Feature-Based Full-Frame Image Stabilization

Chih-Yuan Chung and Homer H. Chen
National Taiwan University
alvinsay@gmail.com, homer@cc.ee.ntu.edu.tw

Abstract

Digital image stabilization usually discards boundary pixels and outputs a smaller video. In this paper, we present a new digital image stabilization algorithm that preserves the frame size of output video by pixel filling. The proposed algorithm eliminates the accumulation error by directly estimating the global motions in a transformation chain with reference to a fixed frame. A feature matching method is adopted to save the computational cost of the global motion estimation and to handle large motions. The experimental results show that the proposed algorithm produces stabilized full-frame video sequences with better frame alignment.

1. Introduction

One of the important features of digital video camera is image stabilization that removes the undesired camera motion caused by hand shaking. Image stabilization not only makes video sequences more pleasing to human eyes but also is useful for many applications such as moving object detection, high dynamic range video, and segmentation of foreground objects.

Digital image stabilizer, which is more cost-effective than the optical stabilizer, is accomplished by moving the video frames to compensate for the hand shaking. In this process, however, undefined pixels are resulted when the image is moved out of the video frame. To solve the problem, one popular approach crops the image to keep the undefined pixels outside the viewing region of the image. By doing so, the output image size is reduced, and a good part of the original image data is wasted. Another approach constructs the output video frame in a mosaic fashion from the current and neighboring frames to preserve frame size of the output video. In this paper, we proposed a full-frame stabilization algorithm based on the mosaicing approach.

1.1. Related Work

The image stabilizer consists of three steps: motion estimation, motion compensation, and image warping. First, the global motions of adjacent frames are estimated and cascaded to obtain a transformation chain. In the second step, a transformation is computed for each frame to compensate for the hand shaking. Finally, video frames are warped according to the transformations.

Approaches to global motion estimation (GME) can be divided into two categories: direct and indirect. Direct approaches [1] are computationally much more expensive than indirect approaches because they involve the minimization of prediction errors in the pixel domain. Indirect approaches first use block-based [2] or feature-based [3], [4], [5] methods to estimate local motion vectors and then obtain the global motion by minimizing the sum of prediction errors of local motion vectors. The previous proposed stabilization algorithms [6], [7] use the direct approaches, and [8], [9] use the indirect approaches.

After image warping, the undefined pixels must be filled to preserve the image size. The digital inpainting technique described in [10], [11] recovers the undefined pixels by solving partial differential equations numerically. However, it has high complexity and inconsistent performance between frames. Some approaches [12], [13] fill the undefined pixels by sampling spatio-temporal volume patches. Such approaches require a long video sequence to increase the success rate. Matsushita et al. [7] proposed motion inpainting that warps neighboring frames and propagates optical flow to stitch multiple images. Although the optical flow may compensate the GME failure in some cases, the inaccuracy of the optical flow algorithm may still cause serious artifacts.

1.2. Proposed Approach

Accurate global motion estimation is required for digital image stabilization systems to generate full-frame stabilized videos. Therefore, the proposed approach aims at improving the accuracy of GME and the performance of image stitching. In previous work, the global motion between non-adjacent frames is computed by cascading the transformations of adjacent frames. The concatenation of transformations would cause accumulation errors, which lead to image misalignment and lower the quality of pixel filling undefined pixels. To handle this problem, the proposed al-

