The Rotational Motion Stabilization Using Simple Estimation of the Rotation Center and Angle

Ho-Dong Seok.*, Do-Jong Kim* and Joon Lyou**

* Agency for Defense Development, Taejeon, Korea

(Tel : +81-42-821-3123; E-mail: hodong@add.re.kr)

(Tel : +81-42-821-3168; E-mail: djkim@hanfos.com)

**Department of Electronics Engineering, Chungnam National University, Taejeon, Korea

(Tel : +82-42-821-5669;; E-mail: jlyou@cnu.ac.kr)

Abstract: This paper presents a simple approach on the rotational motion estimation and correction for the roll stabilization of the sight system. The algorithm first computes the rotational center from the selected local velocity vectors of related pixels by least square methods. And then, rotational angle is found from the special subset of the motion vector. Finally, motion correction is performed by the nearest neighbor interpolation technique. In order to show the performance of the algorithm, the evaluation for the synthetic and real image was performed. The test results show good performance compared with previous approach.

Keywords: Digital image stabilization**,** Motion estimation, Block matching algorithm, Peak signal to noise ratio

1. INTRODUCTION

Sight system mounted on the vehicle such as main battle tanks, requires high stabilization function which removes all of the angular motion disturbances under the stationary or moving conditions. In order to reject the disturbances which come from vehicles engine, cooling pan, irregular terrains, the conventional system adopts just 2-axes mechanical stabilization using inertial sensors because three axes mechanical stabilization is too bulky and expensive. However, modern fire control system demands more accurate 3-axes stabilization performance including pitch, yaw, roll axes for the longer range observation and the higher kill probability. As a simple approach for the 3-axis stabilization, we are considering digital image stabilization (DIS) for the roll axis which can be added for the conventional 2-axes mechanical stabilization system.

DIS is the process of generating compensated video sequences where any and all unwanted camera motion is subtracted from the original input. Several studies on the DIS have been presented such as global motion estimation using local motion vectors [1], motion estimation based on edge pattern matching[2], fast motion estimation based on bit-plane and gray-coded bit-plane matching[3,4], phase correlation-based global motion estimation[5]. The methods can only estimate the translational movement and theses approaches produce poor performance when the image fluctuation contains rotational motion dominantly.

In addition, iterative multi-resolution motion estimation scheme[6] and multi-resolution feature based motion estimation scheme[7] have been presented for the rotational motion estimation. However, the methods does not ensure sufficient bandwidth because of computational load or required to predetermine dominant features such as the horizontal line. Chang[8] proposed digital image translational and rotational motion stabilization using optical flow technique. The algorithm estimates global rotation and translation from the estimation of angular frequency and rotational center. However, it contains time consuming process since it finds the rotational center by searching basis and it is not easy to apply the real time application.

In this paper, we present a new approach on the rotational motion estimation and correction for the roll stabilization of sight system. The algorithm first computes the rotational center from the selected local velocity vectors of related pixels by least square methods. And then, rotational angle is found from the special subset of the motion vector. Finally, motion correction is performed by the nearest neighbor interpolation technique. In order to show the performance of the algorithm, the evaluation for the synthetic and real image was performed. The results show good performance compared with previous approach.

2. DIGITAL IMAGE STABILIZATION : ROLL STABILIZATION

2.1 Global motion estimation containing both translation and rotation motion

Chang[8] presented an image translational and rotational stabilization algorithm using optical flow technique. The algorithm estimates the giggle motion detected. The global motion detection module extracts the rotational angular frequency from the local motion vectors and then it finds the rotational center using search technique and translational motion.

