Full-Frame Video Stabilization with Motion Inpainting

Yasuyuki Matsushita, *Member*, *IEEE*, Eyal Ofek, *Member*, *IEEE*, Weina Ge, Xiaoou Tang, *Senior Member*, *IEEE*, and Heung-Yeung Shum, *Fellow*, *IEEE*

Abstract—Video stabilization is an important video enhancement technology which aims at removing annoying shaky motion from videos. We propose a practical and robust approach of video stabilization that produces full-frame stabilized videos with good visual quality. While most previous methods end up with producing smaller size stabilized videos, our completion method can produce full-frame videos by naturally filling in missing image parts by locally aligning image data of neighboring frames. To achieve this, **motion inpainting** is proposed to enforce spatial and temporal consistency of the completion in both static and dynamic image areas. In addition, image quality in the stabilized video is enhanced with a new practical deblurring algorithm. Instead of estimating point spread functions, our method transfers and interpolates sharper image pixels of neighboring frames to increase the sharpness of the frame. The proposed video completion and deblurring methods enabled us to develop a complete video stabilizer which can naturally keep the original image quality in the stabilized videos. The effectiveness of our method is confirmed by extensive experiments over a wide variety of videos.

Index Terms—Video analysis, video stabilization, video completion, motion inpainting, sharpning and deblurring, video enhancement.

1 Introduction

Video enhancement has been steadily gaining in importance with the increasing prevalence of digital visual media. One of the most important enhancements is video stabilization, which is the process for generating a new compensated video sequence where undesirable image motion is removed. Often, home videos captured with a handheld video camera suffer from a significant amount of unexpected image motion caused by unintentional shake of a human hand. Given an unstable video, the goal of video stabilization is to synthesize a new image sequence as seen from a new stabilized camera trajectory. A stabilized video is sometimes defined as a motionless video where the camera motion is completely removed. In this paper, we refer to stabilized video as a motion compensated video where only high frequency camera motion is removed.

In general, digital video stabilization involves motion compensation and, therefore, it produces missing image pixels, i.e., pixels which were originally not observed in the frame. Previously, this problem had been handled by either trimming the video to obtain the portion that appears in all frames, or constructing image mosaics by accumulating neighboring frames to fill up the missing image areas (see Fig. 1). In this paper, we refer to mosaicing as the image stitching with a global transformation.

- *Y. Matsushita, E. Ofek, X. Tang, and H. Shum are with Microsoft Research Asia, 3/F, Beijing Sigma Center, No. 49, Zhichun Road, Hai Dian District, Beijing, China 100080. E-mail: {yasumat, eyalofek, xitang, hshum}@microsoft.com.*
- *W. Ge is with Department of Computer Science and Engineering, Pennsylvania State University, 310 IST Building, University Park, PA 16802. E-mail: ge@cse.psu.edu.*

Manuscript received 22 Aug. 2005; revised 5 Dec. 2005; accepted 19 Dec. 2005; published online 11 May 2006.
Recommended for acceptance by H. Sawhney.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0451-0805.

The trimming approach has the problem of reducing the original video frame size since it cuts off the missing image areas. Moreover, sometimes due to severe camera-shake, there might be a very small common area among neighboring frames which results in very small video frames. On the other hand, mosaicing works well for static and planar scenes, but produces visible artifacts for dynamic or nonplanar scenes. This is due to the fact that mosaicing methods usually register images by a global geometric transformation model which is not sufficient to represent local geometric deformations. Therefore, they often generate unnatural discontinuity in the frame as shown in the midbottom of Fig. 1.

In this paper, we propose an efficient video completion method which aims at generating *full-frame* stabilized videos with good visual quality. At the heart of the completion algorithm, we propose a new technique, *motion inpainting*, to propagate local motion information which is used for natural stitching of multiple images. We also propose a practical motion deblurring method in order to reduce the motion blur caused by the original camera motion in the video. These methods enable us to develop a high-quality video stabilizer that maintains the visual quality of the original video after stabilization.

1.1 Prior Work

There are typically three major stages constituting a video stabilization process: camera motion estimation, motion smoothing, and image warping. The video stabilization algorithms can be distinguished by the methods adopted in these stages. After briefly overviewing prior methods by following these steps, we review prior work on video completion and motion deblurring.

Video stabilization is achieved by first estimating the interframe motion of adjacent frames. The interframe motion describes the image motion which is also called global motion. The accuracy of the global motion estimation is crucial as the first step of the stabilization.

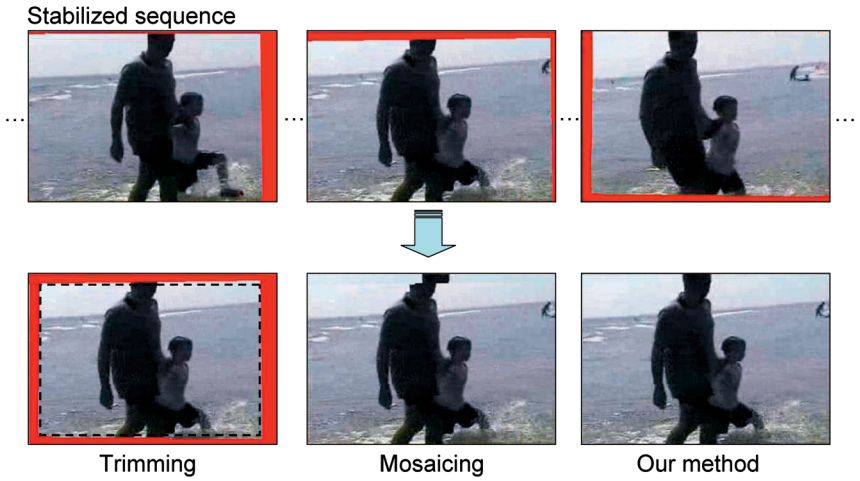


Fig. 1. Top row: stabilized image sequence. The red area represents the missing image area due to the motion compensation. Bottom row: from left to right, result of trimming (dotted rectangle becomes the final frame area), mosaicing, and our method.

There are two major approaches for global motion estimation. One is the feature-based approach [1], [2], [3], [4]; the other is the global intensity alignment approach [5], [6], [7], [8]. Feature-based methods are generally faster than global intensity alignment approaches, while they are more prone to local effects. A good survey on image registration is found in [9].

After computing the global transform chain, the second step is removing the annoying irregular perturbations. There are approaches which assume a camera motion model [3], [10], [11], [12], e.g., camera is fixed, or camera moves on a planar surface. It works well when the assumed camera motion model is correct; however, it is preferable to use a more general camera motion model since there exist many situations where camera motion cannot be approximated by such simple models, e.g., handheld camera motion. More recently, Litvin et al. [14] proposed to use a Kalman filter to smooth the general camera motion path, and Pilu [13] proposed an optimal motion path smoothing algorithm with constraints imposed by a finite image sensor size.

Filling in missing image areas in a video is called *video completion*. In [14], mosaicing is used to fill up the missing image areas in the context of video stabilization. However, the method does not address the problem of nonplanar scenes and moving objects that may appear at the boundary of the video frames, which might cause significant artifacts. Wexler et al. [15] filled in the holes in a video by sampling spatio-temporal volume patches from different portions of the same video. This nonparametric sampling-based approach produces a good result; however, it is extremely computationally intensive. Also, it requires a long video sequence of a similar scene to increase the chance of finding correct matches, which is not often available in the context of video stabilization. Jia et al. [16] took a different approach to solve the same problem by segmenting the video into two layers, i.e., a moving object layer and a static background layer. One limitation of this approach is that the moving object needs to be observed for a long time, at least for a single period of its periodic motion. In other words, it requires the cyclic transition of the color patterns in order to find good matching. Therefore, the method is not suitable for filling in the video boundaries where a sufficient amount of observation is not guaranteed. More recently, Cheung et al. showed an effective video fill-in result in their video epitomes framework [17]. The method again requires a similar video patch in the different portion of the same video; therefore, it fundamentally requires the periodic motion in order to achieve completion.

Motion blur is another problem in video stabilization as the original camera motion trajectory is replaced with a motion compensated trajectory in the stabilized video. Motion blur is a fundamental image degradation process which is caused by moving scene points traverse several pixels during the exposure time. *Motion deblur* has been studied extensively in the literature. In many single frame deblurring techniques, motion deblur is achieved by image deconvolution; however, the point spread function (PSF) is necessary to be estimated. Image deconvolution without knowing the PSF is called blind image deconvolution [18], [19], [20], [21], [22], relying on either image statistics or an assumption of very simple motion models. Although these methods are effective under specific conditions, these assumptions, e.g., a simple motion model, do not hold in many practical situations. Image deblurring methods by deconvolution require very accurate PSFs that are often hard to obtain. One exception is Ben-Ezra and Nayar's method [23] which explicitly measures motion PSFs with a special hardware to avoid difficulty in PSF estimation.

We take a different approach of reducing image blurriness caused by motion blur. Our method is similar to Adelson [24] and Bergen [25] which were proposed independently. In their methods, a sharper image is produced from source images taken of an object at substantially identical fields of view but with different focuses. The image composition is performed by evaluating the energy levels of the high frequency component in the images, by taking the image pixels which have greater energy levels (sharper pixels). In our context, since we need to deal with a dynamic scene captured with a moving camera, we integrate the global image alignment and an adaptive pixel selection process in order to achieve the pixel transfer. Recently, Ben-Ezra et al. [26] have developed a superresolution method with a jitter camera by applying an adaptive sampling of stationary image blocks and compositing them together. Although it requires a special hardware, it is able to increase the resolution of the original videos.