0-7695-3058-3/07 $25.00 © 2007 IEEE
DOI 10.1109/ISM.2007.44

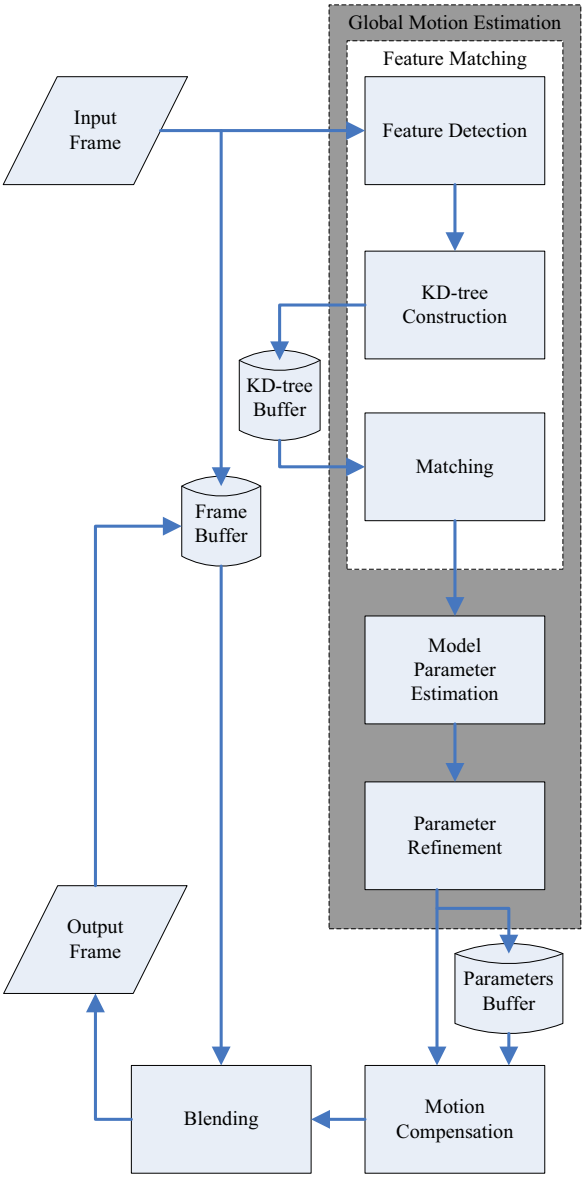

Figure 1. Overview of the proposed algorithm.

gorithm estimates the global motion of a frame with respect to a number of post frames.

Recently, a robust and accurate feature matching technique [4], [5] has been developed and widely used in many applications. We adopt this technique in our proposed image stabilizer for the following two reasons. First it can reuse the extracted image information to reduce the computational cost of local motion estimation for multiple target frames. Second, it works well when the overlapped region

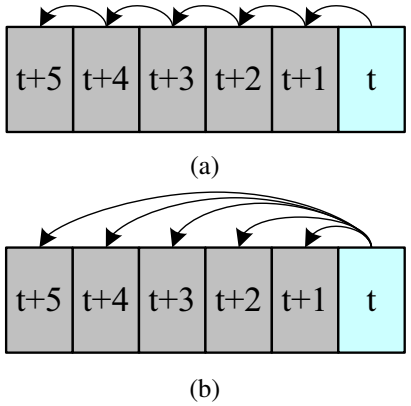

Figure 2. (a)GME is performed only for adjacent frames in the traditional image stabilizer. (b) The proposed algorithm extracts feature descriptors and constructs KD-trees for each frame and then estimates the global motions for different target images.

of the reference and target images is small.

After motion compensation, the proposed algorithm uses transition band blending to stitch frames and stabilize full-frame video sequences. Here, we propose a backward composition scheme to reduce the number of undefined pixels after image composition.

2. Overview of the Proposed Algorithm

The block diagram of our algorithm is shown in Fig. 1. Feature detection is applied to the current input frame, and then the feature descriptor (a vector) for each feature point is constructed. The KD-trees of the descriptors are built for efficiently matching feature points between frames. For each matched feature point, a local motion vector is computed. Then the resulting local motion vectors are input to the RANSAC algorithm to estimate the global motion. In the parameter refinement step, the global motion is further refined by using the Newton-Raphson method. Once the global motion is determined, a global transformation is computed to compensate the undesired hand shaking. Finally, image blending is performed to generate the output frame.

3. Global Motion Estimation

Fig. 2 illustrates our motion estimation scheme in which the global motions between the current frame I_t and every neighboring frame, instead of only the global motions of adjacent frames, are computed. The set of indices of neighboring frames is defined as

$$N_t = \{s | s \neq t, |s - t| < R\},$$

where R is the range of the neighborhood.

