Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion

Junlan Yang, *Student Member, IEEE*, Dan Schonfeld, *Senior Member, IEEE*, and Magdi Mohamed, *Associate Member, IEEE*

***Abstract*— Video stabilization is an important technique in digital cameras. Its impact increases rapidly with the rising popularity of handheld cameras and cameras mounted on moving platforms (e.g., cars). Stabilization of two images can be viewed as an image registration problem. However, to ensure the visual quality of the whole video, video stabilization has a particular emphasis on the accuracy and robustness over long image sequences. In this paper, we propose a novel technique for video stabilization based on the particle filtering framework. We extend the traditional use of particle filters in object tracking to tracking of the projected affine model of the camera motions. We rely on the inverse of the resulting image transform to obtain a stable video sequence. The correspondence between scale-invariant feature transform points is used to obtain a crude estimate of the projected camera motion. We subsequently postprocess the crude estimate with particle filters to obtain a smooth estimate. It is shown both theoretically and experimentally that particle filtering can reduce the error variance compared to estimation without particle filtering. The superior performance of our algorithm over other methods for video stabilization is demonstrated through computer simulated experiments.**

***Index Terms*— Bootstrap filtering, Monte Carlo methods, motion analysis, particle filtering, video stabilization.**

I. Introduction

VIDEO CAMERAS mounted on handheld devices and mobile platforms have become increasingly popular in the consumer market over the past few years due to a dramatic decrease in the cost of such devices. Rattled camera motion and platform vibrations can be difficult to avoid when using handheld cameras, which will generate unstable video images. Video stabilization is, therefore, becoming an indispensable technique in advanced digital cameras and camcorders.

Camera motion estimation is an essential step toward video stabilization. Stabilization methods exploit the fact that camera motion causes the affine transform of the frames, which can be inverted to obtain stable frames. The first step of video stabilization is, therefore, to identify the global affine transformation. However, in video stabilization, unlike most motion estimation techniques, the robustness of the estimation is critical due to the fact that an incorrect estimate will yield a sudden jitter in the video sequence. In particular, the estimation should be performed consistently in the presence of outliers, blurring, or illumination variance. We will demonstrate that the use of corresponding feature points in a particle filtering framework can be used to obtain robust tracking of the camera motion, which is superior to existing approaches.

In most cases, videos captured by mobile cameras observe users' desired motion. It is therefore necessary to preserve the intentional camera motion while removing the undesired motion due to an unsteady and vibrating platform. This is generally accomplished with adaptive or nonadaptive recursive filters. We will employ a motion separation scheme based on the Kalman filter. Combination of camera motion estimation with motion separation determines the undesired motion, which is compensated and thus stable video sequences are produced.

In this paper, we propose the use of particle filters [1] to estimate the global camera motion between successive frames. Particle filters have been widely introduced as a powerful and flexible tool to accurately model nonlinear physical systems. We will adapt particle filters for video stabilization by using them to estimate the affine transformation model of the global camera motion from corresponding feature points. We will demonstrate, theoretically and experimentally, that particle filters can be used to provide a smooth estimate with low error variance, which is critical in video stabilization. The resulting motion estimation algorithm yields accurate and robust estimate of the affine transform model.

The rest of the paper is organized as follows. In Section II, we provide a summary of the related work in the areas of video stabilization and particle filtering. In Section III, we discuss the theoretical issues used to motivate and model the proposed approach to video stabilization. We present several valuable properties of particle filters in the motion estimation framework. The complete video stabilization system is proposed in Section IV and the performance of our approach compared to existing methods is demonstrated through computer simulations in Section V. Finally, we provide a brief summary in Section VI.

II. Related Work

Most of video stabilization algorithms rely on specific image motion models. In earlier efforts, 2-D models [2],

Manuscript received February 13, 2007; revised July 3, 2008. First version published April 7, 2009; current version published July 22, 2009. This paper was recommended by Associate Editor X. Li.

J. Yang and D. Schonfeld are with the Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607 (e-mail: jyang24@uic.edu; dans@uic.edu).

M. Mohamed is with the Physical and Digital Realization Research Center of Excellence, Motorola Labs, Schaumburg, IL, 60196 (e-mail: magdimohamed2007@yahoo.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2020252

[3], 2-D affine models [4]–[6], 2.5-D model [7], and 3-D models [8]–[10] have been employed to represent the video stabilization problem. The 2-D model parameters are uniform for each point in the scene, while, on the contrary, a 3-D model has spatial variant parameters relating to depth information. A pure 2-D model, with a 2-D translation vector and one rotation angle, is not capable of describing the 3-D camera motion, which includes rotation out of the image plane and translation along the optical axis. A 3-D model is valuable in determining depth changes; however, it introduces tremendous complexity and challenges due to the loss of scenes' depth information in the projected images. Though the 2.5-D model has a tempting feature of introducing partial depth information into the 2-D model, the algorithm is practically difficult to realize. The 2-D affine model with six parameters provides an attractive representation of the camera motion. It achieves a good balance between accuracy and computational cost for video stabilization purposes [4]. We will adopt a variation of the 2-D affine model for the camera motion representation. In particular, we introduce a constraint due to the orthogonality of the rotation matrix projected onto the plane, which is critical to the camera model. This constraint, however, has not been recognized previously in video stabilization.

Critical to the success of video stabilization is global camera motion estimation. Methods used for global motion estimation can be classified mainly into two categories: 1) intensity-based motion estimation and 2) feature-based motion estimation. For example, in [4], the six-parameter affine motion is estimated by minimizing a p-norm cost function based on the grayscale values of the image pixels. More examples of intensity-based approaches include optical flow technique [11], gray-code bit-plane matching [12], and sub-image phase correlation [2]. The image intensity-based motion estimation methods have an advantage of being inherently robust to outliers and illumination changes. Some effort has also been devoted to global motion estimation based on image features. Popular features include edge patterns [13] and corners [9]. For algorithms proposed for special scenarios, for example in on-road video stabilization [10], the author uses the lane lines and road vanishing point as global features. It is proposed in [5] to use features on the horizon for off-road situations. Feature-based motion estimation algorithms are, in general, more accurate but less robust, compared with intensity-based motion estimation methods. Therefore, some have sought to gain advantage from both approaches by utilizing both the image intensity and features of the image. Examples include techniques using color information to match characteristic curves [14] and combined texture and correlation measurements for matching of small blocks [7]. In this paper, the global motion estimation is based on image features by tracking scale-invariant feature transform (SIFT) points. However, we further integrate the intensity information of the images by filtering the SIFT points estimate with particle filters.

Particle filtering and sequential Monte Carlo (SMC) methods in general have emerged in recent years as powerful approaches to visual tracking and pose estimation [15]. It has been proposed to overcome the limitation of Kalman filters posed by its foundation on linear and Gaussian models [16]. Pioneering works in particle filters include bootstrap filters [17], CONDENSATION filters [18], and sequential importance sampling (SIS) particle filters [19], [20]. Moreover, the smoothing Monte Carlo methods discussed in [21]–[23]show that particle smoothing can be used as a nonlinear, non-Gaussian counterpart of a Kalman smoother.