2.1.1 The rotational angular frequency extraction

Based on the local motion field of the image, the velocity $(u(x,y), v(x,y))$ at any point (x,y) in the image, including translational and rotational angular frequency ? as well, is given by

$$u(x,y) = u' - w(y - y_0), \quad v(x,y) = v' - w(x - x_0) \qquad (1)$$

where u', v' are translational motion in horizontal and vertical direction, and (x_0, y_0) is rotational center point. For any tow points (x_{k1}, y_{k1}), (x_{k2}, y_{k2}) in the image frame, they should satisfy Eq. (1) and their velocity differences $\Delta u = u(x_{k1}, y_{k1}) - u(x_{k2}, y_{k2})$ and $\Delta v = v(x_{k1}, y_{k1}) - v(x_{k2}, y_{k2})$ take the following forms

$$\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}=\begin{bmatrix}-\Delta y\\ \Delta x\end{bmatrix}w \quad (2)$$

where $\Delta x = x_1 - x_2$, $\Delta y = y_{k1} - y_{k2}$. Eq. (2) is in a form of $Aw = b$. So, the least square estimate of ? can be obtained by $\hat{w} = (A^T A)^{-1} A^T b$.

2.1.2 The rotational center and translational velocity extraction

For a given rotational center (x_0, y_0), the velocity components, $u(x_i, y_i)$ and $v(x_i, y_i)$ at point (x_i, y_i) can be estimated by

$$\hat{u}(x_i, y_i) = \mathrm{u}(x_0, y_0) - \hat{w}(y_i - y_0)$$
$$\hat{v}(x_i, y_i) = \mathrm{v}(x_0, y_0) + \hat{w}(x_i - x_0) \quad (3)$$

Then the error function e can be defined as follows

$$e = \sum_{x_0-p\le x_i\le x_0+p}\ \sum_{y_0-p\le y_i\le y_0+p} \{[\hat{u}(x_i, y_i) - u(x_i, y_i)]^2 + [\hat{v}(x_i, y_i) - v(x_i, y_i)]^2\} \quad (4)$$

By using a searching basis, a point that can produce a smallest e measure over pixels in a certain region centered at this point would be the rotational center of the image frame. Then the translational velocity components in the x, y direction can be directly obtained from the motion vectors of the rotational center since the rotational component in the rotational center is zero.

2.2 Global Motion Estimation

2.2.1 Local motion Estimation

Block matching can be considered as the most popular method for practical local motion estimation method due to its lesser hardware complexity. As a result it is widely available in VLSI and almost all H.262 and MPEG codecs are utilizing block matching for motion estimation. The matching of the blocks can be quantified according to various criteria including the maximum cross correlation, the minimum mean square error (MSE), the minimum mean absolute difference (MAD), and maximum matching pixel count (MPC). Since the one of the most popular choice for the practical implementations is MAD, MAD is used as the criteria for the local motion estimation and as follows:

$$e = \frac{1}{N_1 N_2}\sum_{(x,y)\in B} \left| s(x, y, k) - s(x + d_1, y + d_2, k + 1) \right| \quad (5)$$

Then motion estimates is given by

$$[d_1, d_2]^T = \arg\min_{(d_1, d_2)} MAD(d_1, d_2) \quad (6)$$

Finding the best matching block requires optimizing the matching criterion over all possible candidate motion vectors at each pixel. This can be accomplished by the full search which evaluates the matching for all values of at each pixel, and is extremely time consuming. In order to reduce the computational burden, we limited the search area to a search window on the consideration of the image dynamics. The output of block matching is the velocity field $V(x, y) = (u(x, y), v(x, y))$ of each block at (x, y). For the practical cases, the motion vectors from the real images are usually corrupted by the noises of image. In order to reduce the motion noises, median filter is applied to the local motion fields.

2.2.2 Rotation Center Estimation

The block motion and median filtering result is the local velocities of each pixel in the current image frame and the velocity vectors at any point (x, y) are expressed as follows:

$$u = \frac{dx}{dt} \quad \text{and} \quad v = \frac{dy}{dt} \quad (7)$$

For the case of the pure rotational motion, the motion vectors at any points coincide with the tangential line of the circle which has a center point identical with the image rotational center. Therefore, all the perpendicular lines against motion vectors in any arbitrary points cross at one point, i.e., rotational center as described in Fig. 1.

