IS&T/SPIE Symposium on Electronic Imaging, Image and Video Communications and Proc., Jan. 20-24, 2003, Santa Clara, CA, USA

Probabilistic video stabilization using Kalman filtering and mosaicking

Andrew Litvin, Janusz Konrad, William C. Karl
ECE department, Boston University, 8 St. Mary's Street, Boston, MA, 02215

ABSTRACT

The removal of unwanted, parasitic vibrations in a video sequence induced by camera motion is an essential part of video acquisition in industrial, military and consumer applications. In this paper, we present a new image processing method to remove such vibrations and reconstruct a video sequence void of sudden camera movements. Our approach to separating unwanted vibrations from intentional camera motion is based on a probabilistic estimation framework. We treat estimated parameters of interframe camera motion as noisy observations of the intentional camera motion parameters. We construct a physics-based state-space model of these interframe motion parameters and use recursive Kalman filtering to perform stabilized camera position estimation. A six-parameter affine model is used to describe the interframe transformation, allowing quite accurate description of typical scene changes due to camera motion. The model parameters are estimated using a *p*-norm-based multi-resolution approach. This approach is robust to model mismatch and to object motion within the scene (which are treated as outliers). We use mosaicking in order to reconstruct undefined areas that result from motion compensation applied to each video frame. Registration between distant frames is performed efficiently by cascading interframe affine transformation parameters. We compare our method's performance with that of a commercial product on real-life video sequences, and show a significant improvement in stabilization quality for our method.

Keywords: video stabilization, mosaicking, multiresolution registration

1. INTRODUCTION

In the last decade the use of video acquisition devices increased dramatically due to the decrease in the cost of such devices and dramatic improvements in computer performance, making storage and processing of large data volumes possible. Video cameras became extremely popular in consumer market and their use in industry and the military is quickly growing. Unwanted, parasitic vibrations in video sequences, which are inherent to a handheld and mobile video acquisition device, harm the performance and value of such devices significantly. Both hardware and image processing approaches to video stabilization have been developed. The first approach, called optical stabilization, consists of implementing an optical system that compensates for unwanted camera motion using motion sensors and active optical system. This approach is potentially the most powerful, but makes video cameras significantly more expensive and, therefore, is not chosen for a broad class of devices.

The second approach, which is a focus of this paper, consists in performing post-processing of the video sequence to eliminate unwanted motion in the video (swings and twists) caused by a person holding the camera or mechanical vibration. In this paper we follow the general stabilization framework. In the absence of blurring, unwanted movements of the camera do not modify the individual frame content, but only shift and rotate the image. Therefore, the stabilization must be carried out in the domain of global motion description, such as the set of geometric transform parameters of the frame with respect to an anchor frame. In order to compensate for unwanted motion, intentional component of the transformation between frames has to be estimated. Intentional motion, such as zooming the image, panning, translational or dolly motion with respect to the scene, is slow and smooth compared with unwanted, parasitic camera movements. For this reason, recovering the intentional motion parameters from the motion parameters of the sequence is a low-pass filtering operation in nature. After the intentional motion parameters are estimated, by "subtracting" them from observed motion parameters, the unwanted, noisy component can be obtained and used to transform (warp) the frames. A correcting transform applied to each frame inevitably renders undefined certain regions around edges, causing artifacts. We now consider the steps of the algorithm in greater detail, and we position it with respect to the literature.

At the first step of stabilization, modeling scene change due to camera is performed. Estimating a full 3D model of the scene including depth, while desirable, generally leads to ill-posed, complex problem that forms a field of research on its own. Consequently, we use a 2D global motion model, which is commonly done. Two-parameter translation model is used in [8, 10], four parameter 2D rigid motion model is used in [6], and a six parameter affine model is used in [3, 6]. An analysis of possible motion patterns is carried out in [9], leading to a 2.5D motion model that requires user interaction to choose the dominant character of the motion in the analyzed video sequence. We seek an automated technique, making an affine motion model a good choice. The affine model describes accurately pure rotation, panning, and small translations of the camera in a scene with small relative depth variations and zooming effects. For many outdoor and indoor scenes the above conditions are satisfied, and any errors resulting from model mismatch are attenuated by the proper choice of a cost function used in registration (finding transformation between frames). Several techniques to estimate this interframe transformation have been proposed. Phase correlation was used in [11] and feature tracking in [12]. The most precise approach is to minimize a global cost function constructed using the information from frames being registered. Such a matching cost function can be constructed from characteristic features extracted from the frames [6, 11, 12] or on image intensities [11]. In order to have a tractable registration procedure we use a cost function defined on image intensities. Multiscale optimization of the cost function coupled with gradient descent is used.

Several approaches have been proposed to estimate intentional motion parameters once image transformation caused by motion is modeled. In [2, 7, 12], under the assumption of a static scene and absence of the intentional motion, a mosaic is constructed from video frames and displayed as a stabilized output. However these assumptions do not hold for most real life sequences. Camera motion is assumed to be linear between turn points in [4], leading to jumps in stabilized video. Kinetic camera motion model requiring precise knowledge of the mechanical system is used in [5]. Inertial model of the intentional camera motion is used in [9], effectively low-pass filtering motion parameters. This approach is based on a physical model, which is not realizable in practice, and filter design is carried out in ad-hoc manner. To this end, we propose a probabilistic and physics-based approach to estimate intentional motion parameters. By analyzing the statistical nature of intentional camera motion, we describe intentional motion parameters as functional random processes using physically meaningful state-space model. Unwanted movements are described as random noise processes. Kalman filtering framework is used to obtain optimal estimates of the intentional motion parameters. Tradeoffs between estimator performance and system storage requirements can be achieved by choosing different Kalman filtering.

After the correcting transform is applied to each frame, some regions of the video frames become undefined, resulting in visual degradation of frames. No efficient method to deal with this problem has been reported in the literature, while image truncation and zooming are used in commercial software products. In these methods, edges in all frames are trimmed in such a way that no undefined pixels appear in any frame while preserving rectangular area of fixed size, followed by magnification to keep the original size of the frame. This process causes loss of information, resolution degradation and limits the range of possible correcting transformation due to the fact that large transformations require large cuts and significant scaling. To this end we propose to use mosaicking. We fill undefined regions by pixel values from neighboring frames using the fact that nearby frames have similar content when properly aligned. Although constructing a mosaic from all frames in the stabilized video assuming stationary scene has been performed in [2, 7, 12], mosaicking was not used to perform reconstruction of the video frames in the process of video stabilization.