Classical particle filtering theory allows particles to be sampled from any density function. Traditional implementations of particle filters generally rely only on the state transition information for sampling. However, it has recently been shown that much more effective sampling schemes can be attained by sampling from a proposal density which takes into account the observation data; e.g., ICONDENSATION [24], the unscented particle filter [25], motion-based particle filters [26], and models with embedded-motion [27]. This evolution in particle filters has had a profound impact on the development and popularity of various methods for object tracking in video sequences. The state representation used in most object tracking implementations has been restricted to model a parametric representation of the object's contours, e.g., an ellipsoidal model. In this paper, we extend the application of particle filters to tracking the 2-D affine transform parameters of the global camera motion. Moreover, we borrow from the recent developments of efficient particle filtering by proposing a novel feature-based importance density function such that the sampling scheme utilizes current observations. The proposed scheme can ensure that the algorithm works effectively and efficiently with a low computational cost.

Tracking of the camera motion can be used to stabilize the video sequence by inversion of the 2-D affine model. However, in many circumstances, the user of a mobile camera wishes to capture its intended movements throughout the video scene. Kalman filters [4], [2] and extended Kalman filters (EKF) [6] have been used to distinguish and isolate intended and unintended camera motion. Inertial filters [7] and discrete Fourier transform (DFT) filters [3] have also been proposed for eliminating high-frequency vibrations, when offline processing delay can be tolerated. We will adopt a "constant-velocity" Kalman filter [4] where the intended motion parameters change over time with constant speed. This assumption helps to distinguish and preserve the intended camera motion.

In practice, the stabilized video images often do not overlap perfectly with the boundary of the desired image frame and therefore, missing regions are formed and the resolution is impaired. It is common to adapt a mosaicking technique in order to interpolate the missing image regions [4], [6]. A more comprehensive postprocessing is proposed in [23], where the combination of motion inpainting and image deblurring provides effective enhancement on quality of the stabilized image sequences.

III. Theoretical Foundations

This section addresses theoretical issues of our algorithm, including the camera models and the properties of particle filtering estimation.

A. Camera Model

In video stabilization, the camera model can be derived as follows. Assume that there is one point P in the scene whose coordinates in camera coordinate system at time t_0 is $[x_0, y_0, z_0]^T$. In time t_1, camera has been moved by a rotation and a translation, while the point P remains in the same position in world coordinates. The new coordinates at time t_1 is $[x_1, y_1, z_1]^T$ (also in camera coordinate system). These two vectors of coordinates can be related by the equation

$$\begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T = R_{3\times 3} * \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T + T_{3\times 1} \quad (1)$$

where $R_{3\times 3}$, $T_{3\times 1}$ are the opposite transform of the camera's 3-D rotation and translation, repectively. By projection, the image coordinates of P in time t_0 and t_1 are given by

$$\begin{bmatrix} u_0 & v_0 & \lambda \end{bmatrix}^T = \frac{\lambda}{z_0} * \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T \quad (2)$$

$$\begin{bmatrix} u_1 & v_1 & \lambda \end{bmatrix}^T = \frac{\lambda}{z_1} * \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T \quad (3)$$

where λ is the image plane-to-lens distance of the camera. A detailed description and illustration of above imaging model can be found in [29]. With (1), (2), (3) and by rewriting the rotation matrix $R_{3\times 3}$ and translation vector $T_{3\times 1}$ to show their entries, we can get

$$\begin{bmatrix} u_1 \\ v_1 \\ \lambda \end{bmatrix} = \frac{z_0}{z_1} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \\ \lambda \end{bmatrix} + \frac{\lambda}{z_1} * \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \quad (4)$$

The first two columns of (4) yield the following 2-D form

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = s \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (5)$$

where we define $s \triangleq z_0/z_1$, $t_x \triangleq sR_{13}\lambda + (\lambda/z_1)T_x$ and $t_y \triangleq sR_{23}\lambda + (\lambda/z_1)T_y$. In general, the scaling factor s and translations t_x, t_y vary for objects with different depths. However, it is realistic to assume that in most real scenes the background (at which the stabilization algorithm aim) has small relative depth variation, compared to the distance between camera and the scene [4]. Thus, by assuming the uniformity of the scaling and translation, we can use this 2-D affine transform to approximate the 3-D camera motion. The 2-D affine model is shown by experiments to be an accurate enough model for stabilization purposes.

Moreover, the rotation matrix $R_{3\times 3}$ is orthonormal, i.e., it is constrained by the following [30]

$$R_{11}^2 + R_{12}^2 + R_{13}^2 = 1; \quad R_{21}^2 + R_{22}^2 + R_{23}^2 = 1$$
$$R_{11}R_{21} + R_{12}R_{22} + R_{13}R_{23} = 0.$$

Therefore, we get the relationship between the following four rotation parameters as

$$R_{11}R_{21} + R_{12}R_{22} + \sqrt{1 - R_{11}^2 - R_{12}^2}\sqrt{1 - R_{21}^2 - R_{22}^2} = 0. \quad (6)$$

Since R_{22} can be determined from (6) given R_{11}, R_{12}, and R_{21}, the affine model (5) has six degrees of freedom, which is equivalent to the usual affine model [4]. However, the proposed model is valuable in explicitly expressing the actual physical meanings of the parameters. We can further obtain the depth change and three rotation angels from these parameters, which the usual affine model is not capable of doing.

Our task in global motion estimation is to determine the six parameters s, R_{11}, R_{12}, R_{21}, t_x, and t_y for every successive frame. Also, note that these parameters represent six kinds of motion which can take place independently. Therefore, it is reasonable to assume that these parameters are statistically independent of each other. This observation eases the construction of particle filters, as will be shown in the following sections.

B. Particle Filtering Estimation

The above situation can be considered as a Bayesian tracking problem, where a Markov discrete-time state-space model can be introduced. The state-space model is defined with a state vector at time k, $\mathbf{x}_k \triangleq \left[s_k, t_{xk}, t_{yk}, R_{11k}, R_{12k}, R_{21k}\right]^T$, and observations $\mathbf{z}$. The fundamental idea of particle filter estimation is to recursively approximate the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ by a set of particles $\{\mathbf{x}_k^i, i = 1, \ldots, N\}$ with associated weights $\{w_k^i, i = 1, \ldots, N\}$, where N is the number of particles and k is the time step. In our case, each frame is considered as a step of time. The particles $\mathbf{x}_k^i \sim q(\cdot)$ are random vectors drawn from a proposal $q(\cdot)$, which is referred to as importance density. Then, a weighted approximation to the posterior $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is given by [1]

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (7)$$

where the normalized weights are defined as

$$w_k^i \propto p(\mathbf{x}_k^i|\mathbf{z}_{1:k})/q(\mathbf{x}_k^i|\mathbf{z}_{1:k}).$$

We can then use the expectation of the state density as an estimate of current state [21], [23]

$$\hat{\mathbf{x}}_k \triangleq E\left[\mathbf{x}_k\right] = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k$$
$$\approx \int \mathbf{x}_k \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) d\mathbf{x}_k = \sum_{i=1}^{N} w_k^i \mathbf{x}_k^i. \quad (8)$$

It can be proved that as $N \to \infty$, the approximation (7) converges to the true posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ in mean-square sense and the convergence rate is $1/N$ [31]. The complexity of particle filtering increases exponentially with the increment of state dimension, and so does the required particle number [32]. The significance of realizing R_{22} is determined by R_{11}, R_{12}, R_{21} in that it eliminates 1-D of the state, so the complexity of the system is largely reduced.

