ROBUST 3-D CAMERA MOTION PARAMETER ESTIMATION WITH APPLICATIONS TO VIDEO CODING

Eung-Tae Kim

Department of Electronics Engineering
Korea Polytechnic University
221 Jungwang-Dong, Shihung-City, Kyonggi-Do 429-793, KOREA
etkim@kpu.ac.kr

ABSTRACT

In this paper, we present the estimation method of global motion parameters corresponding to 3D camera motion in the noisy situation. Total least squares problem is first formulated to represent the global motion parameters estimation procedure from the noise-corrupted image coordinates. Then, a recursive total least squares (RTLS) algorithm is proposed to estimate 3D camera motion parameters in image sequences. The algorithm is proposed based on a five camera parameter model: zoom, focal length, pan, tilt, and swing. The experimental results show that the proposed RTLS algorithm has better performance than the conventional linear algorithms in the measurement noisy environments.

1. INTRODUCTION

Estimating the relative camera motion between two image frames is an important research topic in the areas of computer vision and image coding.It has been shown that in video coding, global motion as the movement due to camera motion can be modeled using a few parameters [1]-[4]. Therefore, many researchers have studied the global motion compensation which can improve motion prediction and remove the motion side information greatly.

The conventional linear least squares (LS) estimation methods [2, 3] for global motion parameters suffer from the undesirable measurement errors such as spatial quantization errors and feature matching errors. Several researchers proposed the algorithms which reduce the effect of measurement errors [1, 4]. However, their performances are severely degraded by the disturbance of independently moving objects. Therefore, it needs the more effective and robust techniques to estimate global motion parameters in the presence of local motion.

In this paper, the total least squares (TLS) problem is formulated to describe the relationship between the global motion parameters and the image coordinates contaminated with the measurement errors. These errors can be regarded as matching noise whose distribution is supposed to be nonstationary Gaussian. In the presence of the nonstationary matching noise, the more effective recursive algorithm is proposed to solve the TLS problem.

This work is financially supported by the Ministry of Education and Human Resources Development (MOE) and the Ministry of Commerce, Industry and Energy (MOCIE) through the fostering project of the Industrial-Academic Cooperation Centered University.

In Section 2, a recursive total least squares algorithm (RTLS) for estimating the camera motion parameters is described. In Section 3, the performance of the proposed RTLS algorithm is evaluated.

2. MOTION PARAMETER ESTIMATION USING RECURSIVE TOTAL LEAST SQUARES ALGORITHM

In perspective imaging, the relationship between the image coordinate (X, Y) before the camera motion and the image coordinate (X', Y') after the camera motion is described [4] as

$$
\begin{aligned}
X' &= F_2 \frac{r_{11}X + r_{12}Y + r_{13}F_1}{r_{31}X + r_{32}Y + r_{33}F_1} \\
Y' &= F_2 \frac{r_{21}X + r_{22}Y + r_{23}F_1}{r_{31}X + r_{32}Y + r_{33}F_1}.
\end{aligned} \tag{1}
$$

where F_1, F_2 are the focal lengths of the camera before and after zoom $s = F_2/F_1$, respectively, and r_{ij} for $i, j = 1, 2, 3$ are elements of a 3D rotation matrix **R**. The parameters of equation (1) is composed of five 3D camera motion parameters: zoom factor s, pan angle α, tilt angle β, swing angle γ, and focal length F_1.

Let $\vec{U}_i = (X'_i, Y'_i)^T$ and $\vec{V}_i = (X_i, Y_i)^T$, $\forall i$, denote the 2-D image plane vectors after the camera motion and before the camera motion, respectively. Since the coordinates of features can not be measured exactly due to measurement errors such as spatial quantization errors, feature detector errors, and the matching noise caused by local motion [1, 4], we define a random variable $\vec{\delta} = (\delta_x, \delta_y)^T$ as the measurement error in the image coordinate. We assume that the noises at the different points are uncorrelated, and the noises in the two components of the same coordinates are uncorrelated.