The primary contributions of this paper are:

1) constructing physics-based state-space model to describe the dynamics of intentional and unwanted motion parameters, and Kalman filtering estimation framework to perform intentional motion parameters estimation
2) using mosaicking to fill undefined regions in the stabilized frames

Both of these ideas, to the best of our knowledge, have not been applied to the video stabilization problem thus far. We present an integrated, end-to-end approach to performing stabilization. We report a comparison of our method's performance with existing products' performance on real-life video sequences showing significantly better results obtained by the proposed method. We show that our mosaicking technique gives excellent results, making truncation and zooming obsolete. Our paper is organized as follows. In Section 2 we give detailed description of our approach, and some of our results are presented in Section 3. We refer the reader to the project web page for all our results. Section 4 concludes this paper.

2. VIDEO STABILIZATION AND RECONSTRUCTION FRAMEWORK

The challenge of compensation for unwanted, parasitic motion in a video sequence is rooted in the difficulty of separating unwanted motion between video frames from the motion inherent to the video sequence. Existing approaches to video stabilization concentrate on efficiently registering the frames, but with respect to compensation for unwanted motion they use either overly simple assumptions about camera motion or excessively complicated physics-based models matched to a particular application. Another difficulty is that the transformations applied to the video frames in order to compensate for unwanted motion create undefined image regions. Frame cropping and magnification are common tools to recover intensity in those regions and maintain rectangular frame shape and original size.

In this paper we propose a new, integrated approach to solve the problem of video stabilization. A solution to both of above-mentioned issues constitutes two main contributions of this paper:

1. We propose a probabilistic approach to model interframe motion parameters. The proposed dynamic motion model realistically describes interframe motion caused by camera movements. By using results from recursive estimation theory, we achieve optimal estimation of the intentional image transformation parameters. Our approach is flexible and allows the inclusion of available prior information about the intentional camera motion and characteristics of unwanted camera motion through modifications of the state-space model.
2. Mosaicking is used to reconstruct undefined regions in each warped frame using information from neighboring frames, thus exploiting time correlations between neighboring frames in the video. The stabilization and undefined regions reconstruction steps of the algorithm are integrated by reusing estimates of transformation between adjacent frames.

The overall algorithm consists of the following steps:

1. Video sequence stabilization (unwanted motion compensation)
    1) Estimation of the pair-wise transformations between adjacent frames
    2) Estimation of the intentional motion parameters (Kalman filtering in time).
    3) Compensation of each frame for unwanted motion (frame warping)
2. Reconstruction of undefined regions using mosaicking
    1) Estimation of the transformation between distant frames
    2) Warping distant frames and constructing mosaic for undefined regions in each frame

The block diagram of our overall approach is shown in Figure 1 followed by the detailed description of our algorithm.

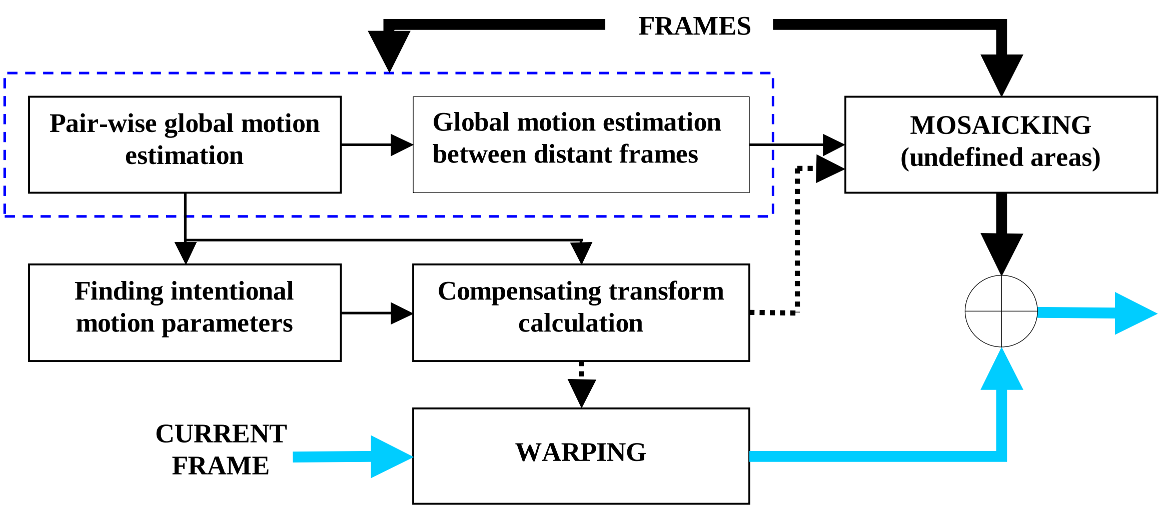

Figure 1: Video stabilization algorithm. Flow of frames (intensities) is shown by thick arrows.

2.1 Video sequence stabilization (unwanted motion compensation)

At this stage, intentional motion parameters in the video are estimated and subtracted from the motion parameters derived from the video sequence, yielding unwanted motion component, which is compensated for in the video sequence by applying proper transformation to each frame.

2.1.1 Estimation of the transformation between adjacent frames

The first part of the algorithm consists of estimating transformations between adjacent frames in order to bring them into alignment. These transformations are needed to model the scene changes due to motion. We use the common approach of describing the motion by a set of parameters assuming that the observed scene undergoes a geometrical transformation between frames. Only static, planar scene transformation due to camera motion can be described accurately by such a geometric transform. In real case, background usually has small relative depth variation, and therefore its transformation between frames in the video can be approximated by such a geometric transform. We use an affine transformation model as a balance between model complexity and its descriptive capability discussed in the first section of this paper. Scene change and mismatch caused by depth changes in the scene are taken care of by combining this affine model with a robust cost function. Under an affine transformation, pixel locations $\boldsymbol{x}=(x,y)$ in frames I^n and I^{n+m} are related by a transformation $(\mathbf{A}_n^m, \boldsymbol{b}_n^m)$ given by

$$\boldsymbol{x}_{n+m} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{A}_n^m \boldsymbol{x}_n + \boldsymbol{b}_n^m \tag{1}$$

where $\boldsymbol{x}_n$ and $\boldsymbol{x}_{n+m}$ are pixel coordinates before and after transformation respectively. Transform $(\mathbf{A}_n^1, \boldsymbol{b}_n^1)$, aligning frames I^n and I^{n+1}, is estimated by minimizing the following cost function with respect to $(\mathbf{A}_n^1, \boldsymbol{b}_n^1)$:

$$E(\mathrm{I}^n, \mathrm{I}^{n+m}, \mathbf{A}_n^m, \boldsymbol{b}_n^m) = \sum_{x \in \chi} \varphi\left(\mathrm{I}^n(\boldsymbol{x}) - \mathrm{I}^{n+m}(\mathbf{A}_n^m \boldsymbol{x} + \boldsymbol{b}_n^m)\right) \tag{2}$$

