Pacific Graphics 2008
T. Igarashi, N. Max, and F. Sillion
(Guest Editors)

Volume 27 (2008), Number 7

Capturing Intention-based Full-Frame Video Stabilization

Bing-Yu Chen[†] Ken-Yi Lee[‡] Wei-Ting Huang[‡] Jong-Shan Lin[§]

[†‡§]National Taiwan University [§]CyberLink Corp.

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, but the stabilized videos resulting from some current methods usually have decreased resolution and are still not so stable. In this paper, we propose a robust and practical method of full-frame video stabilization while considering user's capturing intention to remove not only the high frequency shaky motions but also the low frequency unexpected movements. To guess the user's capturing intention, we first consider the regions of interest in the video to estimate which regions or objects the user wants to capture, and then use a polyline to estimate a new stable camcorder motion path while avoiding the user's interested regions or objects being cut out. 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 and the important regions and objects can also be preserved.

Categories and Subject Descriptors (according to ACM CCS): I.4.4 [Image Processing and Computer Vision]: Restoration I.4.3 [Image Processing and Computer Vision]: Enhancement

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 is an unavoidable effect due to the hand shakes. 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 camcorder 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 quality of the stabilized videos is usually decreased due to the truncation.

In this paper, we propose a robust and practical method of full-frame video stabilization while considering user's capturing intention. To guess the user's capturing intention, we first consider the regions of interest (ROI) in the original captured video to estimate which regions or objects the user really wants to capture, and then use a polyline to estimate a new stable camcorder motion path while avoiding the user's interested regions or objects being cut out, since the camcorder motion path of the video captured with a tripod is like a polyline. Hence, the resulted video is much stable and much close to the video that the user wants to capture, since the capturing regions and objects are preserved and the camcorder motion path is stabilized as capturing with a tripod. To align the video frames along the stabilized camcorder motion path causes some missing areas, which need to be completed. While estimating the camcorder motion path, we also take the possibility of missing area completion into consideration.

[†] e-mail:robin@ntu.edu.tw

[‡] e-mail:{kez, weiting}@cmlab.csie.ntu.edu.tw

[§] e-mail:maruko_lin@gocyberlink.com

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

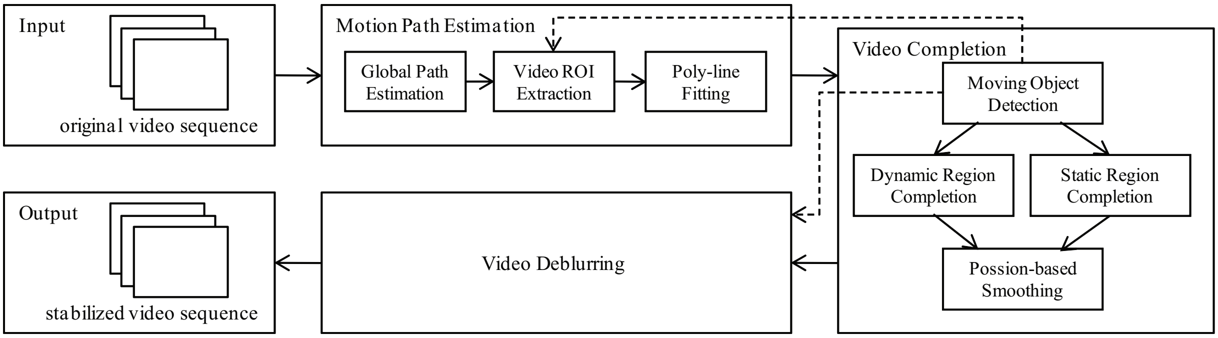

Figure 1: *System framework.*

After aligning the video frames, we fill the dynamic and static missing areas respectively. Since we use a polyline to fit the camcorder motion path rather than using a parametric curve, the missing areas are usually large and can not be easily completed by neighboring frames. To fill the missing areas using the frames far from the current one may cause discontinuity at the boundary of the filled areas, since the intensity of each video frame is usually not necessarily the same. Hence, we smooth the discontinuous boundaries by using a three-dimensional Poisson-based method while taking both of the spatial and temporal consistency into consideration, so that it can result seamless stitching spatially and temporally.

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 [BBM01]. For estimating the camcorder motion path, Litvin *et al.* estimated a new camcorder motion path by altering the camera parameters [LKK03], and Matsushita *et al.* smoothed the camcorder motion path to reduce the high frequency shaky motions [MOTS05]. However, although the high frequency shaky motions can be easily reduced, the stabilized videos still have low frequency unexpected movements. Gleicher and Liu stabilized the camcorder motion to be piecewise constant [GL07], which is similar with our method, but we also take the ROI and the possibility of missing area completion into consideration.