If the measured points, $\hat{X}_i, \hat{Y}_i, \hat{X}'_i$, and $\hat{Y}'_i$ have additive errors $\delta_{x_i}, \delta_{y_i}, \delta_{x'_i}$, and $\delta_{y'_i}$, respectively, for $i = 1, \cdots, N$, the observed image coordinate (X'_i, Y'_i) after camera motion can be expressed as

$$\vec{U}_i + \Delta\vec{U}_i = \hat{\boldsymbol{H}}_i \vec{a},\ for\ i = 1, \cdots, N \tag{2}$$

where $\vec{a}$ consists of the eight motoin parameters, i.e.,

$$\begin{aligned} a_1 = s \cdot r_{11}/r_{33}, a_2 = s \cdot r_{12}/r_{33}, a_3 = s \cdot F_1 r_{13}/r_{33}, \\ a_4 = s \cdot r_{21}/r_{33}, a_5 = s \cdot r_{22}/r_{33}, a_6 = s \cdot F_1 r_{23}/r_{33}, \\ a_7 = r_{31}/(F_1 r_{33}), a_8 = r_{32}/(F_1 r_{33}), \end{aligned} \tag{3}$$

and $\boldsymbol{H}_i$ is a 2×8 matrix whose entities are functions of image coordinates (X_i, Y_i) and (X'_i, Y'_i) as

$$\boldsymbol{H}_i = \begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -X'_i X_i & -X'_i Y_i \\ 0 & 0 & 0 & X_i & Y_i & 1 & -Y'_i X_i & -Y'_i Y_i \end{bmatrix}. \tag{4}$$

Here, $\hat{\boldsymbol{H}}_i = \boldsymbol{H}_i + \Delta\boldsymbol{H}_i$, $\Delta\boldsymbol{H}_i$ is the noise matrix of $\boldsymbol{H}_i$, and $\Delta\vec{U}_i$ is the noise vector of $\vec{U}_i$. If a feature point $(\hat{X}'_i, \hat{Y}'_i)^T$ corresponds to the local moving object having different motion from the camera motion, the noise term $\delta_{x'_i} = \hat{X}'_i - X'_i$ may have a large magnitude in each image coordinates. Therefore, unlike what the conventional methods [1, 4] assume, i.e., the local motion can be regarded as matching noise whose distribution is supposed to be Gaussian as a stationary noise, these errors rather seem to be nonstationary noise. Furthermore, they introduce a bias to the estimated camera motion parameter. Thus, this effect has to be considered.

Given k point correspondences $(\hat{X}_i, \hat{Y}_i)$ and $(\hat{X}'_i, \hat{Y}'_i)$, $i = 1, 2, \cdots, k$, the equation (2) can be expressed as

$$\vec{\boldsymbol{U}}_k + \Delta\vec{\boldsymbol{U}}_k = [\boldsymbol{\mathcal{H}}_k + \Delta\boldsymbol{\mathcal{H}}_k]\vec{a} \tag{5}$$

where

$$\begin{aligned} \vec{\boldsymbol{U}}_k &= [\vec{U}_1^T, \cdots, \vec{U}_k^T]^T \\ \boldsymbol{\mathcal{H}}_k &= [\boldsymbol{H}_1^T, \cdots, \boldsymbol{H}_k^T]^T \\ \Delta\boldsymbol{\mathcal{H}}_k &= [\Delta\boldsymbol{H}_1^T, \cdots, \Delta\boldsymbol{H}_k^T]^T \\ \Delta\vec{\boldsymbol{U}}_k &= [\Delta\vec{U}_1^T, \cdots, \Delta\vec{U}_k^T]^T. \end{aligned}$$

The total least squares (TLS) problem in (5) can be rewritten as

$$\begin{gathered} \min \|\boldsymbol{W}_k\|_F \\ subject\ to\ [\Phi_k + \boldsymbol{W}_k]\vec{q}_k = 0. \end{gathered} \tag{6}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\vec{q}_k \triangleq \begin{pmatrix} 1 \\ -\vec{a} \end{pmatrix} = [1 - a(1), \cdots, -a(8)]^T$, and

$$\Phi_k \triangleq [\vec{\boldsymbol{U}}_k, \boldsymbol{\mathcal{H}}_k],\ \ \boldsymbol{W}_k \triangleq [\Delta\vec{\boldsymbol{U}}_k, \Delta\boldsymbol{\mathcal{H}}_k].$$

In our approach, a recursive procedure is proposed to efficiently estimate the TLS solution in the presence of the nonstationary noise.