where $m=1$ and χ is the set of all locations in the image plane for which transformed coordinates lie in the limits of the valid image coordinates. Cost function (2) is minimized by gradient descent using analytical expressions (not given here) for partial derivatives with respect to the parameters of the transformation. The choice of function $\varphi(x)$ is crucial for robustness of the transformation estimate to the motion of objects in the scene and to mismatch of the affine model due to depth variability. An error in estimating transformation parameters between two frames causes incorrect compensating transformations to be applied to a sequence of frames, resulting in annoying jumps in the processed video. Many such non-quadratic, slowly rising or redescending functions for $\varphi(x)$ have been proposed. Here we use an approximation to the l_p-norm given by

$$\varphi(x) = (x^2 + \beta)^{p/2} \tag{3}$$

with $\beta = 0.01$. The non-zero β insures differentiability of the cost function near zero, needed for proper functioning of the gradient descent approach. In our experiments we use $p=1$ chosen empirically from several test sequences by minimizing occurrences of wrong estimated transformation parameters. In order to avoid local minima of the cost function (2) and accelerate the convergence we use a multiscale implementation. The result at a coarser scale is used to initialize a solution at the next finer scale. At scales 8,4 and 2, frames are low-pass filtered and subsampled by a factor of 8,4 and 2, respectively, before applying gradient descent approach to minimize (2). Solution for each scale serves as initialization for the next finer scale.

2.1.2 Estimating intentional motion parameters

In order to compensate the video sequence for transformation caused by unwanted camera movements, transformations caused by intentional motion of the camera should be identified using interframe motion parameters estimated at the previous stage of the algorithm and prior information on motion dynamics. Instead of modeling motion of the camera we introduce a closely related parameterization of the ongoing image transformation in the natural form of "cumulative" transform defined as registration of the current frame with respect to the first frame, obtained by cascading individual interframe transformations. The cumulative transform for frame n, denoted by $(\tilde{\mathbf{A}}_n, \tilde{\boldsymbol{b}}_n)$, can be obtained as follows:

$$\boldsymbol{x}_n = \mathbf{A}_n \boldsymbol{x}_{n-1} + \boldsymbol{b}_n = \ldots = \left(\prod_{k=n}^{1} \mathbf{A}_k\right) \boldsymbol{x}_1 + \sum_{k=1}^{n} \left(\prod_{m=n}^{k+1} \mathbf{A}_m\right) \boldsymbol{b}_k = \tilde{\mathbf{A}}_n \boldsymbol{x}_1 + \tilde{\boldsymbol{b}}_n \tag{4}$$

Note that the cumulative transform parameters do not necessarily describe a transform allowing the reconstruction of a current frame from the first frame because of the ongoing scene changes and accumulation of errors, but rather to represent the continuously changing parameters whose "increments" describe image transformation between adjacent frames. Elements of matrix $\mathbf{A}$ describe zoom, rotation and dolly motion of the camera, and vector $\boldsymbol{b}$ describes panning and tracking motion. Similarly, we describe the image transform parameters representing intentional motion in terms of intentional cumulative transform $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$. The difference between $(\tilde{\mathbf{A}}_n, \tilde{\boldsymbol{b}}_n)$ and $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$ is attributed to unwanted, parasitic camera motion.

Optimal estimation of $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$ is carried out using a recursive Kalman filtering algorithm. We treat $(\tilde{\mathbf{A}}_n, \tilde{\boldsymbol{b}}_n)$ as noisy observations of intentional cumulative transform parameters obeying physics-based dynamic model. The noise component in observed cumulative motion parameters is attributed to the unwanted camera motion.

The state model for each of the parameters $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$ depends on the real-life expected behavior of these parameters. Two distinct behavior patterns can be identified leading to different dynamic models for different parameters.

1. Diagonal values $\hat{a}_1$ and $\hat{a}_4$ are primarily responsible for zooming and dolly motion. Both types of motion usually take place with constant velocity that is subject to random perturbations. The same reasoning applies to parameters $\hat{b}_1$ and $\hat{b}_2$ that describe translation. Translation is typically caused by camera panning or tracking motion which is likely to take place with constant velocity. The common approach to build a dynamic model for a variable θ changing at a nearly constant rate is to introduce velocity variable θ^v that is constant subject to random perturbations. An increment of the variable at time t, $\delta\theta(t)$ is equal to the value of the velocity variable $\theta^v(t)$. We introduce velocity variables $\hat{a}_1^v, \hat{a}_4^v, \hat{b}_1^v, \hat{b}_2^v$ for each of $\hat{a}_1, \hat{a}_4, \hat{b}_1, \hat{b}_2$, respectively. It is reasonable to assume the independence of dynamic models for each of the 4 parameters. For example, parameters $\hat{a}_1$ and $\hat{a}_1^v$ follow the dynamic model given by
$$\begin{pmatrix} \hat{a}_1 \\ \hat{a}_1^v \end{pmatrix}^{t+1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{a}_1 \\ \hat{a}_1^v \end{pmatrix}^{t} + \begin{pmatrix} 0 \\ N(0,\sigma_z) \end{pmatrix} \tag{5}$$
where $N(0,\sigma_z)$ is white Gaussian noise with variance σ_z.
2. The remaining parameters $\hat{a}_2$ and $\hat{a}_3$ primarily describe rotation of the image caused by camera tilts. We can assume these parameters to be constant in the absence of noise, where perturbations themselves are random. This leads to a simple dynamic model for $\hat{a}_2$ and $\hat{a}_3$. For example, for $\hat{a}_2$, we have
$$\left(\hat{a}_2\right)^{t+1} = \left(\hat{a}_2\right)^{t} + N(0,\sigma_r) \tag{6}$$

The overall state-space model for the intentional cumulative transform parameters is given by
$$\begin{pmatrix} \hat{a}_1 \\ \hat{a}_1^v \\ \hat{a}_4 \\ \hat{a}_4^v \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{b}_1 \\ \hat{b}_1^v \\ \hat{b}_2 \\ \hat{b}_2^v \end{pmatrix}^{t+1} = \begin{pmatrix} 1 & 1 & & & & \cdots & & & & 0 \\ & 1 & & & & & & & & \\ & & 1 & 1 & & & & & & \\ & & & 1 & & & & & & \\ \vdots & & & & 1 & & & & & \vdots \\ & & & & & 1 & & & & \\ & & & & & & 1 & 1 & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1 & 1 \\ 0 & & & & & \cdots & & & & 1 \end{pmatrix} \begin{pmatrix} \hat{a}_1 \\ \hat{a}_1^v \\ \hat{a}_4 \\ \hat{a}_4^v \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{b}_1 \\ \hat{b}_1^v \\ \hat{b}_2 \\ \hat{b}_2^v \end{pmatrix}^{t} + \begin{pmatrix} 0 \\ N(0,\sigma_z) \\ 0 \\ N(0,\sigma_z) \\ N(0,\sigma_r) \\ N(0,\sigma_r) \\ 0 \\ N(0,\sigma_b) \\ 0 \\ N(0,\sigma_b) \end{pmatrix} \tag{7}$$