Traditionally, the importance density is chosen as the prior density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ [18]. However, this choice is, in general, not efficient enough due to the fact that it does not take into account the current observation $\mathbf{z}_k$. Instead, we propose an importance density whose mean vector $\bar{\mathbf{x}}_k$ is itself an estimate of the state obtained from the current observation $\mathbf{z}_k$. We then draw particles from a rather simple distribution $q(\cdot)$ determined by the mean $\bar{\mathbf{x}}_k$ and covariance matrix Σ_1

$$\mathbf{x}_k^i \sim q(\bar{\mathbf{x}}_k, \Sigma_1). \quad (9)$$

Due to the independence assumption of state variables, Σ_1 is set to be diagonal. $\bar{\mathbf{x}}_k$ is obtained in this paper from a particular feature-based motion estimation technique which is detailed in the following section. However, the proposed system works with any other 2-D affine motion estimation techniques. Also, we consider the applied techniques to give a fine estimate in the sense that $\bar{\mathbf{x}}_k$ is an unbiased estimate of the true state $\mathbf{x}_k$, with diagonal covariance matrix Σ_2. Nonetheless, we reply on the weighted sum of samples instead of using directly $\bar{\mathbf{x}}_k$ due to the desirable property of the former. It can be proved that for large particle numbers, the estimation $\hat{\mathbf{x}}_k$ gives lower error variance than $\bar{\mathbf{x}}_k$, as stated in the following.

Let the error vectors associated with two estimations $\hat{\mathbf{x}}_k$ and $\bar{\mathbf{x}}_k$ be given by $\varepsilon_k \triangleq \hat{\mathbf{x}}_k - \mathbf{x}_k$ and $\mathbf{e}_k \triangleq \bar{\mathbf{x}}_k - \mathbf{x}_k$, respectively. And the covariance matrices of errors are defined as $Cov(\varepsilon_k, \varepsilon_k)$ and $Cov(\mathbf{e}_k, \mathbf{e}_k)$. Note that $Cov(\mathbf{e}_k, \mathbf{e}_k) = \Sigma_2$ given unbiasedness assumption. We have the following lemma.

Lemma 1: $Cov(\varepsilon_k, \varepsilon_k) = \frac{1}{N}(\Sigma_1 + \Sigma_2)c_k$, where c_k is some constant and N is the number of samples.

The proof of Lemma 1 can be found in the Appendix. It is then easy to further observe the following:

Proposition 1: For each time step k, there exists an N_k^o such that if $N > N_k^o$, the error covariance matrices have $Cov(\varepsilon_k, \varepsilon_k) \preceq Cov(\mathbf{e}_k, \mathbf{e}_k) = \Sigma_2$, for every dimension. Moreover, $Cov(\varepsilon_k, \varepsilon_k) \to 0$ when $N \to \infty$ with rate $1/N$.

Proof: Let M denote the dimension of the state vector. In order that particle estimation error is lowered in every dimension, we require that $Cov(\varepsilon_k, \varepsilon_k)_{jj} < \Sigma_{2\,jj}, \forall j = 1, \ldots, M$. It is equivalent to requiring $\frac{1}{N}(\sigma_{1j}^2 + \sigma_{2j}^2)c_k < \sigma_{2j}^2, \forall j = 1, \ldots, M$, where σ_{1j}^2 and σ_{2j}^2 are diagonal elements of Σ_1 and Σ_2, respectively. Therefore, it suffices with N satisfying $N > \max_{j=1\ldots M}\left[c_k(\sigma_{1j}^2 + \sigma_{2j}^2)/\sigma_{2j}^2\right] \triangleq N_k^o$. Directly from Lemma 1, $Cov(\varepsilon_k, \varepsilon_k) \to 0$ when $N \to \infty$ with rate $1/N$. ■

Proposition 1 shows that given an unbiased estimate, the particle filter can produce another unbiased estimation with lower error variance, if the number of particles is sufficiently large. We refer to this property as the smoothing property. We also show that the variance of estimation error converges to zero asymptotically with rate $1/N$, which is consistent with the convergence property of the particle filter in [31].

IV. Video Stabilization

In this section, we describe our complete system of video stabilization under the particle filter framework. We first introduce the importance density function based on SIFT [33] algorithm. We then describe the particle filter algorithm for global motion estimation. The Kalman filter is introduced for obtaining intentional motion through smoothing the global motion. The undesired motion is then extracted for motion compensation.

A. Importance Density Using Scale-Invariant Features

The choice of a good importance density is a crucial step in the design of the particle filter. We would like to draw particles from an importance density that is close to true posterior to make the filtering algorithm more effective. A technique is proposed here to encourage particles to be in the right place.

Fig. 1. SIFT correspondence from frame 200, 201 in outdoor sequence *STREET*. The lines across images represent the connections of features in two images, while the lines in the left frame represent the motion vector between them. The rectangles denote incorrect correspondences in the background. The circles denote correspondences from the moving objects.

We use feature tracking to get the mean vector $\bar{\mathbf{x}}_k \triangleq \left[\bar{s}_k, \bar{tx}_k, \bar{ty}_k, \bar{R}_{11k}, \bar{R}_{12k}, \bar{R}_{21k}\right]^T$ for constructing the proposal density $q(\cdot)$. The feature points we use are obtained based on the SIFT algorithm [33]. SIFT extracts and connects feature points in images which are invariant to image scale, rotation, and changes in illumination. Moreover, it provides distinctive descriptors which enable us to find the correspondences between features in different images. An example of feature correspondence of two images found by SIFT is shown in Fig. 1. Once we have corresponding pairs, we can use them to determine the transform matrix between two images. Equation (5) can be rewritten as

$$\begin{bmatrix} u_k & v_k \\ .. & .. \end{bmatrix} = \begin{bmatrix} u_{k-1} & v_{k-1} & 1 \\ .. & .. & 1 \end{bmatrix} \begin{bmatrix} \bar{s}_k\bar{R}_{11k} & \bar{s}_k\bar{R}_{12k} \\ \bar{s}_k\bar{R}_{21k} & \bar{s}_k\bar{R}_{22k} \\ \bar{tx}_k & \bar{ty}_k \end{bmatrix} \tag{10}$$

where $\left[u_{k-1}, v_{k-1}\right]^T$ and $[u_k, v_k]^T$ are one pair of corresponding feature points. We need only three pairs to determine a unique solution. However, more matches can be added as shown in (10). The over-determined system is in the form of $\mathbf{Y} = \mathbf{XA}$, which can be solved easily under least-square criteria by $\mathbf{A} = \left[\mathbf{X}^T\mathbf{X}\right]^{-1}\mathbf{X}^T\mathbf{Y}$. Together with (6), we can further form the mean vector $\bar{\mathbf{x}}_k$ from matrix $\mathbf{A}$. We then generate particles according to the importance density $q(\cdot)$ of a six-dimensional Gaussian distribution

$$\begin{aligned} \mathbf{x}_k^i &\sim q_G(\bar{\mathbf{x}}_k, \Sigma_1) \\ &= \frac{1}{\sqrt{(2\pi)^6 \mid \Sigma_1 \mid}} exp\left[-\frac{1}{2}\left(\mathbf{x}_k^i - \bar{\mathbf{x}}_k\right)^T \Sigma_1^{-1}\left(\mathbf{x}_k^i - \bar{\mathbf{x}}_k\right)\right] \end{aligned} \tag{11}$$

where $q_G(\bar{\mathbf{x}}_k, \Sigma_1)$ refers to the Gaussian function with mean $\bar{\mathbf{x}}_k$ and covariance matrix Σ_1. Σ_1 is selected by the system designer for efficient particle sampling in various situations. This proposal density helps to avoid generating useless particles and, hence, keep the computation cost low. In experiments, we successfully reduce the number of particles to only 30 while acquiring the same or even better quality as using 300 particles when the importance density is a prior distribution.

