MVA2011 IAPR Conference on Machine Vision Applications, June 13-15, 2011, Nara, JAPAN

Real-Time Video Stabilization for Unmanned Aerial Vehicles

Yue Wang, ZuJun Hou, Karianto Leman and Richard Chang
Institute for Infocomm Research, A*Star (Agency for Science, Technology and Research), Singapore
ywang, zhou, karianto, rpchang@i2r.a-star.edu.sg

Abstract

This paper presents a new real-time video stabilization method for Unmanned Aerial Vehicles (UAV). It mainly consists of three steps. Firstly, the keypoints are located based on FAST corner detection and preliminarily matched. Secondly, the matched keypoints are then involved for estimation of affine transform to reduce false matching keypoints. Finally, motion estimation is performed based on affine transform model and the compensation for vibration is conducted based on Spline smoothing. The experiments show that our method performs well and processes up to 30 fps.

1. Introduction

Unmanned Aerial Vehicles (UAV) is a very good tool for earth surface exploration and ground target surveillance. As the hardware technology developed, an increasing number of UAV equipped with streaming video cameras have been employed for immediate observation and object detection and tracking. However, due to the fact that the size and structure limitation, UAV is very hard to avoid the mechanical vibration caused by the engine, thus the video received from mobile UAV cameras may not be stable by the unstable motion of the camera sensors. Therefore, it is very difficult to detect and track targets of interest in such unstable video, thus all of these video sequences should be stabilized before being processed in the next stage.

Video stabilization techniques have been studied by researchers for decades, and many video stabilization schemes have been proposed. These methods can be classified into two categories [1]: (1) mechanical based video stabilization and (2) image processing based stabilization. Mechanical based video stabilization employs sensor to detect the camera motion and then shift the image sensor to compensate the vibration by moving to opposite direction of vibration, although this method is able to achieve a good result in some applications like Digital camera, it can't remove all video vibration and moreover, its capability could be limited due to its complex hardware with requirement on some special modules. As for image processing based methods, video stabilization is attained through image processing techniques to estimate the camera motion by computing the degree of geometric transform parameters between consecutive frames of video, smooth the parameters and compensate the deviation of images. Comparing these two methods, image processing based video stabilization is an ideal video stabilization technique which is more precise, flexible, inexpensive and easy to operate.

This paper mainly focuses on the image processing based video stabilization.

The remaining structure of this paper is arranged as follows. Section 2 gives a very brief introduction of related works on video stabilization. Our method is detailed in Section 3. Experimental results and discussion are presented in Section 4. Finally, the conclusion of this paper is presented in Section 5.

2. Related Works

Here we only briefly review those algorithms that have been proposed in recently published literatures. A real–time smoothing methodology for the stabilization of videos captured from small robotic helicopter platforms is introduced in [2]. It uses Lucas–Kanade feature tracker to detect the regions and then estimate the transformation between two consecutive frames. Unintended motion compensation is accomplished by adjusting for extra rotation and displacements that generate vibrations. This method is able to achieve an average speed between 20 and 28 fps with 3 frames delay.

A fast video stabilization algorithm presented in [3] uses circular block to search and match the key places, then the affine transform is estimated, followed by a process of parameter smoothing by the polynomial fitting and prediction method (PFPM). The speed can probably achieve 10 fps.

A method is proposed in [4] to remove the annoying shaky motion and reconstruct a stabilized video sequence with good visual quality. The scale invariant features (SIFT) is applied to estimate the camera motion. The unwanted vibrations are separated from the intentional camera motion with the combination of Gaussian kernel filtering and parabolic fitting. In addition, to reconstruct the undefined areas, resulting from motion compensation, the mosaicing method with Dynamic Programming was adopted. However, no processing speed has been mentioned in this paper.

PCA-SIFT is applied in [5] to detect the feature points in video frames, and then a block is defined for each PCA-SIFT and a cost function is proposed to filter out foreground object pixels. The processing speed of this method is rather slow, less than 2 fps.