In our implementation we make no assumption as to any coordination between different modes of camera motion (panning in different directions or panning and zoom), that leads to adapting the independence assumption for noise in different state-space model variables. Variance of the noise terms is different for each kind of variable: (σ_z, σ_r and σ_b). The observed cumulative transform parameters $(\tilde{\mathbf{A}}_n, \tilde{\boldsymbol{b}}_n)$ are treated as noisy observations of the intentional cumulative transform parameters. The velocity variables $\hat{a}_1^v, \hat{a}_4^v, \hat{b}_1^v, \hat{b}_2^v$ are auxiliary variables which are not observed. The observation model for each parameter is independent, leading to observation model

$$\begin{pmatrix} \tilde{a}_1 \\ \tilde{a}_2 \\ \tilde{a}_3 \\ \tilde{a}_4 \\ \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix} = \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \\ \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} + \begin{pmatrix} N(0,\sigma_z^{obs}) \\ N(0,\sigma_r^{obs}) \\ N(0,\sigma_r^{obs}) \\ N(0,\sigma_z^{obs}) \\ N(0,\sigma_b^{obs}) \\ N(0,\sigma_b^{obs}) \end{pmatrix} \tag{8}$$

Here we also assume independence of the white Gaussian observation noise for each variable in Eq. (8). We discuss more realistic assumptions in the last section of this paper. Variances of noise processes $\sigma_z, \sigma_r, \sigma_b$ in Eq. (7) are determined by the desired degree of smoothness of the intentional camera motion. Increasing variance for a given parameter means increasing variability of the corresponding state variable, leading to more random estimate for the corresponding intentional state variable, and a more jerky stabilized video. In the limiting case of zero variance, the corresponding parameter is assumed to be constant, leading to a stabilized video with no allowed motion. We call this case "total compensation" (of motion). Observation noise variances $\sigma_z^{obs}, \sigma_r^{obs}, \sigma_b^{obs}$ in Eq. (8) describe the variability of unwanted transformations between frames. The effect of these values on the resulting estimates $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$ is opposite to the effect of $\sigma_z, \sigma_r, \sigma_b$. In the limiting case of zero observation noise, parameters $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$ are set equal to $(\tilde{\mathbf{A}}_n, \tilde{\boldsymbol{b}}_n)$ resulting in the identity compensating transforms and no stabilization. In practice all state model and observation noise variances are set empirically to achieve acceptable stabilization results. In our experiments this is done without actually performing the stabilization, but rather by looking at the filtered sequences of $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$ trading off smoothness and fidelity of estimates to the observations (large differences between $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$ and $(\tilde{\mathbf{A}}_n, \tilde{\boldsymbol{b}}_n)$ lead to significant compensating transformations and possible visual artifacts). A possibility of an automatic selection of noise variances is discussed in the last section.

Kalman filtering using Eq. (7) and (8) can be performed using different schemes depending on the order in which observations are processed, among them recursive filtering, fixed-point and fixed-lag filtering. Depending on the application different option can be used. In real-time applications, recursive or fixed lag estimation is used in which the data are processed sequentially as they arrive. In post-processing applications without storage constraints, fixed point filtering can be used to get full advantage of all observations available prior to the processing. In producing results given in this paper we use fixed-point (smoothing) Kalman filter.

2.1.3 Compensation of each frame for unwanted transformation (frame warping)

Given the observed cumulative transform parameters $(\tilde{\mathbf{A}}_n, \tilde{\boldsymbol{b}}_n)$ and the estimates of intentional cumulative transform parameters $(\hat{\mathbf{A}}_n, \hat{\boldsymbol{b}}_n)$, the difference between them is attributed to unwanted transformations and has to be compensated for. This can be achieved by inverting the former and applying the latter transform. Resulting transform $(\overline{\mathbf{A}}_n, \overline{\boldsymbol{b}}_n)$ is given by

$$\overline{\boldsymbol{x}}_n = \hat{\mathbf{A}}_n \boldsymbol{x}_1 + \hat{\boldsymbol{b}}_n = \left(\hat{\mathbf{A}}_n \left(\tilde{\mathbf{A}}_n\right)^{-1}\right)\boldsymbol{x}_n + \left(-\hat{\mathbf{A}}_n \left(\tilde{\mathbf{A}}_n\right)^{-1}\tilde{\boldsymbol{b}}_n + \hat{\boldsymbol{b}}_n\right) = \overline{\mathbf{A}}_n \boldsymbol{x}_n + \overline{\boldsymbol{b}}_n \tag{9}$$

where $\boldsymbol{x}_n$ and $\overline{\boldsymbol{x}}_n$ are initial and transformed coordinates in frame n, respectively. Using (9), a warped frame $\overline{\mathrm{I}}^n$ is computed as follows

$$\overline{\mathrm{I}}^n(\boldsymbol{x}) = \mathrm{I}^n\left(\left(\overline{\mathbf{A}}_n\right)^{-1}\boldsymbol{x} - \left(\overline{\mathbf{A}}_n\right)^{-1}\overline{\boldsymbol{b}}_n\right) \tag{10}$$

In our implementation, computing image values at non-integer locations in (10) is carried out by cubic interpolation.