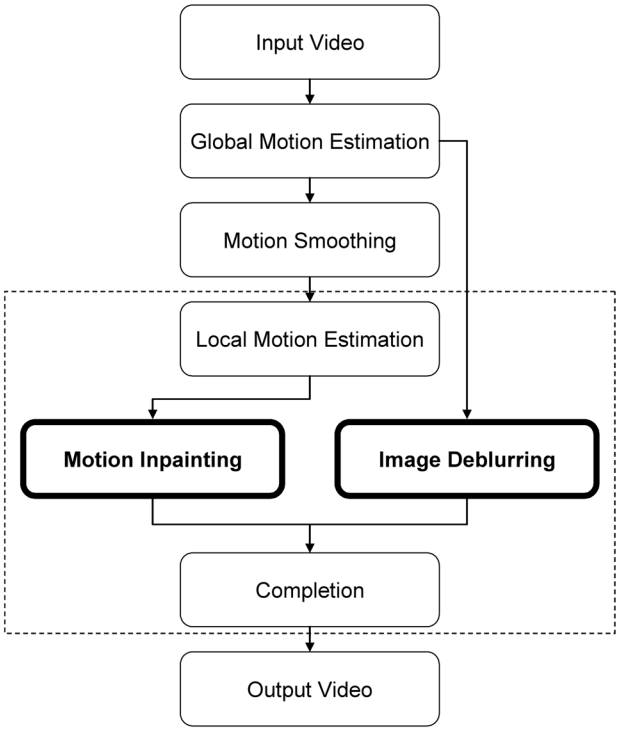


Fig. 2. Flow chart of the full-frame video stabilization.

1.2 Proposed Approach

The limitations of the previous approaches and practical demands motivated us to develop effective completion and deblurring methods for generating full-frame stabilized videos with good visual quality. This paper has two primary contributions in the process of video stabilization (Fig. 2).

1.2.1 Video Completion with Motion Inpainting

First, a new video completion method is proposed which is based on motion inpainting. The idea of motion inpainting is propagating local motion, instead of color/intensity as in image inpainting [27], [28], into the missing image areas. The propagated motion field is then used to help naturally fill up missing image areas even for scene regions that are nonplanar and dynamic. Using the propagated local motion field as a guide, image data from neighboring frames are locally warped to maintain spatial and temporal continuities of the stitched images. Image warping based on local motion was used in the deghosting algorithm for panoramic image construction by Shum and Szeliski [29]. Our method is different from theirs in that we propagate the local motion into an area where the local motion cannot be directly computed.

1.2.2 Practical Motion Deblurring Method

Second, we address the problem of motion blur in the stabilized videos. While motion blur in original videos looks natural, it becomes a visible artifact in stabilized videos because it does not correspond to the compensated camera motion. Furthermore, image stitching without appropriate deblurring results in inconsistent stitching of blurry and sharp images. To solve this problem, we developed a practical deblurring method which does not require accurate point spread functions (PSFs) which are often hard to obtain. Instead of estimating PSFs, we propose a method to transfer sharper pixels to corresponding blurry pixels to increase the sharpness and to generate a video of consistent sharpness as done by Adelson [24] and Bergen [25]. The proposed deblurring method is different from superresolution methods such as [30], [19] in that our method transfers pixels from sharper frames and replaces pixels by weighted interpolation. Therefore, our method does not increase the resolution of the frames, but restores resolution of blurry frames using other frames.

In the rest of this paper, Section 2 describes global and local motion estimation and smoothing methods which are used in our deblurring and completion methods. The video completion algorithm based on motion inpainting is described in Section 4. Section 3 presents the proposed image deblurring method. In Section 5, we show results of both stabilization and additional video enhancement applications. Conclusions are drawn in Section 6.

2 Motion Estimation and Smoothing

This section describes global and local motion estimation methods which are used in the proposed method. Section 2.1 describes the method to estimate interframe image transformation or global motion. Local motion, which deviates from the global motion, is estimated separately after global image alignment as described in Section 2.2. The global motion is used for two purposes, stabilization and image deblurring, while the local motion is used mainly for video completion. Section 2.3 describes the motion smoothing algorithm which is essential for stabilizing global motion.

2.1 Global Motion Estimation

We first explain the method of estimating global motion between consecutive images. In the case that a geometric transformation between two images can be described by a homography (or 2D perspective transformation), the relationship between two overlapping images $I(\mathbf{p})$ and $I'(\mathbf{p}')$ can be written by $\mathbf{p} \sim \mathbf{T}\mathbf{p}'$. $\mathbf{p} = [x\ y\ 1]^T$ and $\mathbf{p}' = [x'\ y'\ 1]^T$ are pixel locations in projective coordinates, and $\sim$ indicates equality up to scale since the 3×3 matrix $\mathbf{T}$ is invariant to scaling.

Global motion is estimated by aligning pair-wise adjacent frames assuming a geometric transformation as detailed in [29]. In our method, an affine model is assumed between consecutive images. We use the hierarchical motion estimation framework, where an image pyramid is first constructed in order to reduce the area of search by starting computation with the coarsest level [5], [31]. By applying the parameter estimation for every pair of adjacent frames, a global transformation chain is obtained.

Throughout this paper, we denote the discrete pixel locations in the image coordinate I_t as $\mathbf{p}_t = \{\mathbf{p}_t^i = (x^i, y^i)\}$. The subscript t indicates the index of the frame. We also denote the global transformation $\mathbf{T}_i^j$ to represent the coordinate transform from frame i to j. Therefore, the transformation of image I_t to the I_{t-1} coordinate can be described as $I_t(\mathbf{T}_t^{t-1}\mathbf{p}_t)$. Note that transformation $\mathbf{T}$ only describes the coordinate transform; hence, $I_{t-1}(\mathbf{T}_t^{t-1}\mathbf{p}_t)$ has the pixel values of frame $t-1$ in the coordinates of frame t, for instance.

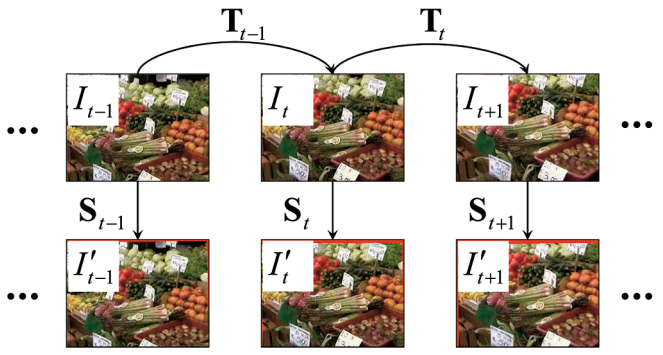


Fig. 3. Illustration of the global transformation chain $\mathbf{T}$ defined over the original video frames I_i, and the transformation $\mathbf{S}$ from the original path to the smoothed path. The bottom frame sequence is the motion compensated sequence.

2.2 Local Motion Estimation

Local motion describes the motion which deviates from the global motion model, e.g., motion of moving objects or image motion due to nonplanar scenes. Local motion is estimated by computing optical flow between frames after applying a global transformation, using only the common coverage areas between the frames.

A pyramidal version of Lucas-Kanade optical flow computation [32], [33] is applied to obtain the local motion field $\mathbf{F}_t^{t'}(\mathbf{p}_t) = [u(\mathbf{p}_t)v(\mathbf{p}_t)]^T$. $\mathbf{F}_t^{t'}(\mathbf{p}_t)$ represents the optical flow field from frame $I_t(\mathbf{p}_t)$ to $I_{t'}(\mathbf{T}_{t'}^t\mathbf{p}'_t)$, and u and v are the flow vector elements along the x and y-direction, respectively.

2.3 Removal of Undesired Motion

A stabilized motion path is obtained by removing undesired motion fluctuation. As assumed in [14], the intentional image motion in videos is usually slow and smooth; therefore, we treat the high frequency component in the global motion chain as the unintentional motion.

Previous motion smoothing methods smooth out the transformation chain itself or the cumulative transformation chain with an anchoring frame. Our method, on the other hand, smoothes *temporally local transformations* in order to accomplish motion smoothing.

When smoothing is applied to the original transformation chain $\mathbf{T}_0^1, \ldots, \mathbf{T}_{i-1}^i$ as it is done in prior works, the smoothed transformation chain $\tilde{\mathbf{T}}_0^1, \ldots, \tilde{\mathbf{T}}_{i-1}^i$ is obtained. In this case, a motion compensated frame I'_i is obtained by transforming I_i with $\prod_{n=0}^{i} \mathbf{T}_{n+1}^n \tilde{\mathbf{T}}_n^{n+1}$. This cascade of the original and smoothed transformation chain often generates accumulation error. In contrast, our method is free from accumulative error because our method locally smoothes displacement from the current frame to the neighboring frames.

Instead of smoothing out the transformation chain along the video, we directly compute the transformation $\mathbf{S}$ from a frame to the corresponding motion compensated frame using only the neighoring transformation matrices. We denote the indices of neighboring frames as $\mathcal{N}_t = \{j : t-k \le j \le t+k\}$. Let us assume that frame I_t is located at the origin of the image coordinate, aligned with the major axes.

