TERM PAPER REPORT

ROBUST VIDEO STABILIZATION BASED ON PARTICLE FILTER TRACKING OF PROJECTED CAMERA MOTION

EE 608A DIGITAL VIDEO PROCESSING : TERM PAPER REPORT

ROBUST VIDEO STABILIZATION BASED ON PARTICLE FILTER TRACKING OF PROJECTED CAMERA MOTION

Squadron Leader Sandeep Sharma (15104046)

Badri Narain Patro (15204267)

1. Objective

The objective of this paper is to implement video stabilization based on particle filtering framework. To show the use of particle filters to track he projected affine motion of camera model.

2. Introduction

VIDEO CAMERAS mounted on handheld devices and mobile platforms have become increasingly popular in the consumer market over the past few years. The undesired 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. Its impact increases rapidly with the rising popularity of handheld cameras and cameras mounted on moving platforms (e.g., cars).

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. Additionally, the use of corresponding feature points in a particle filtering framework helps to obtain robust tracking of the camera motion.

In this paper, we implement particle filters [1] to estimate the global camera motion between successive frames. 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. The 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.

3. Literature Survey

In earlier efforts, 2-D models [2]-[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 had adopted a variation of the 2-D affine model for the camera motion representation.

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:

(a) Intensity-based motion estimation. The image intensity-based motion estimation methods have an advantage of being inherently robust to outliers and illumination changes.
(b) feature-based motion estimation. Popular features include edge patterns [11] and corners [9]. Feature-based motion estimation algorithms are, in general, more accurate but less robust, compared with intensity-based motion estimation methods. In this paper, the global motion estimation is based on image features by tracking scale-invariant feature transform (SIFT) points.

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. This evolution in particle filters has had a profound impact on the development and popularity of various methods for object tracking in video sequences. In this paper, we extend the application of particle filters to tracking the 2-D affine transform parameters of the global camera motion alongwith feature-based importance density function. The proposed scheme can ensure that the algorithm works effectively and efficiently with a low computational cost.

4. Theoretical Background

Here, we will discuss theoretical issues of algorithm, including camera models and properties of particle filtering estimation.

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 t0 is $[x0, y0, z0]^T$. In time t1, 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 t1 is $[x1, y1, z1]^T$ (also in camera coordinate system). These two vectors of coordinates can be related by the equation

$$[x1\ y1\ z1]^T = R_{3\times3} * [x0\ y0\ z0]^T + T_{3\times1} \quad (1)$$

where $R_{3\times3}$, $T_{3\times1}$ are the opposite transform of the camera's 3-D rotation and translation, respectively. By projection, the image coordinates of P in time t0 and t1 are given by

$$[u0\ v0\ \lambda]^T = (\lambda/z0) * [x0\ y0\ z0]^T \quad (2)$$

$$[u1\ v1\ \lambda]^T = (\lambda/z1) * [x1\ y1\ z1]^T \quad (3)$$

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 [13]. With (1), (2), (3) and by rewriting the rotation matrix $R_{3\times3}$ and translation vector $T_{3\times1}$ 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 = z0/z1, tx = s R_{13}λ + (λ/z1)T_x and t_y = s R_{23}λ + (λ/z1)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-Daffine 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 $R_{3\times3}$ is orthonormal, i.e., it is constrained by the following [14]

$$R_{11}^2 + R_{12}^2 + R_{13}^2 = 1; \quad (6)$$

$$R_{21}^2 + R_{22}^2 + R_{23}^2 = 1 \quad (7)$$

$$R_{11}R_{21} + R_{12}R_{22} + R_{13}R_{23} = 0. \quad (8)$$

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

$$R_{11}R_{21} + R_{12}R_{22} + \sqrt{1 - R_{11}^2 - R_{12}^2}\sqrt{1 - R_{21}^2 - R_{22}^2} = 0. \quad (9)$$

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.

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 [15] algorithm. We then describe the particle filter algorithm for global motion estimation.

(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. We use feature tracking to get the mean vector for constructing the proposal density q(·).

$$\mathbf{x_k} = \left[\bar{s}_k, \bar{tx}_k, \bar{ty}_k, \bar{R}_{11k}, \bar{R}_{12k}, \bar{R}_{21k}\right]^T \tag{10}$$

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. Once we have corresponding pairs, we can use them to determine the transform matrix between two images. Equation (5) can be rewritten as

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

where $[u_{k-1}, v_{k-1}]^T$ 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. The over-determined system is in the form of $\mathbf{Y} = \mathbf{XA}$, which can be solved easily under least-square criteria by $\mathbf{A} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{Y}$. We can further form the mean vector $\mathbf{x_k}$ from matrix $\mathbf{A}$. We then generate particles according to the importance density q(·) of a six-dimensional Gaussian distribution.