Let $\boldsymbol{r}_j$ denote a 9×2 input matrix at index j as follows.

$$\boldsymbol{r}_j \triangleq [\vec{U}_j, \boldsymbol{H}_j]^T = [\vec{r}_{1j}, \vec{r}_{2j}]. \tag{7}$$

It is well-known that the minimization problem of (6) can be associated with the equivalent minimization problem of Rayleigh quotient

$$\mu(\vec{q}_k) = \min \|\boldsymbol{W}_k\|^2 = \min_{\vec{q}_k} \frac{\vec{q}_k^T \Phi_k^T \Phi_k \vec{q}_k}{\vec{q}_k^T D \vec{q}_k} = \min_{\vec{q}_k} \frac{\vec{q}_k^T R_k \vec{q}_k}{\vec{q}_k^T D \vec{q}_k} \tag{8}$$

where R_k satisfies the Hermitian matrix which is defined as

$$R_k \triangleq \Phi_k^T \Phi_k = \sum_{j=1}^{k} \boldsymbol{r}_j \boldsymbol{r}_j^T \tag{9}$$

and D is a 9×9 symmetric nonnegative matrix which is defined as

$$D = \begin{bmatrix} \boldsymbol{I}_{3\times3} & & & & \\ & 0 & & & \\ & & \boldsymbol{I}_{2\times2} & & \\ & & & 0 & \\ & & & & \boldsymbol{I}_{2\times2} \end{bmatrix}. \tag{10}$$

It is easy to see that 3rd, 6th columns of $\Delta\boldsymbol{\mathcal{H}}$ are all zeros. Therefore, Φ_k has the exactly known columns, 4th and 7th columns.

For a real-valued symmetric matrix R_k, given the previous eigenvector $\vec{q}_{k-1}$, we update it to obtain $\vec{q}_k$ by

$$\vec{q}_k = \vec{q}_{k-1} + \Psi_k \vec{\alpha} \tag{11}$$

where Ψ_k is a 9×2 correction matrix and $\vec{\alpha} = [\alpha_1, \alpha_2]^T$. In the gradient method, Ψ_k is chosen as the gradient of $\mu(\vec{q}_k)$, which gives a poor convergence speed. In the Newton's method, R_k is chosen as a Hessian matrix of (8). But it is difficult to compute a second derivatives of (8) and guarantee positive definite of Hessian matrix in practical situations.

For the algorithm to get good adaptation to the input signal, Ψ_k is chosen to be the Kalman gain matrix,

$$\Psi_k = \boldsymbol{K}_k = R_k^{-1} \boldsymbol{r}_k = [\vec{\psi}_{1k}, \vec{\psi}_{2k}]. \tag{12}$$

The value of $\vec{\alpha}$ can be found by substituting (11) to (8) and differentiating with respect to α_1, α_2, respectively. But, it is difficult to solve α_1 and α_2 simultaneously. So, we can solve them alternately after one value is fixed. This procedure means that the input matrix of (7) is divided into each column vectors, i.e.,

$$\vec{r}_k = \vec{r}_{mk}, \quad m = 1, 2 \tag{13}$$

and update the parameter $\vec{q}_k$ successively. First we select $\vec{r}_{1k}$. Then, (11) can be reduced into

$$\vec{q}_k = \vec{q}_{k-1} + \alpha \vec{\psi}_k \tag{14}$$

where $\vec{\psi}_k$ is the Kalman gain vector,

$$\vec{\psi}_k = R_k^{-1}\vec{r}_k. \tag{15}$$

In order to reduce the computational complexity, we use the matrix inversion lemma to update the Kalman gain.

