

# FULL-FRAME VIDEO STABILIZATION WITH A POLYLINE-FITTED CAMCORDER PATH

<sup>1</sup>Jong-Shan Lin (林蓉珊), <sup>1</sup>Wei-Ting Huang (黃惟婷)  
<sup>2</sup>Bing-Yu Chen (陳炳宇), <sup>3</sup>Ming Ouhyoung (歐陽明)

National Taiwan University

E-mail: <sup>1</sup>{maruko,weiting}@cmlab.csie.ntu.edu.tw, <sup>2</sup>robin@ntu.edu.tw, <sup>3</sup>ming@csie.ntu.edu.tw

## ABSTRACT

Annoying shaky motion is one of the significant problems in home videos, since hand shake is an unavoidable effect when capturing by using a hand-held camcorder. Video stabilization is an important technique to solve this problem. However, the stabilized videos resulted by current methods usually have decreased resolution and are still not so stable. In this paper, we propose a novel, robust, and practical method of video stabilization with a polyline-fitted camcorder path. Our method can produce full-frame stabilized videos, and not only has the high frequency shaky motioned but also the low frequency unexpected movements are removed. To achieve this, we use a polyline to estimate a new stable camcorder motion path, and then we fill the dynamic and static missing areas caused by frame alignment from other frames to keep the same resolution and quality as the original video. Furthermore, we smooth the discontinuous regions by using a three-dimensional Poisson-based method. After the above automatic operations, a full-frame stabilized video can be achieved.

## 1. INTRODUCTION

As the use of digital camcorders grows, to capture videos using hand-held camcorders becomes more and more convenient than before. However, since most people usually do not bring a tripod with their camcorders, unwanted vibration in video sequence is an unavoidable effect due to the handshakes. To avoid or remove the annoying shaky motion is one of the significant problems in home videos, and video stabilization is an important technique to solve this problem. Many existed video stabilization applications result a stabilized video by smoothing the camcorder motion path and then truncating the missing areas after aligning the video frames along the smoothed motion path. Hence, the stabilized videos still have many

unexpected movements, since only high frequency shaky motions are removed during the smoothing stage. Moreover, the video qualities of the stabilized videos are usually decreased due to the truncating stage.

In this paper, we propose a novel, robust, and practical method of full-frame video stabilization with a polyline-fitted camcorder path. To achieve this, we use a polyline to estimate a new stable camcorder motion. Hence, the resulted videos are much stable and much close to the videos that the users want to capture. Once we obtain a new stable camcorder motion path of the video, the video frames are aligned along the new motion path to form a stabilized video.

Due to the frame alignment, there are several missing areas in the new stabilized video. Unlike other trimming approaches, we fill the dynamic and static missing areas respectively by using motion inpainting and warped neighboring video frames. This completion method works well even if there are some moving objects located at the boundary of the video frames while keeping the same resolution and quality as the original video. However, since we use a polyline to fit the camcorder motion path rather than using a parametric curve to smooth the motion path, the missing areas are usually large and cannot be easily completed by neighboring frames. To fill the missing areas using the pixels on the frames far from the current frame may cause the discontinuity at the boundaries of the filled areas, since the intensity of each video frame is usually not necessarily the same. Hence, we smooth the discontinuity boundaries by using a three-dimensional Poisson-based method which takes both of the spatial and temporal consistency into consideration and can result seamless stitching spatially and temporally.

After stabilizing the videos, the blurry video frames also caused by the handshakes become much noticeable. Hence, we detect the blurry frames and transfer the pixels from neighboring sharper frames. Therefore, our method can produce full-frame stabilized videos, and not only has the high frequency shaky motioned but also the low frequency unexpected movements are removed. The stabilized videos are stable, comfortable, and much more close to the videos which the users

really want to capture if they bring a tripod with them when capturing.

## 2. RELATED WORK

Video stabilization is an important research topic in multimedia, image processing, computer vision, and computer graphics. Buehler *et al.* proposed an image-based rendering (IBR) method to stabilize videos [4]. Recently, image processing methods are widely used for video stabilization. For estimating the camcorder motion path, Litvin *et al.* estimated a new camcorder motion path by altering camera parameter [9], and Matsushita *et al.* smoothed the camcorder motion path to reduce the high frequency shaky motions [11]. However, although the high frequency shaky motions can be easily reduced, the stabilized videos still have low frequency unexpected movements.

