ONLINE VIDEO STABILIZATION BASED ON PARTICLE FILTERS

*Junlan Yang, Dan Schonfeld, Chong Chen**

Multimedia Communication Lab
ECE Dept., University of Illinois
Chicago, IL, 60607
{jyang24,dans,cchen49}@uic.edu

Magdi Mohamed

Physical Realization Research Center of Excellence
Motorola Labs
Schaumburg, IL, 60196
Magdi. Mohamed@motorola.com

ABSTRACT

Particle filters have been introduced as a powerful tool to estimate the posterior density of nonlinear systems. These filters are also capable of processing data online as required in many practical applications. In this paper, we propose a novel technique for video stabilization based on the particle filtering framework. Scale-invariant feature points are extracted to form a rough estimate which is used to model the importance density. We use a constant-velocity Kalman filter model to estimate intentional camera movement. We also prove that the particle filtering estimate will lower the error variance. The superior performance and robustness of our algorithm is demonstrated by computer simulations.

Index Terms— Image motion analysis, Image sequence analysis, Particle tracking, Monte Carlo methods

1. INTRODUCTION

Cameras mounted on hand-held devices and mobile platforms such as cars and planes capture unstable images. Rattled camera motion and platform vibrations degrade the visual quality of video images. Different video stabilization techniques have been proposed to construct stable images. A critical step in stabilization is motion estimation. In [1], a six-parameter affine model is used to describe the inter-frame transformation. The model parameters are estimated by minimizing a p-norm-based cost function. Sung-Jea Ko et.al employs a gray-coded bit-plane matching [2] to estimate fast motion. Sub-image phase correlation based global motion estimation is proposed by S. Erturk [3], to find the translation along x and y axis. These are intensity based algorithms, while in [4], C. Morimoto and R. Chellappa solve the 2D motion equations using features on the horizon, which is a strong visual cue in off-road situations. In [5], the authors deal with on-road situation by detecting the lane lines and road vanishing point in the image as global features.

Our system employs a particle filtering [6] framework for global motion estimation. Particle filters are commonly used in video tracking and pose estimation, but not yet applied to video stabilization. It can be seen in this paper that particle filter allows us to solve the problem with both feature matching and intensity optimization method. Therefore, our algorithm is accurate due to the effective feature points extracted and more robust than common feature tracking method.

The rest of this paper is organized as follows. In Section 2, we describe the framework of estimating global affine motion using particle filters. Then in Section 3, we discuss the way of dealing with intentional motion and motion compensation. Experimental results are presented in Section 4 and Section 5 draws the conclusion.

2. PARTICLE FILTERING FOR GLOBAL AFFINE MOTION ESTIMATION

Assume a hand-held camera is looking at a static scene; however, the video frames obtained suffer from jitters produced by the camera's 3D rotation $R_{3\times 3}$ and translation $L_{3\times 1}$ due to the unsteady hand. Under the 3D camera model, pixel locations (x, y) in image frame k and (x', y') in frame $k+1$ are related by

$$\begin{bmatrix} x' \\ y' \\ \lambda \end{bmatrix} = s \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ \lambda \end{bmatrix} + \begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix} \quad (1)$$

where s is a scale factor and λ is the focal length. From the first two columns of the matrix, we get

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} sR_{11} & sR_{12} \\ sR_{21} & sR_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} sR_{13}\lambda + l_x \\ sR_{23}\lambda + l_y \end{bmatrix} \quad (2)$$

Equ. (2) shows that using a 2D affine transformation one can get the same visual effects as using a 3D transformation, with different parameters. With the assumption that the rotation angle outside the image plane is quite small between successive frames, and the scene is far away from the camera thus $s = 1$, the above model can be simplified as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} Tx_k \\ Ty_k \end{bmatrix} \quad (3)$$

*The authors would like to thank the Physical Realization Research Center of Excellence, Motorola Labs, for supporting this work.

where θ_k denotes the rotation angel in the image plane, and Tx_k, Ty_k are translation displacements along x and y axis respectively. Denote the first frame to be the stable reference frame, our task in motion estimation is to determine the above three parameters referring to the reference frame.