In practice, we observe that SIFT sometimes provides incorrect corresponding feature points especially when the pictures are blurred due to rapid camera movements. Rectangles in Fig. 1 show examples of incorrect correspondences in the background scene. In such cases, the resulting $\bar{\mathbf{x}}_k$ will be an estimate with large error variance. However, we can see later

in the experiment that particle filtering is robust in the sense that it will save us from trapping in the wrong estimate by relying on weighted sum of samples. This result agrees with the smoothing property of particle filtering. The fundamental reason is in the weighting process; particle filtering explores the underlying density by viewing observations in different aspects such as image intensity and image features, and hence is more robust than a single criteria of analyzing images.

Though there is no need to be concerned about outliers in the background, we have to deal with moving objects. As we can see in Fig. 1, SIFT connects features from the moving objects (yellow circles), which correspond to local motion, and should be separated when calculating the global motion. We propose to identify these features by the difference in moving velocities between objects and the background. It can be assumed that the velocities of local motions within the images are much larger than that of the global image motion. As we can see in Fig. 1, the lengths of the motion vectors from the car's features are, in general, larger than the ones belonging to background. When comparing two successive frames, the features which have moved a large distance should be isolated before estimating the global motion and intentional motion. This isolation can be done, since in general, we consider that the moving objects occupy only a small portion of the scene so that background features dominate.

B. Particle Filtering for Global Motion Estimation Between Successive Frames

We employ here the bootstrap filtering proposed in [17] with slight modifications. At time k, we first generate $\mathbf{x}_k^i$ from an importance density $q_G(\bar{\mathbf{x}}_k, \Sigma_1)$, as in (11). We then need to assign weights to these particles. The desired weights should perform as an evaluation on the "quality" of the particles. "Quality" here represents how close to true state each particle's state is. In our case, N particles suggest N proposals of the transformation matrix, so we can apply the N inverse transforms to frame k and get N candidate images $\mathbf{A}_i$. (Here we temporarily drop the frame index k for the simplicity of the notations.) Then we compare these images with $k-1$ frame $\mathbf{A}_0$ to determine the similarity between them. The particle weights are hence decided according to the similarities, i.e., the higher the similarity, the larger the weight. We choose mean square error (MSE) and feature distance as two measures of similarity. MSE comparison calculates the difference of the gray-scale from pixel to pixel between two images and then computes the square and mean to get an MSE value M_i. The MSE likelihood is then given by

$$P_{MSE}^i \propto \frac{1}{\sqrt{2\pi}\sigma_M} \exp\left\{-\frac{{M_i}^2}{2\sigma_M^2}\right\}. \tag{12}$$

The feature distance comparison employs the SIFT feature points extracted in calculating the importance density. As long as we have features in image $\mathbf{A}_i$ and $\mathbf{A}_0$, we can calculate the distances of all the corresponding feature points. Denote the average distance to be D_i. The feature likelihood is given by

$$P_{\text{feature}}^i \propto \frac{1}{\sqrt{2\pi}\sigma_F} \exp\left\{-\frac{{D_i}^2}{2\sigma_F^2}\right\} \tag{13}$$

where σ_M and σ_F in (12) and (13) are adjustable standard deviations which can be chosen experimentally. The normalized weight for particle $\mathbf{x}_k^i$ is then given by

$$w_k^i = \frac{P_{MSE}^i P_{\text{feature}}^i}{\sum_{i=1}^N P_{MSE}^i P_{\text{feature}}^i}. \tag{14}$$

Once we obtain the weight for each particle, we will approach the true state by a discrete weighted approximation as in (8): $\hat{\mathbf{x}}_k = \sum_{i=1}^N w_k^i \mathbf{x}_k^i$, where the estimated state tells the estimated values of global affine motion parameters $\hat{\mathbf{x}}_k = [\hat{s}_k, \hat{tx}_k, \hat{ty}_k, \hat{R}_{11k}, \hat{R}_{12k}, \hat{R}_{21k}]^T$. $\hat{R}_{22k}$ can be calculated from (6).

Now assume that the first frame of the video sequence is stable, and denote it to be the reference frame. Then the accumulative scaling factor, accumulative rotation matrix R_k^A, and translation displacement T_k^A with respect to the reference frame are given by

$$s_k^A = s_{k-1}^A \cdot \hat{s}_k, \qquad R_k^A = R_{k-1}^A \begin{bmatrix} \hat{R}_{11k} & \hat{R}_{12k} \\ \hat{R}_{21k} & \hat{R}_{22k} \end{bmatrix}, \tag{15}$$

$$T_k^A = \hat{s}_k \begin{bmatrix} \hat{R}_{11k} & \hat{R}_{12k} \\ \hat{R}_{21k} & \hat{R}_{22k} \end{bmatrix} T_{k-1}^A + \begin{bmatrix} \hat{tx}_k \\ \hat{ty}_k \end{bmatrix}. \tag{16}$$

C. Intentional Motion Estimation and Motion Compensation

When the camera moves with the user, i.e., the frames in the video observe an intentional motion, we should not compensate for the global motion directly. Instead, we should estimate the desired motion caused by the user and compensate only for the undesired motion caused by camera vibration. For intentional motion estimation, we apply the Kalman filter-based technique proposed in [4]. The input of the Kalman filter is the accumulative rotation matrix R_k^A, translation vector T_k^A, and scaling factor s_k^A. With the assumption that the desired motion takes place with constant velocity, the velocity variables for all six parameters are introduced into the state-space model. Following the independence assumption made above, the six parameters can be modeled separately, which leads to simple state transition and observation models. For example, translation along x-axis Tx and the velocity of translation Tx^v follow the state transition process given by

$$\begin{bmatrix} Tx_k \\ Tx_k^v \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Tx_{k-1} \\ Tx_{k-1}^v \end{bmatrix} + \begin{bmatrix} 0 \\ n \end{bmatrix} \tag{17}$$

where n is the state noise. The observation matrix is set to be identity, so the observation equals estimation plus observation noise.

Implementing Kalman filtering is a recursive way to get the intentional rotation matrix R_k^D and translation vector T_k^D, scaling s_k^D for frame k. We can then compensate for the unwanted motion by applying the following inverse transform

$$\begin{bmatrix} u_s \\ v_s \end{bmatrix} = \frac{s_k^D}{s_k^A} R_k^D \left(R_k^A\right)^{-1} \left(\begin{bmatrix} u_k \\ v_k \end{bmatrix} - T_k^A\right) + T_k^D \tag{18}$$

where $[u_k, v_k]$ and $[u_s, v_s]$ are pixel locations of unstable frame k and the stabilized output, respectively.