When filling up the missing image areas, there are some image inpainting approaches developed for recovering the missing holes in the image [1, 5, 8]. Although these approaches can complete the missing regions with correct structure, but there will be obvious discontinuity if we recover each video frame respectively. Litvin *et al.* used mosaic method to fill up the missing areas in the stabilized video [9], however they did not consider the moving objects may appear at the boundary of the video. Wexler *et al.* and Shiratori *et al.* filled up the missing holes by sampling the spatio-temporal volume patches from other portion of the video volume [15, 16]. The former approach used the most similar patch in color space for completing the missing holes and the later one used the patch with similar motion vector. The drawback of these methods is that they need large computing time for searching a proper patch. Jia *et al.* and Patwardhan *et al.* segmented the video into two layers and recovered them individually [7, 13]. These methods focused on long and periodic observed time of the moving objects, but this is not guaranteed in common home videos.

## 3. OVERVIEW

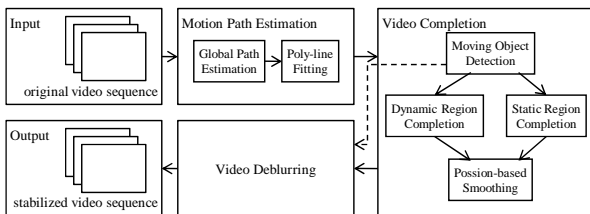


Fig. 1: System framework.

Fig. 1 shows the system framework of our algorithm. The input of our system is a video sequence captured by a hand-held camcorder without using a tripod. Hence, the video sequence has much annoying shaky motions due to the handshakes. The first process of our system is

called motion path estimation (Sec. 4). In this process, the camcorder motion path of the original video is estimated and changed to be a stabilized one. There are three steps contained in this process. In the first step (Sec. 4.1), we find out the transformation between the consecutive frames and combine all of the transformations to obtain the global camcorder motion path of the original video. In the second step (Sec. 4.2), the estimated global camcorder motion path is approximated by a polyline, since the camcorder motion path of the video captured by using a tripod is like a polyline.

After the stabilized camcorder motion path is achieved, the video completion process is applied (Sec. 5). Because the position of each frame is changed according to the frame alignment along the new motion path, there are some missing areas within each aligned frame. The first step in the video completion is to detect if there exists moving objects and where they are (Sec. 5.1). In the second step, we separate the moving objects as the dynamic foreground regions from the static background regions and complete the missing areas of them by different methods (Sec. 5.2 and 5.3). To fill the missing areas using the pixels on the frames far from the current frame may cause the discontinuity at the boundaries of the filled areas, since the intensity of each video frame is usually not necessarily the same. In order to make a seamless stitching, we apply a three-dimensional Poisson-based smoothing method on the discontinuous regions (Sec. 5.4).

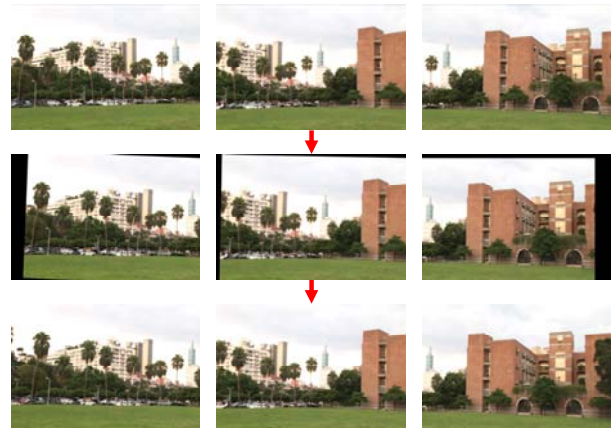


Fig. 2: Top row: Three frames of the original video. There are annoying shaky motions in the frames. Middle row: Stabilized frames. Black areas show the missing areas of the frames. Bottom row: Completed frames. This is the result of our method; the shaky motions of the video are removed.

The last process is video deburring (Sec. 6). Because the motion blur of each frame may not be matched in the stabilized camcorder motion path, the blurry frames become much noticeable in the new stabilized video. Instead of finding the accurate point spread function for image deburring, we choose a video

deblurring method by transferring the pixel values from neighboring sharper frames to the blurry frames. After the above automatic processes, the output will be a stabilized video with stable camcorder motion path while keeping the same resolution and quality as the original one. Fig. 2 shows three frames of an input video and their stabilized results before and after the video completion and deblurring processes.

#### 4. MOTION PATH ESTIMATION

In order to generate stabilized videos, we first estimate the camcorder motion path of the original video (Sec. 4.1). Then, the original camcorder motion path is stabilized by using a polyline-based motion path (Sec. 4.2), so that the undesirable motion caused by hand shake can be removed.