2.1. Particle Filtering Framework

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 $\mathbf{x}_k = [Tx_k, Ty_k, \theta_k]^T$ and a measurement $\mathbf{z}$. The fundamental idea of particle filter estimation is to recursively approximate the posterior pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ by a set of particles $\{\mathbf{x}_k^i, i = 0, 1, ...N\}$ with associated weights $\{w_k^i, i = 1, 2, ...N\}$, 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 $\mathbf{x}^i \sim q(\mathbf{x})$ are drawn from a proposal $q(\cdot)$, which is called importance density. Here we generate samples $\mathbf{x}_k^i$ based on a Gaussian importance density stated as follows:

$$\mathbf{x}_k^i = \hat{\mathbf{x}}_{k-1} + \mathbf{n}_k^i \qquad (4)$$

$$\mathbf{n}_k^i \sim N_G(\Delta\mathbf{x}_k, \Sigma) \qquad (5)$$

where $\hat{\mathbf{x}}_{k-1}$ is the estimation of the state in time step $k-1$. We add the previous motion because we want all the parameters estimated are relative to the reference frame, which can directly been used for compensation. $N_G(\Delta\mathbf{x}_k, \Sigma)$ refers to the Gaussian function, with mean $\Delta\mathbf{x}_k$ and covariance matrix Σ. In this problem, it is reasonable to assume that the three dimensions of Gaussian function are independent. The mean of Gaussian function, $\Delta\mathbf{x}_k = [\Delta Tx_k, \Delta Ty_k, \Delta\theta_k]^T$ can be simply set to zero in the case of a prior distribution. We will further explore this issue in Section 2.2.

The desired weights should perform as an evaluation on how close the state suggested by each particle is to the true state. Since we have N guesses of the transformation matrix, we apply all the N inverse transforms to an original image and get N candidate images $\mathbf{A}_i$. Then we compare these images with the reference image $\mathbf{A}_0$ to tell the similarities between them, hence the weights of particles. We choose Mean-Square-Error and edge wrapping technique for comparison. MSE comparison calculates the mean-square error M_i of the grayscale from pixel to pixel between $\mathbf{A}_i$ and $\mathbf{A}_0$. The likelihood is a decreasing function of M_i given by

$$P_{MSE}^i \propto \frac{1}{\sqrt{2\pi}\sigma_M} \exp\{-\frac{{M_i}^2}{2\sigma_M^2}\} \qquad (6)$$

The edge wrapping comparison employs the edge detection technique proposed in [7]. We can construct two images containing only edge portions from both $\mathbf{A}_i$ and $\mathbf{A}_0$, respectively. Then we calculate the correlation R_i of two edge images. The likelihood of edge wrapping is given by

$$P_{edge}^i \propto \frac{1}{\sqrt{2\pi}\sigma_E} \exp\{-\frac{{R_i}^2}{2\sigma_E^2}\} \qquad (7)$$

where σ_M and σ_E in Equ.(6) and Equ.(7) are adjustable standard deviations. Therefore, the normalized weights are given by

$$w_k^i = \frac{P_{MSE}^i P_{edge}^i}{\sum_{i=1}^N P_{MSE}^i P_{edge}^i} \qquad (8)$$

Once we obtain the weight for each particle, we will approach the true state by a discrete weighted approximation,

$$\hat{\mathbf{x}}_k = \sum_{i=1}^N w_k^i \mathbf{x}_k^i \qquad (9)$$

where the estimated state tells the estimated values of global affine motion parameters, $\hat{\mathbf{x}}_k = [\hat{Tx}_k, \hat{Ty}_k, \hat{\theta}_k]^T$.

2.2. Importance Density Using Scale-Invariant Features

The choice of a good importance density is the crucial step in the design of particle filter. A technique is proposed here to encourage particles be generated close to true posterior density.