We can calculate the spatial position of each neighboring frame I_s, relative to frame I_t, by the global transformation $\mathbf{T}_t^s$. We seek the correcting transformation $\mathbf{S}$ from the original frame I_t to the motion compensated frame I'_t by

$$\mathbf{S}_t = \sum_{i \in \mathcal{N}_t} \mathbf{T}_t^i \star G(k), \tag{1}$$

where $G(k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-k^2/2\sigma^2}$ is a Gaussian kernel, and $\star$ is the convolution operator, and $\sigma = \sqrt{k}$ is used. Using the obtained matrices $\mathbf{S}_0, \ldots, \mathbf{S}_t$, the original video frames can be warped to the motion compensated video frames (see Fig. 3) by

$$I'_t(\mathbf{p}'_t) \leftarrow I_t(\mathbf{S}_t\mathbf{p}_t). \tag{2}$$

Fig. 4 shows the result of our motion smoothing method with $k = 6$ in (1). In Fig. 4, x and y-translation elements of the camera motion path are displayed. As we can see in Fig. 4, abrupt displacements which are considered to be unwanted camera motion are well reduced by our motion smoothing. The smoothness of the new camera motion path can be controlled by changing k, with a larger k yielding a smoother result. We found that annoying high frequency motion is well removed by setting $k = 6$, i.e., about 0.5 *sec* with NTSC. k can be increased when a smoother video is preferred.

3 Image Deblurring

After stabilization, motion blur, which is not associated to the new motion of the video, becomes a noticeable noise that needs to be removed. As mentioned in Section 1, it is often difficult to obtain accurate PSFs from a free-motion camera; therefore, image deblurring using deconvolution is unsuitable for our case. In order to sharpen blurry frames without using PSFs, we developed a new interpolation-based deblurring method. The key idea of our method is transferring sharper image pixels from neighboring frames to corresponding blurry image pixels.

Our method first evaluates the "relative blurriness" of the image by measuring how much of the high frequency component has been removed from the frame in comparison to the neighboring frames. Image sharpness, which is the inverse of blurriness, has been long studied in the field of microscopic imaging where accurate focus is essential [34], [35]. We use the inverse of the *sum of squared gradient measure* to evaluate the relative blurriness because of its robustness to image alignment error and computational efficiency. By denoting two derivative filters along the x and y-directions by f_x and f_y, respectively, the blurriness measure is defined by

$$b_t = \frac{1}{\sum_{\mathbf{p}_t} \left\{ \left((f_x \star I_t)(\mathbf{p}_t)\right)^2 + \left((f_y \star I_t)(\mathbf{p}_t)\right)^2 \right\}}. \tag{3}$$

This blurriness measure does not give an absolute evaluation of image blurriness, but yields relative image blurriness among similar images when compared to the blurriness of other images. Therefore, we restrict the measure to be used in a limited number of neighboring frames where significant scene change is not observed. Also, the blurriness is computed using a common coverage area which is observed in all neighboring frames. Relatively blurry frames are determined by examining $b_t/b_{t'}, t' \in \mathcal{N}_t$, e.g., when $b_t/b_{t'}$ is larger than 1, frame $I_{t'}$ is considered to be sharper than frame I_t.

Once relative blurriness is determined, blurry frames are sharpened by transferring and interpolating corresponding pixels from sharper frames. To reduce reliance on pixels where a moving object is observed, a weight

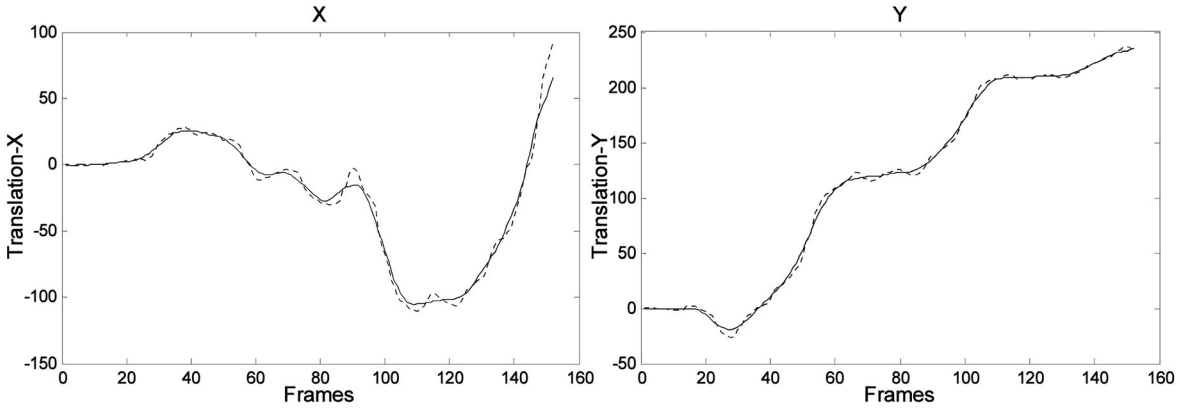


Fig. 4. Original motion path (dotted line) and smoothed motion path (solid line) with our displacement smoothing. Translations along X and Y direction are displayed.

factor which is computed by a pixel-wise alignment error $E_{t'}^{t}$ from $I_{t'}$ to I_t is used:

$$E_{t'}^{t}(\mathbf{p}_t) = \left| I_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t) - I_t(\mathbf{p}_t)) \right|. \tag{4}$$

High alignment error is caused by either moving objects or error in the global transformation. Using the inverse of pixel-wise alignment error E as a weight factor for the interpolation, blurry pixels are replaced by interpolating sharper pixels. The deblurring can be described by

$$\hat{I}_t(\mathbf{p}_t) = \frac{I_t(\mathbf{p}_t) + \sum_{t' \in \mathcal{N}} w_{t'}^{t}(\mathbf{p}_t) I_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t)}{1 + \sum_{t' \in \mathcal{N}} w_{t'}^{t}(\mathbf{p}_t)}, \tag{5}$$

where w is the weight factor which consists of the pixel-wise alignment error $E_{t'}^{t}$ and relative blurriness $b_t/b_{t'}$, expressed as

$$w_{t'}^{t}(\mathbf{p}_t) = \begin{cases} 0 & \text{if } \frac{b_t}{b_{t'}} < 1 \\ \frac{b_t}{b_{t'}} \frac{\alpha}{E_{t'}^{t}(\mathbf{p}_t) + \alpha} & \text{otherwise.} \end{cases} \tag{6}$$

$\alpha \in [0, \infty]$ controls the sensitivity on the alignment error, e.g., by increasing α, the alignment error contributes less to the weight. The weighting factor is defined in a way the interpolation uses only frames which are sharper than the current frame. This nonlinear operation approximates a temporal bilateral filter [36] which avoids oversmoothing caused by irrelevant image pixels.

Fig. 5 shows the result of our deblurring method. As we can see in Fig. 5, blurry frames in the top row are well sharpened in the bottom row. Note that since our method considers the pixel-wise alignment error, moving objects are well preserved without yielding ghost effects, which are often observed with simple frame interpolation methods.

4 Video Completion with Motion Inpainting

Our video completion method locally adjusts image pixels from neighboring frames using the local motion field in order to obtain seamless stitching of the images in the missing image areas. At the heart of our algorithm, *motion inpainting* is proposed to propagate the motion field into the missing image areas where local motion cannot be directly computed. The underlying assumption is that the local motion in the missing image areas is similar to that of adjoining image areas. The flow chart of the algorithm is illustrated in Fig. 6. First, the local motion from the neighboring frame is estimated over the common coverage image area. The local motion field is then propagated into missing image areas. Note that unlike prior image inpainting works, we do not propagate color but propagate local motion. Finally, the propagated local motion is used as a guide to locally warp image pixels to achieve smooth stitching of the images.

Let $\mathcal{M}_t$ be the missing pixels, or undefined image pixels, in the frame I_t. We wish to complete $\mathcal{M}_t$ for every frame t while maintaining visually plausible video quality.

4.1 Mosaicing with Consistency Constraint

As a first step of video completion, we attempt to cover the static and planar part of the missing image area by mosaicing with an evaluation of its validity. When the global transformation is correct and the scene in the missing image area is static and planar, mosaics generated by warping from different neighboring frames should be consistent with each other in the missing area. Therefore, it is possible to evaluate the validity of the mosaic by testing the consistency of the multiple mosaics which cover the same pixels. We use the variance of the mosaic pixel values to measure the consistency; when the variance is high, the mosaic is less reliable at the pixel. For each pixel $\mathbf{p}_t$ in the missing image area $\mathcal{M}_t$, the variance of the mosaic pixel values is evaluated by

$$v_t(\mathbf{p}_t) = \frac{1}{n-1} \sum_{t' \in \mathcal{N}_t} \left(I_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t) - \overline{I}_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t) \right)^2, \tag{7}$$

where

$$\overline{I}_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t) = \frac{1}{n} \sum_{t' \in \mathcal{N}_t} I_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t), \tag{8}$$

and n is the number of neighboring frames. For color images, we use the intensity value of the pixel which is computed by $0.30R + 0.59G + 0.11B$ [37]. A pixel $\mathbf{p}_t$ is filled in by the median of the warped pixels only when the computed variance is lower than a predefined threshold T:

$$I_t(\mathbf{p}_t) = \begin{cases} \text{median}_{t'}\left(I_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t)\right) & \text{if } v_t < T \\ \text{keep it as missing} & \text{otherwise.} \end{cases} \tag{9}$$

If all missing pixels $\mathcal{M}_t$ are filled with this mosaicing step, we can skip the following steps and move to the next frame.