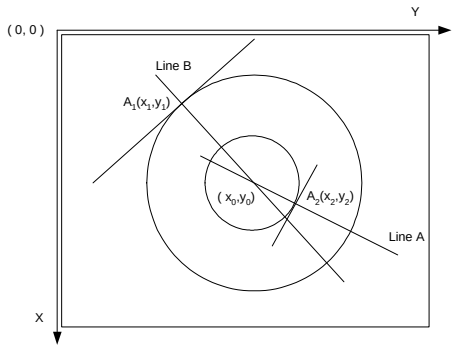


Fig. 1. The analysis of finding rotational center.

The equations of the perpendicular line at points $A_1 = (x_1, y_1), A_2 = (x_2, y_2)$ can be expressed as Eqs. (8-9).

For the point A_1,

$$y = a_1 x + b_1 \quad (8)$$

Where, $a_1 = -\frac{u_1}{v_1}$, $b_1 = y_1 - a_1 x_1$, and u_1, v_1 are velocity vectors at point A_1. For the point A_2,

$$y = a_2 x + b_2 \quad (9)$$

Where, $a_2 = -\frac{u_2}{v_2}$, $b_2 = y_2 - a_2 x_2$ and u_2, v_2 are velocity vectors at point A_2.

Although, two distinct points are enough for the computation of the rotational center, noises and insufficient image contrast often cause erroneous results. In order to guarantee the robustness of the computation results under the noise environments, over-determined systems are utilized using more than 3 points.

For n points case, we get Eq. (10) for finding rotation

center (x_0, y_0)

$$\begin{bmatrix} -a_1 & 1 \\ -a_2 & 1 \\ \vdots & \vdots \\ -a_n & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (10)$$

Eq. (10) can be rewritten as matrix vector form as following:

$$Ax = b \quad (11)$$

Then, the rotational center can be solved as Eq. (12)

$$x = (A^T A)^{-1} A^T b \quad (12)$$

2.2.3 Rotation Angle Estimation

After finding the rotational center, rotational angle q can be computed through the another least square approach.

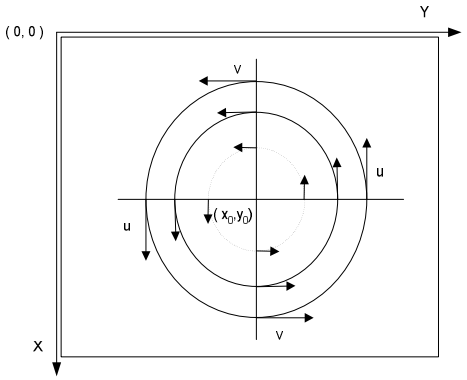


Fig. 2. The analysis of finding rotational angle

As shown by Fig. 2 all points on perpendicular line at center point against y-axis have only y-directional motion vector. However all points on horizontal line at center point against x-axis have only x-directional motion vector. Therefore rotational angle can be estimated by following equation.

$q = \frac{du}{dx}$ for points on horizontal line and $\frac{dv}{dy}$ for points on perpendicular line

where dx, dy are distances between related pixel and rotation center and du, dv are motion vector at each point on horizontal and vertical lines.

As a same reason to rotation center estimation, we can consider least square solution for angle estimation. For n points case, we get Eq. (13) for finding rotation angle q

$$\begin{bmatrix} dv_1 \\ dv_2 \\ \vdots \\ dv_N \\ du_1 \\ du_2 \\ \vdots \\ du_N \end{bmatrix} = \begin{bmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_N \\ dy_1 \\ dy_2 \\ \vdots \\ dy_N \end{bmatrix} q \quad (13)$$

where dx_1, dx_2, $\cdots$ dx_N are the distance from rotational center to each points on the horizontal line, dy_1, dy_2, $\cdots$ dy_N are the distance from rotational center to each points on the vertical line, dv_1, dv_2, $\cdots$ dv_N are the amplitude of motion vector at each points on the horizontal line, and du_1, du_2, $\cdots$ du_N are the amplitude of motion vector at each points on the vertical line.