To describe the global motion, we employ the perspective transformation that models an image as a 2D planes in 3D space. The perspective transformation can be represented by

$$T = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{pmatrix}.$$

Let (x, y) and (x', y'), respectively, be the coordinates of a point in the current frame and the target frame. We have

$$x' = \frac{m_{11}x + m_{12}y + m_{13}}{m_{31}x + m_{32}y + 1}, \tag{1a}$$

$$y' = \frac{m_{21}x + m_{22}y + m_{23}}{m_{31}x + m_{32}y + 1}. \tag{1b}$$

The advantage of the perspective model is that it can describe the global motion more accurately than geometric and affine model since it has more degrees of freedom.

3.1. Feature Matching and Local Motion Vectors

We use local motion vectors to estimate the global motions of neighboring frames. To find local motion vectors between frames, we use a two-stage feature matching algorithm [5]. In the first stage, it scans the current image and the associated target image to detect feature points and generates feature descriptors. In the second stage, the matching algorithm finds the corresponding feature point in the target image for each feature point in the current image. The local motion vectors are computed from the pairs of corresponding feature points. Note that once feature descriptors are extracted from the current image, they can be repeatedly utilized to find the motion vectors for different target images. In comparison to the block-based methods, which require performing whole GME again when the current or target image is changed, the computational cost is saved in the proposed approach.

In the proposed algorithm, we adopt the SIFT [5] algorithm to detect feature points and generate feature descriptors. The features detected by these algorithms are highly invariant to illumination changes and image transformations. Thus, the proposed algorithm is stable even for frames with large motion.

For each feature point extracted from the current frame, the corresponding feature point can be found by searching the nearest neighbor in the feature descriptor space. A KD-tree [14], [4] data structure is used to index feature descriptors for each frame in video sequence to accelerate searching speed. With the help of the KD-tree, the cost of searching target can be significantly reduced to a small fraction of the exhaustive search.

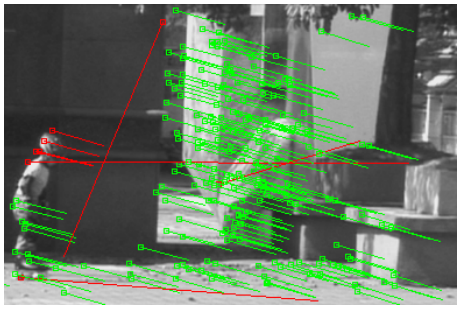

Figure 3. The result of RANSAC. Squares and lines, respectively, represent the feature points and the local motion vectors. Inliers are marked in green, and outliers in red.

3.2. Model Parameters Estimation

In [2], the Newton-Raphson method is used to find global motion parameters by minimizing the cost function

$$E = \sum_p E_p,$$

$$E_p = (x'_p - x_p)^2 + (y'_p - y_p)^2,$$

where p is the index of the local motion vectors, (x'_p, y'_p) is the coordinates of the feature point in the target frame , and (x_p, y_p) is the coordinates of the corresponding point computed from the estimated GME parameters. However, few motion vectors resulted from false matching would fail the initial guess step of the Newton-Raphson method. We use RANSAC (Random Sample Consensus) algorithm to initialize the parameters of the global motion model. It repeatedly chooses enough samples to solve the linear equations derived from the global motion model. Then the result that has minimum outliers is used as the initial value of the parameters. For the perspective global motion model, 4 independent motion vectors are randomly chosen in each run. Each motion vector is substituted into the following equations deduced from equations (1a) and (1b)

$$x' = m_{11}x + m_{12}y + m_{13} - m_{31}x'x - m_{32}x'y, \tag{2a}$$

$$y' = m_{21}x + m_{22}y + m_{23} - m_{31}y'x - m_{32}y'y. \tag{2b}$$

Given enough motion vectors, m_{ij} can be estimated by singular value decomposition (SVD). Fig. 3 shows the result of RANSAC.

The estimated parameters are further refined using a nonlinear approach. We mark the points which satisfy $E_p \leq E_{th}$ as inliers. E_{th} is the threshold of the deviations of local motion vectors. In the proposed algorithm, we set E_{th} to 9. To make all inliers contribute to the final GME parameters, the Newton-Raphson method is then applied to the inliers.