When filling up the missing image areas, there are some image inpainting approaches developed for recovering the missing holes in an image [BSCB00, CPT03, LZW03]. Although these approaches can complete the missing regions with correct structure, but there will be obvious temporal discontinuity if we recover each video frame individually. Litvin *et al.* used a mosaic method to fill up the missing areas in the stabilized video [LKK03], however they did not consider the moving objects may appear at the boundary of the video frames. Wexler *et al.* and Shiratori *et al.* filled up the missing holes by sampling the spatio-temporal volume patches from other portions of the video volume [WSI04,SMTK06]. The former approach used the most similar patch in color space for completing the missing areas 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. Matsushita *et al.* also provided motion inpainting to complete the moving objects appeared at the boundary of the video frames [MOTS05]. Jia *et al.* and Patwardhan *et al.* segmented the video into two layers and recovered them individually [JWTT04, PSB07]. These methods focused on long and periodic observed time of the moving objects, but this is not guaranteed in common home videos.

3. Overview

Figure 1 shows the system framework of our algorithm. The input of our system is a video captured by a hand-held camcorder without using a tripod. Hence, the video has much annoying shaky motions due to the hand shakes. The first process of our system is motion path estimation (Section 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 (Section 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 (Section 4.2), we extract the video ROI from the video by considering both of the spatial and temporal regions of interests. In the third step (Section 4.3), the estimated global camcorder motion path is approximated by a polyline. When the estimated camcorder motion path is fitted by a polyline, the extracted video ROI and the possibility of missing area completion are also taken into consideration in order to avoid the user's interested regions or objects being cut out and make the stabilized camcorder motion path as stable as possible.

After the stabilized camcorder motion path is achieved,

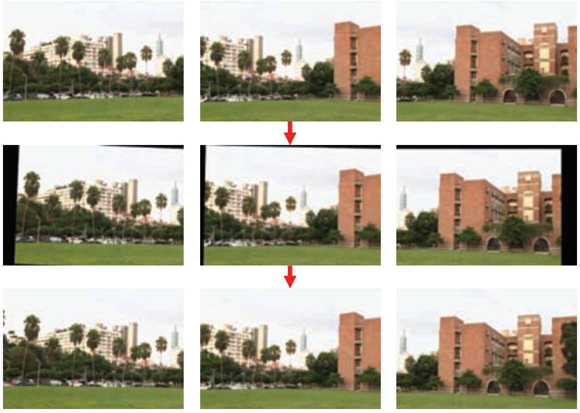

Figure 2: *Top row: Three frames of the original video. There are annoying shaky motions in the video. Middle row: Aligned frames, where the black regions show the missing areas. Bottom row: Completed frames, which are the result of our method; the shaky motions is stabilized.*

the video completion process is applied (Section 5). Because the position of each frame is changed according to the frame alignment along the new camcorder motion path, there are some missing areas within each aligned frame. The first step is to detect if there exists moving objects and where they are (Section 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 (Sections 5.2 and 5.3). To fill the missing areas using the frames far from the current one may cause discontinuity at the boundary 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 to smooth the discontinuous regions (Section 5.4).

The last process is video deblurring (Section 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 stabilized video. Instead of finding the accurate point spread function (PSF) for image deblurring, we use a video deblurring method by transferring the pixels from neighboring sharper frames to the blurry ones. After the above automatic processes, the output ia a stabilized video with a stable camcorder motion path while keeping the same resolution and quality as the original one. Figure 2 shows three frames of an input video and its stabilized result before and after the video completion process.

4. Motion Path Estimation

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) [Low99], which is invariant to scaling and rotation of the image. The feature points on every consecutive frames are matched if the distances between the feature descriptions are small enough and RANSAC (RANdom SAmple Consensus) [FB81] is used to select the inliers of the matched feature pairs. Then, 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. If the affine transformation matrix $\mathbf{T}_i$ between frames i and $i+1$ is constructed, the pixel p_i on frame i and its corresponding pixel p_{i+1} on frame $i+1$ will have the following relationship: $p_{i+1} = \mathbf{T}_i \cdot p_i$. Once the transformation matrices between the consecutive frames are obtained, all of the transformations can be combined to derive a global transformation chain.

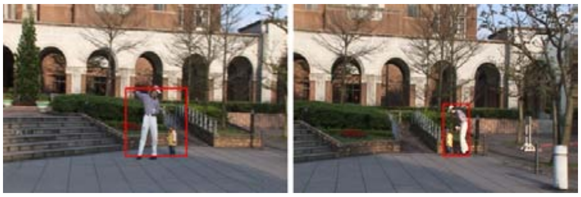

Figure 3: *The detected moving objects.*

4.2. Video ROI Extraction

To extract the video ROI from the input video, we take the temporal and spatial attention models into consideration to produce the spatio-temporal saliency maps. The spatial attention model is based on an image ROI extraction method proposed by Itti *et al.* [IKN98], and the temporal attention model is extracted by considering the moving objects in the video, which is detected by using the local motion vectors obtained in Section 5.1.

In order to detect the moving objects in current frame i with whatever small or large motion, we detect the moving objects by checking the local motion vectors from the previous n frames to the next n ones, where n is a frame window size. After detecting the moving objects in the $2n+1$ frames, the frame window size n is set to be $2n$ to detect the moving objects again in order to detect large motion. To generate the temporal saliency map $SalT(i)$ of frame i, we combine the temporal saliency maps $SalT(i)_n$ in different frame window sizes n by taking the union of the temporal saliency maps. Figure 3 shows the detected result.

