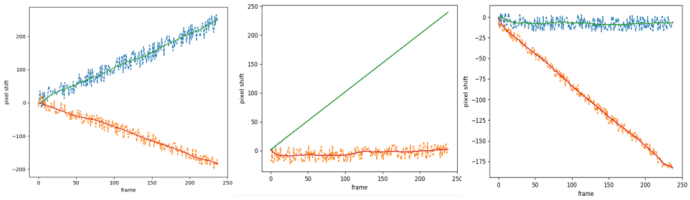

Figure 4. *Original Camera Motion Estimate vs Filtered Camera Motion* (A) Camera moving along x and y axis with noise present on both (B) Camera moving only at the y axis but noise is present at x axis (C) Camera moving along the x axis with noise present on both axes

The optimized camera flow is compared to the original camera estimate and the difference of each motion point is calculated. The amount of change equates to the amount of pixel shift that will be introduced during each frame transformation. Finally, the transformed frames are converted back into a video format. The system writes back the frames to video with similar format as the type of video encoding of the original video.

Videos composed of simulated videos and shaky footage obtained from YouTube. Original videos for simulation include an even motion on a certain axis. A random movement is added to the motion and the results were compared to the original. Minimum relative error was attained and is

perceptibly negligible when the video is played. The compensated motion has provided a smooth camera flow with significant reduction on the shaky motion that was applied.

This method was tested on several videos searched on YouTube to check the effectiveness of the proposed algorithm. Some unwanted motion factors were identified to be caused by shaky hand movement/motion affected by mobility, environmental conditions such as wind, and other external factors such as the car's engine.

Significant reduction on the shaky motion was achieved when the method was implemented and is more evident when the video suffers from a high shaky movement. However, the method cannot compensate for rolling shutter problems in certain videos where wobbling of frames is present since no image warping method has been incorporated. Sudden image rotation was also found to be treated by the algorithm as part of the camera flow, thereby only compensating any irregularities during sudden angular motion. This is only relevant if the stabilization goal is to provide a still image of an object or a still scene throughout the entire video. This implementation also requires prominent features on the image sequence and background that aren't translating on the z axis. Nevertheless, the proposed system can successfully stabilize shaky motion that is relevant during video capturing of handheld devices as observed on the videos post processed with this method.

IV. Conclusion

The paper introduced a simple and robust algorithm to reduce shaky movement during video recording which is a common problem when handheld devices are used. Camera motion estimation was obtained using feature mask comparison among frames and the resulting motion estimate passed through a Kalman filter to provide compensation on the shift that the frames had suffered due to irregularities in movement during recording process, thereby creating a smooth camera flow. The proposed system presents a simpler implementation to stabilize a video with results that are still acceptable.

References

[1] S.Akhila , H. Lokesha and K. Reddy. "A survey on video stabilization algorithms". International Journal of Advanced Information Science and Technology (IJAIST), 2014, pp. 167-170.

[2] S. Guptal, S. Mazumdar. "Sobel Edge Detection Algorithm". International Journal of Computer Science and Management Research Vol 2 Issue 2, February 2013, pp. 1579-1583.

[3] J. Shi, C. Tomasi. "Good Features to Track". IEEE Conference on Computer Vision and Pattern Recognition. June 1994 .

[4] N. Nguyen, D. Laurendeau, A. Albu. "A Robust Method for Camera Motion Estimation in Movies Based on Optical Flow". The 6th International Conference on Information Technology and Applications . 2009.

[5] S. Ertuk. "Real-Time Digital Image Stabilization Using Kalman Filters". Real Time Imaging Vol 8 Issue No. 4. August 2002, pp. 317-328.

[6] R. Hu, R. Shi, I. Shen, and W. Chen. "Video Stabilization Using Scale-Invariant Features". 11th International Conference Information Visualization. 2007.