We use feature tracking to get the means of the Gaussian density function, ΔTx_k, ΔTy_k, $\Delta\theta_k$, which have been mentioned in Section 2.1. The feature points we use are obtained based on SIFT algorithm [8]. SIFT extracts and connects feature points in images which are invariant to image scale, rotation and changes in illumination. Once we have corresponding pairs, we can use them to determine the transform matrix between two images. Equ.(3) can be rewritten as

$$\begin{bmatrix} x' & y' \\ .. & .. \end{bmatrix} = \begin{bmatrix} x & y & 1 \\ .. & .. & 1 \end{bmatrix} \begin{bmatrix} \cos\Delta\theta_k & -\sin\Delta\theta_k \\ \sin\Delta\theta_k & \cos\Delta\theta_k \\ \Delta Tx_k & \Delta Ty_k \end{bmatrix} \qquad (10)$$

where (x, y) and (x', y') are corresponding feature points. We need only two pairs to determine an 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 easily solved under Least-Square criteria by $\mathbf{A} = [\mathbf{X}^T\mathbf{X}]^{-1}\mathbf{X}^T\mathbf{Y}$.

The values of ΔTx_k, ΔTy_k and $\Delta\theta_k$ serve as a rough estimation and hence the means of importance density function. It helps us to avoid generating useless particles and hence to keep the computation cost low enough to achieve online performance.

2.3. Properties of Particle Filtering Estimation

Instead of using directly the rough estimation $\Delta\mathbf{x}_k$ yielded from (10), we use an advanced particle filtering algorithm for estimation due to its desirable properties stated below.

1) Smoothing property of particle filters. In the situation we described above, the errors between true state and two estimations:$\varepsilon_k = \hat{\mathbf{x}}_k - \mathbf{x}_k$ and $\mathbf{e}_k = \Delta\mathbf{x}_k - \mathbf{x}_k$ is evaluated by calculating their variance $Var(\varepsilon_k)$ and $Var(\mathbf{e}_k)$. It can be shown that, by satisfying that

$$\sum_{i=1}^{N}(w_k^i)^2 \leq c_k \leq 1, \tag{11}$$

we can achieve $Var(\varepsilon_k) < Var(e_k)$, in all the three dimensions of the state. c_k is a constant related to the importance density and w_k^i is the normalized weight obtained from (8). And we can also prove that the above requirement is always satisfied by a sufficiently large N, under the constraint that$\sum_{i=1}^{N} w_k^i = 1$. It means that particle filtering gives estimation with lower error variance, hence the estimation is smoother over time.

2) *Convergence property of particle filters.* In [9], it is shown that the estimation of particle filter converges in mean square sense, and the rate of convergence is in $1/N$. Therefore, in every frame, the estimation of the motion vector converges very fast to the true values.

3) *Robustness of particle filters.* SIFT algorithm might connect wrong feature points, especially when pictures are blurred by rapid shakes of the camera. Computing the transform matrix from incorrect correspondences will produce bad results. On the contrary, particle filter performs well even when the output of SIFT is inaccurate. Since particle filter relies on samples around the SIFT output rather than the output itself, and incorporates different properties of the images, it is more resistent to mistakes that single algorithm would make. The practical necessity will be proved by the experimental results reported in Section 4.

3. INTENTIONAL MOTION ESTIMATION AND MOTION COMPENSATION

When the camera moves with the user, namely, the frames in the video observe an intentional motion, we cannot compensate for the global motion directly. Instead, we should estimate the desired motion and compensate for the unwanted motion caused by camera vibration. We use the Kalman filter based estimation technique proposed in [1]. Once we get the intentional motion vector $[Tx_k^d, Ty_k^d, \theta_k^d]^T$, we can compensate for the unwanted motion by applying the following inverse transform

$$\begin{bmatrix} x^s \\ y^s \end{bmatrix} = \begin{bmatrix} \cos\beta_k & -\sin\beta_k \\ \sin\beta_k & \cos\beta_k \end{bmatrix} \left(\begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} Tx_k^d - \hat{Tx}_k \\ Ty_k^d - \hat{Ty}_k \end{bmatrix} \right) \tag{12}$$