To obtain the spatio-temporal attention model by combining the temporal and spatial attention models, we have to set the fusion methodology first. According to some observations, if the motion of the moving objects is large in the video, the spatio-temporal attention model should incorporate the temporal attention model more, otherwise it should incorporate the temporal attention model less. Then, the

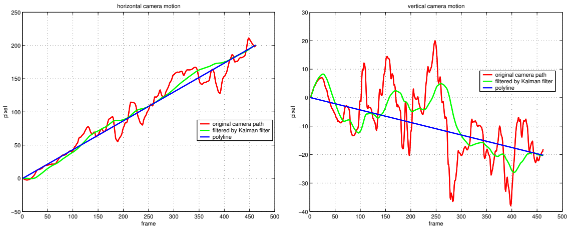

Figure 4: *The original camcorder motion path (red curve) and the estimated ones after applying the Kalman filter (green curve) and fitting by a polyline (blue straight line) for horizontal (Left) and vertical (Right) directions.*

spatio-temporal saliency map $Sal(i)$ is defined as $Sal(i) = kt_i \times SalT(i) + ks_i \times SalS(i)$ [ZS06], where $SalT(i)$ and $SalS(i)$ are the temporal and spatial saliency maps of frame i, and the weighting parameters kt_i and ks_i are defined as

$$kt_i = \frac{\alpha_i}{\alpha_i + \beta}, ks_i = \frac{\beta}{\alpha_i + \beta}, \tag{1}$$

where $\beta \in (0,1)$ is a constant value and

$$\alpha_i = \frac{SalT(i)}{\max(SalT(i)) - \min(SalT(i))}. \tag{2}$$

4.3. 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, since the camcorder motion path of the video captured with a tripod is like a polyline. We first separate the camcorder motion path estimated from Section 4.1 to be horizontal and vertical ones, and operate them respectively. Then, Kalman filter is employed to estimate a smooth camcorder motion path [PN04] while considering the video ROI extracted from Section 4.2. The camcorder motion path smoothed by the Kalman filter (Kalman path) is shown as the green curves in Figure 4.

Then, we use a polyline to fit the Kalman path while considering the possibility of missing area completion. To fill the missing areas, ideally it can be done by copying pixels from other frames. Once it cannot be simply achieved, we will use image inpainting to fill the monotonous missing areas to make the stabilized camcorder motion path as stable as possible. Hence, the possibility of missing area completion is evaluated by using the gradient of each frame's boundary areas. Then, the camcorder motion path is fitted by a polyline while taking the video ROI and the possibility of missing area completion into consideration as shown as the blue polyline in Figure 4.

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 $\mathbf{M}_i$, then the i-th frame is aligned to $\mathbf{M}_i \cdot \prod_{j=i-1}^{0} \mathbf{T}_j^{-1} \cdot p_i$, where p_i means the pixels on the i-th frame and $\mathbf{T}_j$ represented the 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. The top row of Figure 5 shows the frames aligned by a polyline-fitted camcorder motion path without taking video ROI into consideration. Hence, the building in the original video which the user wants to capture is cut out. By considering the saliency maps of the original video as shown in the bottom row of Figure 5, we can find a proper polyline-based camcorder motion path and the stabilized result is shown in the bottom row of Figure 2.

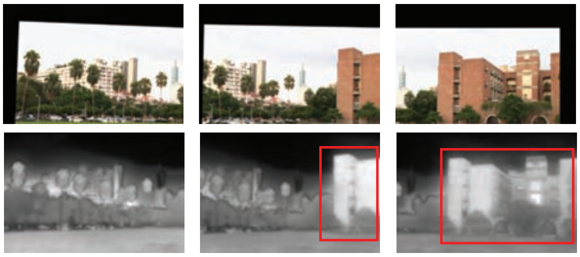

Figure 5: *Top row: Aligned frames of the top row of Figure 2 without considering video ROI. The black regions show the missing areas and the building is cut out. Bottom row: The saliency maps of the top row of Figure 2.*

5. Video Completion

After aligning the video frames along the stabilized camcorder motion path, there are several missing areas in the stabilized video. 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 (Section 5.1). Then, we complete the missing areas by filling dynamic regions (Section 5.2) and static regions (Section 5.3) respectively.

5.1. Moving Object Detection

In order to detect the moving objects, we first align every pair of adjacent frames by using the affine transformation obtained in Section 4.1. Then, we evaluate the optical flow of them [BA96] to obtain the motion vector of each pixel. The motion vector of pixel p_i can be described as $\mathbf{F}_i(p_i)$ which represents the motion flow at pixel p_i from frame i to $i+1$, and the length of the motion vector shows the motion value. Hence, the pixel p_i on frame i and its corresponding pixel p_{i+1} on frame $i+1$ according to the motion vector have the relationship: $p_{i+1} = \mathbf{T}_i \cdot \mathbf{F}_i(p_i)$, since the motion vector is obtained after aligning the frame according to the affine transformation matrix $\mathbf{T}_i$.