Eq. (13) can be rewritten as matrix vector form as following:

$$b = Aq \quad (14)$$

Then, the rotational angler can be solved as Eq. (15)

$$q = (A^T A)^{-1} A^T b \quad (15)$$

2.3 Motion Compensation

The result of the motion estimation process described in the previous section is capable of computing the motion between two frames. The objective of motion compensation is to keep some kind of history of the motion estimates in order to create a stabilized sequence.

The way motion estimation and compensation are performed defines the structure of the stabilization algorithm. We can consider two possible implementation: one that always use two consecutive frames from the input image sequence to estimate the motion parameters, which so called as the frame-to-frame algorithm(FFA), and a second on that keeps a reference image and uses it to estimate the motion between the reference and the current input image, which so called as the frame-to-reference algorithm.[7].

In this paper, the frame-to-frame algorithm is used because the frame to reference algorithm sometimes can not find corresponding point when motion is large.

Before the motion compensation, a source coordination of the pixel is computed using the rotation center and angle parameters as following:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} \begin{bmatrix} \cos q & \sin q \\ -\sin q & \cos q \end{bmatrix}^{-1} \quad (16)$$

where (x_c, y_c) is center of rotation, (x, y) is position after rotation and (x', y') is position before rotation.

Then, the motion compensation is performed by the nearest neighbor interpolation.

$$f(x, y) = f(f_{ix}(x'), f_{ix}(y')) \quad (17)$$

where, $f(x, y)$ is the gray level at the (x, y) and $f_{ix}(x)$ bis rounding operation which convert the float number x to integer.

3. Experiments

In order to verify the proposed algorithm, the experiments

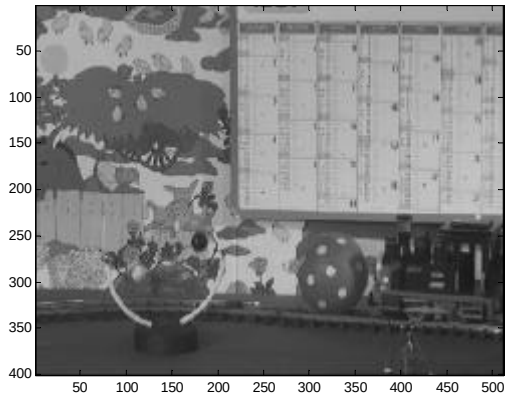
were performed for the two kinds of images, the first one is synthetic image and the second one is real image. And we evaluate the fidelity of proposed image stabilization techniques using peak signal-to-noise ratio(PSNR) between stabilized frames[9]. The PSNR between consecutive frame I_1 and I_0 is define as

$$PSNR(I_1, I_0) = 10 \log \frac{255^2}{MSE(I_1, I_0)} \quad (18)$$

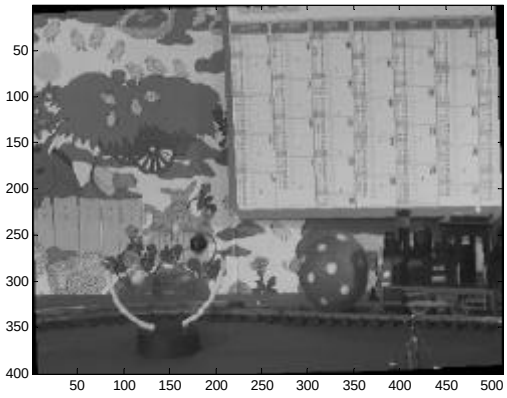
where the mean squared error(MSE) is a measure of the average departure per pixel from the desired stabilized result. The PSNR gives a relation between the desired output and the residual image, in terms of their powers(for gray images with a maximum intensity of 255). The higher the PSNR between two stabilized frames, the better the fidelity of the system.