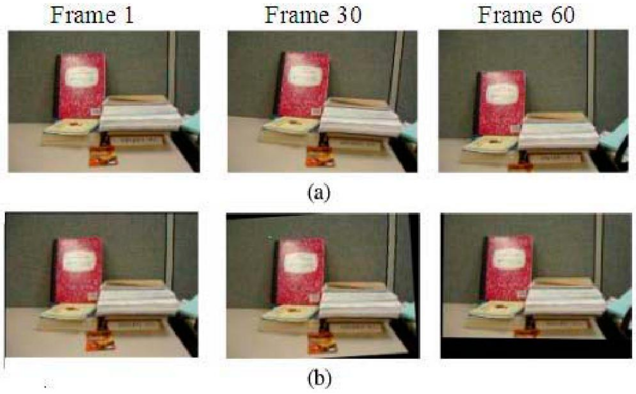
where $\beta_k = \theta_k^d - \hat{\theta}_k$. (x', y') and (x^s, y^s) is the pixel locations of unstable and stabilized image, respectively.

Another important issue here is how to deal with moving objects. In a situation that background moves with the shaking camera, we can tell the moving objects by noticing the moving velocities of different parts in the images. It can be assumed that the velocity of the moving objects is much larger than that of the background. Therefore, by comparing two successive frames, the parts which have moved an extraordinarily large distance should be isolated before estimating the global motion and intentional motion.

4. EXPERIMENTAL RESULTS

We test the effectiveness of our algorithm in several real-life video sequences captured by a hand held digital camera without any image stabilization technique. Here we report some of the experimental results.

Fig.1 shows the frames 1 (reference frame), frame 30, frame 60 of an indoor static scene sequence, with both original images (Fig.1 (a)) and stabilized images (Fig.1 (b)). Note that the stabilized images remain motionless with respect to the reference frame, regardless of the rotation and translation of the camera.



(a)

(b)

Fig. 1. Original (a) and stabilized (b) images for an indoor sequence

Fig.2 (a) shows the frame 1 (reference frame), frame 50, frame 60 of an outdoor scene sequence. In this sequence, not only the camera is vibrating, but a car is also passing by in the scene. Fig.2 (b) is a test output using simply the parameters obtained from SIFT points, $\Delta\mathbf{x}_k$. Fig.2 (c) is the stabilization output of our complete particle filter algorithm. It can be noticed that when SIFT tracks wrong feature points due to the blurring, the estimation is incorrect. Yet, as shown in (c), particle filter can recover itself and give steady performance.

Fig.3 shows three frames of a road scene sequence observing intentional motion, with both original images (Fig.3 (a)) and stabilized images (Fig.3 (b)). The red cross tags the road vanishing point in the first frame, and the location is fixed through all the frames. As we can see, the red cross remains in the road vanishing point in all the three frames of the stabilized video.

Fig.4 shows the ground truth of the global motion, the estimation results of our global and intentional motion estimation. We also compute in this case the RMSE between originally stable image and unstable images, and the RMSE

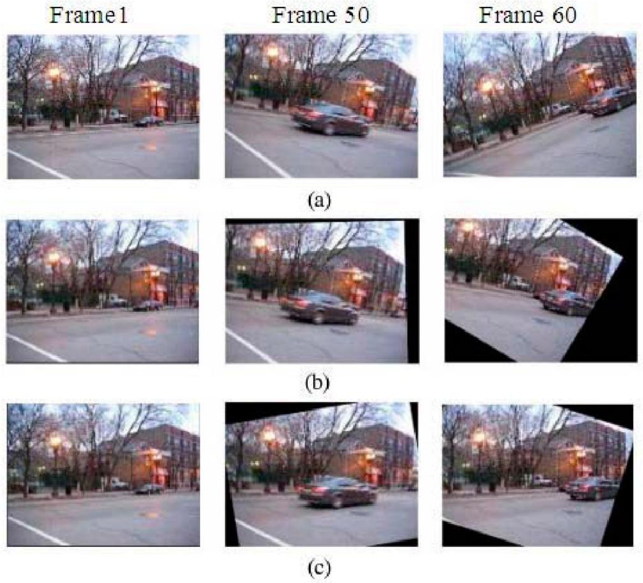


Fig. 2. Original (a) and stabilized (c) images for an outdoor sequence, with a comparison to SIFT output (b)