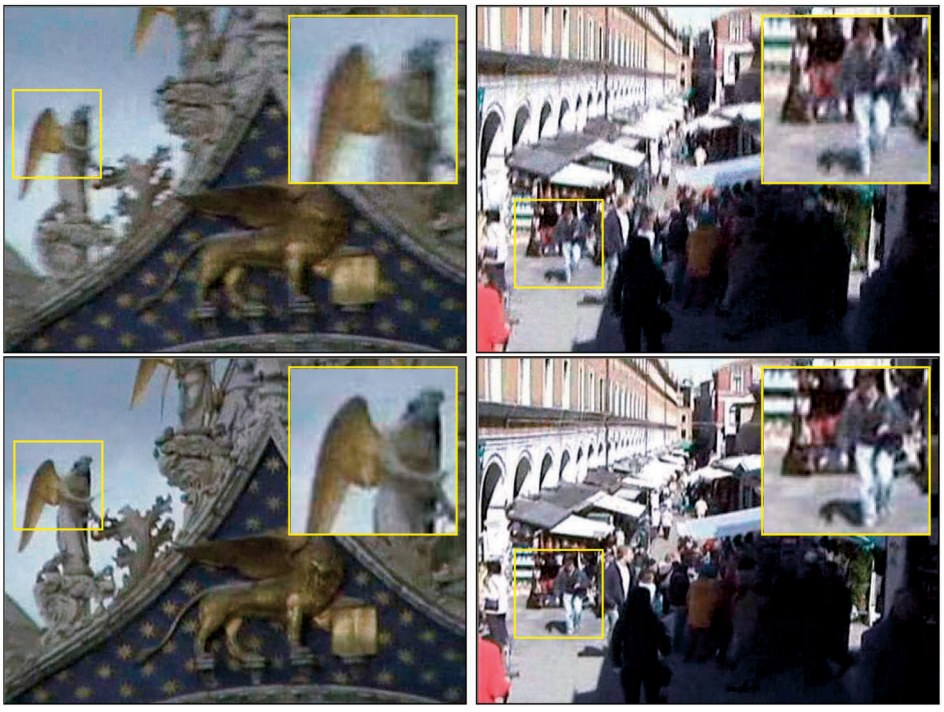


Fig. 5. The result of image deblurring. Top of the image pairs: original blurry images, and bottom: deblurred images with our method.

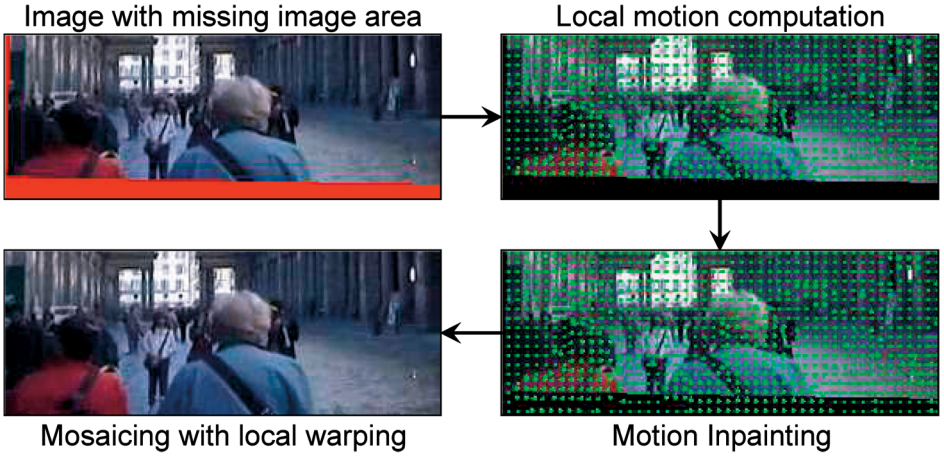


Fig. 6. Video completion. Local motion is first computed between the current frame and a neighboring frame. Computed local motion is then propagated with motion inpainting method. The propagated motion is finally used to locally adjust image pixels.

4.2 Local Motion Computation

From this step, each neighboring frame $I_{t'}$ is assigned a priority to be processed based on its alignment error. It is often observed that the nearer frame shows a smaller alignment error, and thus has a higher processing priority. The alignment error is computed using the common coverage area of $I_t(\mathbf{p}_t)$ and $I_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t)$ by

$$e_{t'}^{t} = \sum_{\mathbf{p}_t} \left| I_t(\mathbf{p}_t) - I_{t'}(\mathbf{T}_t^{t'}\mathbf{p}_t) \right|. \tag{10}$$

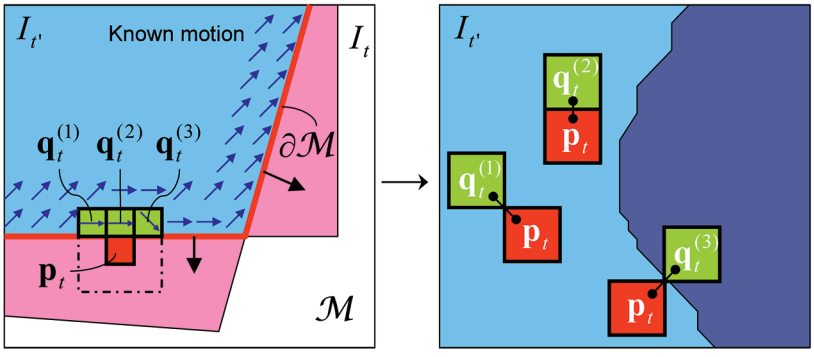
Local motion is estimated by the method described in Section 2.2.

4.3 Motion Inpainting

In this step, the local motion data in the known image areas is propagated into the missing image areas. The propagation starts at pixels on the boundary of the missing image area. Using motion values of neighboring known pixels, motion values on the boundary are defined, and the boundary gradually advances into the missing area $\mathcal{M}$ until it is completely filled as illustrated in Fig. 7.

Suppose $\mathbf{p}_t$ is a pixel in a missing area $\mathcal{M}$ on image I_t. Let $q_t \in \mathcal{H}(\mathbf{p}_t)$ be the pixels of the neighborhood of $\mathbf{p}_t$, that already has a defined motion value by either the initial local motion computation or prior extrapolation of motion data. With an assumption that the local motion variation is *locally* small, the local motion $\mathbf{F}(\mathbf{p}_t)$ can be written as the following equation using the local motion defined on neighboring pixel $\mathbf{q}_t$:

$$\begin{aligned} \mathbf{F}(\mathbf{p}_t;\mathbf{q}_t) &\approx \mathbf{F}(\mathbf{q}_t) + \begin{bmatrix} \frac{\partial F_1(\mathbf{q}_t)}{\partial x} & \frac{\partial F_1(\mathbf{q}_t)}{\partial y} \\ \frac{\partial F_2(\mathbf{q}_t)}{\partial x} & \frac{\partial F_2(\mathbf{q}_t)}{\partial y} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\ &= \mathbf{F}(\mathbf{q}_t) + \nabla\mathbf{F}(\mathbf{q}_t)(\mathbf{p}_t - \mathbf{q}_t), \end{aligned} \tag{11}$$



(a) (b)

Fig. 7. Motion inpainting. (a) Motion field is propagated on the advancing front $\partial\mathcal{M}$ into $\mathcal{M}$. (b) The color similarities between $\mathbf{p}_t$ and its neighbors $\mathbf{q}_t$ are measured in the neighboring frame $I_{t'}$ after warped by local motion of $\mathbf{q}_t$. The computed color similarities are then used as weight factors for the motion interpolation.

by the first order approximation of Taylor series expansion. In (11), we denote $\mathbf{F} = \mathbf{F}_t^{t'}$ and use F_1 and F_2 to represent the first and second (x-direction and y-direction) components in the motion vector, respectively. Also, we use $[x\ y]^T$ for the pixel coordinate of $\mathbf{q}_t$ and $[u\ v]^T$ to represent displacement from pixel $\mathbf{q}_t$ to $\mathbf{p}_t$, i.e., $[u\ v]^T = \mathbf{p}_t - \mathbf{q}_t$.

The motion value for pixel $\mathbf{p}_t$ is generated by a weighted average of the motion vectors of the pixels $\mathcal{H}(\mathbf{p}_t)$:

$$\mathbf{F}(\mathbf{p}_t) = \frac{\sum_{\mathbf{q}_t \in \mathcal{H}(\mathbf{p}_t)} w(\mathbf{p}_t, \mathbf{q}_t)\big(\mathbf{F}(\mathbf{q}_t) + \nabla\mathbf{F}(\mathbf{q}_t)(\mathbf{p}_t - \mathbf{q}_t)\big)}{\sum_{\mathbf{q}_t \in \mathcal{H}(\mathbf{p}_t)} w(\mathbf{p}_t, \mathbf{q}_t)}, \tag{12}$$

where the weighting factor $w(\mathbf{p}_t, \mathbf{q}_t)$ controls the contribution of the motion value of $\mathbf{q}_t \in \mathcal{H}(\mathbf{p}_t)$ to pixel $\mathbf{p}_t$.