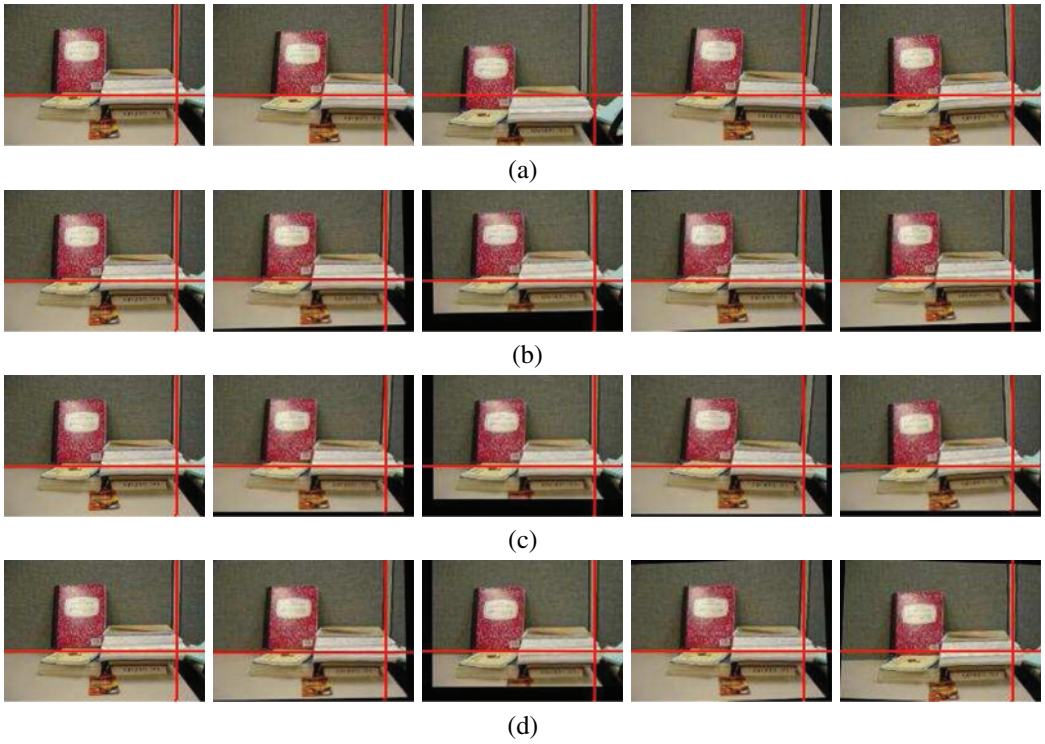

Fig. 2. Indoor scenario: Comparison of video stabilization methods for the ONDESK video sequence: (a) Original image. (b) Matched-feature-based motion estimation (MFME). (c) p-norm cost function-based motion estimation (CFME). (d) Proposed particle filtering-based motion estimation (PFME).

To summarize, the algorithm for each time step at time k is given as follows.

1) Load two successive frames: frame k and frame $k-1$.
2) Extract and match SIFT feature points in the two images.
3) Reject the features that might corresponds to moving objects by detecting the motion speed.
4) Compute $\bar{\mathbf{x}}_k$ vector from SIFT feature matches as in (10).
5) Use particle filtering framework to estimate the global motion between frame k and frame $k-1$:
   a) for $i = 1 : N$, generate particles from an Gaussian importance density, as in (11);
   b) for $i = 1 : N$, calculate the normalized weight from the importance sampling, as in (14);
   c) form the estimation using the weighted sum of samples, as in (8).
6) Calculate the accumulative motion as in (15) and (16).
7) Use Kalman filter to estimate intentional motion from accumulative motion.
8) Reconstruct the stabilized image, as in (18).

V. Experimental Results

We test the effectiveness of our algorithm in stabilization of several real-life video sequences captured by a handheld digital camera. The experiments are performed using MATLAB on a 3.2 GHz Pentium IV PC. We compare three methods for video stabilization: 1) global motion estimation with p-norm cost function-based motion estimation (CFME) [4]; 2) global motion estimation with matched feature-based motion estimation (MFME) [5]; and 3) the proposed approach to global motion estimation with particle filtering-based motion estimation (PFME). In order to ascertain the impact of particle filtering on global motion estimation, we modify the MFME approach to use the same feature points proposed in our approach to PFME, i.e., SIFT points. All three video stabilization techniques are implemented on a frame-to-frame basis. The particle filters use 30 particles per frame in all cases. In the stabilized sequences, black areas along the boundary result from undefined regions that are beyond the field of view of the camera. However, the use of mosaicking to extend the field of view is beyond the scope of this paper. Besides, we use red markings for easy visualization of the video stabilization results. Also note that some of the scenes (e.g., in video ONROAD) have nonuniform depths, which in principle violates the assumption of the 2-D affine model. However, the results still have good quality, which in turn shows the 2-D affine model is an accurate approximation of the 3-D motion.

The video clip ONDESK has been captured with an indoor background and no moving objects. It has 120 frames with resolution 160×120 and frame rate 15 frames/s. A comparison of video stabilization based on MFME and CFME with the proposed video stabilization method based on PFME is depicted in Fig. 2. Snapshot images of the original video sequence corresponding to Frames 1, 13, 50, 75, and 114 are illustrated in Fig. 2(a). The vertical red line passes through a gray line on the wall, and the horizontal red line represents the intersection of the wall and the desk in the first frame. The results of video stabilization using the methods based on MFME, CFME, and

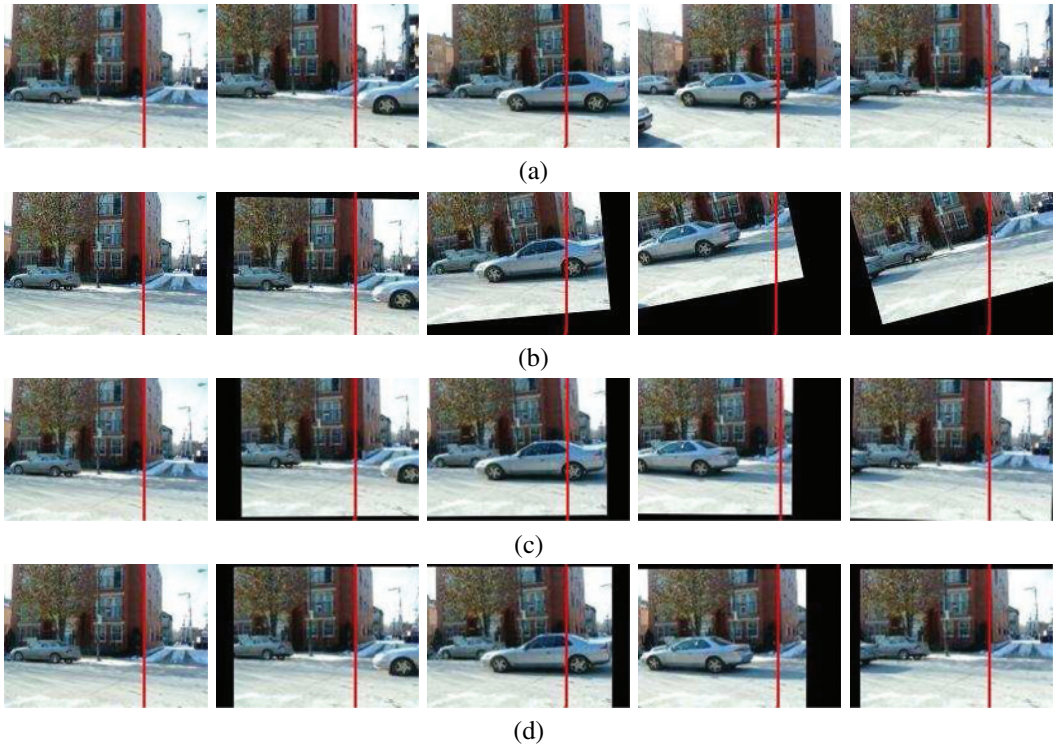

Fig. 3. Outdoor scenario I: Comparison of video stabilization methods for the STREET video sequence: (a) Original image. (b) Matched-feature-based motion estimation (MFME). (c) p-norm cost function-based motion estimation (CFME). (d) Proposed particle filtering-based motion estimation (PFME).

