Improving video stabilization in the presence of motion blur

Manish Okade
Department of Electronics and
Electrical Communication Engineering
Indian Institute of Technology (IIT)
Kharagpur, India 721302
okade.manish@ece.iitkgp.ernet.in

P. K. Biswas
Department of Electronics and
Electrical Communication Engineering
Indian Institute of Technology (IIT)
Kharagpur, India 721302
pkb@ece.iitkgp.ernet.in

***Abstract*—In this paper we propose the idea of deblurring for efficient feature matching in the context of video stabilization. This is achieved by incorporating a motion deblurring block prior to the feature extraction and matching stage in the stabilization pipeline. The effect of motion blur on feature extraction and matching has not been investigated so far as most of the approaches have looked into deblurring as a post processing step. After preprocessing the blurred frames, the scale invariant feature transform points are used to find correspondence to get an estimate of the camera motion. Smoothing of the motion parameters to retain the desired motion is performed using a gaussian filter. Finally, inverse of the resulting image transform is carried out to obtain a stable video sequence. We compare our method to existing approaches and show how the inserted block improves the video stabilization performance. Interframe Transformation Fidelity (ITF) is used to show the superiority of our proposed approach.**

***Index Terms*—Video stabilization, motion blur, SIFT, feature matching, deblurring.**

I. Introduction

Hand held video camcorders are now gaining popularity due to the decreasing cost of these devices. Sudden hand jerks and platform vibrations are a common problem in these devices. Motion estimation, motion filtering and motion compensation are three essential steps required to perform the stabilization task. Motion estimation is a very important step in stabilization and computationally, the most expensive. It can be performed either using intensity based or feature based techniques. Intensity based methods include gray coded bit-plane matching [1], optical flow [2] and sub-image phase correlation methods [3] while feature based methods include corner matching [4], edge pattern matching [5] and SIFT point matching [6]. Once global interframe motion is estimated then motion filtering is performed using techniques like motion vector integration [1], gaussian filtering [7], kalman filtering [8] followed by motion compensation to construct the stabilized frame. Video stabilization in the presence of motion blur has not been investigated so far but an interesting work of visual tracking in the presence of motion blur was investigated by Favaro et al. [9].

In this paper we use feature based technique to perform motion estimation. We propose the idea of accurate feature extraction by first incorporating a motion deblurring block prior to the feature extraction stage. Our idea is based on the assumption that any hand-held camera experiencing severe jitter motion is subjected to motion blur. If this is not processed then feature matching between successive frames will be inaccurate which will hamper the global motion estimation. Incorrect estimation of the global motion parameters will result in inaccurate alignment between successive frames which defeats our fundamental goal of stabilization. We use the Scale Invariant Feature Transform (SIFT) [10] points as features for finding correspondence. SIFT features have been used previously for video stabilization purposes but the motion blur resulting in the image acquisition process has not been analyzed and pre-treated. Significant gain in Interframe Transformation Fidelity (ITF) [11] is achieved by our proposed approach as shown by the experimental results.

II. Camera Model

We use the 2-D affine motion model for estimating the camera motion given by

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

$$p' \sim Tp \tag{2}$$

where, (x, y) is the point p in reference frame and (x', y') is the point p' after the frame has undergone affine motion represented by the transformation matrix T. Parameters $\{m_1, m_2, m_4, m_5\}$ control rotation, scaling and sheering effects of the transform while $\{m_3, m_6\}$ correspond to translation parameters in the horizontal and vertical directions, respectively. Our task in camera motion estimation is to determine the six parameters $\{m_1, m_2, m_3, m_4, m_5, m_6\}$ for every successive frame.