The weighting factor $w(\cdot,\cdot)$ is designed to reflect two important factors: geometric distance and the *pseudo*similarity of colors between $\mathbf{p}$ and $\mathbf{q}$. The geometric distance controls the contribution to the new local motion vector, e.g., the nearer $\mathbf{q}$ governs the new local motion on $\mathbf{p}$ more. We define the geometric distance factor g by

$$g(\mathbf{p}_t, \mathbf{q}_t) = \frac{1}{||\mathbf{p}_t - \mathbf{q}_t||}, \tag{13}$$

which is evaluated on I_t image plane. The second factor, pseudosimilarity of colors, attempts to rely on the motion vector where its pixel color is similar to color value of the target pixel. The factor is considered as a measure for motion similarity, assuming that neighboring pixels of similar colors belong to the same object in the scene and, thus, they will likely move in a similar motion. Since the color of pixel $\mathbf{p}_t$ is unknown in frame I_t, we use the neighboring frame $I_{t'}$ for the estimation of pseudosimilarity of colors. As illustrated in Fig. 7, $\mathbf{q}_{t'}$ are first located in the neighboring image $I_{t'}$ in Fig. 7b using $\mathbf{q}_t$ and their local motion. Using the geometric relationship between $\mathbf{q}_t$ and $\mathbf{p}_t$ in Fig. 7a, $\mathbf{p}_{t'}$ are tentatively determined in $I_{t'}$. Using $\mathbf{p}_{t'}$ and $\mathbf{q}_{t'}$, we measure the pseudosimilarity of colors by

$$c(\mathbf{p}_t, \mathbf{q}_t) = \frac{1}{||I_{t'}(\mathbf{q}_{t'} + \mathbf{p}_t - \mathbf{q}_t) - I_{t'}(\mathbf{q}_{t'})|| + \epsilon}, \tag{14}$$

where ϵ is a small value for avoiding division by zero. When $\mathbf{q}_{t'}$ or $\mathbf{q}_{t'} + \mathbf{p}_t - \mathbf{q}_t$ also fall into the missing region, we leave $\vec{p}_t$ missing and proceed to the next missing pixel. Although the measure does not capture the *exact* color similarity between $\mathbf{p}_t$ and $\mathbf{q}_t$, the color similarity between $\mathbf{p}_{t'}$ and $\mathbf{q}_{t'}$ on image $I_{t'}$ gives an approximation. For this approximation, we rely on the fact that the local pixel arrangement does not vary dramatically in the small time frame. We are currently using the l^2-norm for the color difference in RGB space for the sake of computation speed, but different color spaces, richer measures or intensity difference could alternatively be used.

Using these two factors g and c, we define the weighting factor by their product:

$$w(\mathbf{p}_t, \mathbf{q}_t) = g(\mathbf{p}_t, \mathbf{q}_t)c(\mathbf{p}_t, \mathbf{q}_t). \tag{15}$$

To summarize, with the geometric factor g, the effect from distant pixels decreases. On the other hand, pseudosimilarity of colors c approximates anisotropic propagation of local motion using the color similarity measure on the neighboring frame $I_{t'}$.

The actual scanning and composition in the missing area $\mathcal{M}$ is achieved using the Fast Marching Method (FMM) [38] as described by [39] in the context of image inpainting. Let $\partial\mathcal{M}$ be the group of all boundary pixels of missing image area M (pixels which have a defined neighbor). Using FMM, we are able to visit each undefined pixel only once, starting with pixels of $\partial\mathcal{M}$, and advancing the boundary inside $\mathcal{M}$ until all undefined pixels are assigned motion values as shown in Fig. 7. The pixels are processed in ascending distance order from the initial boundary $\partial\mathcal{M}$, such that pixels close to the known area are filled first. The result of this process is a smooth extrapolation of the local motion flow to the undefined area in a manner that preserves object boundaries with color similarity measure.

4.4 Local Adjustment with Local Warping

Once the optical flow field in the missing image area $\mathcal{M}_t$ is obtained, we use it as a guide to locally warp $I_{t'}$ in order to generate a smooth stitching even including moving objects.

$$I_t(\mathbf{p}_t) \leftarrow I_{t'}(\mathbf{T}_t^{t'}(\mathbf{F}_t^{t'}\mathbf{p}_t)). \tag{16}$$

If some missing pixels still exist in I_t, the algorithm goes back to Step 1 and uses the next neighboring frame.

After the loop of Steps (a)-(c), most of the cases of missing pixels are filled; however, sometimes there still

Fig. 8. Result of video stabilization #1. Top row: Original input sequence. Middle row: stabilized sequence which still has missing image areas. Bottom row: stabilized and completed sequence. The grid is overlaid for better visualization.

remain missing image pixels which are not covered by warped neighboring images. In fact, the coverage of the missing image pixels is not guaranteed. Experimentally, even though such remaining areas exist, they are always small; therefore, we simply interpolate the neighboring pixels to fill up the areas. Richer methods such as nonparametric sampling [40], [15] or diffusion methods can also be used to produce higher quality completion than blurring, with additional computational cost.

4.5 Summary of the Algorithm

To summarize, we have the following algorithm for filling in the missing image area $\mathcal{M}$.

Goal: Let $\mathcal{M}_t$ be a set of missing image pixels in image I_t. Fill in $\mathcal{M}_t$ using the neighboring frames $\{I_{t-k}, \ldots, I_{t+k}\}$.
Mosaic with consistency checking
Compute $e^t_{t'}$ where $t' : t-k \leq t' \leq t+k$
While $\mathcal{M}_t \neq \mathbf{0}$
  for $t = \arg_{t'} \min(e^t_{t'})$ to $\arg_{t'} \max(e^t_{t'})$
    Compute local motion from I_t to I'_t
    Motion Inpainting to obtain full motion field $\mathbf{F}^{t'}_t$
    Fill in M_t by I'_t with local motion information $\mathbf{F}^{t'}_t$
  end of for-loop for t
end of while-loop for $\mathcal{M}_t$

At the heart of the algorithm, motion inpainting ensures spatial and temporal consistency of the stitched video.

5 Experimental Results

To evaluate the performance of the proposed method, we have conducted extensive experiments on 30 video clips (about 80 minutes in total) to cover different type of scenes. We set the number of neighboring frames to be $2k = 12$ throughout the experiment.

In the following, we first show the result of video completion in Section 5.1. Quantitative evaluation of the quality of resulting videos and computational performance analysis are also described in Section 5.2 and Section 5.3, respectively. We also show practical applications of our video completion method in Section 5.4.

5.1 Video Completion Results

In our experiment, a 5×5 size filter h is used to perform motion inpainting. In Fig. 8 and Fig. 9, the result of video stabilization and completion is shown. In the two figures, the top row shows the original input images, and the stabilized result is in the middle row which contains a significant amount of missing image areas. The missing image areas are naturally filled in with our video completion method as shown in the bottom row.

Fig. 10 shows a comparison result. Fig. 10a shows the result of our method, and the result of direct mosaicing is shown in Fig. 10b. As we can see clearly in Fig. 10b, the mosaicing result looks jaggy on the moving object (since multiple mosaics are used), while our result Fig. 10a looks more natural and smoother.

Fig. 11 shows our video completion results over different scenes. The top-left pair of images shows a successful result in a scene containing a fast moving plane. The top-right pair shows the case of a nonplanar scene, and in the left-bottom pair, an ocean wave is naturally composited with our method. Similar to Fig. 10a, our method accurately handles local motion caused by either moving objects or nonplanar scenes.

5.1.1 Failure Cases

Our method sometimes produces visible artifacts. Most of the failure cases are observed when the global and/or local motion estimation fails. Fig. 12 shows the failure example due to the incorrect estimation of the global motion. The filled-in image area of the completion result shown in Fig. 12b becomes skewed due to the inaccurate stitching. Another source of artifacts is the limited ability of the motion inpainting, i.e., it is not capable of producing abrupt changes of motion in the missing image areas. Fig. 13 shows such an example. In the figure, a boy swings down his arms, but this motion is not observed in the input image sequence (top row) due to the limited field-of-view. The local motion is computed over the input image sequence; however, it does not contain sufficient information to describe the arms' motion. As a result, the resulting images contain visible artifacts as shown in the middle row. The ground truth images are shown in the bottom row for the comparison.

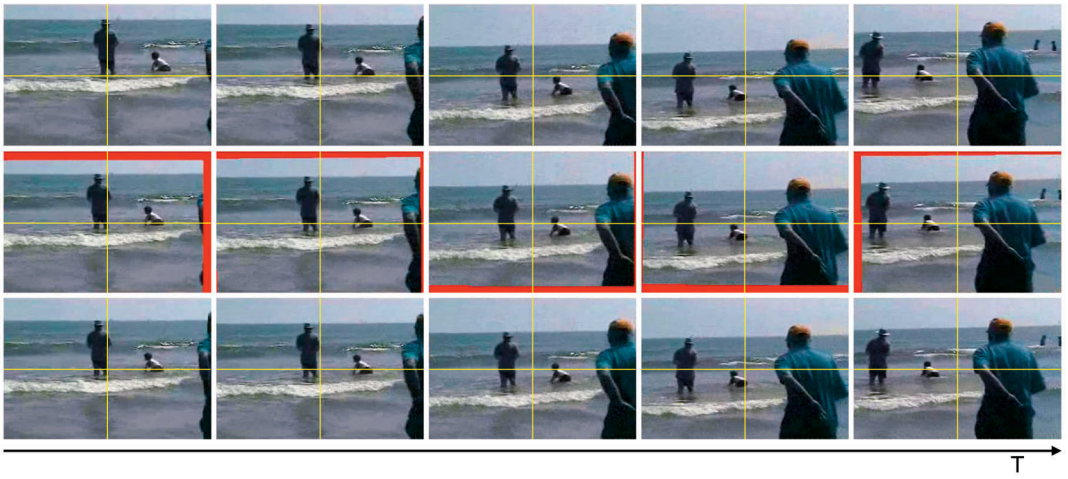


Fig. 9. Result of video stabilization #2. Top row: Original input sequence. Middle row: stabilized sequence which still has missing image areas. Bottom row: stabilized and completed sequence. The grid is overlaid for better visualization.