A dual pass video stabilization system in [6] uses an iterative method for global motion estimation and an adaptive window smoothing for the intentional motion estimation. This method is only for off-line processing of video stabilization. The processing speed can reach 25 fps.

3. Proposed Algorithm for Video Stabilization

The flowchart of our method is illustrated in Figure 1. First of all, corner based features are extracted by FAST corner detector [7] and matching pairs are determined. Next, motion between two consecutive frames is estimated based on an affine transform model. Subsequently, the estimated motion parameters are cumulated and smoothed by a spline model. Finally, the frames are compensated based on smoothed parameters and form a stable video. The details are explained as follows.

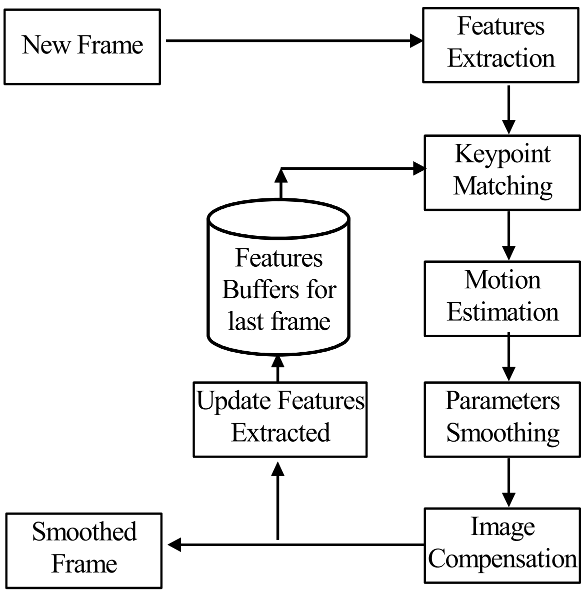

Figure 1. Diagram of the proposed algorithm

3.1. Fast corner points detection and matching

There are many methods exist for keypoint detection, these include SUSAN, Harris, SIFT, SURF and various extensions. To speed up keypoint detection, the corner point detection algorithm of Features from Accelerated Segment Test (FAST) [7] had been employed in this step.

For each point, a 12-by-12 block centered around it is extracted from its respective image frame. Due to the fact that the corresponding points are closed in the consecutive frames, the search areas are constrained with (image_width/5) x (image_height/5), which is for reducing the region and thus reducing the processing time, and also removing the significant outliers. The Sum of Squared Differences (SSD) between their respective image regions is calculated to measure the matching degree between points. Each point in frame A is corresponded with the point in frame B with the lowest matching cost.

3.2. Motion estimation

For numerical simplicity and stability, a simpler-rotation-translation 2D affine transformation with only four unknown parameters is adopted here to describe geometric transformation between two images. Suppose $P(x,y)$ and $P'(x',y')$ to be the pixel location of corresponding points in consecutive video frames, the relationship between these two locations can be expressed by following transform:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

$$A = \begin{vmatrix} S\cdot\cos(\theta) & -S\cdot\sin(\theta) & T_x \\ S\cdot\sin(\theta) & S\cdot\cos(\theta) & T_y \\ 0 & 0 & 1 \end{vmatrix} \tag{2}$$

where S is the scale, θ is the rotation and T_X and T_Y are the translations. This has only four free parameters compared to the full affine transform's six: one scale factor, one angle, and two translations. The parameters of affine transform in Equation (1) are estimated based on the method presented in [8][9], which utilizes the affine invariant property of the ratio of areas for affine transform parameter estimation.

3.3. Image compensation

To get a stable video, we need to compensate the current frame to obtain stable images, which is also the final step for video stabilization. Compensation of the images can not be calculated directly from the parameters calculated in Equation (1), since undesired motion of the sensors and normal motion of the UAV should be separated ahead of time. Normal motion of the UAV is apparently different from the undesired motion of the sensors; the former is slow and follows certain rules, while the latter is fast and random yet unpredictable. In view of different characters mentioned above, in this step, traditional methods usually adopt a low-pass filter to smooth motion parameters to maintain the low-frequency motion and reject the high-frequency motion. However, this method always leads to an undesired effect, i.e., the stabilized images are several intervals falling behind the real video.