III. Video Stabilization System Framework

A. Overview

Fig. 1 shows the system framework of our proposed approach. The input to our system is a video captured by a hand-held video camera which we assume consists of annoying

978-0-7695-4599-8/11 $26.00 © 2011 IEEE
DOI 10.1109/NCVPRIPG.2011.25

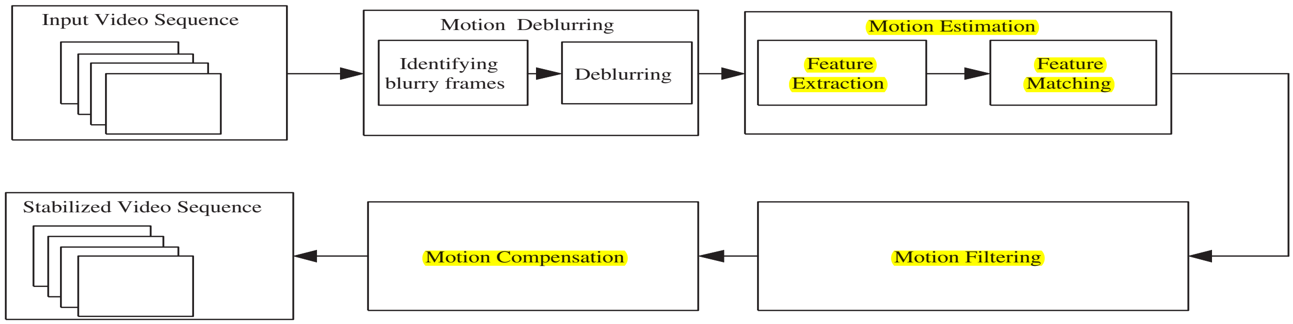


Fig. 1: System framework

shaky hand motions. We propose to perform motion deblurring prior to feature extraction and matching stage, to account for the motion blur which might be present in the input frames. After this preprocessing step we extract the SIFT points and match them to get an estimate of the correspondence between successive frames. This estimate gives the global motion that has taken place between successive frames. Next, we use a gaussian smoothing filter to retain the desired motions. Finally, motion compensation is performed by image warping operation.

B. Motion Deblurring

SIFT has been designed for extracting highly distinctive invariant features from images. These features are robust to changes in scale, orientation and illumination but their performance falls down in case of blurred scenes [6]. Fig. 2 shows two consecutive frames (sequence LAB) affected by motion blur and how they result in incorrect correspondences of the feature points. Our proposed method first performs motion deblurring, prior to feature extraction and matching stage to account for the motion blur as described below.

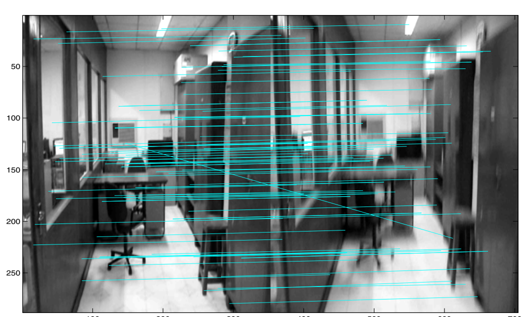


Fig. 2: In-correct correspondence due to motion blur (sequence LAB)

1) Identifying Blurry frames: We note that blurring occurs only under certain conditions like severe jitter, fast object motion or long exposure time of the camera. Hence not all frames are affected by this degradation and there is a need to identify the blurry frames. We propose a method as described by the flowchart in Fig. 3 to recognize the blurry frames in the video sequence. We firstly evaluate the relative blurriness of

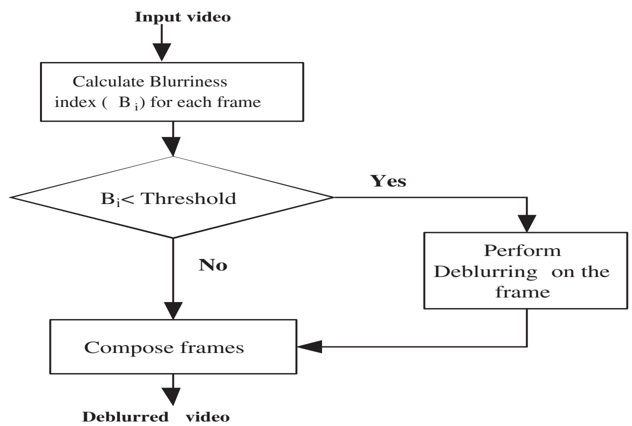


Fig. 3: Flowchart for identifying and processing blurry frames

each frame by calculating the gradient of the frame. Generally, the gradient of blurry image is smaller than that of sharper one at the same regions. With this assumption the blurriness of frame i is defined as