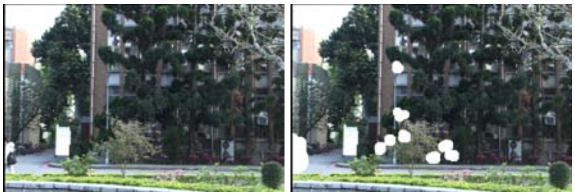

Figure 6: *Left: The frame after aligning to the stabilized camcorder motion path. Right: The mask of detected moving objects (white regions).*

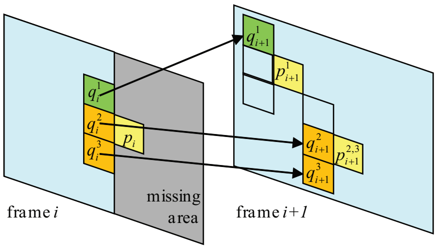

Figure 7: *The motion vector of pixel p_i at the missing area of frame i is determined by weighted averaging the motion vectors of its neighboring pixels q_i^j, $j = 1,2,3$ at the known area. The weight is defined by the color difference between p_i and q_i^j. Hence, even if the directions of the motion vectors of the neighboring pixels are not the same (like q_{i+1}^1), according to the color similarity of p_{i+1} and q_{i+1}^j, the motion vector to $p_{i+1}^{2,3}$ will be used and that to p_{i+1}^1 will be almost ignored.*

The motion values in the moving object regions are considered to be relatively larger than those in the static background regions. Hence, we can get a simple mask to show the regions with large motion values by a simple threshold as shown in Figure 6. The dynamic regions are obtained by evaluating the dilation of the mask, which can help to guarantee the boundary of the moving objects are involved in the dynamic regions. If the missing area falls in the region where the neighboring pixels have been masked as the dynamic one, this area is treated as the dynamic region and motion inpainting (Section 5.2) is used to complete the area, otherwise we recover the area by mosaicing (Section 5.3).

5.2. Dynamic Region Completion

For the dynamic missing regions, instead of filling the color values from other frames directly, we want to fill them 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 areas obtained in Section 5.1 are propagated to the dynamic missing areas as [MOTS05].

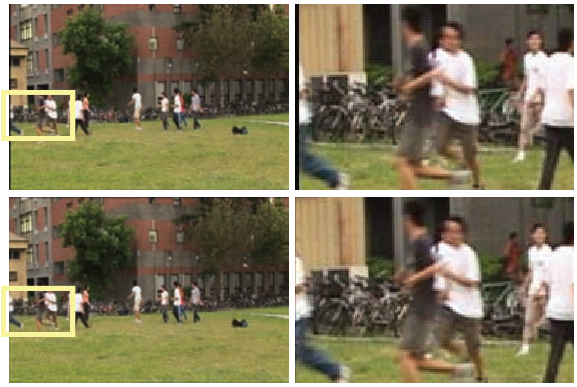

Figure 8: *Upper-Left: The frame after aligning to the stabilized camcorder motion path. There is a missing area at the left side and a moving object across the missing area. Lower-Left: The result of dynamic region completion. Right Column: The close-up view of the yellow rectangles in the Left Column.*

First, the local motion vectors are estimated by computing the optical flow between the stabilized video frames [BA96]. The propagation starts at the pixel on the boundary of the dynamic missing areas, its local motion vector is calculated as a weighted average of the motion vectors of its neighboring pixels at known areas. 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

$$\mathbf{F}_i(p_i) = \frac{\sum_{q_i \in N_{p_i}} w(p_i, q_i)\mathbf{F}_i(q_i)}{\sum_{q_i \in N_{p_i}} w(p_i, q_i)}, \tag{3}$$

where $w(p_i, q_i)$ determines the contribution of the motion vector $\mathbf{F}_i(q_i)$ of pixel q_i, and N_{p_i} denotes the neighboring pixels of p_i. 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 pixel p_{i+1} can also be determined as illustrated in Figure 7. 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 p_{i+1} and q_{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

$$w(p_i, q_i) = 1/(ClrD(p_{i+1}, q_{i+1}) + \varepsilon), \tag{4}$$

where ε 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 p_{i+1} and q_{i+1}. This weight term guarantees that the contribution of the motion vector in different objects is small. Figure 8 shows the result.

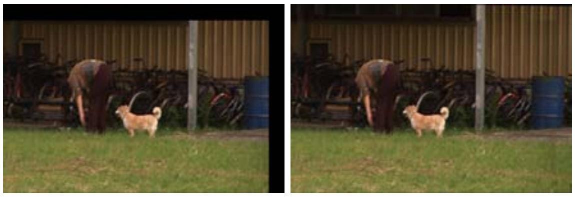

Figure 9: *Left: The frame after changing the position according to the stabilized camcorder motion path. There is a missing area at the right side and upper side. Right: The result of static region completion. Since the missing area is large, there is a discontinuity boundary between the recovered pixels and the original frame.*

5.3. Static Region Completion

After completing the dynamic regions, we then recover the static ones by its neighboring frames which are wrapped according to the affine transformation obtained in Section 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 $p_{i'}$ to the missing pixel p_i. Figure 9 shows the static region completion result.

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 missing pixels in the static missing areas are completed or all frames are searched. Finally, if there are still some missing areas, we then use image inpainting to complete them. Since the polyline-fitted camcorder motion path is determined by considering the gradient of each frame's boundary areas, the rest missing areas can always be completed.