Given a set of video frames F_i, i=0,1,2…, we can now use the above procedure to estimate the distortion between all frames F_i and F_{i+1}, as affine transforms, H_i. Thus the cumulative distortion of a frame i relative to the first frame will be the product of all the preceding inter-frame transforms, or

$$A_i^{cumulative} = \Pi_{j=0}^{i-1} A_i \tag{3}$$

To smooth the video, at each step we calculate the transform A between the present frames. Then we combine this cumulative transform, $A^{cumulative}$, which describes all camera motion after the first frame. We evaluate this transform's scale, rotation, and translation parameters and store them for each time step. To get a smoothed frame, these parameters need to be smoothed and reconstituted into a full transform, $A_i^{cumulative,smoothed}$. The method for smoothing these parameters is presented in next section. To smooth the video we then warp the present frame by

$$Frame_i^{smoothed} = A_i^{cumulative,smoothed} \cdot (A^{cumulative})^{-1} \cdot Frame_i \qquad (4)$$

3.4. Parameter smoothing

There are many ways to smooth a cumulative sequence of image transforms, including numerical optimization [10] and Kalman filtering [11]. The traditional methods usually adopt a low-pass filter to smooth motion parameters to maintain the low-frequency motion and reject the high-frequency motion, such as using a Gaussian kernel to conduct a convolution and a smoothed cumulative transform parameters. However, this method always leads to an undesired result that the stabilized images are several intervals (depend on window width for smoothing) falling behind the real video. Here a cubic spline [12][13] based smoothing approach is used for the simpler approach of convolving the time sequence of the cumulative transform parameters. This has the effect of removing high-frequency noise (or camera jitter) while passing any cumulative effect, such as from a continuous pan or from motion tracking. The cubic smoothing spline $g(t)$ is generated to minimize

$$\mu \sum_{j=0}^{n-1} w(j)\left|P_j - g(j)\right|^2 + (1-\mu)\int \lambda(t)\left|D^2 g(t)\right|^2 dt \qquad (5)$$

where μ is the smoothing parameter, w is the weight for error measure, λ is for the piecewise constant weight function in the roughness measure. In our application, μ, w and λ are set to 0.2, 1 and 1, respectively. Four parameters in Equation (1) are passed to Equation (5) for smoothing respectively.

4. Experimental Results

This section presents our experimental results when testing our method with video captured from a RC UAV that transmits 2.4 Ghz wireless video from a micro–camera for field experiments. All tests were run on a DELL T5500 Workstation with an Intel Xeon 2.26 GHz processor and 6 GB of RAM. Images were processed with a resolution of 320 × 240 pixels. The programming is using C++ and Matlab. Some basic image processing modules are from OpenCV.

The algorithm of keypoint detection and matching is firstly tested in the experiment, followed by the test of effectiveness of proposed video stabilization algorithm on the real videos captured from UAV.

4.1. Keypoint detection and matching

The source code of FAST corner detector is downloaded from [14]. In FAST detector, we use 12-point detector [7] and set the threshold to 60, the maximum number of corner is limited to 300. The corner detection results are shown in the top row in Figure 2, two images are the original consecutive frames in video. The matching results of keypoints are shown in the bottom row. It can be observed that the algorithm works well in locating the corner points and matching.

4.2. Motion estimation and video stabilization

The proposed algorithm has been applied to real video sequences. Our RC UAV is controlled remotely and transmits wireless video to a receiver in ground. Tests are run on a number of video sequences and each is closed to 10 minutes (19740 frames) with diverse content. Aerial footage included a car, buildings and people moving around a big open yard, as shown in Figure 3.