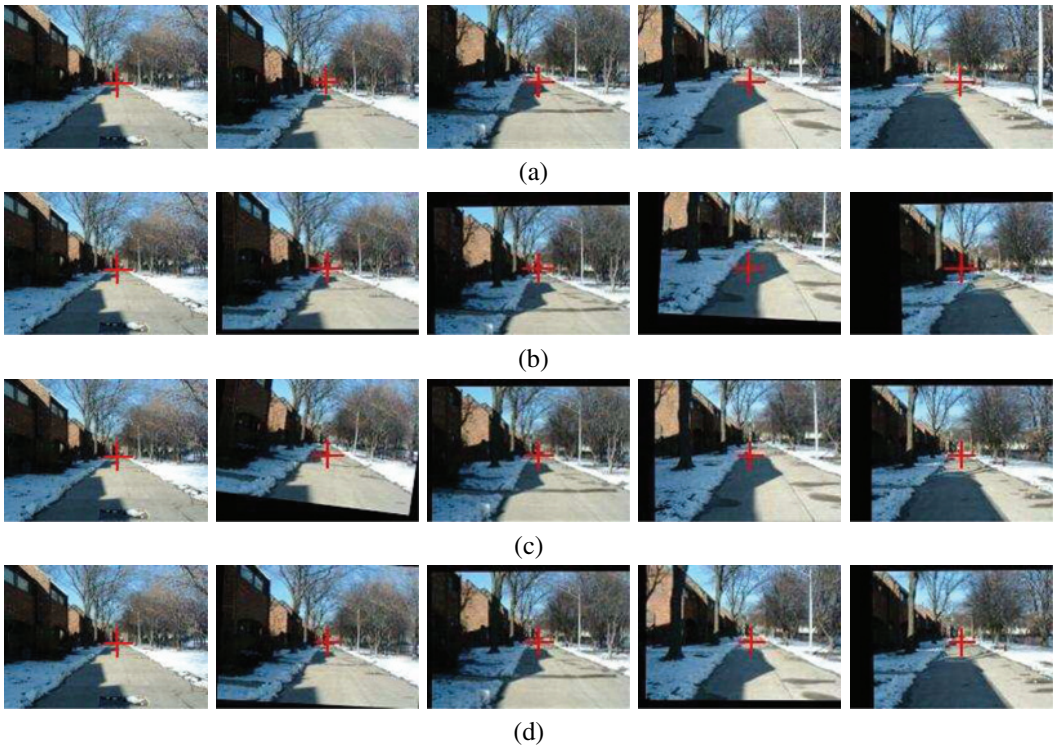

Fig. 4. Outdoor scenario II: Comparison of video stabilization methods for the ONROAD video sequence: (a) Original image. (b) Matched-feature-based motion estimation (MFME). (c) p-norm cost function-based motion estimation (CFME). (d) Proposed particle filtering-based motion estimation (PFME).

PFME are depicted in Fig. 2(b)–(d), respectively. We observe that all three output sequences remove the unwanted motion of the camera and improve the video images. We further observe that when the camera rotates (frames 75 and 114), the rotation angles of the recovered images using the CFME are less accurate than the corresponding images for the MFME and PFME methods. In this scenario, we conclude that when the feature correspondences are correct, the MFME and PFME

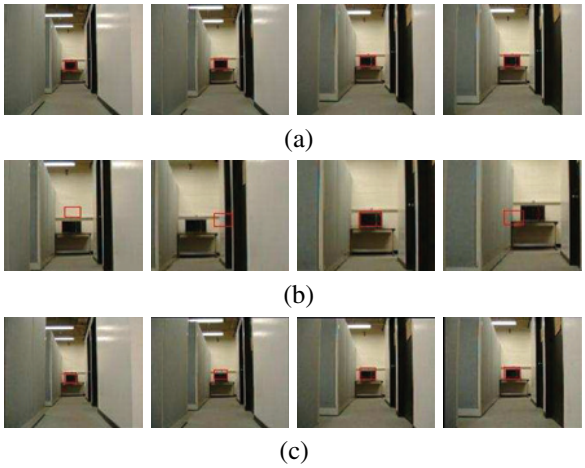

Fig. 5. Synthetic video: Performance of video stabilization algorithm. (a) Original LAB_Stable video sequence (ground truth). (b) Synthetic LAB_Unstable video sequence. (c) Proposed particle filtering-based motion estimation (PFME).

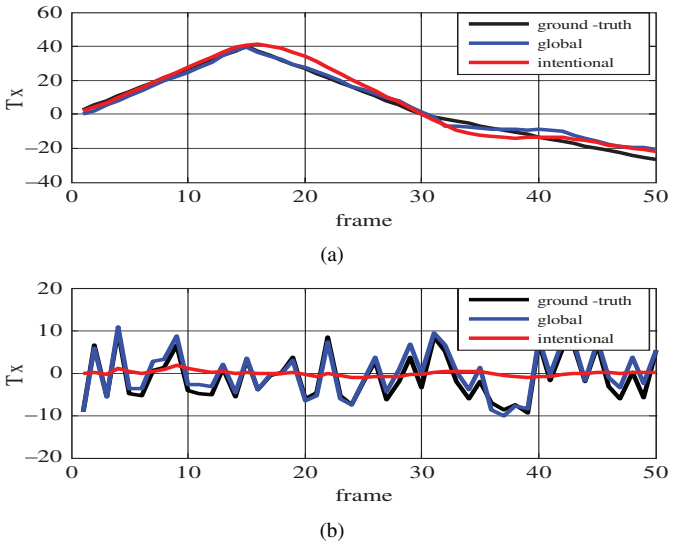

Fig. 6. Comparison between the ground-truth representation of the global camera motion (black line), the estimated global camera motion using the PFME method (blue line), and the estimated intentional camera motion (red line), for translation (in pixels) in both the horizontal and vertical directions. (a) Motion in horizontal direction. (b) Motion in vertical direction.

provide more accuracy compared to the CFME-based method, whereas the performance of the MFME and the proposed PFME is nearly identical.

We capture an outdoor video STREET (200 frames, 320 × 240, 15 frames/s), which contains a car moving across the camera and a building in the background. This scene is challenging due to the presence of moving objects and image blurring due to fast camera vibrations. Snapshots of the original video clip corresponding to Frames 1, 104, 110, 114 and 120 are depicted in Fig. 3(a). The vertical red line marks the front edge of the building in Frame 1. Stabilization results based on MFME, CFME, and the proposed PFME are presented in Fig. 3(b)–(d), respectively. We observe that the MFME-based stabilization loses the ability to stabilize the scene since the global motion estimation suffers from incorrect feature matching due to blurring. One illustration of the incorrect matchings in the background has been provided earlier in Fig. 1. On the other hand, despite the fact that the particle filter relies on the same incorrect feature matching, PFME is still able to recover the correct global camera motion and stabilize the video sequence. This experiment demonstrates that particle filtering is resistant to errors due to incorrect feature matching and yields a robust method for video stabilization. Also, we can see that CFME is robust with outliers and gives almost the same performance as PFME except for being less accurate (e.g. frame 114).