$$\begin{aligned} \mathbf{x}_k^i &\sim q_G(\bar{\mathbf{x}}_k, \Sigma_1) \\ &= \frac{1}{\sqrt{(2\pi)^6 \mid \Sigma_1 \mid}} exp\left[-\frac{1}{2}\left(\mathbf{x}_k^i - \bar{\mathbf{x}}_k\right)^T \Sigma_1^{-1} \left(\mathbf{x}_k^i - \bar{\mathbf{x}}_k\right)\right] \end{aligned} \tag{12}$$

where $q_G(\mathbf{x_k}, \sum 1)$ refers to the Gaussian function with mean $\mathbf{x_k}$ and covariance matrix $\sum 1$. $\sum 1$ 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 practice, we observe that SIFT sometimes provides incorrect corresponding feature points especially when the pictures are blurred due to rapid camera movements. In such cases, the resulting $\mathbf{x_k}$ 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.

(b) **Particle Filtering for Global Motion Estimation Between Successive Frames.** We employ here the bootstrap filtering proposed in [12] with slight modifications. At time k, we first generate $\mathbf{x}_k^i$ from an importance density $q_G(\mathbf{x}_k, \sum 1)$. We then need to assign weights to these particles. The desired weights should perform as an evaluation on the "quality" of the particles. 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 **A**i. 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.

(i) **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 Mi. The MSE likelihood is then given by

$$P_{MSE}^{i} \propto \frac{1}{\sqrt{2\pi}\sigma_M} \exp\left\{-\frac{M_i^2}{2\sigma_M^2}\right\}. \quad (13)$$

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

$$P_{\text{feature}}^{i} \propto \frac{1}{\sqrt{2\pi}\sigma_F} \exp\left\{-\frac{D_i^2}{2\sigma_F^2}\right\} \quad (14)$$

where σM and σF are adjustable standard deviations which can be chosen experimentally. The normalized weight for particle $\mathbf{x}_k^i$ is then given by

$$w_k^i = \frac{P_{MSE}^i P_{\text{feature}}^i}{\sum_{i=1}^N P_{MSE}^i P_{\text{feature}}^i}. \tag{15}$$

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

$$\hat{\mathbf{x}}_k = \sum_{i=1}^N w_k^i \mathbf{x}_k^i \tag{16}$$

where the estimated state tells the estimated values of global affine motion parameters. 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

$$s_k^A = s_{k-1}^A \cdot \hat{s}_k, \qquad R_k^A = R_{k-1}^A \begin{bmatrix} \hat{R}_{11k} & \hat{R}_{12k} \\ \hat{R}_{21k} & \hat{R}_{22k} \end{bmatrix},$$
$$T_k^A = \hat{s}_k \begin{bmatrix} \hat{R}_{11k} & \hat{R}_{12k} \\ \hat{R}_{21k} & \hat{R}_{22k} \end{bmatrix} T_{k-1}^A + \begin{bmatrix} \hat{tx}_k \\ \hat{ty}_k \end{bmatrix}. \tag{17}$$

(c) **Algorithm.** To summarize, the algorithm for each time step at time *k* is given as follows.

(i) Load two successive frames: frame *k* and frame *k* − 1.

(ii) Extract and match SIFT feature points in the two images.

(iii) Reject the features that might corresponds to moving objects by detecting the motion speed.

(iv) Compute x_k vector from SIFT feature matches.

(v) Use particle filtering framework to estimate the global motion between frame *k* and frame *k* − 1:

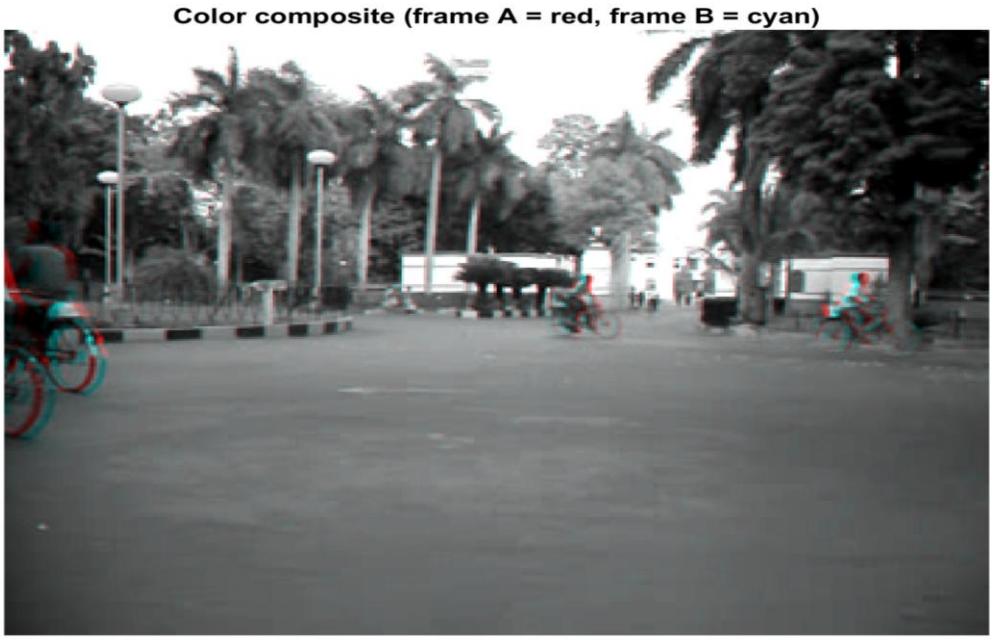
(aa) for *i* = 1 : *N*, generate particles from an Gaussian importance density.
(ab) for *i* = 1 : *N*, calculate the normalized weight from the importance sampling.
(ac) form the estimation using the weighted sum of samples.

(vi) Calculate the accumulative motion.

(vii) Reconstruct the stabilized image.

5. Experimental Results

(a) **Step 1. Read Frames from a Movie File** Here we read in the first two frames of a video sequence. We read them as intensity images since color is not necessary for the stabilization algorithm, and because using grayscale images improves speed. Below we show both frames side by side, and we produce a red-cyan color composite to illustrate the pixel-wise difference between them.



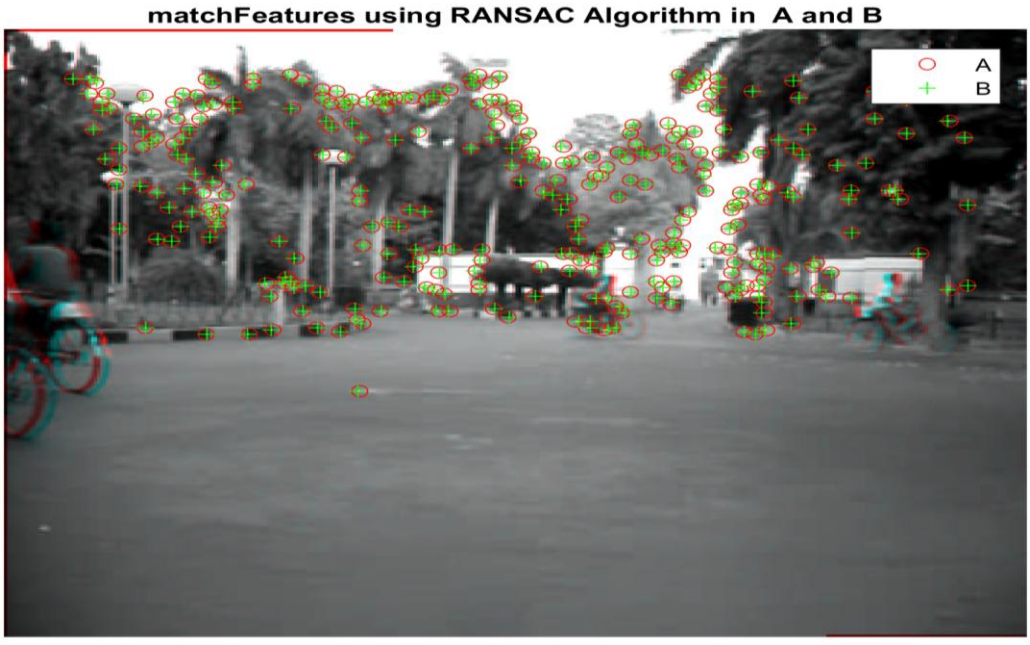
(b) **Step 2. Collect Salient Points from Each Frame** Our goal is to determine a transformation that will correct for the distortion between the two frames. As input we must provide this function with a set of point correspondences between the two frames. To generate these correspondences, we first collect points of interest from both frames, then select likely correspondences between them. In this step we produce these candidate points for each frame. To have the best chance that these points will have corresponding points in the other frame, we want points around salient image features such as corners. The detected points from both frames are shown in the figure below.

Corners Interest Point in A

Corners Interest Point in B

(c) **Step 3. Select Correspondences Between Points** Next we pick correspondences between the points derived above. For each point, we extract a Fast Retina Keypoint (FREAK) descriptor centered around it. Points in frame A and frame B are matched putatively. Point from frame B can correspond to multiple points in frame A. Match features which were found in the current and the previous frames. The image below shows the same color composite given above, but added are the points from frame A in red, and the points from frame B in green. Yellow lines are drawn between points to show the correspondences selected by the above procedure.