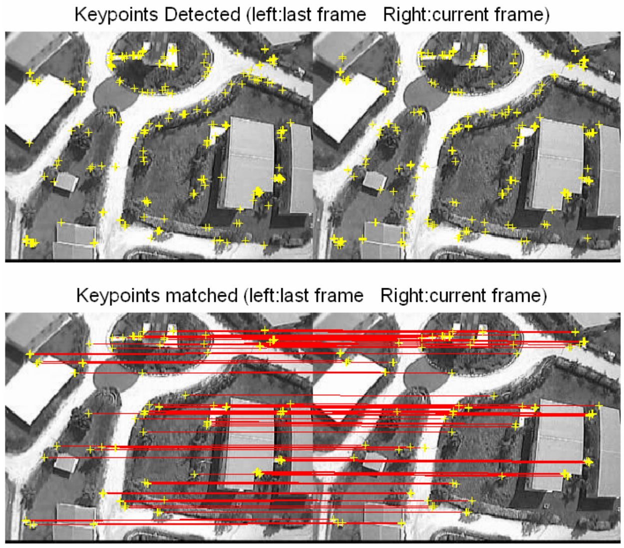

Figure 2. Keypoint detection and matching on two consecutive frames.

Figure 3. Videos for real-time video stabilization evaluation.

The examples of four smoothed parameters in $A_i^{cumulative,smoothed}$ are shown in Figure 4, with the smoothed parameters using Gaussian smooth algorithm. The window for the smoothing is set to 10. It is not surprised that there is some delay in the smooth process in Gaussian smooth algorithm.

The more straightforward way to observe the stable capability is through mean images generated from a number of consecutive frames. Figure 5 shows two examples, left columns are two means of 10-frames from original video, while right columns are from stabilized video. It is clear showing that mean images of original video are more blur than that of stabilized video. This means the stabilization algorithm is working fine. The speed can reach up to 30fps. An example of video stabilization of our method can be accessed at our server: http://www1.i2r.a-star.edu.sg/~ywang/demo/video_stabilization.avi.

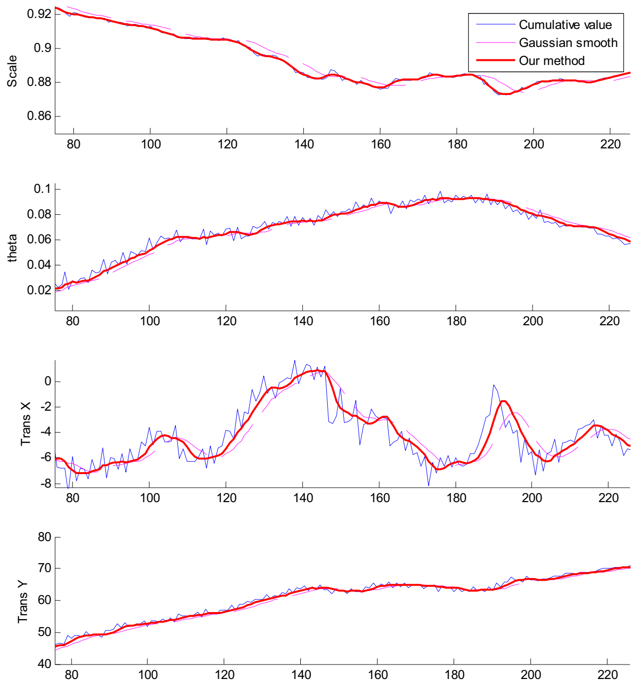

Figure 4. Smoothed parameters

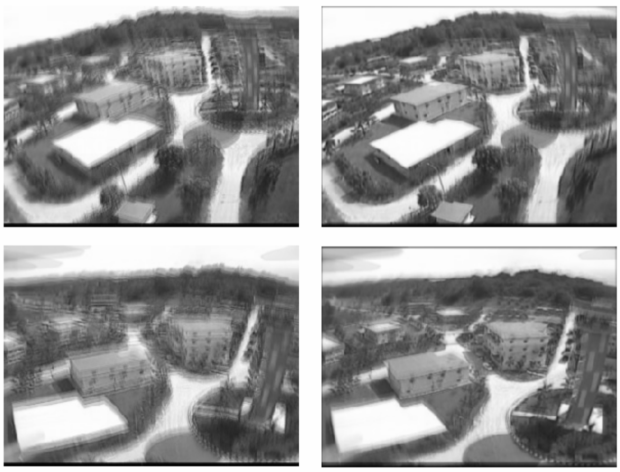

Figure 5. Mean images of original video (left) and stabilized video (right).