5.4. 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. In order to keep the spatial and temporal continuity, we provide a three-dimensional Poisson-based smoothing method, which is extended from [PGB03].

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 ones. We first apply the Poisson equation in the spatial domain which is written as: For all $p \in \Omega$,

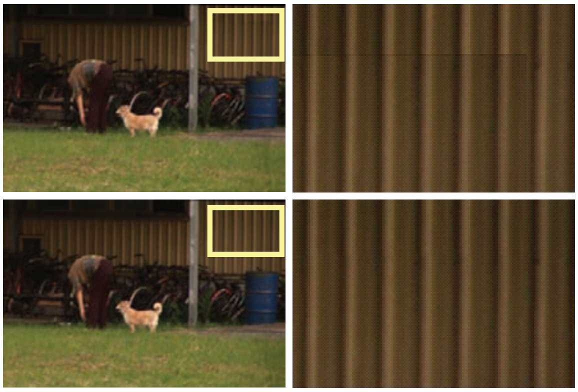

Figure 10: *Upper-Left: The frame after video completion. Since the missing area is large, there is a discontinuous boundary between the recovered pixels and the original frame. Lower-Left: The result of video completion with Poisson-based smoothing. Right Column: The close-up view of the yellow rectangles in the Left Column.*

$$|N_p|f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}, \qquad (5)$$

where Ω denotes the missing area, p is a pixel in the missing area Ω, 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 Ω in the known 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 the known ones. 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 by considering not only the spatial neighboring pixels but also the temporal neighboring ones. Hence, the Poisson equation is the same as Eq. (5), but N_p includes all neighboring pixels of pixel p in the video volume. Figure 10 shows the result.

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 [MOTS05], but we separate the moving objects from static background first and deal with them separatively as the video completion process. The main idea of this method is to copy the pixels from neighboring sharper frames to the blurry ones. We first evaluate the "relative blurriness" of each frame by calculating

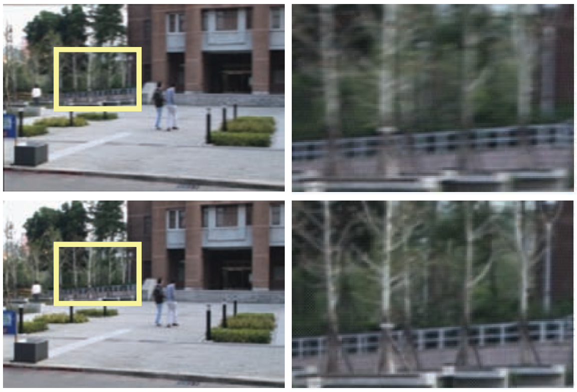

Figure 11: *Upper-Left: A blurry frame. Lower-Left: The result of video deblurring. Right Column: The close-up view of the yellow rectangles in the Left Column.*

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:

$$B_i = \sum_{p_i} (g_x(p_i)^2 + g_y(p_i)^2), \tag{6}$$

where p_i is the pixel on 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 ones 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 the frames i' to i by

$$\tilde{p}_i = \frac{p_i + \sum_{i' \in N_i} w^i_{i'} p_{i'}}{1 + \sum_{i' \in N_i} w^i_{i'}}, \tag{7}$$

where $\tilde{p}_i$ and p_i are the same pixel on frame i after and before the deblurring operation, N_i denotes the neighboring frames of current frame i, $p_{i'}$ is the corresponding pixel of p_i according to affine transformation $\mathbf{T}^i_{i'}$ and local motion vector $\mathbf{F}^i_{i'}(p_{i'})$ from frame $i' \in N_i$ to i, i.e., $p_i = \mathbf{T}^i_{i'} p_{i'}$ for static regions and $p_i = \mathbf{T}^i_{i'} \mathbf{F}^i_{i'}(p'_i)$ for dynamic ones, and $w^i_{i'}$ is a weighting factor between i' and i which is defined as:

$$w^i_{i'} = \begin{cases} 0 & \text{if } B_{i'}/B_i < 1 \\ B_{i'}/B_i & \text{otherwise} \end{cases}. \tag{8}$$

Figure 11 shows the result.

7. Result

All of the videos used in this paper was captured by using a hand-held camcorder without using a tripod, and the resolution of the videos are 720×480. The resolution of all resulted (stabilized) videos are the same as the input ones. Figure 2 and Figures 12∼15 show our results. In Figure 2, the user wants to use the hand-held camcorder to capture a panorama view. Without a tripod, the captured video are shaky due to the hand shakes. Although the camcorder motion path can be stabilized by a polyline-based camcorder motion path, without taking video ROI into consideration, the stabilized motion path may cause the building to be cut out as the top row of Figure 5. The bottom row of Figure 2 shows our result which is stabilized as captured by using a tripod and the building could be preserved in the stabilized video.

In Figure 12, the user wants to use the hand-held camcorder to capture a static scene. Although the shaky camcorder motion path can be smoothed by using other smoothness methods, the second row of Figure 12 still has some unwanted motions. The bottom row of Figure 12 shows our result. The estimated camcorder position is fixed as captured by using a tripod. For comparison, the forth row of Figure 12 shows the result of truncating the missing areas and the resolution is reduced.