$$R_k^{-1} = [R_{k-1} + \vec{r}_k\vec{r}_k^T]^{-1} = R_{k-1}^{-1} - \frac{R_{k-1}^{-1}\vec{r}_k\vec{r}_k^T R_{k-1}^{-1}}{1 + \vec{r}_k^T R_{k-1}^{-1}\vec{r}_k}.$$

So, R_k^{-1} can be simply updated from the previous value. In the presence of noise, the $rank(R_k)$ is generally full since the independent noises are added to the coordinates which are the elements of R_k and each coordinate before camera motion is measured at the distinct points.

Instead of (11), substituting (14) to (8), differentiating the resulting equation with respect to a scalar α, and setting it to zero will result in the following quadratic equation

$$a\alpha^2 + b\alpha + c = 0 \tag{16}$$

where

$$\begin{aligned} a &= \vec{q}_{k-1}^T R_k \vec{\psi}_k \vec{\psi}_k^T D \vec{\psi}_k - \vec{\psi}_k^T R_k \vec{\psi}_k \vec{q}_{k-1}^T D \vec{\psi}_k \\ b &= \vec{q}_{k-1}^T R_k \vec{q}_{k-1} \vec{\psi}_k^T D \vec{\psi}_k - \vec{\psi}_k^T R_k \vec{\psi}_k \vec{q}_{k-1}^T D \vec{q}_{k-1} \\ c &= \vec{q}_{k-1}^T R_k \vec{q}_{k-1} \vec{q}_{k-1}^T D \vec{\psi}_k - \vec{q}_{k-1}^T R_k \vec{\psi}_k \vec{q}_{k-1}^T D \vec{q}_{k-1} \end{aligned}$$

Among the coefficients a, b, c, the quadratic forms are computed efficiently by using $\vec{q}_{k-1}^T R_k \vec{\psi}_k = \vec{q}_{k-1}^T \vec{r}_k$ and $\vec{\psi}_k^T R_k \vec{\psi}_k = \vec{\psi}_k^T \vec{r}_k$. Also, $\vec{q}_{k-1}^T R_k \vec{q}_{k-1}$ can be computed simply as follows.

$$\begin{aligned} \vec{q}_{k-1}^T R_k \vec{q}_{k-1} &= \vec{q}_{k-1}^T R_{k-1} \vec{q}_{k-1} + \vec{q}_{k-1}^T \vec{r}_k \vec{r}_k^T \vec{q}_{k-1} \\ &= \lambda_{min}(k-1) \vec{q}_{k-1}^T D \vec{q}_{k-1} + (\vec{q}_{k-1}^T \vec{r}_k)^2 \end{aligned}$$

The minimum value of $\mu(\vec{q}_k)$, $\lambda_{min}(k)$, can be obtained by

$$\lambda_{min}(k) = \frac{\delta + a \cdot \alpha}{d} \tag{17}$$

where $d = \vec{\psi}_k^T D \vec{\psi}_k \vec{q}_{k-1}^T D \vec{q}_{k-1} - (\vec{q}_{k-1}^T D \vec{\psi}_k)^2$ and $\delta = \vec{q}_{k-1}^T R_k \vec{q}_{k-1} \vec{\psi}_k^T D \vec{\psi}_k - \vec{q}_{k-1}^T R_k \vec{\psi}_k \vec{q}_{k-1}^T D \vec{\psi}_k$.

As in [5], choosing the smallest root of (16), we obtain the update vector $\vec{q}_k$ for (14). Once more, $\vec{q}_k$ is updated by the above procedures for $\vec{r}_{2k}$ in (13). Finally, the solution of $\vec{a}$ can be obtained

$$a(i-1) = -\frac{q_N(i)}{q_N(1)},\ i = 2, \cdots, 9 \tag{18}$$

where $q_N(i)$ is the ith element of the vector $\vec{q}_N$. The proposed algorithm is summarized as follow.