5. Conclusion

A video stabilization algorithm has been presented in this paper, which consists of three steps. Firstly, the keyponits are detected and matched. Then, the motion between two consecutive frames is estimated. And finally, the cumulative distortion of a frame relative to the first frame is generated and the parameters are smoothed by cubic spline smoothing. The proposed method was tested for video stabilization in the real video captured by camera installed on UAV, the experimental results show the efficiency and accuracy of the proposed algorithm.

It is worth noting that the accuracy of estimated transform between two consecutive images is highly dependent on the accuracy of keypoints detected. In the case of noise or blur image due to camera quality and transmit distortion, the algorithm for video stabilization may not be robust. Our future work will focus on the following aspects to improve our method: (1) currently the keypoint matching is based on grey image. Color information can be involved for a robust multi-channel region matching strategy. This will help to increase the accuracy of matching and thus the affine transform estimation; (2) more local and global features, such as object contour and geometrical relationship, can be applied to trade off noise and significant image distortion. A different descriptor for feature point has to be constructed for this purpose. However, for above mentioned improvement, we have to balance between the processing speed and algorithm complexity and robustness.

References

[1] Y. Shen, P. Guturu, T. Damarla, B. P. Buckles and K. R. Namuduri, “Video Stabilization Using Principal Component Analysis and Scale Invariant Feature Transform in Particle Filter Framework,” *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 3, pp. 1714-1721, 2009.

[2] M. Vazquez and C. Chang, “Marynel Vazquez and Carolina Chang,” *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4019-4024, 2009.

[3] H. Shen, Q. Pan, Y. Cheng and Y. Yu, “Fast Video Stabilization Algorithm for UAV,” *ICIS*, pp. 542-546, 2009.

[4] R. Hu, R. Shi, I. Shen, W. Chen, “Video Stabilization Using Scale-Invariant Features,” *The 11th International Conference Information Visualization*, pp. 871-877, 2007.

[5] Y. Shen, P. Guturu, T. Damarla, B. Buckles and K. Namuduri, “Video Stabilization Using Principal Component Analysis and Scale Invariant Feature Transform in Particle Filter Framework,” *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 3, pp. 1714-1721, 2009.

[6] P. Pan, A. Minagawa, J. Sun, Y. Hotta and S. Naoi, “A Dual Pass Video Stabilization System Using Iterative Motion Estimation and Adaptive Motion Smoothing,” ICPR 2010, pp. 2298-2301, 2010.

[7] E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection,” *European Conference on Computer Vision 2006*, vol. 1, pp. 430-443, 2006.

[8] Y. Wang, Z. Hou and K. Leman, "Combination of Local and Global Feature for Near-Duplicate Detection," *The 17th Intl Conf on Multimedia Modeling*, pp. 328-338, Jan 2011.

[9] Y. Wang, Z. Hou and K. Leman, "Keypoint-Based Near-Duplicate Images Detection Using Affine Invariant Feature and Color Matching," *ICASSP 2011*, Prague, Czech Republic, May 2011.

[10] K. Lee, Y. Chuang, B. Chen, M. Ouhyoung, "Video Stabilization Using Robust Feature Trajectories," National Taiwan University, 2009.

[11] A. Litvin, J. Konrad and W. Karl, "Probabilistic Video Stabilization Using Kalman Filtering and Mosaicking," *IS&T/SPIE Symposium on Electronic Imaging, Image and Video Communications and Proc.*, 2003.

[12] Y. Wang, E.K. Teoh, Z. Hou and J. Wang, "Object Boundary Extraction Using Active B-Snake Model," *ICPR 2008*, pp. 1199-1202, 2008.

[13] Y. Wang, E.K. Teoh and D. Shen, "A B-Snake Model Using Statistical and Geometric Information - Application to Medical Images," *ICARCV 2002*, pp.793-797, 2002.

[14] http://mi.eng.cam.ac.uk/~er258/work/fast.html