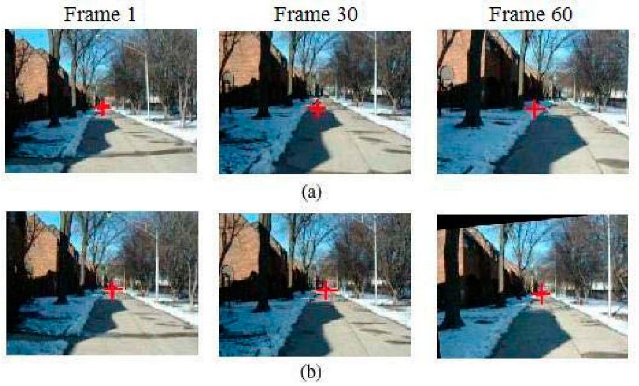


Fig. 3. Original (a) and stabilized (b) images for a road scene sequence

between originally stable image and stabilized images. The former is 362.69 in average and the latter is 59.36 in average for 200 frames. The error is reduced largely in stabilized sequence, which indicates effectiveness of the algorithm.

5. CONCLUSION

In this paper, we have presented a novel approach for video stabilization. We use particle filters to estimate the global motion between adjacent frames. Desired motion estimation has also been implemented to extract abrupt movements for compensation. Experiments have proved that our algorithm yields robust results. The flexibility of particle filters allows the algorithm be further developed and applied to variant situations.

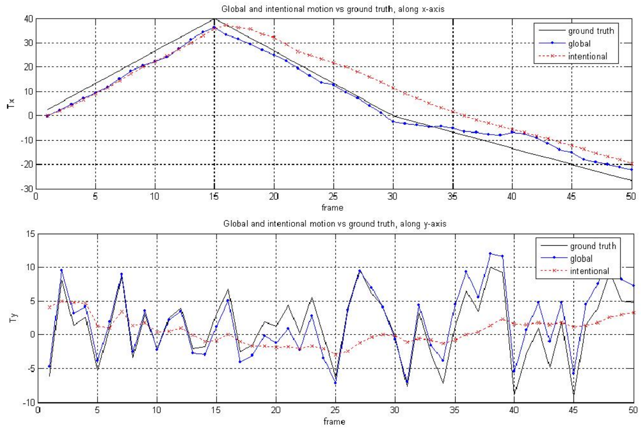


Fig. 4. A comparison between ground truth(solid line) and the global motion (solid line with marker'.') and intentional motion (dashed line with marker'+'), for both T_x and T_y respectively.

6. REFERENCES

[1] A. Litvin, J. Konrad, and W. Karl, "Probabilistic video stabilization using kalman filtering and mosaicking," *IS&T/SPIE symposium on Electronic Imaging, Image and Video Communication and Proc.*, Jan 20-24, 2003.

[2] S. Ko, S. Lee, S.Jeon, and E. Kang, "Fast digital image stabilizer based on gray-coded bit-plane matching," *IEEE Trans. on Consumer Electronics*, Aug. 1999.

[3] S.Erturk, "Digital image stabilization with sub-image phase correlation based global motion estimation," *IEEE Trans. on Consumer Electronics*, Nov. 2003.

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

[5] Y. Liang, H. Tyan, and S. Chen et.al, "Video stabilization for a camcorder mounted on a moving vehicle," *IEEE Trans. on Vehicular Technology*, Vol.53, No. 6, 2004.

[6] N. Gordon, M. Arulampalam, S. Maskell, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Processing*, Vol. 50, No. 2, 2002.

[7] P. Kovesi, "Phase congruency detects corners and edges," *Proceedings DICTA*, Dec, 2003.

[8] D.Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No.2, pp.91-110, 2004.

[9] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, 2002.