2.2 Reconstruction of undefined regions using mosaicking

After the compensating transformation is applied to each frame, undefined regions appear near the edge of each frame. The extent of these regions varies from frame to frame and presents unacceptable visual artifacts. To our knowledge the solution to this problem has not been discussed in the literature, while available software products use frame trimming

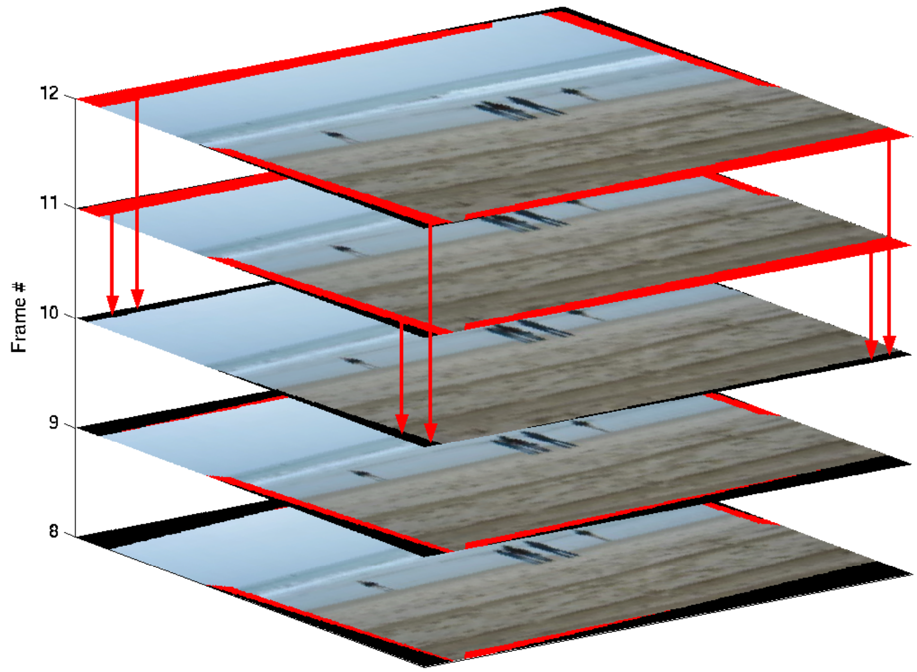

Figure 2: Mosaicking illustration for a typical frame with 2 previous and 2 future aligned frames shown. Red (grey in b/w version) areas correspond to missing areas in the reconstructed frame (with 3 pixel overlap over defined areas in central frame), black areas correspond to undefined areas.

and magnification or filling by a constant value. These methods lead to severe quality degradation of the resulting video and limit the range of possible correcting transformations (significant transformations lead to large undefined areas). Here we propose to use mosaicking for each frame in order to exploit temporal correlations between frames. Figure 2 illustrates the mosaicking process for one frame of our test sequence. Five consecutive frames are shown with the frame being reconstructed shown in the center. Two previous and two future frames are aligned with the transformed middle frame. Black regions in the frames are undefined regions. Red (grey in b/w version) regions correspond to defined pixels in neighboring frames collocated with undefined pixels in the middle frame (with overlap). These pixels are used to assign a value to a corresponding pixel in the reconstructed frame. Our experiments show that in most cases for each undefined pixel, several of the aligned neighboring frames contain a valid pixel value. Note that in Figure 2, not all pixels can be reconstructed from just four neighboring frames and more frames are needed. We emphasize the difference between our method, in which separate mosaic is constructed for each frame, and existing techniques constructing mosaic from all frames in the video, relying on scene stationarity. Such methods fail when the scene undergoes any change. Our method is detailed below.

2.2.1 Estimation of the transformation between distant frames

In order to properly align up to M future and past frames with respect to the current warped frame n, we need to find registration parameters of these frames with respect to the current frame. For a given frame n, using interframe motion parameters $(\mathbf{A}_n^1, \boldsymbol{b}_n^1)$,determined in the initial stage of our stabilization algorithm, as initial conditions, we sequentially estimate the global transform parameters between frames n and $n \pm m$ where $2 \le m \le M$. Simply cascading interframe transformations leads to error accumulation and hence, potential misalignment of different frames composing the mosaic. Instead, for each m, cascaded transforms $(\mathbf{A}_n^m, \boldsymbol{b}_n^m)$ and $(\mathbf{A}_{n+m}^1, \boldsymbol{b}_{n+m}^1)$ are used to initialize the solution for $(\mathbf{A}_n^{m+1}, \boldsymbol{b}_n^{m+1})$. The coordinate transformation obtained using cascaded transforms is given by

$$\boldsymbol{x}_{n+m+1} = \mathbf{A}_{n+m}^1 \boldsymbol{x}_{n+m} + \boldsymbol{b}_{n+m}^1 = \mathbf{A}_{n+m}^1 \left(\mathbf{A}_n^m \boldsymbol{x}_n + \boldsymbol{b}_n^m \right) + \boldsymbol{b}_{n+m}^1 = \left(\mathbf{A}_{n+m}^1 \mathbf{A}_n^m \right) \boldsymbol{x}_n + \left(\mathbf{A}_{n+m}^1 \boldsymbol{b}_n^m + \boldsymbol{b}_{n+m}^1 \right) \tag{11}$$

The transform $(\mathbf{A}_n^{m+1}, \boldsymbol{b}_n^{m+1})$ is then estimated by minimizing (2) as described above with only a few iterations of the gradient descent at the finest scale needed because the initial solution given by (11) is very close to the final solution. After M future frames are registered with respect to each frame, no additional computation is needed to register past

M frames with respect to any frame. For instance, the registration transformation for past frame $(-m)$ with respect to frame n can be found by inverting the registration transform of frame n as a future frame with respect to frame $(n-m)$

$$(\mathbf{A}_n^{-m}, \boldsymbol{b}_n^{-m}) = \left((\mathbf{A}_{n-m}^{m})^{-1}, -(\mathbf{A}_{n-m}^{m})^{-1} \boldsymbol{b}_{n-m}^{m} \right) \tag{12}$$

2.2.2 Warping distant frames and composing mosaic for undefined regions in each frame

Each frame out of $2M$ neighboring frames is aligned with respect to the warped current frame $\overline{\mathrm{I}}^n$ given by (10). Aligning transform for frame I^{n+m} is formed by cascading inverted registration transform $(\mathbf{A}_n^m, \boldsymbol{b}_n^m)$ with the correcting transform $(\overline{\mathbf{A}}_n, \overline{\boldsymbol{b}}_n)$ defined in (9). The resulting warping transform is given by

$$\left(\overline{\overline{\mathbf{A}}}_n, \overline{\overline{\boldsymbol{b}}}_n \right) = \left(\overline{\mathbf{A}}_n \left(\mathbf{A}_n^m \right)^{-1}, -\overline{\mathbf{A}}_n \left(\mathbf{A}_n^m \right)^{-1} \boldsymbol{b}_n^m + \overline{\boldsymbol{b}}_n \right) \tag{13}$$

and the warped frame $\overline{\overline{\mathrm{I}}}^n$ is computed as follows

$$\overline{\overline{\mathrm{I}}}^n(\boldsymbol{x}) = \mathrm{I}^n \left(\left(\overline{\overline{\mathbf{A}}}_n \right)^{-1} \boldsymbol{x} - \left(\overline{\overline{\mathbf{A}}}_n \right)^{-1} \overline{\overline{\boldsymbol{b}}}_n \right) \tag{14}$$