##### 4.1. Global Path Estimation

To estimate the global camcorder motion path, we first extract the feature points of each frame by SIFT (Scale Invariant Feature Transform) [10], which is invariant to scaling and rotation of the image. The feature points on every consecutive frame are matched if the distances between the feature descriptions are small enough and RANSAC (RANdom SAMple Consensus) [6] is used to select the inliers of the matched feature pairs. For the accuracy, an over-constrained system is applied to find out the least square solution between these matched feature pairs and derive the affine transformation between the two consecutive frames. The affine transformation is represented by a  $3 \times 3$  affine model which contains six parameters. That means if we find out the transformation matrix  $T_i$  between frames  $i$  and  $i+1$ , the corresponding pixel  $p_i$  on the frame  $i$  and pixel  $p_{i+1}$  on the frame  $i+1$  have the following relationship:  $p_{i+1} = T_i \cdot p_i$ . Once the transformation matrices between the consecutive frames are obtained, all of the transformations can be combined to derive a global transformation chain.

##### 4.2. Motion Path Fitting

To obtain a stabilized camcorder motion path without not only the high frequency shaky motions but also the low frequency unexpected movements, we use a polyline to fit the estimated global camcorder motion path. We first separate the camcorder motion path estimated from Sec. 4.1 to be horizontal and vertical ones  $M_i = [x, y]$ , and operate them respectively. Then, Kalman filter is employed to estimate a smooth camcorder motion path  $K_i$  of  $M_i$  [12] which is represented in two-dimensional space as

$$M_i = HK_i + v, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$K_{i+1} = \Phi K_i + w, \Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where  $\Phi$  is the state transition model which is applied to the previous state  $K_i$  to approximate the next state  $K_{i+1}$ ,  $H$  is the observation model which maps the true state space  $M_i$  into the observed space  $K_i$ ,  $v \sim N(0, R)$  is the observation noise which is assumed to be a zero mean Gaussian white noise with covariance  $R$ , and  $w \sim N(0, Q)$  is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance  $Q$ . The camcorder motion path smoothed by the Kalman filter is shown as the green curves in Fig. 3.

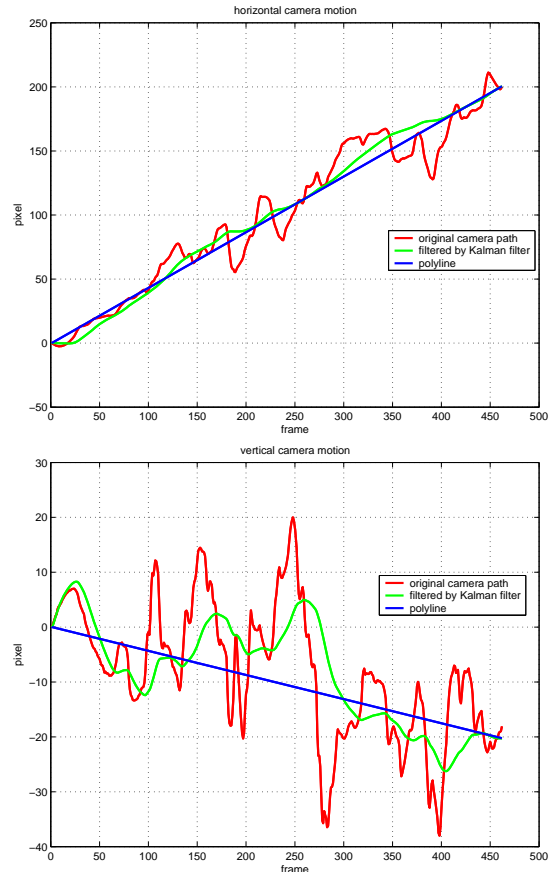


Fig. 3: The original camcorder motion path (red curve) and the estimated camcorder motion path after applying Kalman filter (green curve) and fitting by a polyline (blue straight line) for horizontal (Upper) and vertical (Lower) motion paths.

After applying the Kalman filter, we can obtain the smoothed camcorder motion path (Kalman path). Then, we use a polyline to fit the Kalman path. We first assume there is only one straight line connected the camcorder position of the first frame and that of the last frame on the Kalman path. Then, we calculate the difference between the camcorder position in the original motion path of each frame and the temporary motion path (the straight line) and find the maximal difference. If the maximal difference is larger than the a threshold, the straight line is broken by connecting the camcorder position of this frame on the Kalman path and the endpoints of the straight line, and hence the temporary camcorder motion path (the straight line) becomes a polyline with two straight line segments. This step is performed iteratively until the temporary camcorder motion path can preserve all important regions in the video. The final camcorder motion path is shown as the blue polyline in Fig. 3.