$$B_i = \sum_{p_i} (g_x^2(p_i) + g_y^2(p_i)) \tag{3}$$

where, B_i is the blurriness index, p_i is the pixel on frame i, g_x and g_y are the gradients of x- and y- directions, respectively. Fig. 4 shows the plot of blurriness index B_i versus frame number for the LAB sequence. The blurriness index is calculated for each frame of the input video sequence and compared with a threshold ($threshold = 23$ for LAB sequence). Those frames whose blurriness index B_i fall below the threshold are treated by the deblurring block. Other frames are sharper as they exceed the threshold and are hence not de-blurred. Thus, we selectively deblur only those frames which have a significant amount of blur.

2) Deblurring: The deblurring is challenging because the blur kernel is not known in advance. Deblurring is performed on the selected frame using the approach adopted by Fergus et al. [12] which consists of two steps. First, the blur kernel is estimated from the input image and a known image prior in a coarse-to-fine fashion to avoid local minima. Next, using the estimated kernel, a standard deconvolution algorithm is applied to estimate the unblurred image.

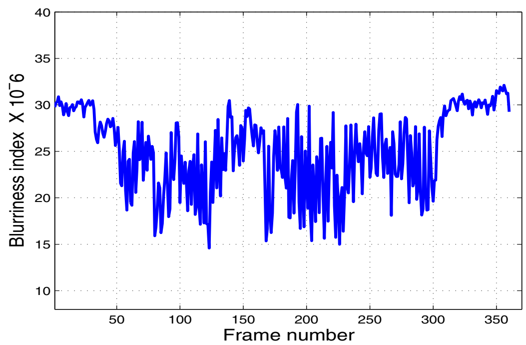


Fig. 4: Plot of Blurriness index versus frame number (sequence LAB)

C. Feature Extraction and Matching

SIFT keypoints are used as features for estimating the global interframe motion vector. SIFT keypoints are extracted from two successive frames and then the two sets of feature points are matched to obtain a local motion vector. Mathematically speaking since absolute positions $p = (x_k, y_k, 1)$ and $p' = ({x_k}', {y_k}', 1)$ of both first and second keypoint in both frames are known, the local motion vector v_k can be estimated and represents how the feature has supposedly moved from the previous frame to the next one. The whole set of feature local motion vectors do not contain useful information for effective motion compensation as it includes matches relating to moving objects in the scene. We eliminate moving objects by assuming that the velocity of moving objects is very large as compared to other motions by using a fixed threshold. Thus the transformation matrix T as described in Section II is obtained.

D. Motion Filtering (Smoothing)

Desired camera motions (like panning, zooming) in a video sequence must be retained and not compensated. They are assumed to be smooth with slow inter-frame variations while unwanted camera motion are assumed to be random. Given the original transformation chain T_t, from frame 1 to frame t in a video sequence, we obtain the accumulated smoothed transformation denoted by S_t as

$$S_t = \sum_{i \in N_t} T_t^i \star G(k_t) \tag{4}$$

where $N_t = \{j \mid t - k_t \leq j \leq t + k_t\}$, denotes the temporal neighborhood of frame t and $G(k_t) = \frac{1}{\sqrt{2\Pi}\sigma} \exp(\frac{-k_t^2}{2\sigma^2})$ is a gaussian filter with $\sigma = \sqrt{k_t}$. The window size k_t varies over time and enables the smoothing operation to be adaptive.

E. Motion Compensation

Using the smoothed matrices $S_0, \ldots, S_t$, the original video frame I_t is warped to the motion compensated video frame $I_t^{'}$ by $I_t^{'} \leftarrow I_t(S_t)$.