The outdoor video clip ONROAD (400 frames, 320 × 240, 30 frames/s) has been captured with a camera moving in the optical direction, which is a popular example of intentional camera motion. Frames 1, 96, 166, 244, and 324 of the original video sequence are illustrated in Fig. 4(a). The red cross marks the vanishing point of the road in the first frame, and its location is fixed throughout the snapshots presented. We once again compare the results of video stabilization using MFME, CFME, and the proposed PFME in Fig. 4(b)–(d), respectively. As can be seen, in this scenario both MFME and CFME produce errors in some of the video frames, which yield annoying vibrations in the resulting video sequence. Similar as in STREET, the errors of MFME are resulting from incorrect feature matching, and CFME is less accurate. The red cross remains in the road vanishing point in all of the frames of the PFME, which points to the superiority of the PFME-based video stabilization algorithm.

We test our algorithm on synthetic videos where we record a stable video LAB_Stable (150 frames, 320 × 240, 15 frames/s) with the camera observing only intentional motion along the optical axis toward the object and no vibrations. It is used as the ground-truth video. We artificially produce an unstable video sequence LAB_Unstable by applying predetermined affine transforms to every frame in LAB_Stable, which serve as ground-truth for image motions. A quantitative evaluation of the performance of the various video stabilization algorithms are given. Fig. 5(a) illustrates Frames 30, 60, 90, and 120 of the original stable video LAB_Stable. The corresponding and enlarged snapshots of the unstable video LAB_Unstable generated from original video sequence is depicted in Fig. 5(b). The results of video stabilization using the proposed PFME are provided in Fig. 5(c). The red rectangle denotes the location of a microwave oven in each frame of the original stable video sequence. As we can see in Fig. 5(c), the red rectangle outlines the image of the microwave oven nearly perfectly throughout the stabilized video sequence using the PFME-based method. This illustration shows that the output sequence using the proposed video stabilization algorithm is very close to the original stable sequence, and demonstrates the efficiency of our system.

Fig. 6 provides a graphical illustration of the ground truth of the global camera motion, the estimated global camera motion using PFME, and the estimated intentional camera motion, for horizontal and vertical translations of the first 50 frames of the LAB video sequence. We observe that the global motion estimation using the proposed method is very close to the ground truth. Following the evaluation measures proposed in [34], we also compute the MSE and PSNR for LAB_Unstable,

TABLE I
COMPARISON OF AVERAGE MSE AND PSNR FOR STABILIZED OUTPUT

Statistics	Unstable	MFME	CFME	PFME
Average MSE	4238.9	2282.7	2373.5	1332.9
Average PSNR (dB)	11.858	15.549	15.494	17.500

as well as for the stabilized output produced by three methods (CFME, MFME, and PFME). A summary of the average MSE and average PSNR results of the whole 150 frames is provided in Table I. We observe that PSNR for the stabilized outputs produced by the three methods are all higher than those of the unstable video. Among the three, PFME gives the highest PSNR. Specifically, the average MSE of PFME is nearly one-half of the corresponding MSE using MFME and CFME, whereas the average PSNR improvement for PFME is more than 50% larger than the corresponding PSNR gains of MFME and CFME.

VI. CONCLUSION

In this paper, we presented a novel approach for robust video stabilization based on particle filter estimation of projected camera motion. Our proposal has been to extend particle filtering to the estimation and tracking of the global camera motion parameters in video sequences. An efficient implementation of particle filters for global motion estimation has been proposed based on carefully designed importance sampling. We relied on corresponding SIFT points to obtain an estimation of the camera motion model. We then generated particles by sampling from a density function characterized by the estimated camera motion model. We also proved analytically that particle filtering can be used to reduce the variance of a time series estimate and thus yield a smooth and more accurate estimate when the number of particles is sufficiently large. We demonstrated experimentally that the proposed particle filtering scheme can be used to obtain an efficient and accurate motion estimation in video sequences.

APPENDIX
PROOF OF LEMMA 1

Without loss of generality, we can denote the true state $\mathbf{x}_k$ as in the origin of the state-space to simplify the notations. In such case, $\varepsilon_k \equiv \hat{\mathbf{x}}_k$ and $\mathbf{e}_k \equiv \bar{\mathbf{x}}_k$. Both of them have zero mean.

Proof: Note that

$$E\left[\mathbf{x}_k^i\right] = E\left[E\left[\mathbf{x}_k^i|\bar{\mathbf{x}}_k\right]\right] = E\left[\bar{\mathbf{x}}_k\right] = 0 \tag{19}$$

$$E\left[\mathbf{x}_k^i \mathbf{x}_k^{i\,T}\right] = E\left[E\left[\mathbf{x}_k^i \mathbf{x}_k^{i\,T}|\bar{\mathbf{x}}_k\right]\right] = E\left[\Sigma_1 + \bar{\mathbf{x}}_k \bar{\mathbf{x}}_k^T\right] = \Sigma_1 + \Sigma_2. \tag{20}$$

In the following derivation, we assume that w_k^i and $\mathbf{x}_k^i$ are statistically independent. The assumption is meaningful when we consider that the posterior function is unknown and particle samples are drawn from an arbitrary proposal density. Therefore, the distribution of the random samples $\mathbf{x}_k^i$ from the proposal density, and the weight w_k^i used to estimate the posterior function are assumed to be independent. From this assumption, the error variance of ε_k can be obtained as

$$Cov(\varepsilon_k, \varepsilon_k) = E\left[\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T\right] = \sum_{i=1}^{N}\sum_{j=1}^{N} E\left[w_k^i w_k^j \mathbf{x}_k^i \mathbf{x}_k^{j\,T}\right] \tag{21}$$

$$= \sum_{i=1}^{N} E\left[\left(w_k^i\right)^2\right](\Sigma_1 + \Sigma_2). \tag{22}$$

Equation (21) comes from (8). We then separate the summation of j to two portion $j = i$ and $j \neq i$. Equation (22) holds due to the independence assumption of w_k^i, and $\mathbf{x}_k^i$, and following from (19) and (20).

Further, the normalized weights w_k^i for frame k are calculated from the importance sampling function, i.e., $w_k^i = \pi_k^i / \sum_{i=1}^{N} \pi_k^i$, where π_k^i are the likelihood computed for i.i.d particles, and hence can be regarded as i.i.d random variables and have finite mean. Denote their mean and variance as m_π and σ_π^2, varying with k. When the number of particles N is large, according to Kolmogorov's Strong Law of Large Numbers [35], the sum of π_k^i can be approximated by $\sum_{i=1}^{N} \pi_k^i \simeq N E\left[\pi_k^i\right] = N m_\pi$. Therefore

$$\sum_{i=1}^{N} E\left[\left(w_k^i\right)^2\right] \simeq \sum_{i=1}^{N} \frac{E\left[\left(\pi_k^i\right)^2\right]}{\left(\sum_{i=1}^{N} \pi_k^i\right)^2} = \sum_{i=1}^{N} \frac{m_\pi^2 + \sigma_\pi^2}{N^2 m_\pi^2} = \frac{m_\pi^2 + \sigma_\pi^2}{N m_\pi^2}. \tag{23}$$

Denote $c_k \triangleq (m_\pi^2 + \sigma_\pi^2)/m_\pi^2$, we can further simplify (22) to $Cov(\varepsilon_k, \varepsilon_k) = \frac{1}{N}(\Sigma_1 + \Sigma_2)c_k$. ■

REFERENCES

[1] N. Gordon, M. Arulampalam, S. Maskell, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[2] S. Erturk, "Digital image stabilization with sub-image phase correlation based global motion estimation," *IEEE Trans. Consumer Electron.*, vol. 49, no. 4, pp. 1320–1325, Nov. 2003.