Once the camcorder motion path is fitted by a polyline, the video frames are aligned along the polyline fitted camcorder motion path. If the global transition matrix from the first frame to the  $i$ -th frame is denoted by  $M_i$ , then the  $i$ -th frame is aligned to  $M_i \cdot \prod_{j=i-1}^0 T_j^{-1} \cdot p_i$ , where  $p_i$  means the pixels on the  $i$ -th frame and  $T_i$  represented affine transformation matrix between  $j$ -th frame and  $j+1$ -th frame. Hence, we can obtain a stabilized video after the polyline fitting and frame alignment.

## 5. VIDEO COMPLETION

After aligning the video frames along the stabilized camcorder motion path, there are several missing areas in the new stabilized video. Traditionally, this problem could be solved by cutting out the missing areas and scaling the stabilized video to its original size, but it may result a stabilized video with worse resolution. Hence, to make the resolution of the stabilized video as good as that of the original one, the missing areas are filled from other frames.

To complete the video, we first detect the moving objects to segment the video to a static background region and some dynamic moving object regions (Sec. 5.1). Then, we complete the missing areas by filling dynamic regions (Sec. 5.2) and static regions (Sec. 5.3) respectively. Since the camcorder motion path is fitted by a polyline, the missing areas may be large and need to be filled by the pixels on the frames far from the current frame, so we provide a three-dimensional Poisson-based smoothing method to smooth the discontinuity stitched areas (Sec. 5.4).

### 5.1. Moving Object Detection

In order to detect moving objects, we first align every pair of adjacent frames by using the affine transformation obtained in Sec. 4.1. Then, we evaluate the optical flow of them by using an efficient and less noisy optical flow approach [2, 3] to obtain the motion vector of each pixel. The motion vector of pixel  $p_i$  can be described as  $F_i(p_i)$  which represents the motion flow at pixel  $p_i$  from frame  $i$  to frame  $i+1$ , and the length of the motion vector shows the motion value.



Fig. 4: Left: The frame after changing the position according to the stabilized camcorder motion path. Right: The mask of detected moving objects (white regions).

The motion values in the moving object regions are considered to be relatively larger than those in the static background region. Hence, we can get a simple mask to show the regions with large motion values by a simple threshold as shown in Fig. 4. The dynamic regions are obtained by evaluating the dilation of the mask, which can help to guarantee the boundary of the moving objects is involved in the dynamic regions. If the missing area falls in the regions where the neighboring pixels have been asked as the dynamic region, this area is treated as the dynamic region and motion inpainting is used to complete the area, otherwise we recover the area by mosaicing.

### 5.2. Dynamic Region Completion

For the dynamic missing regions, instead of filling in the color values from other frames directly, we want to fill them up with correct motion vectors. Once we derive the motion vectors of each pixel in the missing areas, we can get the pixel color from the next frame according to the motion vectors. The local motion vectors in the known image areas are propagated to the dynamic missing areas as described in [11].

First, local motion vectors are estimated by computing the optical flow between the stabilized frames [2, 3]. The propagation starts at the pixel on the boundary of the dynamic missing areas, its motion vector is calculated as a weighted average of the motion vectors of its neighboring pixels. The process will continue until the dynamic missing areas are filled with motion vectors completely. If  $p_i$  is a pixel in the missing area, it will be filled according to its motion vector which is determined by

$$F_i(p_i) = \frac{\sum_{q_i \in N_{p_i}} w(p_i, q_i) F_i(q_i)}{\sum_{q_i \in N_{p_i}} w(p_i, q_i)},$$

where  $w(p_i, q_i)$  determines the contribution of the motion vector of pixel  $q_i$ ,  $N_{p_i}$  denotes the eight neighboring pixels of  $p_i$ , and  $F_i(p_i)$  represents the motion vector at pixel  $p_i$  from frame  $i$  to frame  $i+1$ . Suppose the neighboring pixel  $q_i \in N_{p_i}$  already has a motion vector, according to its motion vector, we can estimate its position on the next frame as  $q_{i+1}$ . By using the geometric relationship between the pixels  $p_i$  and  $q_i$ , the position of the pixel  $p_{i+1}$  in the frame  $i+1$  can also be determined as illustrated in Fig. 5. Since the pixels in the same object have similar color values and move in the same direction, if the difference between the color values of the pixels  $q_{i+1}$  and  $p_{i+1}$  is small, they will likely belong to the same object, and the weight of the motion vector of pixel  $q_i$  is set to be large as

$$w(p_i, q_i) = \frac{1}{(ClrD(p_{i+1}, q_{i+1}) + \varepsilon)},$$

where  $\varepsilon$  is a small value for avoiding the division by zero and  $ClrD(p_{i+1}, q_{i+1})$  is the  $l^2$ -norm color difference in RGB color space of the pixels  $q_{i+1}$  and  $p_{i+1}$ . This weight term guarantees that the contribution of the motion vector in different objects is small. Fig. 6 shows the result.