IV. Results

We have used MATLAB 7 on a 3.2 GHz Pentium Dual Core CPU for our experimentation. We have used the gaussian kernel k_t to be 6 in our experiments. We use both subjective and objective evaluation to substantiate our proposed approach. The sequence LAB (360 frames @30 fps, CIF resolution) is captured by us in an indoor environment. Fig. 5(a) shows three frames of the unstable input sequence corresponding to frame numbers 79, 80 and 81. Fig. 5(b) shows the stabilization results without the deblurring block. Fig. 5(c) shows the stabilization results using our proposed approach which clearly shows the improvement obtained by adding the deblurring block in the stabilization pipeline.

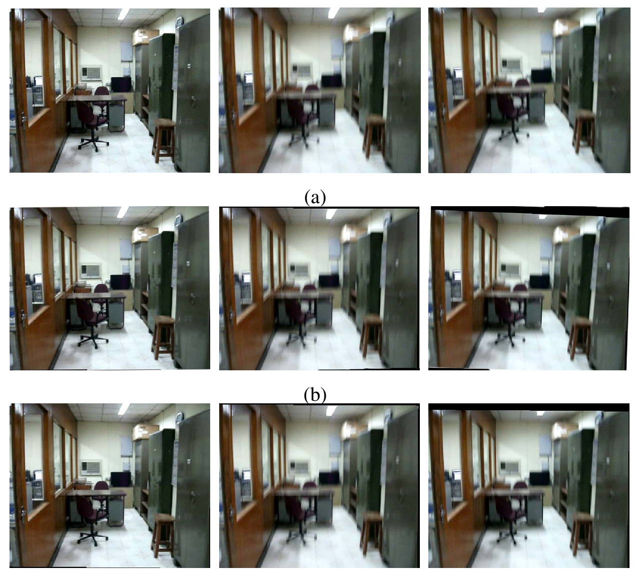
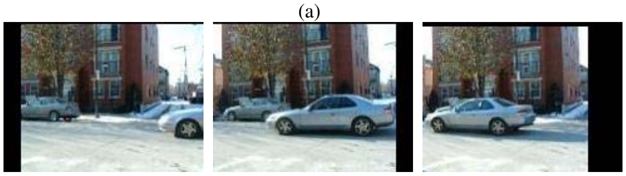
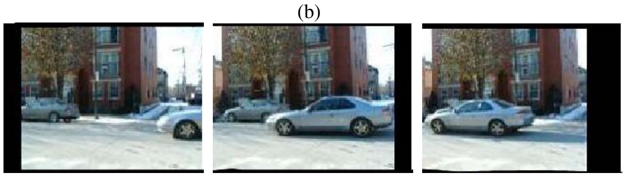
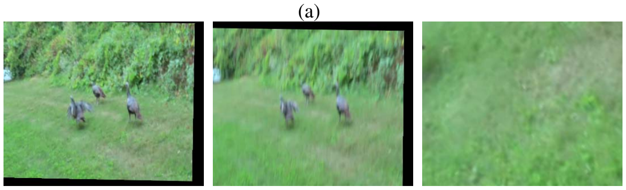
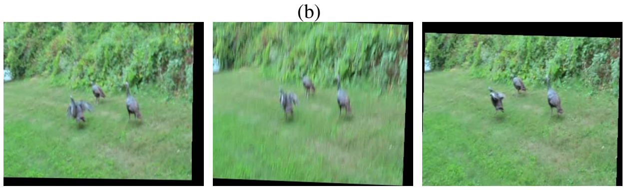

(a) (b) (c)

Fig. 5

We now compare our method with Particle Filtering based Motion Estimation (PFME) using SIFT feature points [6]. The videos were available at http://www.ece.uic.edu/~yjunlan/index.html. The sequence STREET consists of a scene with a car moving across the camera and building in the background. As reported in their paper, this scene is challenging because of moving objects and image blurring due to fast camera vibrations. Fig. 6(a) shows three frames of the unstable input sequence. Fig. 6(b) shows their stabilization results which have been obtained by generating particles and performing importance sampling to counter the blurring effect. Fig. 6(c) shows the stabilization results using our proposed approach. We observe that our results come very close to their approach when compared subjectively. We also compare our method with video sequence from Liu et al. [13] available at http://web.cecs.pdx.edu/~fliu/project/subspace_stabilization/. Fig. 7(a) shows input frames 300, 301 and 302 of the sequence 00016.avi. Due to severe amount of motion blur present in frame number 301 of the sequence, incorrect transformation matrix parameters are obtained thereby degrading the stabilization as shown in Fig. 7(b). The stabilization results using our proposed


(a)

(b)

(c)
Fig. 6


(a)

(b)

(c)
Fig. 7

approach is shown in Fig. 7(c) which shows that minimizing the blur in the input frame is able to achieve better stabilization performance.