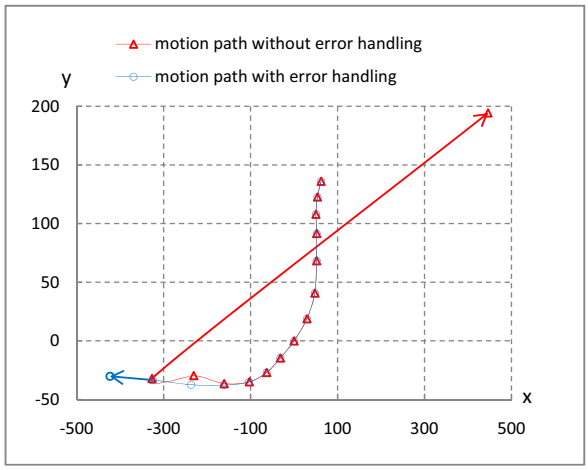

Figure 4. The motion paths estimated with and without error handling. The x- and y-axes, respectively, represent the horizontal and vertical translations.

3.3. Error Detection and Handling

Motion blur in the video may suppress the number of feature points generated by the feature detection algorithm. A camera with acceptable shutter speed can generate videos without extreme blur. However, for low-end cameras or when shooting in dark environments, shutter speed becomes slow. In such cases, the feature matching algorithm would fail due to low feature repeatability.

Two heuristic methods are utilized to detect errors of the feature matching algorithm. One uses the ratio of the number of inliers to the number of all motion vectors to measure the quality of GME. If the ratio is lower than 80%, we consider the global motion to be corrupted.

The other method is based on the assumption that the global motions between adjacent frames are relatively moderate. In the motion compensation stage, the translational components of $T_{t,s}$ are compared with the ones of $T_{t,s-1}$. If any dimension of transitional components is larger than a threshold, the global motion $T_{t,s}$ is considered to be corrupted as well. The threshold is defined as

$$X_{TH} = R \times max(W, H),$$

where W and H is the width and height of the image and R is a constant number that affects the performance of the proposed algorithm. If R is too large, some GME errors may not be detected. If R is too small, the proposed algorithm may suffer from the accumulation error due to the correction of the global motions. In the proposed algorithm, R is empirically set to 0.05.

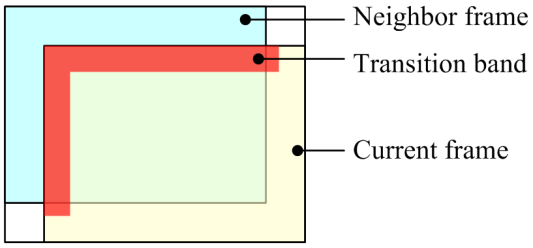

Figure 5. A transition band in the boundary between current frame and transformed neighboring frame is introduced to reduce the artifacts caused by image composition.

The corrupted global motion parameters can be corrected by utilizing other transformations that are related to the current frame and the target frame. For example, the corrupted transform $T_{t,s}$ can be replaced by $T_{t,k}T_{k,s}$ where k can be searched from the range $R = \{x|t < x < s\}$. Fig. 4 shows the motion paths estimated with and without the error handling. It should be noted that without the error handling, there is a sudden change of direction and magnitude at the end of the motion path. With the error handling, there is no such problem.

4. Motion Compensation

Each frame I_t is applied an transformation S_t to compensate for the global motion in the motion compensation stage. S_t can be obtained by the following equation:

$$S_t = P^{-1}(\frac{\sum_{i \in N_t} w(i-t)P(T_{i,t})}{\sum_{i \in N_t} w(i-t)}).$$

$T_{i,j}$ denotes 3×3 homogeneous matrix representing the global motion from the frame I_i to the frame I_j. $P(x)$ is an operator that extracts translation components and rotation angles from the transformation x and formulate them as a vector. $P^{-1}(y)$ recovers the corresponding transformation matrix from the vector y. The weighting function $w(d)$ is defined by the following equation:

$$w(d) = \exp(-\frac{d^2}{\sigma^2}).$$

Making σ range from 3 to 6 results in a good performance.