3.1 Synthetic image

For the synthetic image test, the rotated image is generated from the known rotational center and angle. The original image and the artificially rotated images are shown in Fig. 3. The optical flow vectors of two consecutive images are computed by using block matching method and median filter is used for pre-processing filter in order to reduce motion noise. The motion vectors and median filtered results are shown in Fig. 4.

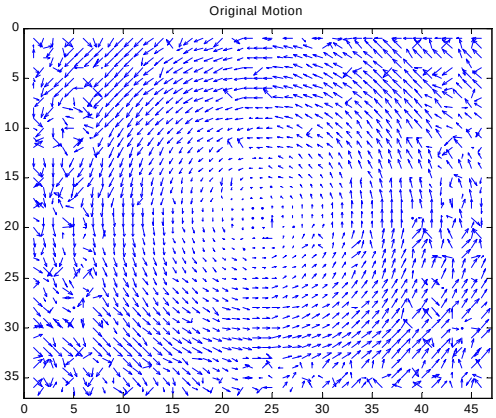


(a)



(b)

Fig 3. two consecutive image for simulation. (a) original image (b) rotated image



(a)

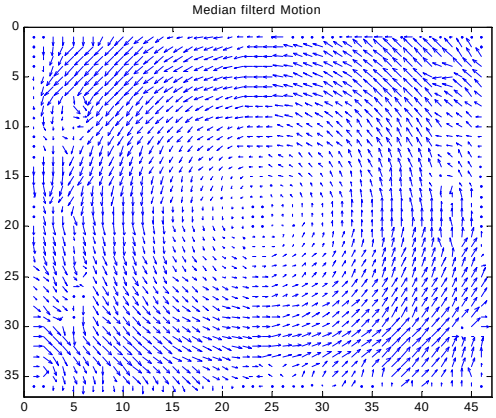


Fig. 4. optical flow vector (a) origin (b) median filtered.

According to proposed algorithm, motion estimation and correction is performed to the synthetic two image sequence. Rotational center and angle estimated by proposed method are compared with the results of Chang' method and real known value. After motion correction based on estimated value, we calculate PSNR between original image and stabilized image. And in order to know the maximum expected performance of image stabilization we find upper bound(UB) of PSNR. UB is defined as the PSNR between original image and stabilized image which is obtained from rotating and de-rotating original image according to known rotation center and angle. The performance of proposed DIS system is compared with Chang's algorithm. All results are described in Table 1. As shown in Table 1. proposed DIS system shows good performance compared with Chang's method.

3.2 Real Image

For the real image test, we get image frames from test bed which is for implementing to real sight system as shown in Fig. 5. The image frame sequence is of size 400×400 pixels and contains 50 frames in length. Fig 6 is sample image frames for simulation. To get rid of the boundary effect, we choose the center 360×360 pixels of the frame as the active block for estimation.

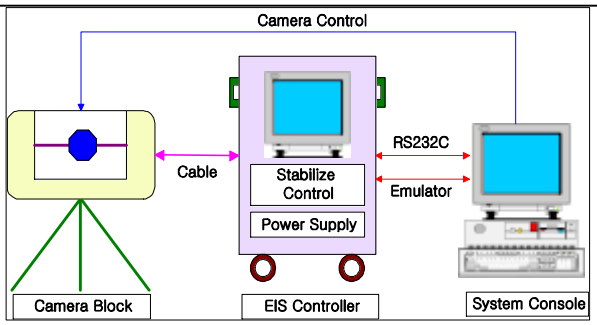
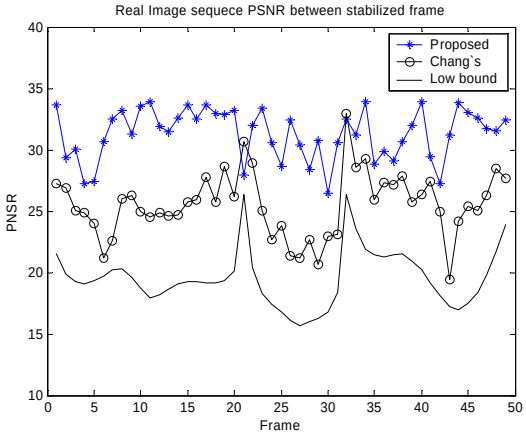