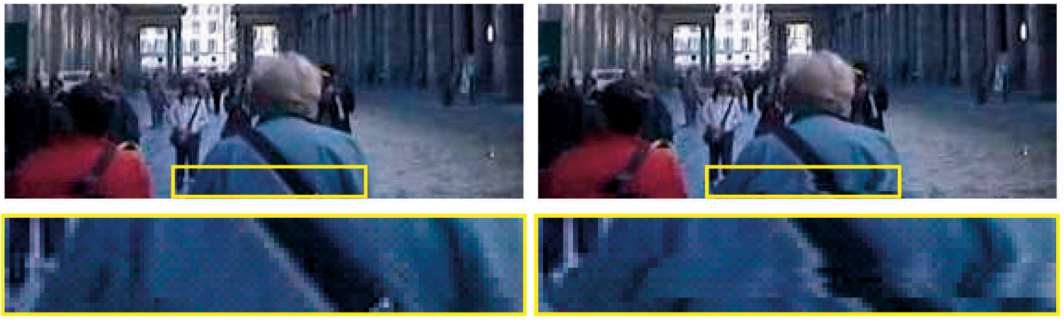


Fig. 10. Comparison of completion results. (a) Our method and (b) mosaicing. The rectangular areas in the images in the top row are closed up in the bottom row.

Fig. 11. Result of video completion over different types of scenes. In each pair, (a) is the stabilized image with missing image area (filled in by red), and (b) is the completion result.

5.2 Quantitative Evaluation of Video Completion

We have measured the quality of video completion in two different ways: 1) deviation from the ground truth and 2) evaluation of spatio-temporal smoothness. When the produced video is close to the ground truth, it is reasonable to say that the video is natural. The second evaluation is performed on the results which are not close to the ground truth, since they may still be "natural" although the deviation from the ground truth is large. The goal of video completion is generating visually natural videos, which is not necessarily equivalent to being close to the ground truth. We measure the "naturalness" by the spatio-temporal smoothness of the video.

5.2.1 Deviation from the Ground Truth

In order to make a comparison with the ground truth, we have cropped captured videos to produce smaller field-of-view videos and applied our video completion technique.

Fig. 12. Failure case #1. The filled-in area is skewed due to the inaccurate global/local motion estimation result. Dashed rectangle areas in the top row are closed-up in the bottom row. (a) Original. (b) Our method. (c) The ground truth.

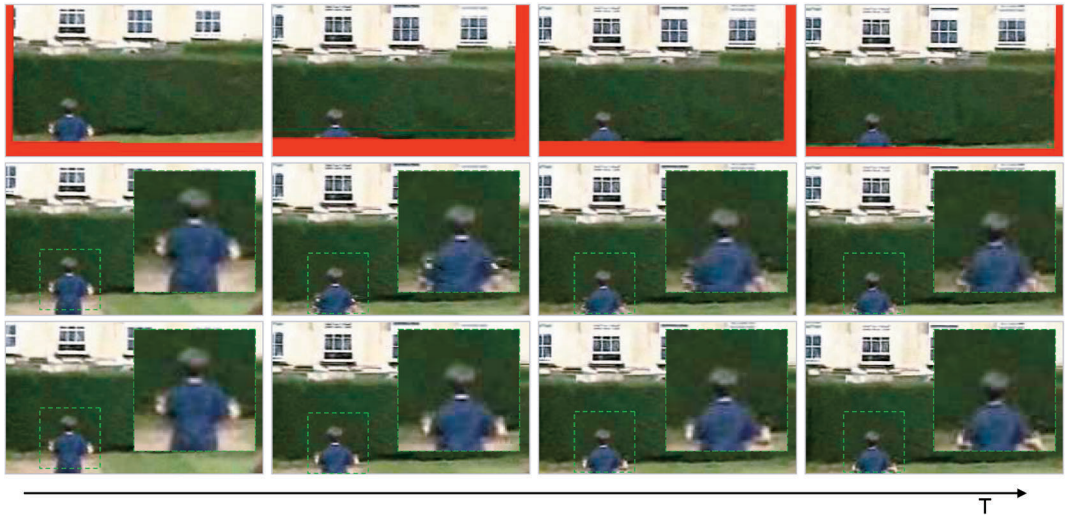


Fig. 13. Failure case #2. Top row: original input images with missing image areas. Middle row: result of video completion. Bottom row: the ground truth. The artifacts are caused by the fact that motion inpainting is not able to produce the sudden changes of local motion in missing image areas.

In this way, we are able to compare the intensity differences in the filled-in image areas between the produced video and the ground truth. Fig. 14 shows the comparison with the ground truth. In this experiment, five different video clips are chosen as shown in the figure. The first column shows the input images with missing image areas, and the second column shows the result of video completion. The ground truth is shown in the third column, and the right-most column shows the intensity difference between the resulting image and the ground truth. The intensity is calculated by $0.30R + 0.59G + 0.11B$ [37]. In Fig. 14, the images where moving objects appear at the boundaries of the missing image areas are selected in order to assess the performance of motion inpainting. Table 1 shows the mean absolute difference (MAD) of intensity compared to the ground truth. The mean is taken using image pixels of the filled-in image areas. As shown in the table, our method outperforms the simple mosaicing method significantly.

5.2.2 Evaluation of Spatio-Temporal Smoothness

Sometimes, the deviation from the ground truth becomes large while the resulting video still looks natural. In fact, the goal of the video completion is producing the visually natural videos, which is not necessarily the same as producing videos which are close to the ground truth. We consider that the naturalness of the video can be partly evaluated by the spatial and temporal consistency. The spatial consistency can be measured by how seamlessly missing image areas are filled in, while the temporal consistency can be evaluated by smoothness of pixel transitions between successive frames at temporal boundaries. If the spatial consistency is not achieved, a video frame may suffer from noticeable unnatural seams. And, if the temporal consistency is violated, temporal artifacts such as flickering may result in the video. We do not consider that the spatial and temporal continuity of the video can fully assess the quality of the resulting video. However, smooth transition of pixel values along spatial and temporal axes is one important aspect of the statistics of natural videos.

TABLE 1
Comparison with the Ground Truth

Mean absolute intensity difference	Our method	Mosaicing
Scene #1	9.87	12.2
Scene #2	4.18	7.83
Scene #3	7.64	8.27
Scene #4	6.65	9.14
Scene #5	10.5	23.6

The numbers show the mean absolute difference of intensity in the filled-in areas compared to the ground truth. Scene number corresponds to the scenes shown in Fig. 14 from top to bottom.

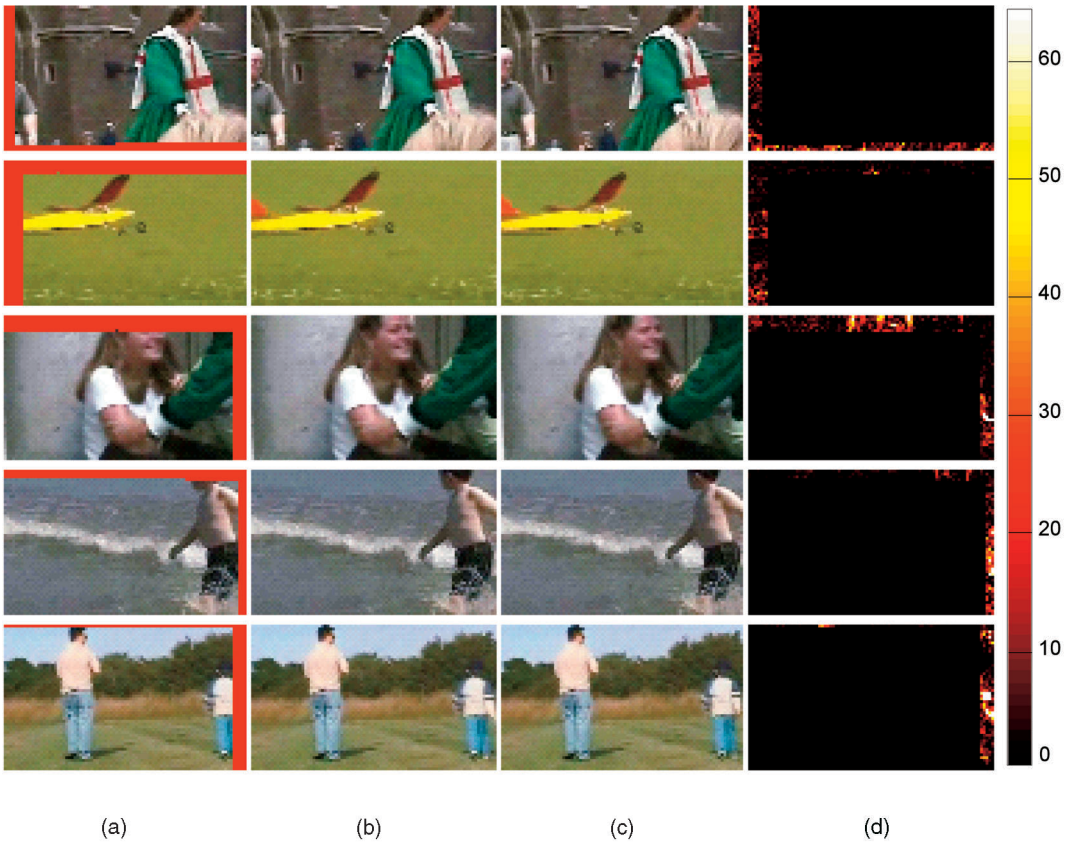


Fig. 14. Comparison with the ground truth. (a) Input images with missing image areas, (b) results of video completion with motion inpainting, (c) the ground truth, and (d) difference between the results and the ground truth (in intensity).