Note that it is not necessary to compute the global motion $T_{s,t}$ where $s < t$ since we can get it by just inversing $T_{t,s}$, which has already been computed.

5. Frame Composition

After video stabilization, the pixels near the frame boundary would be lost due to motion compensation. We

define the region where pixels are not lost as Ω. Each neighboring frame $I_k, k \in N_t$ is blended through the transformation $T_{k,t}S_t$ into the current frame. As shown in Fig. 5 , we define the transition band of width w along the boundary of the overlapped area. The blending weights decrease from inner boundary to outer boundary in the transition band. The blending weight map is defined as follow:

$$W(p) = \begin{cases} g(d(p)), & \text{if } p \in \Omega \text{ and } d(p) < w \\ 0, & \text{if } p \notin \Omega \\ 1, & \text{otherwise} \end{cases},$$

where $d(p)$ is the shortest path from p to the boundary of Ω. Therefore, the blending equation is

$$I'_t(p) \leftarrow W(p)I'_t(p) + (1 - W(p))I^t_s,$$

where I^t_s is the frame transformed from I_s by $T_{s,t}S_t$.

After compositing the current image with the neighboring images, the undefined pixels may still exist. The number of undefined pixels would be reduced, if we increase the neighboring range to cover the entire sequence, but it is computational costly. We propose backward composition that utilizes the results of the previous frames. It composites the current frame with the previous stabilized full-frames within a certain range. Because the previous frames are also composited from their previous frames within certain range, an undefined pixel may acquire the value which actually belongs to the frame that appears long time ago. Fig. 6 compares the composition result without and with backward composition. It can be seen that the undefined pixels are fewer when backward composition is applied.

To deal with remaining undefined pixel, the proposed algorithm diffuses the color from the known pixels to the undefined pixels in the fast marching order [15]. Because the area of the undefined pixels is small in this step, and the fast marching diffusion fills the undefined pixels smoothly, the artifact is hardly sensed by the user.

6. Experimental Results

We tested the proposed algorithm on various video sequences. It was run on Pentium 4, 3.2GHz CPU, at 1 frame/sec by using the SIFT algorithm and 3 frames/sec by using the SURF [16] algorithm instead for CIF sequences.

Raw video sequences were obtained by a DV camera, undergoing translation, rotation, and zooming during the video shooting. Fig. 7 shows the result without and with error handling. The serious distortion due to blur (shown in the 2nd and 4th images in the top row) is corrected by the proposed error handling scheme. Figs. 8 and 9 show the stabilized frames before and after inpainting. We can see that most static objects can be inpainted correctly, and for the objects with slight local motion (the man in the left most image in Fig. 8), there is no noticeable distortion. Fig. 10 compares the inpainted full-frame image generated by our algorithm with that by the image stabilization system based on block-based motion estimation [2]. It can be seen that alignment errors appear in the block-based system, particularly for frames with large motion, but the result of the proposed system does not have such visible artifacts.

Figure 6. The stabilized images without composition (left), with composition (middle), and with backward composition (right).

7. Conclusion and future work

We have presented a robust image stabilization algorithm that preserves the video size. The proposed approach removes the accumulation error of the transformation chain by directly estimating the global motion between non-adjacent frames. We adopt the feature matching algorithm based-on SIFT and SURF to reduce the computational cost of global motion estimation. The algorithm performs well even for videos with large camera motion. Compared with the full-frame image stabilizer based on traditional GME, the proposed algorithm is more robust.

We only consider global motion in this paper and use a perspective model that assumes the whole scene as a plane, so non-planer object and moving object may not be inpainted correctly. Further refinement using local motion information is underway.

References

[1] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *ECCV*, 1992, pp. 237–252.

[2] Y. Su, M.-T. Sun, and V. Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," *IEEE Trans. CSVT*, vol. 15, pp. 232–242, 2005.