For each undefined pixel $\boldsymbol{x}$ in the target frame $\overline{\mathrm{I}}^n$, the reconstructed image value is found as follows

$$\overline{\mathrm{I}}^n(\boldsymbol{x}) = \frac{1}{\sum E(n,m)} \sum_{-M \le m \le M, m \ne 0} E(n,m) \overline{\overline{\mathrm{I}}}^{n+m}(\boldsymbol{x}) \tag{15}$$

where the weights $E(n,m)$ are set to the inverse of the errors of registration $E(\mathrm{I}^n, \mathrm{I}^{n+m}, \mathbf{A}_n^m, \boldsymbol{b}_n^m)$ obtained by minimizing (2). Large error of registration for a particular frame is the sign of a considerable mismatch, therefore the weight for this frame is set to a smaller value. We use additional cross-weighting at the boundary of undefined regions, setting the weight to depend linearly on the distance from the boundary with overlap width of 6 pixels. Taking multiple frames into account in computing the value of a particular pixel and using smooth transition between the defined area and the mosaic help to redistribute error over entire reconstructed region, reducing visual artifacts.

3. RESULTS

We refer the reader to the project web page to view initial video sequences and all our results discussed in this paper (http://iss1.bu.edu/~litvin/stabilization/index.html).

We test our techniques on 3 real-life video sequences (which we call A, B and C) acquired using a hand held video camera without any image stabilization. These three sequences contain increasing amount of vibrations and increasing depth variation and object motion. Sequence A is acquired by a person walking on the beach. It does not contain foreground objects. Sequence B contains foreground as well as background objects and more severe swings and vibrations. Sequence C is acquired from a moving car and contains extensive shear transformations between frames. We first illustrate different aspects of our technique on sequence A, which contains both translational and rotational unwanted motion. It is critical that global motion parameters be estimated accurately between each two consecutive frames, otherwise incorrect compensations are applied which result in annoying discontinuities in the resulting video. Using sequence A, we show how the accuracy of interframe motion parameters can be tested. In order to simplify the task we modify the motion model. First, we assume only translational motion between frames described by vector $\boldsymbol{b}$. Using this model, the cumulative transform parameters are given by

$$\tilde{\boldsymbol{b}}_n = \sum_{k=1}^{n} \boldsymbol{b}_n^{n+1} \tag{16}$$

The components of $\tilde{\boldsymbol{b}}_n$ for sequence A are shown in Figure 3. Spikes correspond to sudden swings of the camera. Assuming static camera (performing “total motion compensation”), the correcting transform becomes

$$\overline{\boldsymbol{x}}_n = \boldsymbol{x}_n - \tilde{\boldsymbol{b}}_n \tag{17}$$

The result of applying such compensating transform is illustrated in Figure 4 and included on the project web page. It can be seen that landmark objects in the corrected sequence do not move with respect to the frame coordinates, while

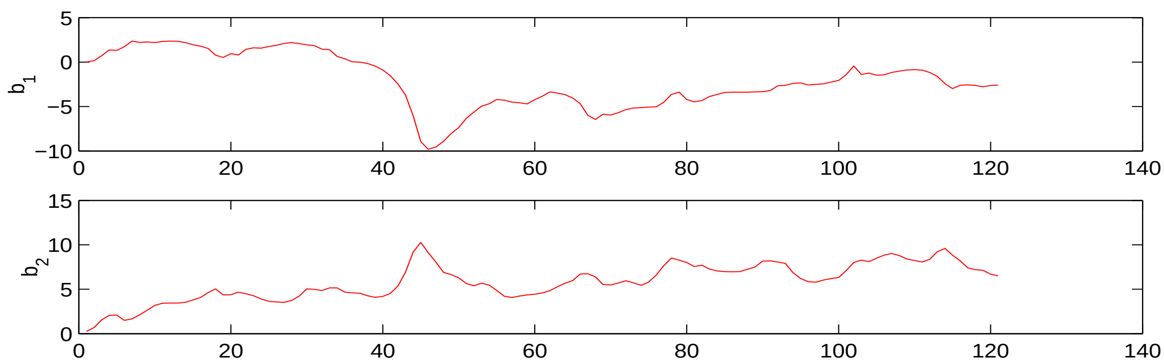

Figure 3: Cumulative motion parameters for sequence A using translational motion model.

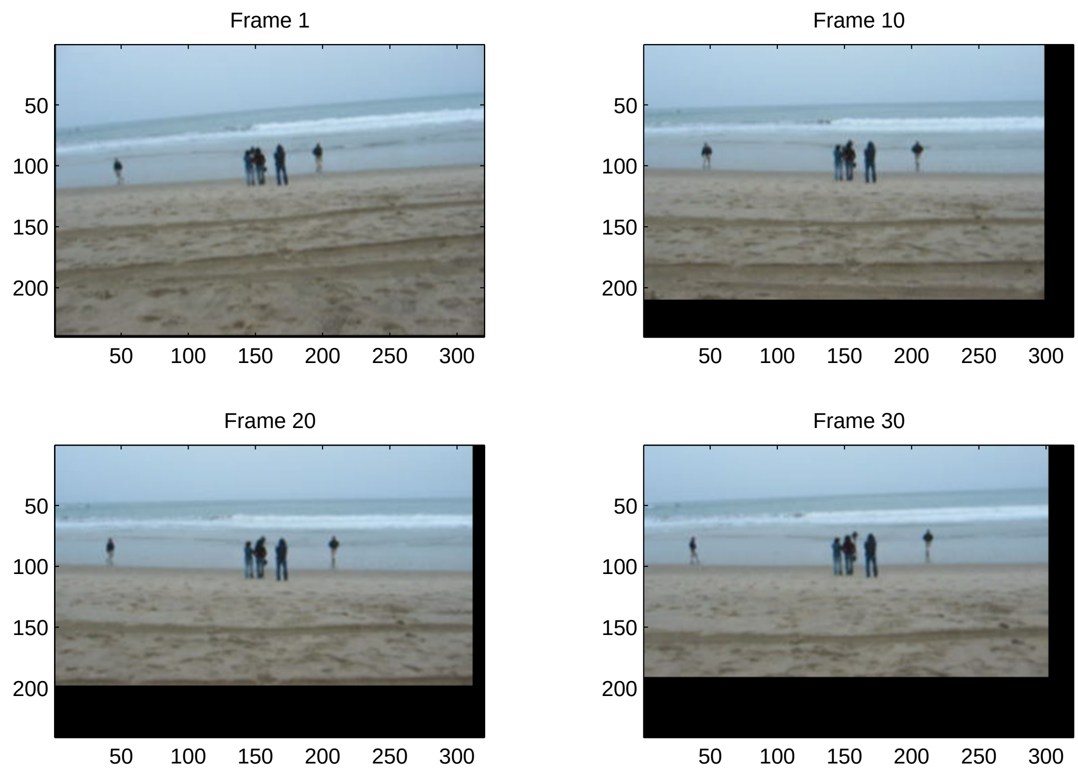

Figure 4: Compensating transform applied to selected frames of sequence A assuming static intentional camera and only translational motion model

rotational vibrations remain uncorrected. No translational jumps are observed, which indicates that using a 2 parameter model an accurate solution was found for interframe motion parameters.