In Figure 13, the user wants to use a hand-held camcorder to capture a man walking with his child. Without a tripod, the captured video are shaky due to the hand shakes. The bottom row of Figure 13 shows our result which is stabilized as captured by using a tripod. In Figure 14, the user wants to use a 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 Figure 14 shows our result and the stabilized camcorder motion path is just like to capture the scene by using a tripod. In Figure 15, the user wants to use a hand-held camcorder to capture some high buildings. The right column of Figure 15 shows out result, and the blurry frame specified by the orange rectangle has also been deblurred.

8. Conclusion and Future Work

A full-frame video stabilization approach is proposed in this paper to obtain a stabilized video while considering the video ROI in the input video. Since we use a polyline to fit the original camcorder motion path, the stabilized camcorder 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 and may cut out some capturing objects, the two problems are solved by applying a three-dimensional Poisson-based smoothing method and taking the video ROI into consideration. 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.

Our limitation is that, if the moving objects occupy too

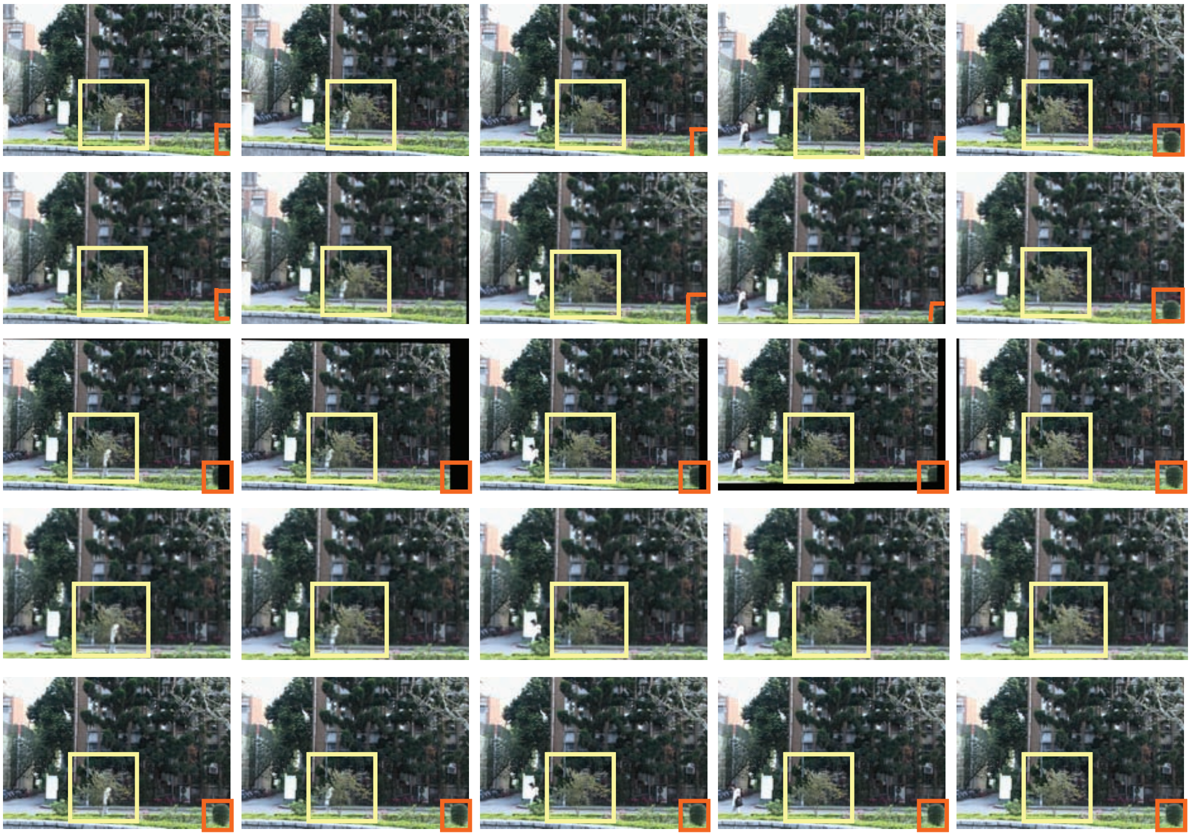

Figure 12: *Top row: Five frames of the original video. Orange and yellow rectangles show two trees in the video. Due to the camcorder hand-shake, the locations of the rectangles in each frame are different. Second row: Stabilized frames resulted by smoothing the camcorder motion path. The black regions show the missing areas. Since only high frequency shaky motions are removed, the rectangles still locate at different place in each frame. Third row: Stabilized frames resulted by polyline-fitted camcorder motion path. The locations of the trees (rectangles) are almost the same, but the missing areas are large. Fourth row: The result produced by truncating the missing areas of the third row. The resolution of the stabilized video is reduced, and the tree specified by the yellow rectangle has been cut out. Bottom row: Our result.*

large area in the video frames, there will be some problems about finding the affine transformation matrix. The inaccuracy transformation matrix would cause the result faulty. To deal with large moving objects is one of our future work. In addition, some problems about filling up missing areas and catching camcorder motion path will appear if the user shakes the camera calculatedly to cause the video juddering. Although the blurry frames are deblurred, our video deblurring method still can not deal with extreme blurry frames.