We use the magnitude of three-dimensional intensity gradients $||\nabla I|| = \sqrt{\nabla I \cdot \nabla I}$, where

$$\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial t} \end{bmatrix} \approx \begin{bmatrix} I(x+1,y,t) - I(x-1,y,t) \\ I(x,y+1,t) - I(x,y-1,t) \\ I(x,y,t+1) - I(x,y,t-1) \end{bmatrix}, \quad (17)$$

in order to measure the spatio-temporal smoothness. We define the normalized discontinuity measure D as the inverse of spatio-temporal smoothness that can be written as:

$$D = \sum_{i}^{n} ||\nabla I_i||/n, \quad (18)$$

where n is the number of pixels.

The discontinuity is measured on the spatio-temporal boundaries where images are stitched together. Since the discontinuity D does not provide absolute measure, we compared the discontinuity between our method and a simple mosaicing method. We denote the discontinuity of our method and mosaicing with D_O and D_M, respectively. In order to assess the quality difference, we pick up boundary pixels $\mathbf{p}$ which are in motion, i.e., where the local motion $\mathbf{F}_t^{t'}(\mathbf{p}) \neq [0\ 0]^T$. We directly compare the spatio-temporal smoothness in our result with that of the mosaicing result in order to see which provides smoother result using the average discontinuity D_A obtained from the surrounding image areas. The average discontinuity D_A is used as a standard discontinuity in the video. The relative smoothness is evaluated by

$$((D_M - D_A) - (D_O - D_A))/(D_M - D_A) = (D_M - D_O)/(D_M - D_A).$$

In this experiment, we have used seven different video clips (about 8,000 frames in total). The measured smoothness was generally higher using our method, and a 11.2 percent smoother result on average compared to the mosaicing method was obtained, ranging in $5.9 \sim 23.5\%$.

5.3 Computational Cost

In order to clarify the bottleneck of the computation for further development, we have measured the computational cost for each algorithm component.

The computational cost of our current research implementation is about 2.2 frames per second for a video in the size of 720×486 with a Pentium4 2.8 GHz CPU. Letting the number of frames in the video be N and the smoothness parameter be k, the computational cost of the algorithm blocks and the number of computations are summarized in Table 2. Note that the computational cost is measured without any hardware acceleration. As shown in Table 2, the computational time is proportional to the number of frames N and is also roughly proportional to the smoothness parameter k. Among the algorithm blocks, most of the computation time is spent on local motion estimation. Therefore, it is important to speed up the local motion estimation part for the realization of real-time implementa-

TABLE 2
Summary of Computational Cost

	Computational Cost (%)	Number of times
Global motion estimation	5.26%	N
Motion smoothing	0.05%	N (using $2k$ motions)
Local motion estimation	84.25 %	$2kN$
Motion inpainting	7.20%	$2kN$
Image warping	1.47 %	(2k+1)N for global warping, $2kN$ for local warping
Image deblurring	1.77%	N (using $2k+1$ images)

The computational time is proportional to the number of frames N. The percentage is measured when the smoothness $k = 6$. The number of times represents the count of operations of each computation block.

(a) (b)

Fig. 15. Sensor dust removal. (a) Spots on the camera lens are visible in the original video. (b) The spots are removed from the entire sequence by masking out the spot areas and applying our video completion method.

tion. Utilizing GPU power as it is done in [41], it will be possible to significantly improve the speed.

5.4 Other Video Enhancement Applications

In addition to video stabilization, the video completion and deblurring algorithms we developed in this paper can also be used in a range of other video enhancement applications. We show two interesting ones here: sensor dust removal from a video, caused by dirt spots on the video lens or broken CCD, and overlaid text/logo removal. They can be considered as a problem of filling up specific image areas which are marked as missing. This can be naturally applied to time-stamp removal from a video. In particular, when a stabilizing process is applied to a video, it is essential to remove these artifacts from the video since they become shaky in the final stabilized video. In this experiment, we manually marked artifacts as missing image areas. The missing image areas are then filled up by our video completion method.

Fig. 15 shows the result of sensor dust removal. Fig. 15a is a frame from the original sequence, and circles indicate the spots on the lens. The resulting video frames are free from these dirt spots as they are filled up naturally as shown in the right image. Fig. 16 shows the result of text removal from a video. The first row shows the original sequence, and some text is overlaid in the second row. Marking the text areas as missing image areas, our video completion method is applied. The bottom row shows the result of the completion. The result looks almost identical to the original images since the missing image areas are naturally filled up. The absolute intensity difference of the original and result images is taken in Fig. 17d. The result image is not identical to the original image; however, the difference is small, and more importantly, visual appearance is well preserved.

6 Discussion and Conclusion

We have proposed video completion and deblurring algorithms for generating full-frame stabilized videos. A new efficient completion algorithm based on motion inpainting is proposed. Motion inpainting propagates motion into missing image areas, and the propagated motion field is then used to seamlessly stitch images. We have also proposed a practical deblurring algorithm which transfers and interpolates sharper pixels of neighboring frames instead of estimating PSFs. The proposed completion method implicitly enforces *spatial* and *temporal* consistency supported by motion inpainting. Spatial smoothness of the image stitch is indirectly guaranteed by the smoothness of the extrapolated optical flow. Also, temporal consistency on both static and dynamic areas is given by optical flow from the neighboring frames. These properties make the resulting videos look natural and coherent.

6.1 Limitations

Our method strongly relies on the result of global motion estimation which may become unstable when a moving object covers large amounts of image area, for example. We are using a robust technique to eliminate outliers; however, it

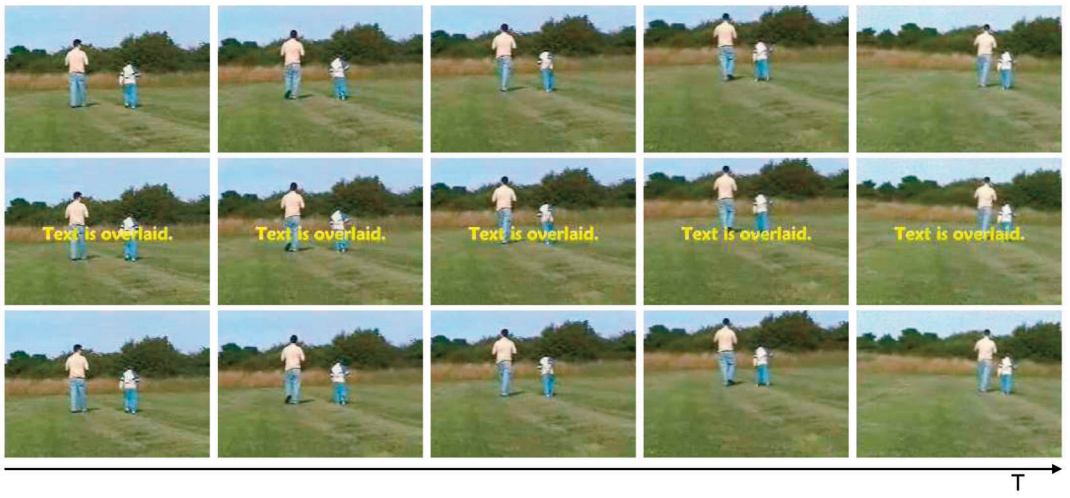


Fig. 16. Result of overlaid text removal from the entire sequence of a video. Top row: original image sequence. Middle row: the input with overlaid text. Bottom row: result of overlaid text removal.

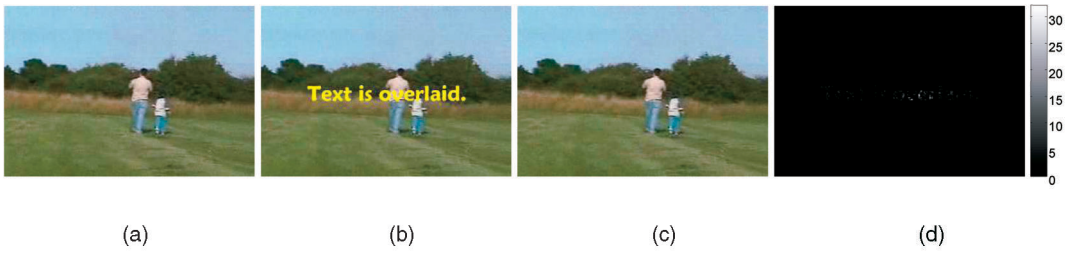


Fig. 17. Comparison of the ground truth and the text removal result. (a) A frame from the original video, (b) a text is overlaid on the original video, (c) result of text removal, and (d) absolute intensity difference between the original and result frame.

fails when more than half the area of the image is occluded by a moving object. Local motion estimation also has limitations, and may generate wrong results for very fast moving objects where the local motion estimation is difficult. In these cases, neighboring frames will not be warped correctly, and there will be visible artifacts at the boundary. To increase the smoothness at the boundaries, we will be able to use further methods, such as boundary matting [42] and graphcut-based synthesis [43], [44] upon our method.