Now we employ the full 6-parameter interframe affine motion model. In Figure 5, we show the resulting cumulative transform parameters calculated according to Eq. (4) and the filtered parameters obtained using a fixed-interval Kalman filtering. In Figure 6, we show four frames of sequence A before and after applying the stabilization algorithm. It can be seen that camera panning and tilting are largely compensated for. Full results are given on the project web page. Our method performs well even for more difficult sequences B and C, although some model mismatch artifacts can be noticed. The solution to these problems can be achieved by using interframe motion model with better descriptive capabilities as well as by adaptively eliminating the impact of foreground and moving objects (for example, by including preliminary step of block motion estimation followed by masking out blocks with fast motion).

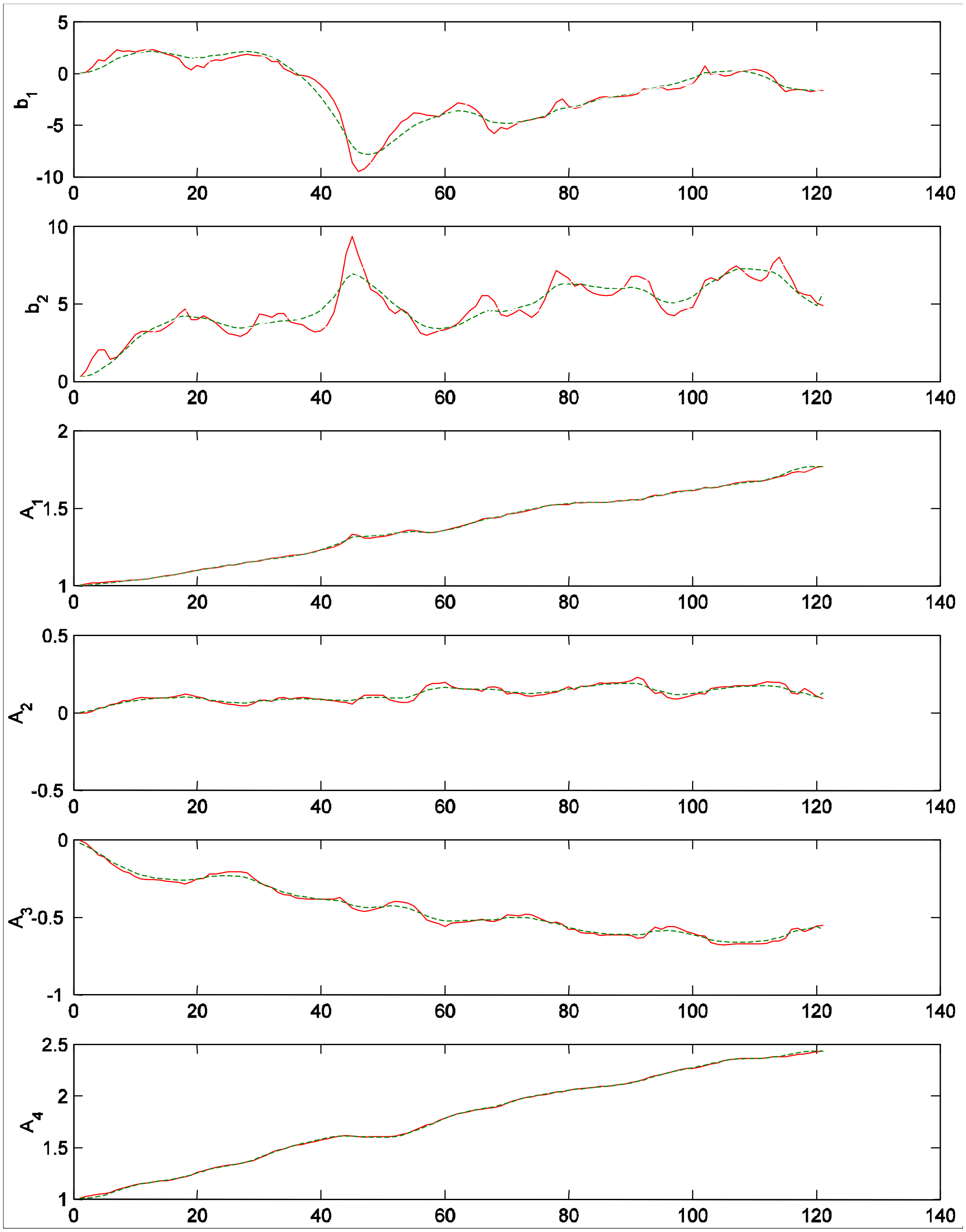

Figure 5: Red straight lines – Inter-frame motion parameters obtained for sequence A; Green dashed lines - intentional cumulative transform parameters estimated using smoothing Kalman filtering

The evaluation and comparison of video stabilization algorithms is a difficult task given that no ground truth is available for real sequences and usefulness of synthetic sequences is limited. Perceptual judgment of stabilization is the best option to evaluate video stabilization algorithms aimed at the human observer. On the project web page we show the