Acknowledgments

We would like to thank anonymous reviewers for their valuable comments. This work was partially supported by the National Science Council of Taiwan under NSC95-2622-E-002-018 and NSC96-2622-E-002-002, and also by the Excellent Research Projects of the National Taiwan University under NTU95R0062-AE00-02.

References

[BA96] BLACK M. J., ANANDAN P.: The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding 63*, 1 (1996), 75–104.

[BBM01] BUEHLER C., BOSSE M., MCMILLAN L.: Non-metric image-based rendering for video stabilization. In *IEEE Computer Vision and Pattern Recognition 2001 Conference Proceedings* (2001), vol. 2, pp. 609–614.

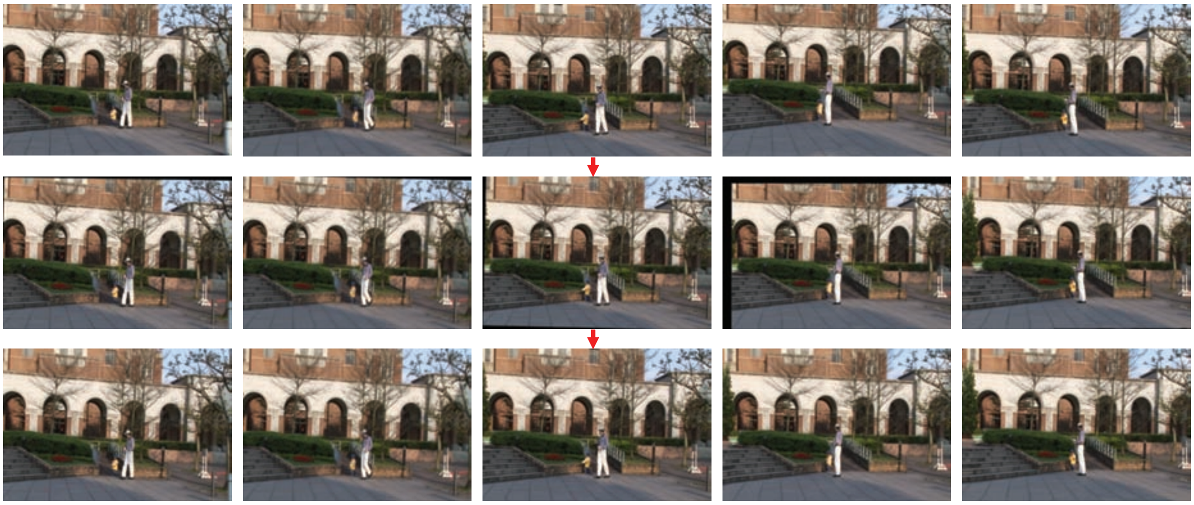

Figure 13: *Top row: Five frames of the original video. Middle row: Stabilized frames. The black regions show the missing areas. Bottom row: Our result.*

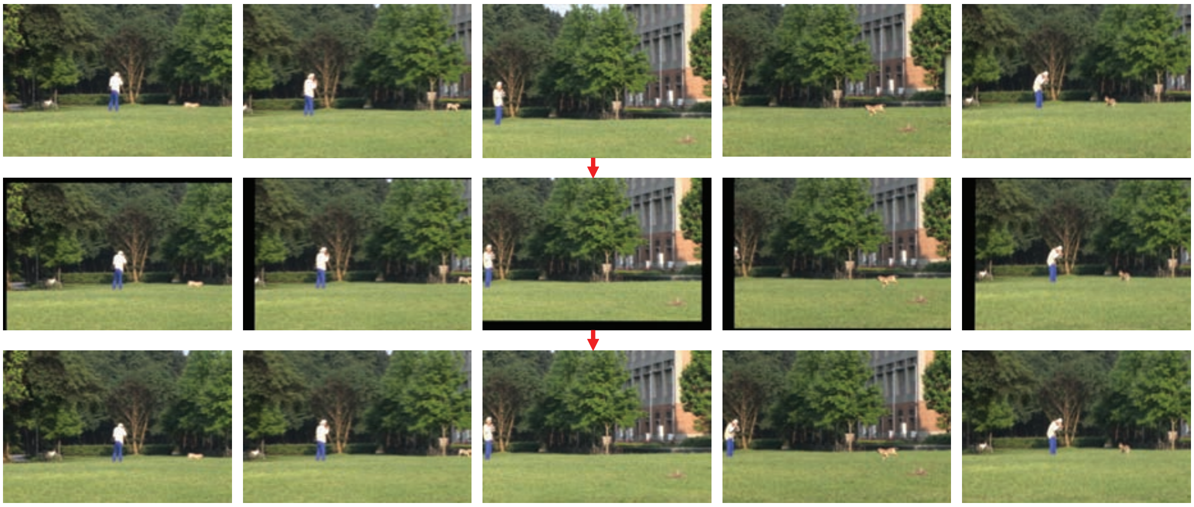

Figure 14: *Top row: Five frames of the original video. Middle row: Stabilized frames. The black regions show the missing areas. Bottom row: Our result.*

[BSCB00] Bertalmio M., Sapiro G., Caselles V., Ballester C.: Image inpainting. In *ACM SIGGRAPH 2000 Conference Proceedings* (2000), pp. 417–424.