Algorithm

Step 1 Permutation : $\vec{a} \rightarrow \vec{q}$

Step 2 Initialize $\vec{q}_0$ and $\vec{\psi}_0$.
For $k = 1, \cdots, N$
For $m = 1, 2$

Step 3 Select input vector $\vec{r}_k = \vec{r}_{mk}$

Step 4 Update $\vec{\psi}_k = R_k^{-1}\vec{r}_k$ by using the matrix inversion lemma

Step 5 Calculate α from (16)

$$\alpha = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Step 6 $\vec{q}_k = \vec{q}_{k-1} + \alpha\vec{\psi}_k$
End
End

Step 7 Inverse permutation : $\vec{q} \rightarrow \vec{a}$
$a(i-1) = -\frac{q_N(i)}{q_N(1)},\ i = 2, \cdots, 9$

3. EXPERIMENTAL RESULTS

3.1. Experiment with Synthetic Data

The proposed method has been tested for the synthetic data in which the camera motion parameters are known. In this simulation, the image size of 480×704 is used. The focal length F_1 and F_2 are set to 100 and 95, respectively. The rotation angle (α, β, γ) is set to $(-0.1°, 0.1°, 0.0°)$.

Each feature points are contaminated with additive Gaussian noise with a mean of zero and a standard deviation of 0.5 [pixel]. To evaluate the robustness of the proposed algorithm to the non-stationary noises, we generated 20% noisy motion fields, which represent the matching errors caused by local motion or undesirable observations, by corrupting the synthetic motion field with additive Gaussian noise which has mean, -2, and standard deviation, 5.

To evaluate the performance of the parameter estimators, *mean square error* (MSE) is measured by $MSE = \frac{1}{N}\sum_{i=1}^{N} \|\vec{u'}(\vec{u}_i, \vec{a}_e) - \vec{u'}(\vec{u}_i, \vec{a}_o)\|^2$ where $\vec{u},\ \vec{u'}$ is a point before and after the camera motion, respectively, $\vec{a}_o$ denotes the true camera parameter vector, and $\vec{a}_e$ denotes the estimated one. Fig. 1 is the average mean squared errors of the estimation methods through 100 trials. As shown in 1, the proposed RTLS algorithm outperforms over the existing algorithms which are Y. T. Tse [3], A. Zakhor [2], 6-parameter method, and MLS-TLS algorithm [4]. Compared with MLS-TLS algorithm, the average estimation accuracy of the proposed algorithm are shown in Fig. 2. The proposed algorithm provides more precise accuracy than MLS-TLS algorithm. In the computational aspects, the computational complexity of the proposed algorithm is $m \cdot (\frac{5}{2}n^2 + 6n)$ for $m = 2N, n = 9$. But the computational complexity of MLS-TLS algorithm is $2mn^2 - 2/3n^3 + 4mn_2^2 + 8n_2^3 + mn_1^2 + n_1^3/3$ for $m = 2N, n = 9, n_2 = 7, n_1 = 2$. Therefore, the computational complexity of the proposed algorithm is less than that of the MLS-TLS algorithm.

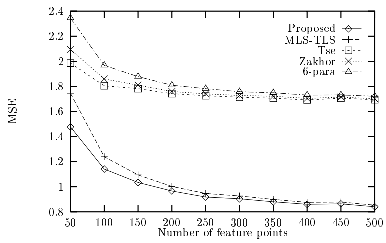

Fig. 1. MSE as the number of feature points varies

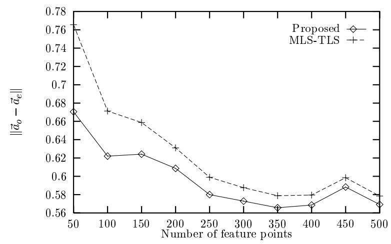

Fig. 2. Estimation accuracy

3.2. Experiment with Real Image Data

In this simulation, a two-stage motion compensation (MC) technique is used in H.263 codec. In the first stage, the proposed global MC (GMC) is used to construct a globally motion compensated frame. In the second stage, a globally motion compensated frame is used as the reference frame in local MC (LMC). The coding structure used is IPPPP..., and the PB-frames mode is not employed. Quantization step size of the DCT coefficients is set to 15. For GMC, block size and search range are set to (8, $-15 \sim +15$) and (16, $-7 \sim +7$), respectively. For LMC, block size and search range are set to 16, $-7 \sim +7$, respectively. In case of H.263, only LMC is used.