[3] D. Farin and P. H. N. de With, "Evaluation of a feature-based global-motion estimation system," in *VCIP*, July 2005, vol. 5960, pp. 1331–1342.

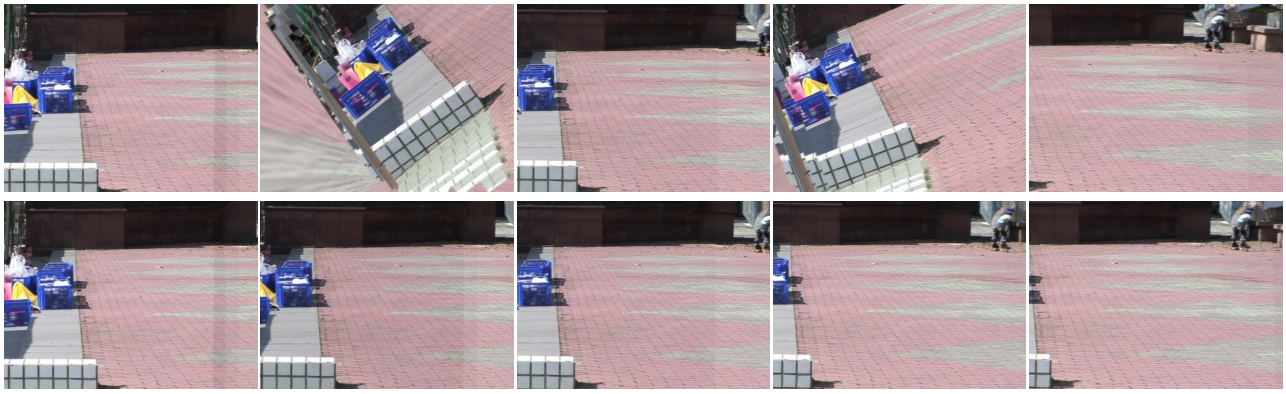

Figure 7. The result without error handling (top row) and with error handling (bottom row).

Figure 8. The stabilized frames (top row) and the inpainted frames (bottom row).

Figure 9. The stabilized frames (top row) and the inpainted frames (bottom row).

[4] M. Brown and D. G. Lowe, "Recognising panoramas," in *ICCV*, 2003, pp. 1218–1227.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[6] A. Litvin, J. Konrad, and W.C. Karl, "Probabilistic video stabilization using kalman filtering and mosaicking," in *IS&T/SPIE Symposium on Electronic Imaging*, 2003, pp. 663–674.

[7] Y. Matsushita, E. Ofek, X. Tang, and H.-Y. Shum, "Full-frame video stabilization," in *CVPR*, 2005, pp. 50–57.

[8] A. Censi, A. Fusiello, and V. Roberto, "Image stabilization by features tracking," in *ICIAP*, 1999, pp. 665–667.

[9] Y.-C. Peng, H.-A. Chang, C.-K. Liang, H. Chen, and C.-J. Kao, "Integration of image stabilizer with video codec for digital video cameras," in *ISCAS*, 2005, pp. 4871–4874.

(a) (b) (c) (d)

Figure 10. In the frame with large global motion, the block-based algorithm would be affected seriously, but the proposed algorithm remains stable. (a) Stabilized frame of block-based algorithm. (b) Full-frame of block-based algorithm. (c) Stabilized frame of the proposed system. (d) Full-frame of the proposed algorithm.

[10] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH*, 2000, pp. 417–424.

[11] C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Processing*, vol. 10, no. 8, pp. 1200–1211, 2001.

[12] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *CVPR*, 2004, pp. 120–127.

[13] K.A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *ICIP*, 2005, vol. 2, pp. 69–72.

[14] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *CVPR*, 1997, pp. 1000–1006.

[15] J. Sethian, "A fast marching level set method for monotonically advancing fronts," in *Nat. Acad. Sci.*, 1996, vol. 93, pp. 1591–1595.

[16] H. Bay, T. Tuytelaars, and L. J. Van Gool, "SURF: Speeded up robust features," in *ECCV*, 2006, pp. 404–417.