[3] S. Erturk and T. Dennis, "Image sequence stabilization based on DFT filtering," *Proc. IEEE Image Vision Signal Process.*, vol. 127, no. 2, pp. 95–102, Apr. 2000.

[4] A. Litvin, J. Konrad, and W. Karl, "Probabilistic video stabilization using kalman filtering and mosaicking," in *Proc. Image Video Commun. IS&T/SPIE Symp. Electron. Imaging*, Santa Clara, CA, pp. 663–674, 2003.

[5] C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization for off-road navigation," *Real-Time Imaging*, vol. 2, pp. 285–296, 1996.

[6] C. Morimoto and R. Chellappa, "Fast 3-D stabilization and mosaic construction," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 660–665.

[7] J. Jin, Z. Zhu, and G. Xu, "Digital video sequence stabilization based on 2.5-D motion estimation inertial motion filtering," *Real-Time Imaging*, vol. 7, pp. 357–365, 2001.

[8] Z. Duric and A. Rosenfeld, "Image sequence stabilization in real time," *Real-Time Imaging*, vol. 2, pp. 271–284, 1996.

[9] A. Censi, A. Fusiello, and V. Roberto, "Image stabilization by feature tracking," in *Proc. Int. Conf. Anal. Process.*, Venice, Italy, 1999, pp. 665–667.

[10] Y. Liang, H. Tyan, and S. Chen, "Video stabilization for a camcorder mounted on a moving vehicle," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1636–1648, Nov. 2004.

[11] J. Chang, W. Hu, M. Cheng, and B. Chang, "Digital image translation and rotation motion stabilization using optical flow technique," *IEEE Trans. Consumer Electron.*, vol. 48, no. 1, pp. 108–115, Feb. 2002.
[12] S. Ko, S. Lee, S. Jeon, and E. Kang, "Fast digital image stabilizer based on gray-coded bit-plane matching," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 598–603, Aug. 1999.
[13] J. Pail, Y. Park, and D. Kim, "An adaptive motion decision system for digital image stabilizer based on edge pattern matching," *IEEE Trans. Consumer Electron.*, vol. 38, no. 3, pp. 607–615, Aug. 1992.
[14] S. Piva, M. Zara, G. Gera, and C. Reegazzoni, "Color-based video stabilization for real-time on-board object detection on high-speed trains," in *Proc. IEEE Conf. Advanced Video Signal Based Surveillance*, Miami, FL, 2003, pp. 299–304.
[15] C. Chen, J. Yang, D. Schonfeld, and M. Mohamed, "Pose estimation from video sequences based on sylvester's equation," in *Proc. SPIE Electron. Imaging, Conf. Visual Commun. Image Process.*, San Jose, CA, 2007, pp. 1S.1–1S.12.
[16] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.* Ser. D, vol. 82, pp. 35–45, 1960.
[17] N. Gordon, D. Salmond, and A. Smith, "Novel approach to non-linear and non-gaussian bayesian state estimation," *Proc. IEEE-F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
[18] M. Isard and A. Blake, "Condensation conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, Aug. 1998.
[19] A. Doucet, J. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. 1st ed. New York: Springer-Verlag, 2001, ch. 2, sec. 2, pp. 15–38.
[20] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statist. Computing*, vol. 10, no. 3, pp. 197–208, 2000.
[21] M. Isard and A. Blake, "A smoothing filter for condensation," in *Proc. 5th Eur. Conf. Comput. Vision*, vol. 1. Freiburg, Germany, 1998, pp. 767–781.
[22] M. Klass, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoother: If I had a million particles," in *Proc. 23rd Int. Conf. Mach. Learning*, Pittsburgh, PA, 2006, pp. 25–29.
[23] W. Fong, S. Godsill, A. Doucet, and M. West, "Monte carlo smoothing with application to audio signal enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 438–449, Feb. 2002.
[24] M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," in *Proc. 5th Eur. Conf. Comput. Vision*, vol. 1. Freiburg, Germany, 1998, pp. 893–908.
[25] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Kauai, Hawaii, 2001, pp. 786–793.
[26] N. Bouaynaya, W. Qu, and D. Schonfeld, "An online motion-based particle filter for head tracking applications," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Philadelphia, PA, 2005, pp. 225–228.
[27] J. M. Odobez and D. Gatica-Perez, "Embedding motion in model-based stochastic tracking," in *Proc. Int. Conf. Pattern Recognition*, vol. 2. Cambridge, U.K., 2004, pp. 815–818.
[28] Y. Matsushita, E. Ofek, X. Tang, and H. Shum, "Full-frame video stabilization," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognition*, San Diego, CA, 2005, pp. 50–57.
[29] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* 1st ed. Englewood Cliffs, NJ: Prentice Hall, 2002, ch. 2, sec. 1.2, pp. 9–16.
[30] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim, "Pose estimation from corresponding point data," *IEEE Trans. Syst., Man. Cybern.*, vol. 19, no. 6, pp. 1426–1446, Dec. 1989.
[31] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 736–746, Mar. 2002.
[32] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proc. ECCV*, vol. 2. Dublin, Ireland, 2000, pp. 3–19.
[33] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Dec. 2004.
[34] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5. Seattle, WA, 1998, pp. 2789–2792.
[35] S. I. Resnick, *A Probability Path*. 1st ed. Cambridge, MA: Birkhauser, 2001, ch. 7, sec. 7.4, pp. 213–219.

Junlan Yang (S'06) received the B.S. in information engineering in 2005 from Zhejiang University, Hangzhou, China. She is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Illinois, Chicago.

She has been a Research Assistant at the University of Illinois since 2005.

Ms. Yang received the IBM Student Paper Award at the IEEE International Conference on Image Processing in 2007.

Dan Schonfeld (M'90–SM'05) received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Johns Hopkins University, Baltimore, MD, in 1986, 1988, and 1990, respectively.

In August 1990, he joined the Department of Electrical Engineering and Computer Science, University of Illinois, Chicago, where he is currently a Professor in the Departments of Electrical and Computer Engineering, Computer Science, and Bioengineering, and Co-Director of the Multimedia Communications Laboratory and member of the Signal and Image Research Laboratory. He has authored over 170 technical papers in various journals and conferences.

Magdi Mohamed (S'93–M'93–A'95) received the B.Sc. degree in electrical engineering from the University of Khartoum, Khartoum, Sudan, and the M.S. degree in computer science and the Ph.D. degree in electrical and computer engineering, both from the University of Missouri, Columbia, in 1983, 1990 and 1995, respectively.

He was a Visiting Professor at the University of Missouri, Columbia from 1995 to 1996. He is currently working as a Principal Staff Engineer at the Physical and Digital Realization Research Center of Excellence, Motorola Labs, Schaumburg, IL, where he has been since 1996.

Dr. Mohamed is a member of Motorola's Science Advisory Board Associates.