The proposed method has been tested on a wide variety of video clips to verify its effectiveness. In addition, we have demonstrated the applicability of the proposed method for practical video enhancement by showing sensor dust removal and text removal results.

REFERENCES

[1] I. Zoghlami, O. Faugeras, and R. Deriche, "Using Geometric Corners to Build a 2D Mosaic from a Set of Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 420-425, 1997.

[2] D. Capel and A. Zisserman, "Automated Mosaicing with Super-Resolution Zoom," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 885-891, 1998.

[3] A. Cenci, A. Fusiello, and V. Roberto, "Image Stabilization by Feature Tracking," *Proc. 10th Int'l Conf. Image Analysis and Processing,* pp. 665-670, 1999.

[4] M. Brown and D. Lowe, "Recognizing Panoramas," *Proc. Int'l Conf. Computer Vision,* pp. 1218-1225, 2003.

[5] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *Proc. Second European Conf. Computer Vision,* pp. 237-252, 1992.

[6] R. Szeliski, "Image Mosaicing for Tele-Reality Applications," technical report, CRL-Digital Equipment Corp., 1994.

[7] T.-J. Cham and R. Cipolla, "A Statistical Framework for Longrange Feature Matching in Uncalibrated Image Mosaicing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 442-447, 1998.

[8] Z. Zhu, G. Xu, Y. Yang, and J. Jin, "Camera Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering," *Proc. IEEE Int'l Conf. Intelligent Vehicles,* vol. 2, pp. 329-334, 1998.

[9] R. Szeliski, "Image Alignment and Stitching: A Tutorial," Technical Report MSR-TR-2004-92, Microsoft Corp., 2004.

[10] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, "Real-Time Scene Stabilization and Mosaic Construction," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 54-62, 1994.

[11] C. Buehler, M. Bosse, and L. McMillan, "Non-Metric Image-Based Rendering for Video Stabilization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 609-614, 2001.

[12] Z. Duric and A. Resenfeld, "Shooting a Smooth Video with a Shaky Camera," *J. Machine Vision and Applications,* vol. 13, pp. 303-313, 2003.

[13] M. Pilu, "Video Stabilization as a Variational Problem and Numerical Solution with the Viterbi Method," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 625-630, 2004.

[14] A. Litvin, J. Konrad, and W. Karl, "Probabilistic Video Stabilization Using Kalman Filtering and Mosaicking," *Proc. IS&T/SPIE Symp. Electronic Imaging, Image, and Video Comm.,* pp. 663-674, 2003.

[15] Y. Wexler, E. Shechtman, and M. Irani, "Space-Time Video Completion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 120-127, 2004.

[16] J. Jia, T. Wu, Y. Tai, and C. Tang, "Video Repairing: Inference of Foreground and Background under Severe Occlusion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 364-371, 2004.

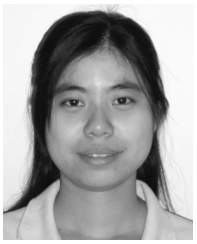
[17] V. Cheung, B.J. Frey, and N. Jojic, "Video Epitomes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 42-49, 2005.

[18] R. Fabian and D. Malah, "Robust Identification of Motion and Out-of-Focus Blur Parameters from Blurred and Noisy Images," *CVGIP: Graphical Models and Image Processing,* vol. 53, no. 5, pp. 403-412, 1991.

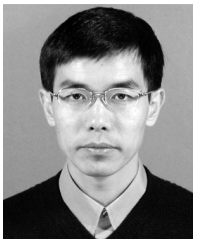
[19] B. Bascle, A. Blake, and A. Zisserman, "Motion Deblurring and Super-Resolution from an Image Sequence," *Proc. Fourth European Conf. Computer Vision,* vol. 2, pp. 573-582, 1996.
[20] P. Jansson, *Deconvolution of Image and Spectra,* second ed. Academic Press, 1997.
[21] A. Rav-Acha and S. Peleg, "Restoration of Multiple Images with Motion Blur in Different Directions," *Proc. Fifth IEEE Workshop Application of Computer Vision,* pp. 22-28, 2000.
[22] Y. Yitzhaky, G. Boshusha, Y. Levy, and N. Kopeika, "Restoration of an Image Degraded by Vibrations Using Only a Single Frame," *Optical Eng.,* vol. 39, no. 8, pp. 2083-2091, 2000.
[23] M. Ben-Ezra and S. Nayar, "Motion-Based Motion Deblurring," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 689-698, June 2004.
[24] E. Adelson, "Depth-of-Focus Image Processing Method," US Patent 4,661,986, 1987.
[25] J.R. Bergen, "Method and Apparatus for Extended Depth of Field Imaging," US Patent 6,201,899, 2001.
[26] M. Ben-Ezra, A. Zomet, and S.K. Nayar, "Video Super-Resolution Using Controlled Subpixel Detector Shifts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 6, pp. 977-987, June 2005.
[27] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image Inpainting," *Proc. SIGGRAPH Conf.,* pp. 417-424, 2000.
[28] A. Criminisi, P. Perez, and K. Toyama, "Object Removal by Exemplar-Based Inpainting," *IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 721-728, 2003.
[29] H.-Y. Shum and R. Szeliski, "Construction of Panoramic Mosaics with Global and Local Alignment," *Int'l J. Computer Vision,* vol. 36, no. 2, pp. 101-130, 2000.
[30] M. Irani and S. Peleg, "Improving Resolution by Image Registration," *CVGIP: Graphical Models and Image Processing,* pp. 231-239, 1991.
[31] P. Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *Int'l J. Computer Vision,* vol. 2, no. 3, pp. 283-310, 1989.
[32] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 674-679, 1981.
[33] J. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm," OpenCV Document, Intel, Microprocessor Research Labs, 2000.
[34] E. Krotkov, "Focusing," *Int'l J. Computer Vision,* vol. 1, no. 3, pp. 223-237, 1987.
[35] N. Zhang, M. Postek, R. Larrabee, A. Vladar, W. Keery, and S. Jones, "Image Sharpness Measurement in the Scanning Electron Microscope," *The J. Scanning Microscopies,* vol. 21, pp. 246-252, 1999.
[36] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. Int'l Conf. Computer Vision,* pp. 836-846, 1998.
[37] J. Foley, A. vanDam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice.* Addison-Wesley, 1996.
[38] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences.* Cambridge Univ. Press, 1996.
[39] A. Telea, "An Image Inpainting Technique Based on the Fast Marching Method," *J. Graphics Tools,* vol. 9, no. 1, pp. 23-34, 2004.
[40] A. Efros and T. Leung, "Texture Synthesis by Nonparametric Sampling," *Proc. Int'l Conf. Computer Vision,* pp. 1033-1038, 1999.
[41] R. Strzodka and C. Garbe, "Real-Time Motion Estimation and Visualization on Graphics Cards," *Proc. IEEE Visualization Conf.,* pp. 545-552, 2004.
[42] C. Rother, V. Kolmogorov, and A. Blake, "'Grabcut': Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 309-314, 2004.
[43] Y. Boykov and M. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proc. Int'l Conf. Computer Vision,* pp. 105-112, 2001.
[44] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A.F. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 277-286, 2003.

Yasuyuki Matsushita received the BEng, MEng, and PhD degrees in electrical engineering from the University of Tokyo in 1998, 2000, and 2003, respectively. Currently, he is a researcher in Visual Computing Group at Microsoft Research Asia. His research interests include photometric methods in computer vision and video improvement algorithms. He is a member of IEEE.

Eyal Ofek received the BS degree in mathematics, physics, and computer science in 1987, the MS degree in computer science in 1992, and the PhD degree in computer science in 2000, all from the Hebrew University. He is a researcher at Microsoft Virtual earth, previously at Microsoft Research Asia. His interests are in IBR, vision-based interaction, and rendering. He is a member of the IEEE.



Weina Ge received the BS degree in computer science from Zhejiang University in 2005. She is currently a PhD candidate and research assistant in the Computer Science and Engineering Department at Penn State University. Her research interests include computer vision, image retrieval, and data mining.



Xiaoou Tang received the BS degree from the University of Science and Technology of China, Hefei, in 1990, the MS degree from the University of Rochester, New York, in 1991, and the PhD degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He was a professor and the director of Multimedia Lab in the Department of Information Engineering, the Chinese University of Hong Kong until 2005. Currently, he is the group manager of the Visual Computing Group at Microsoft Research Asia. He is a local chair of the IEEE International Conference on Computer Vision (ICCV) 2005, an area chair of ICCV07, a program chair of ICCV09, and a general chair of the IEEE ICCV International Workshop on Analysis and Modeling of Faces and Gestures 2005. He is a guest editor of the special issue on underwater image and video processing for the *IEEE Journal on Oceanic Engineering* and the special issue on image and video-based biometrics for *IEEE Transactions on Circuits and Systems for Video Technology*. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*). His research interests include computer vision, pattern recognition, and video processing. He is a senior member of the IEEE.

Heung-Yeung Shum received the PhD degree in robotics from the School of Computer Science, Carnegie Mellon University in 1996. He worked as a researcher for three years in the Vision Technology Group at Microsoft Research Redmond. In 1999, he moved to Microsoft Research Asia where he is currently the managing director. His research interests include computer vision, computer graphics, human computer interaction, pattern recognition, statistical learning, and robotics. He is on the editorial boards of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*), the *International Journal on Computer Vision* (*IJCV*), and *Graphical Models*. He was the general cochair of the 10th International Conference on Computer Vision (ICCV 2005, Beijing). He is a fellow of the IEEE.