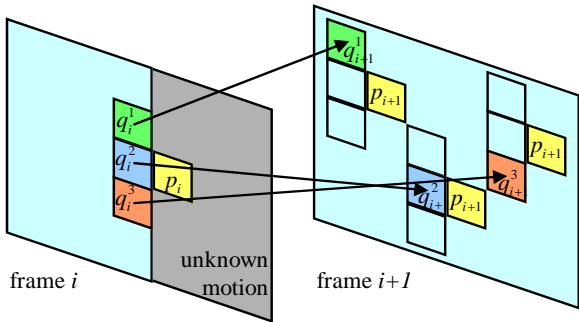


Fig. 5: Suppose the neighboring pixel  $q_i$  already has a motion vector, according to its motion vector, we can estimate its position on the next frame as  $q_{i+1}$ . By using the geometric relationship between the pixels  $p_i$  and  $q_i$ , the position of the pixel  $p_{i+1}$  in the frame  $i+1$  can also be determined.

### 5.3. Static Region Completion

After completing the dynamic regions, we then recover the static ones by its neighboring frames which are

wrapped to the current frame according to the affine transformation obtained in Sec. 4.1. For the pixel  $p_i$  in the static missing area at frame  $i$ , if there exists its corresponding pixel  $p_i$  at the warped neighboring frame  $i'$ , we directly copy the pixel to the missing pixel  $p_i$  by  $p_i = T_i^i(p_i)$ , where  $T_i^i$  represents the transformation from frame  $i'$  to frame  $i$ . Fig. 7 illustrates the process and Fig. 8 shows the static region completion result.



Fig. 6: Upper-Left: The frame after changing the position according to the stabilized camcorder motion path. There is a missing area at the left side and a moving object across the missing area. Lower-Left: The result of dynamic region completion. Right Column: The close up view of the yellow square of the Left Column.

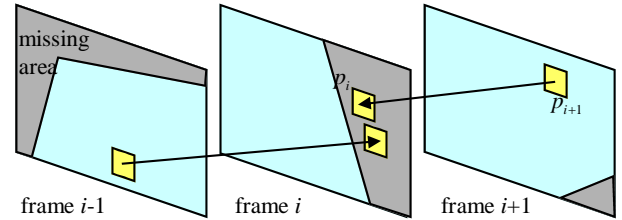


Fig. 7:  $p_i$  is a pixel in the missing area at current frame  $i$  and  $p_{i+1}$  is its corresponding pixel in the next frame  $i+1$ . The pixel value of  $p_{i+1}$  is directly copied to  $p_i$  for recovering the missing pixel.



Fig. 8: Left: The frame after changing the position according to the stabilized camcorder motion path. There is a missing area at the right side and upper side. Right: The result of static region completion. Since the missing area is large, there is a discontinuity boundary between the recovered pixels and the original frame.

To find the corresponding pixel  $p'_i$  of  $p_i$ , we begin the search from the nearest neighboring frame and propagate the search out. For example, if  $i$  is the current frame we want to recover, we search the frames  $i-1$  and  $i+1$  first, if there are missing areas still have not been recovered by the two frames, the following two frames  $i-2$  and  $i+2$  are used to recover the missing areas. We keep the search until all the missing pixels in the static missing areas are completed. Finally, if there are still some missing areas we cannot recover, we use a simple image inpainting approach to complete them.

### 5.3. Poisson-Based Smoothing

Although the missing areas caused by the stabilized camcorder motion path are completed, there may be a discontinuous boundary between the recovered pixels and the original frame, since the missing areas may be large and needed to be filled from the frame far from the current one. This problem may be solved by simply applying some smoothing approaches for the boundary areas; however these simply smoothing operations can only solve the spatial discontinuity problem. When we play the spatially smoothed video, it still has temporal discontinuity problem. In order to keep the spatial and temporal continuity, we provide a three-dimensional Poisson-based smoothing method. Poisson-based smoothing approach is often used in image editing [14], and we extend this approach for video editing.