For objective evaluation we have used the Interframe Transformation Fidelity (ITF) which measures the PSNR between successive frames given by

$$ITF = \frac{1}{N_{frame} - 1} \sum_{k=1}^{N_{frame}-1} PSNR(I_k, I_{k+1}) \qquad (5)$$

where, N_{frame} is the total number of video frames in the video sequence and $PSNR(I_k, I_{k+1})$ is the Peak Signal to Noise Ratio between two successive frames. As shown in Table I our proposed approach is able to achieve better ITF

TABLE I: ITF comparison

Sequence	Original ITF(dB)	Stabilized ITF(dB)	Stabilized ITF(dB) (proposed approach)
LAB.avi	25.23	27.31	28.32
STREET.avi	18.43	18.89	19.25
00016.avi	22.29	21.58	23.31

values as compared to PFME and subspace video stabilization methods.

V. Conclusion

In this paper we have investigated a practical problem in video stabilization applications. Deblurring for efficient SIFT matching is what we have explored. By our experiments we have shown how pre-processing the blur helps in increasing the feature matching accuracy thereby improving the video stabilization performance. Our approach is not computationally expensive as treatment of blur is done only on those frames which show significant amount of motion blur. Our future work will concentrate on performing deblurring using cues from multiple frames for estimating the blur kernel in place of using a single image for the blur kernel estimation. Full frame video stabilization using inpainting is also presently being investigated.

References

[1] S. Ko, S. Lee, S. Jeon, and E. Kang, "Fast digital image stabilizer based on gray-coded bit-plane matching," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 598–603, Aug. 1999.

[2] J. Chang, W. Hu, M. Cheng, and B. Chang, "Digital image translation and rotation motion stabilization using optical flow technique," *IEEE Trans. Consumer Electron.*, vol. 48, no. 1, pp. 108–115, Feb. 2002.

[3] S. Erturk, "Digital image stabilization with sub-image phase correlation based global motion estimation," *IEEE Trans. Consumer Electron.*, vol. 49, no. 4, pp. 1320–1325, Nov. 2003.

[4] A. Censi, A. Fusiello, and V. Roberto, "Image stabilization by feature tracking," in *Proc. Int. Conf. Anal. Process*, vol. 2, Venice, Italy, 1999, pp. 665–667.

[5] J. Pail, Y. Park, and D. Kim, "An adaptive motion decision system for digital image stabilizer based on edge pattern matching," *IEEE Trans. Consumer Electron.*, vol. 38, no. 3, pp. 607–615, Aug. 1992.

[6] J. Yang, D. Schonfeld, and M. Mohamed, "Robust video stabilization based on particle filter tracking of projected camera motion," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 19, no. 7, pp. 945–954, July 2009.

[7] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150–1163, 2006.

[8] A. Litvin, J. Konrad, and W. Karl, "Probabilistic video stabilization using kalman filtering and mosaicking," in *Proc. Image Video Commun. IST/SPIE Symp. Electron. Imaging*, Santa Clara, CA, 2003, pp. 663–674.

[9] P. Favaro, H. Jin, and R. Cipolla, "Visual tracking in the presence of motion blur," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 18–25.

[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] S. Battiato, G. Puglisi, and A. Bruna, "A robust video stabilization system by adaptive motion vectors filtering," in *ICME*, 2008, pp. 373–376.

[12] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics, SIGGRAPH 2006 Conference Proceedings, Boston, MA*, vol. 25, pp. 787–794, 2006.

[13] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Transcations on Graphics*, vol. 30, no. 1, pp. 1–10, 2011.