The test image sequences is 240×352 SIF "Flower garden" sequence (44-90frames) whose frame rates are 15 fps. As shown in Fig. 3, the overall performance of the proposed method becomes considerably better than that of the conventional method [2, 3] as well as that of H.263 (LMC only) even in a large 3D rotating image. The total PSNR and bitrates of the proposed method are slightly better than those of the MLS-TLS method. In the view of GMC, the proposed GMC method has better performance in Fig. 3 (b). In the aspect of computational time, the proposed method significantly outperforms the MLS-TLS method as shown in Fig. 3 (d). The average number of iteration of the proposed method is $1/3$ of the MLS-TLS method. This implies that the proposed algorithm can accurately estimate the camera motion parameters by means of fewer iterations. Thus, the computational effort can be reduced greatly in the estimation procedure.

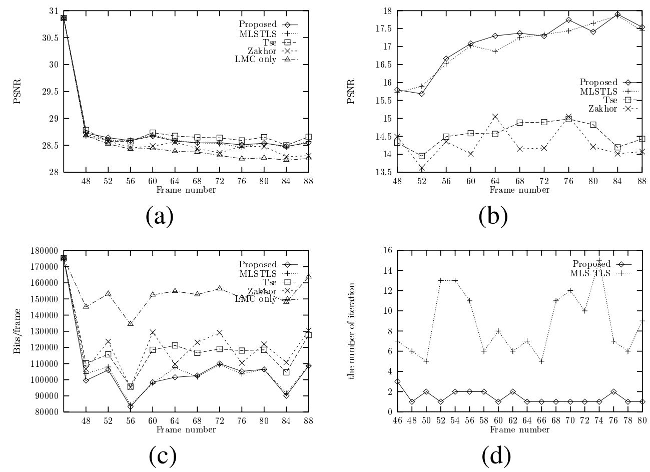

Fig. 3. *Flower garden* sequence (a) PSNR of decoded pictures (b) PSNR of GMC pictures (c) total bitrates (d) the number of iteration

4. CONCLUSIONS

In this paper, we have described a recursive total least squares algorithm for estimating 3D camera motion in image sequences. The proposed RTLS algorithm is based on a five camera parameter model: zoom, focal length, pan, tilt, and swing. The parameter estimation using the RTLS algorithm reduces the effect of the non-stationary noises efficiently. It has been shown in the simulation that the proposed algorithm has better performance than the MLS-TLS algorithm as well as the existing linear least squares algorithms in the presence of measurement errors.

5. REFERENCES

[1] J. Park, N. Yagi, K. Enami, K. Aizawa, and M. Hatori, "Estimation of camera parameters from image sequence for model-based video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 4, No. 3, June 1994, pp. 288-296.

[2] A. Zakhor and F. Lari, "Edge-based 3-d camera motion estimation with application to video coding," *IEEE Trans. on Image Processing*, Vol. 2, No. 4, October 1993, pp. 481-498.

[3] Y. T. Tse and R. L. Baker, "Global zoom/pan estimation and compensation for video compression," *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, 1991, pp. 2725-2728.

[4] E. T. Kim and H. M. Kim, "An efficient linear 3D camera motion estimation method with applications to video coding," *Optical Engineering*, vol. 37, no. 3, pp. 1065-1077, March 1998.

[5] C. E. Davila, "Recursive total least squares algorithm for adaptive filtering," *IEEE Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 1853-1856, May 1991.