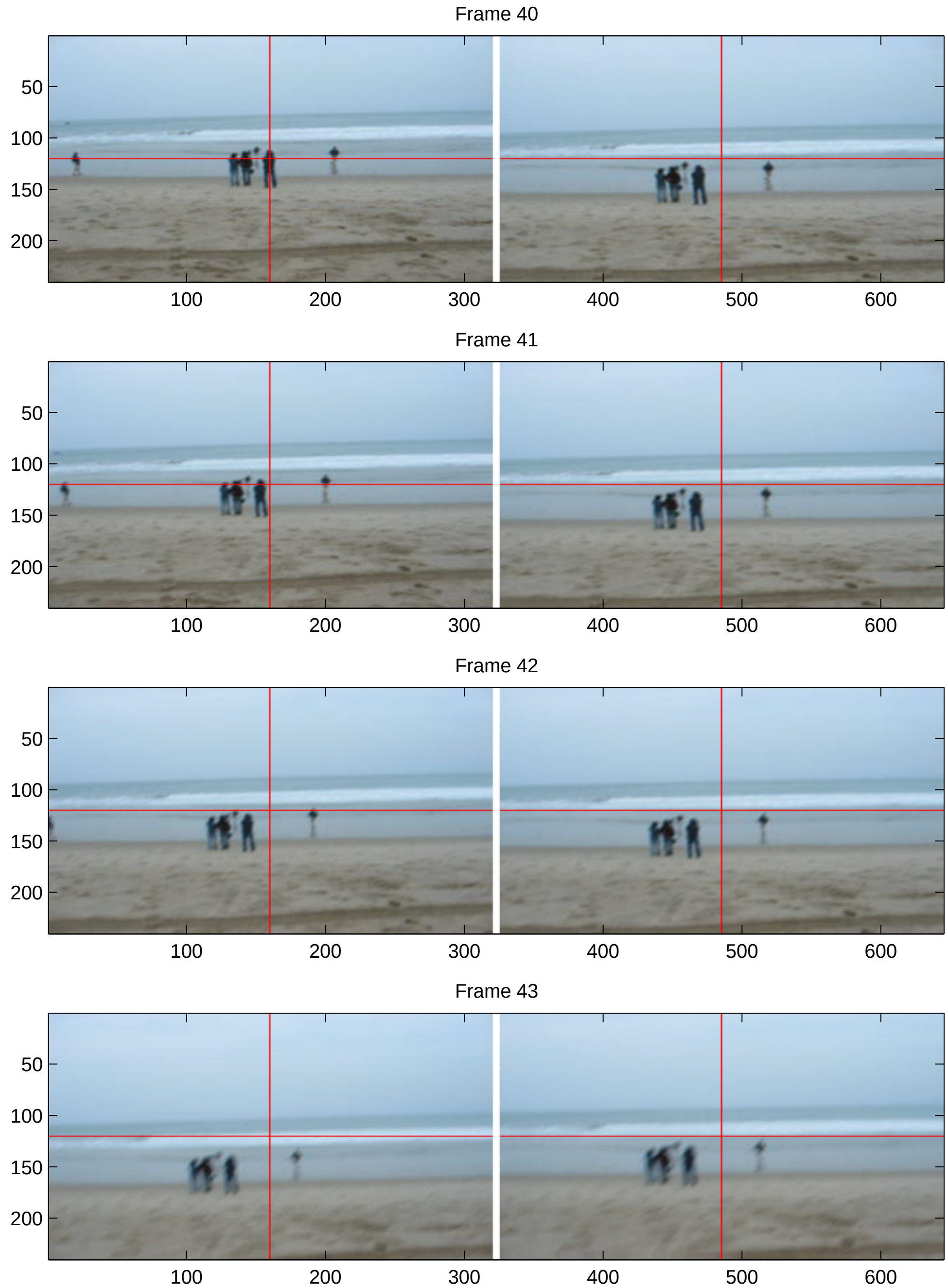

Figure 6: Four frames of sequence A before (left) and after (right) stabilization.

comparison of our method with one of the available software products, “Steady Hand” by a Germany based company DynaPel Inc. A displacement model and global search method are used in this product to estimate interframe transformations. Our results show significantly higher quality of stabilization. Our reconstruction does not change the scale due to our mosaicking-based reconstruction of undefined areas. The evidence of using mosaicking can only be noticed in single frames.

4. CONCLUSIONS

In this paper we presented a novel approach to video stabilization. We used a probabilistic approach to model and separate unwanted, parasitic camera motion parameters from the intentional camera motion parameters estimated out of the video sequence. We used mosaicking to reconstruct undefined regions in the frames caused by the applied compensating transformations.

Using our technique we obtained promising preliminary results on random test sequences with complex motion and severe vibrations. We compared our results with one of commercial products and showed a significant improvement of performance for our technique.

Several aspects of our method can be improved to achieve an even better performance and eliminate the need to adjust dynamic model parameters. A more realistic colored, correlated noise model may be used to estimate the representation of the intentional camera position. For example, noise processes of state variables a_2 and a_3 are correlated because these parameters describe the angle of the camera tilt. Noise is colored for each variable because the time scale of typical camera swing or tilt is usually longer than sampling interval. Full covariance matrices of process and observation noise models can be estimated and tracked dynamically to accommodate changing conditions of the footage (such as speed of the vehicle or person holding the camera). A more efficient method can be employed in estimating interframe transform parameters to achieve more robust and real-time performance. Advanced multiscale mosaicking technique such as one used in [1] can eliminate remaining visual artifacts due to changes in illumination between frames and errors in registration. Our method of stabilization can be easily adapted to perform additional processing, such as sampling rate conversion, static mosaic construction, ego-motion estimation.

REFERENCES

1. P.J. Burt and E.H. Adelson, A multiresolution spline with application to image mosaicking, *ACM Transactions on Graphics*, Vol. 2 No. 4, pp. 217-236, 1983
2. M. Hunsen et al, Real-time scene stabilization and mosaic construction, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 54-62, 1994
3. M. Irani, B. Rousso and S. Peleg, Recovery of ego-motion using image stabilization, *Proc. of Conference on Computer Vision and Pattern Recognition*, 454-460, June 1994
4. Z. Duric and A. Rosenfeld, Stabilization of image sequences, *Technical report CAR-TR-778*, University of Meryland, Jul. 1995
5. Y.S. Yao and R. Chelappa, Electronic stabilization and feature tracking in long video sequences, *Technical report CAR-TR-790*, University of Maryland, Sept. 1995
6. C.H. Morimoto and R. Chellappa. Automatic digital image stabilization, *Proc. of IEEE International Conference on Pattern Recognition*, Vienna, Austria, Aug. 1996
7. C. Morimoto and R. Chellappa, Fast 3D Stabilization and mosaic construction, *Proc.* of *Conference on Computer Vision and Pattern Recognition*, 660-665, 1997
8. K. J. Hwan, Adaptive digital image stabilization using bit-plane matching, *Comm. & Multimedia Electronics Workshop*, pp.47-52, 1998
9. Z. Zhu et al, Camera stabilization based on 2.5D motion model estimation and inertial motion filtering, *IEEE Conference on Iintelligent Vehicles*, 329-334, 1998
10. K. -H. Lee, S. -H. Lee and S. -J. Ko, Digital image stabilizing algorithms based on bit-plane matching, *IEEE Trans. on Consumer Electronics,* vol. 44, no. 3, pp. 617-622, Aug. 1998
11. C. Guestrin, F. Cozman, and E. Krotkov, Fast Software Image Stabilization with Color Registration, *Proc. of IROS 98*, Vol. 1, 19 – 24, Oct. 1998
12. A. Censi, A. Fusiello, and V. Roberto. Image stabilization by features tracking. *Proc. of the 10th International Conference on Image Analysis and Processing (ICIAP `99)*, 665-667, Sept. 1999