To solve the discontinuity problem, before filling in a pixel from other frames, the Poisson equation is applied to obtain a smoothed pixel by considering its neighboring pixels in the same frame and neighboring frames. We first apply the Poisson equation in the spatial domain which is written as: For all  $p \in \Omega$ ,

$$|N_p|f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq},$$

where  $\Omega$  denotes the missing area,  $p$  is a pixel in the missing area  $\Omega$ ,  $N_p$  denotes the neighboring pixels of pixel  $p$ ,  $|N_p|$  is the number of neighboring pixels  $N_p$ ,  $f_p$  and  $f_q$  are the correct pixel values of pixels  $p$  and  $q$  which are what we want to derive,  $v_{pq}$  determines the divergence of pixels  $p$  and  $q$ ,  $\partial\Omega$  is the region surrounding the missing area  $\Omega$  in the known image areas, and  $f_q^*$  denotes the known color value of pixel  $q$  in  $\partial\Omega$ .

The Poisson equation can keep the correct structure in the missing area and achieve a seamless stitching between the recovering areas and known image areas. In order to achieve temporal coherence, after recovering the missing areas of each frame, we correct the pixel

values of the missing areas by apply the Poisson equation again. For the missing areas, now we consider not only the spatial neighboring pixels but also temporal neighboring pixels. Hence, the Poisson equation is the same, but  $N_p$  includes all neighboring pixels of pixel  $p$  in the video volume. Fig. 9 shows the result.



Fig. 9: Upper-Left: The frame after video completion. Since the missing area is large, there is a discontinuous boundary between the recovered pixels and the original frame. Lower-Left: The result of video completion with Poisson-based smoothing. Right Column: The close up view of the yellow square of the Left Column.

## 6. VIDEO DEBLURRING

After video stabilization, the blurry frames which look smooth in the original video become noticeable. Our video deblurring method fundamentally based on [11], but we separate the moving objects from static background first, and deal with them respectively as the video completion process. Since the blur of the static background is much more noticeable than that of the moving objects, in the following explanation, we only focus on the static background deblurring.

The main idea of this method is to copy the pixels of neighboring sharper frames to the blurry frames. We first evaluate the "relative blurriness" of each frame by calculating the gradient of it. 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:

$$B_i = \sum (g_x(p_i)^2 + g_y(p_i)^2),$$

where  $p_i$  is the pixel of the frame  $i$ , and  $g_x$  and  $g_y$  are the gradients of  $x$ - and  $y$ - directions, respectively. We can derive the relative blurriness between the current frame and its neighboring frames by comparing their blurriness  $B_i$ . If the blurriness  $B_i$  of current frame  $i$  is smaller than the blurriness  $B_{i'}$  of its neighboring frames  $i'$ , then the frames  $i'$  are treated to be sharper than the frame  $i$  and we can use the frames

$i'$  to recover the current blurry frame  $i$  by transferring the corresponding pixels from these sharper frames  $i'$  to the blurry frame  $i$  by

$$\tilde{p}_i = \frac{p_i + \sum_{i' \in N_i} w_{i'}^i T_{i'}^i(p_{i'})}{1 + \sum_{i' \in N_i} w_{i'}^i},$$

where  $\tilde{p}_i$  and  $p_i$  are the same pixel in the frame  $i$  after and before the deblurring operation,  $N_i$  denotes the neighboring frames of current frame  $i$ ,  $T_{i'}^i$  represents the transformation from the neighboring frame  $i' \in N_i$  to frame  $i$ , and  $w_{i'}^i$  is a weighting factor between  $i'$  and  $i$  which is defined as:

$$w_{i'}^i = \begin{cases} 0 & \text{if } B_{i'} / B_i < 1 \\ B_{i'} / B_i & \text{otherwise} \end{cases}.$$

Fig. 10 shows the result of this deblurring method.

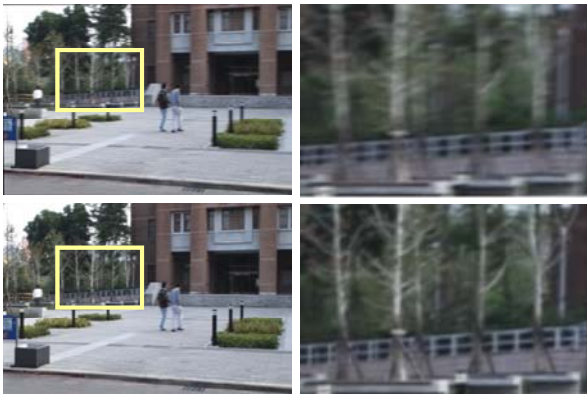


Fig. 10: Upper-Left: A blurry frame. Lower-Left: The result of video deblurring. Right Column: The close up view of the yellow square of the Left Column.