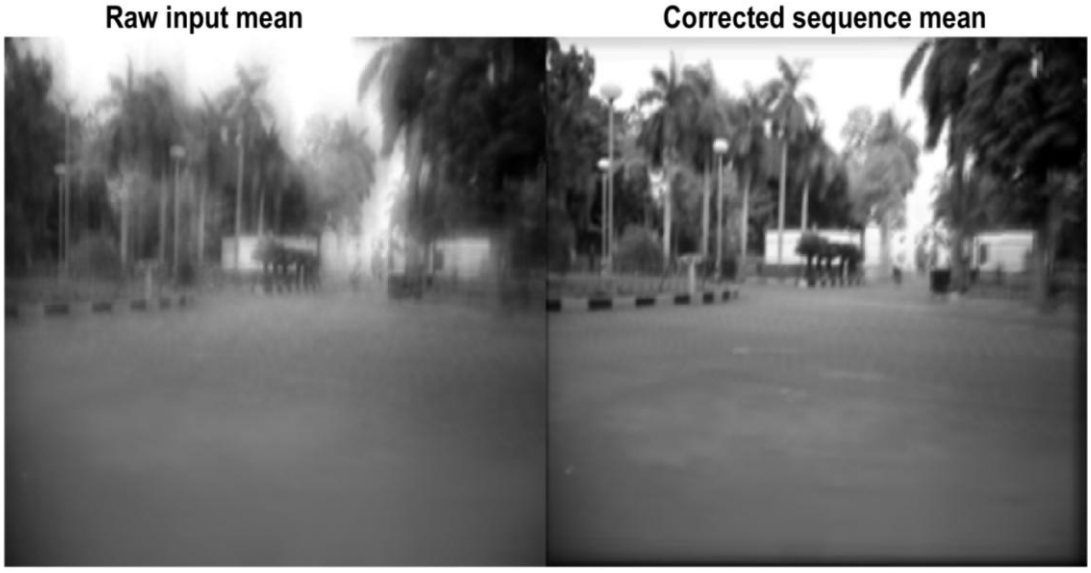
(d) **Step 4. Estimating Transform from Noisy Correspondences** We can derive a robust estimate of the geometric transform between the two images using the M-estimator SAmple Consensus (MSAC) algorithm, which is a variant of the RANSAC algorithm. This function, when given a set of point correspondences, will search for the valid inlier correspondences. From these it will then derive the affine transform that makes the inliers from the first set of points match most closely with the inliers from the second set. Thus, even though the affine transform is limited to altering only the imaging plane, here that is sufficient to align the background planes of both images. Furthermore, if we assume that the background plane has not moved or changed significantly between frames, then this transform is actually capturing the camera motion. Therefore correcting for this will stabilize the video.



(e) **Step 5. Transform Approximation and Smoothing** Now we will estimate the distortion between all frames T_i and T_{i+1} as affine transforms, H_i. Thus the cumulative distortion of a frame i relative to the first frame will be the product of all the preceding inter-frame transforms. For numerical simplicity and stability, we

choose to re-fit the matrix as a simpler scale-rotation-translation transform. This has only four free parameters compared to the full affine transform's six: one scale factor, one angle, and two translations. We show this conversion procedure below by fitting the above-obtained transform H with a scale-rotation-translation equivalent, H_{sRt}. To show that the error of converting the transform is minimal, we re-project frame B with both transforms and show the two images below as a red-cyan color composite.

(f) **Step 6. Run on the Full Video** Now we apply the above steps to smooth a video sequence. The function cvexTformToSRT also converts a general affine transform into a scale-rotation-translation transform. At each step we calculate the transform H between the present frames. We fit this as an s-R-t transform, H_{sRt}. Then we combine this the cumulative transform, $H_{cumulative}$, which describes all camera motion since the first frame. The last two frames of the smoothed video are shown in a Video Player as a red-cyan composite.



6. Discussion

This paper 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. SIFT Interest point depends on threshold in SIFT detector. Also depend on the matching algorithm to remove outliers. We also proved practically 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.

7. References

[0] Junlan Yang, *Student Member, IEEE*, Dan Schonfeld, *Senior Member, IEEE*, and Magdi Mohamed, *Associate Member, IEEE* **"Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion "** IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 19, NO. 7, JULY 2009 945

[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. 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.

[12] 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.

[13] 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.

[14] 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.

[15] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Dec. 2004.