[CPT03] Criminisi A., Perez P., Toyama K.: Object removal by exemplar-based inpainting. In *IEEE Computer Vision and Pattern Recognition 2003 Conference Proceedings* (2003), vol. 2, pp. 721–728.

[FB81] Fischler M. A., Bolles R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395.

[GL07] Gleicher M. L., Liu F.: Re-cinematography: improving the camera dynamics of casual video. In *ACM Multimedia 2007 Conference Proceedings* (2007), pp. 27–36.

[IKN98] Itti L., Koch C., Niebur E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (1998), 1254–1259.

[JWTT04] Jia J., Wu T.-P., Tai Y.-W., Tang C.-K.: Video repairing inference of foreground and background under severe occlusion. In *IEEE Computer Vision and Pattern Recognition 2004 Conference Proceed-*

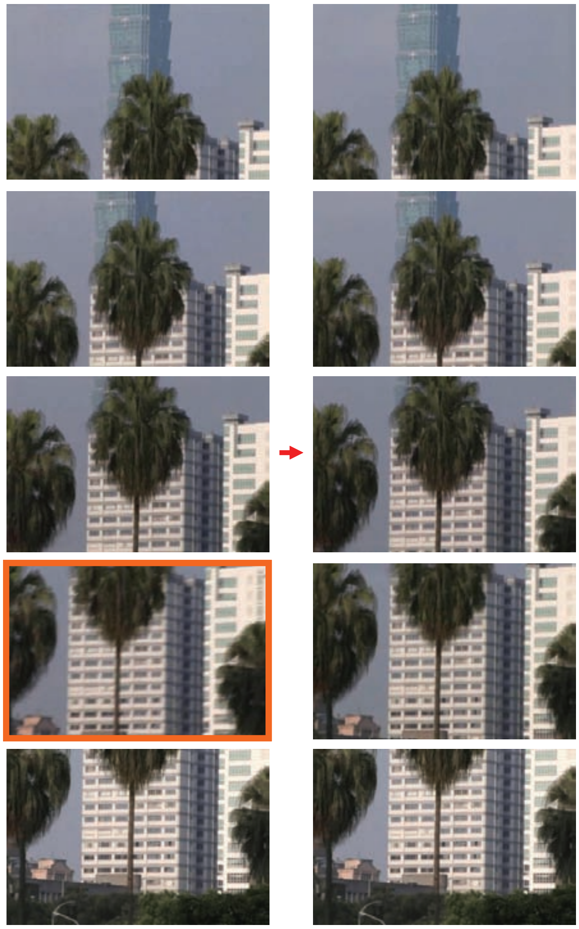

Figure 15: *Left: Five frames of the original video. Right: Our result. The blurry frame specified by the orange rectangle has been deblurred.*

ings (2004), vol. 1, pp. 364–371.

[LKK03] Litvin A., Konrad J., Karl W. C.: Probabilistic video stabilization using Kalman filtering and mosaicking. In *Proceedings of 2003 SPIE Conference on Electronic Imaging* (2003), vol. 5022, pp. 663–674.

[Low99] Lowe D. G.: Object recognition from local scale-invariant features. In *Proceedings of 1999 IEEE International Conference on Computer Vision* (1999), pp. 1150–1157.

[LZW03] Levin A., Zomet A., Weiss Y.: Learning how to inpaint from global image statistics. In *Proceedings of 2003 IEEE International Conference on Computer Vision* (2003), vol. 1, pp. 305–312.

[MOTS05] Matsushita Y., Ofek E., Tang X., Shum H.-Y.: Full-frame video stabilization. In *IEEE Computer Vision and Pattern Recognition 2005 Conference Proceedings* (2005), vol. 1, pp. 50–57.

[PGB03] Pérez P., Gangnet M., Blake A.: Poisson image editing. In *ACM SIGGRAPH 2003 Conference Proceedings* (2003), pp. 313–318.

[PN04] Pan Z., Ngo C.-W.: Structuring home video by snippet detection and pattern parsing. In *ACM SIGMM Multimedia Information Retrieval 2004 Conference Proceedings* (2004), pp. 69–76.

[PSB07] Patwardhan K. A., Sapiro G., Bertalmio M.: Video inpainting under constrained camera motion. *IEEE Transactions On Image Processing 16*, 2 (2007), 545–553.

[SMTK06] Shiratori T., Matsushita Y., Tang X., Kang S. B.: Video completion by motion field transfer. In *IEEE Computer Vision and Pattern Recognition 2006 Conference Proceedings* (2006), vol. 1, pp. 411–418.

[WSI04] Wexler Y., Shechtman E., Irani M.: Space-time video completion. In *IEEE Computer Vision and Pattern Recognition 2004 Conference Proceedings* (2004), vol. 1, pp. 120–127.

[ZS06] Zhai Y., Shah M.: Visual attention detection in video sequences using spatiotemporal cues. In *ACM Multimedia 2006 Conference Proceedings* (2006), pp. 815–824.