## 7. RESULT

All of the videos used in this paper was captured by using a hand-held video camera without using a tripod, and the resolution of the videos are all  $720 \times 480$ . The resolution of all stabilized videos is the same as the input videos. Fig. 2, Fig. 11 and Fig. 12 show our results. In Fig. 2, the user wants to use the hand-held camcorder to capture a panorama view. Without a tripod, the captured video is shaky due to the handshakes. In Fig. 11, the user wants to use the hand-held camcorder to capture a man walking with his child. Without a tripod, the captured video is shaky due to the handshakes. The bottom rows of Fig. 11 shows our result which is stabilized as captured by using a tripod. In Fig. 12, the user wants to use the hand-held camcorder to capture a man playing with his dog, but due to the view angle limitation, the user pans the camcorder a little bit to capture the whole scene. The

bottom row of Fig. 12 shows our result and the stabilized camcorder motion path is just like to capture the scene by using a tripod.

## 8. CONCLUSION AND FUTURE WORK

A full-frame video stabilization approach is proposed in this paper to obtain a stabilized video. Since we use a polyline to fit the original camcorder motion path, the stabilized motion path is much more stable than other smoothness approaches. Hence, in the stabilized video, not only the high frequency shaky motions but also the low frequency unexpected movements are removed. Although using a polyline to estimate the camcorder motion path may cause large missing areas, it is solved by applying a three-dimensional Poisson-based smoothing method. To fill the missing areas from other frames and deal with blurry frames, we separate the moving objects from the static background and deal with them respectively in completion and deblurring processes.

## ACKNOWLEDGEMENT

This paper was partially supported by the National Science Council of Taiwan under NSC 95-2622-E-002-018 and also by the Excellent Research Projects of National Taiwan University under 95R0062-AE00-02.

## REFERENCES

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image Inpainting," *Proc. ACM SIGGRAPH 2000*, pp. 417-424, 2000.
- [2] M. J. Black and P. Anandan, "A Framework for the Robust Estimation of Optical Flow," *Proc. IEEE ICCV 1993*, pp. 231-236, 1993.
- [3] M. J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *CVIU*, Vol. 63, No. 1, pp. 75-104, 1996.
- [4] C. Buehler, M. Bosse, and L. McMillan, "Non-Metric Image-Based Rendering for Video Stabilization," *Proc. IEEE CVPR 2001*, Vol. 2, pp. 609-614, 2001.
- [5] A. Criminisi, P. Perez, and K. Toyama, "Object Removal by Exemplar-Based Inpainting," *Proc. IEEE CVPR 2003*, Vol. 2, pp. 721-728, 2003.
- [6] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *CACM*, Vol. 24, No. 6, pp. 381-395, 1981.
- [7] J. Jia, T.-P. Wu, Y.-W. Tai, and C.-K. Tang, "Video Repairing Inference of Foreground and Background under Severe Occlusion," *Proc. IEEE CVPR 2004*, Vol. 1, pp. 364-371, 2004.

[8] A. Levin, A. Zomet, and Y. Weiss, "Learning How to Inpaint from Global Image Statistics," *Proc. IEEE ICCV 2003*, Vol. 1, pp. 305-312, 2003.

[9] A. Litvin, J. Konrad, and W. C. Karl, "Probabilistic Video Stabilization using Kalman Filtering and Mosaicking," *Proc. SPIE EI 2003*, Vol. 5022, pp. 663-674, 2003.

[10] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. IEEE ICCV 1999*, pp. 1150-1157, 1999.

[11] Y. Matsushita, E. Ofek, X. Tang, and H.-Y. Shum, "Full-Frame Video Stabilization," *Proc. IEEE CVPR 2005*, Vol. 1, pp. 50-57, 2005.

[12] Z. Pan and C.-W. Ngo, "Structuring Home Video by Snippet Detection and Pattern Parsing," *Proc. ACM MIR 2004*, pp. 69-76, 2004.

[13] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video Inpainting under Constrained Camera Motion," *IEEE TIP*, Vol. 16, No. 2, pp. 545-553, 2007.

[14] P. Pérez, M. Gangnet, and A. Blake, "Poisson Image Editing," *Proc. ACM SIGGRAPH 2003*, pp. 313-318, 2003.

[15] T. Shiratori, Y. Matsushita, X. Tang, and S. B. Kang, "Video Completion by Motion Field Transfer," *Proc. IEEE CVPR 2006*, Vol. 1, pp. 411-418, 2006.

[16] Y. Wexler, E. Shechtman, and M. Irani, "Space-Time Video Completion," *Proc. IEEE CVPR 2004*, Vol. 1, pp. 120-127, 2004.

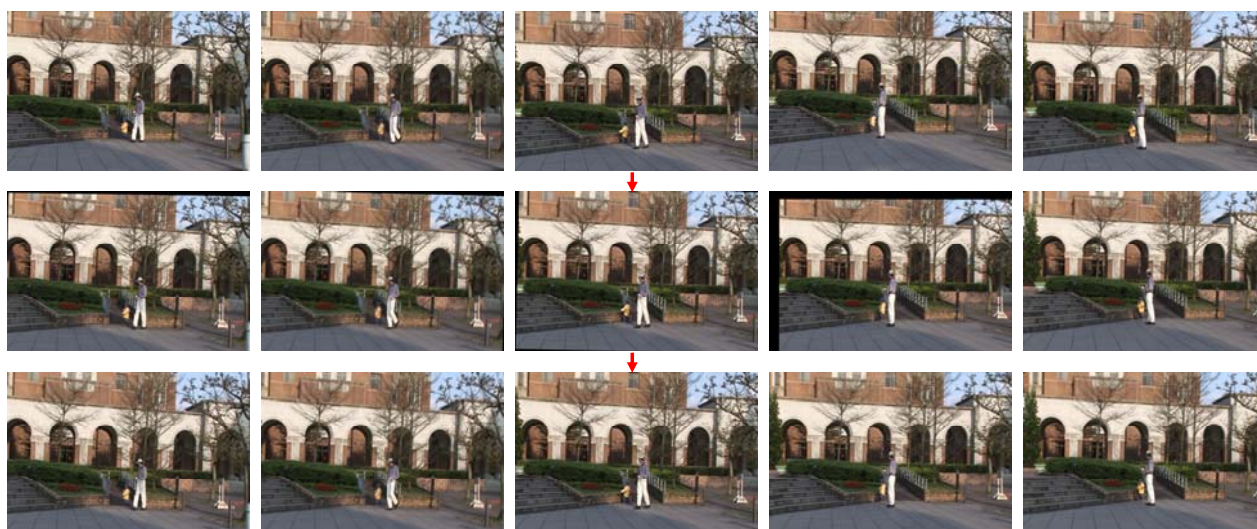


Fig. 11: Top row: Five frames of the original video. Middle row: Stabilized frames. Black areas show the missing areas of the frames. Bottom row: Our result.

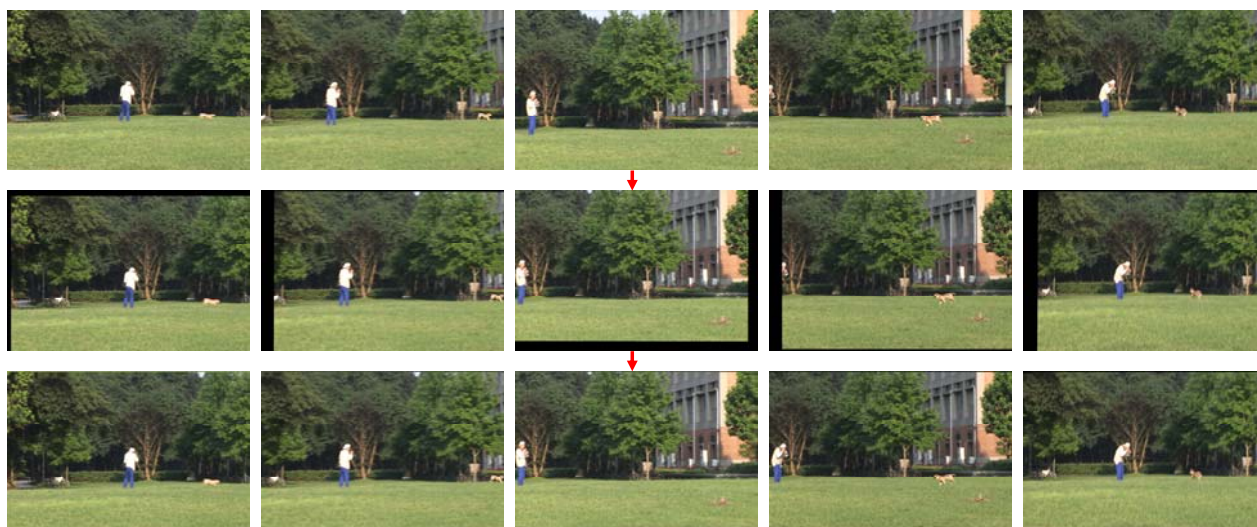


Fig. 12: Top row: Five frames of the original video. Middle row: Stabilized frames. Black areas show the missing areas of the frames. Bottom row: Our result.