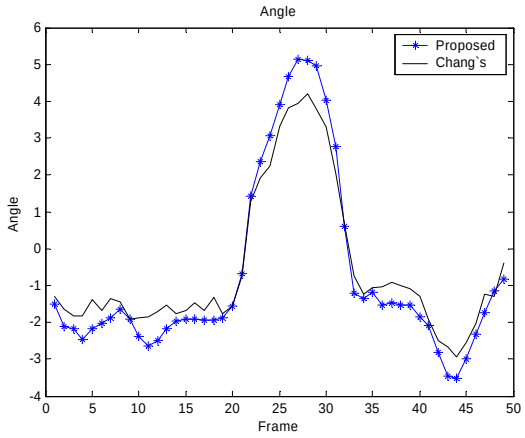
Fig. 5 test environment

Fig. 6 sample of real image frame

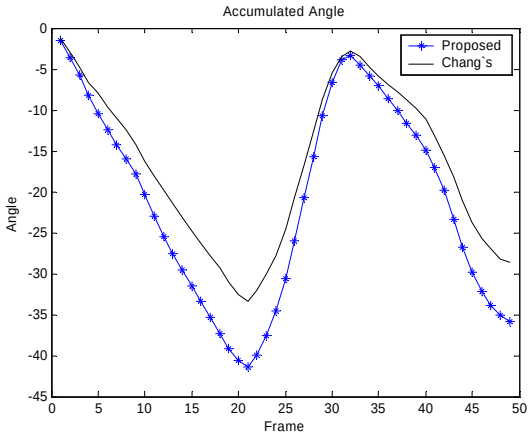
After 50 video frames were processed by the proposed image stabilizer, the results are shown in Fig. 8. In Fig. 8, lower bound(LB) is the PSNR between the reference and current frames without compensation. It means minimum performance of DIS system. Similar to experiment for synthetic image proposed algorithm has good performance compared with previous algorithm.



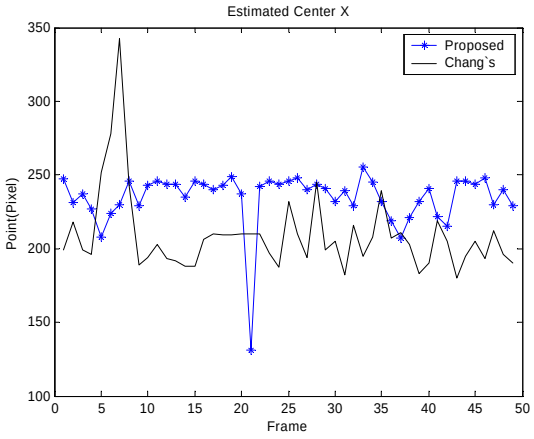
(a)



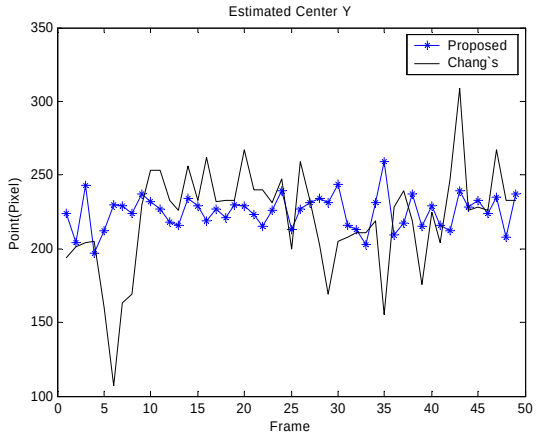
(b)



(c)



(d)



(e)

Fig. 7 experiment results for real image frames: (a) PSNR (b) estimated angle between frames (c) accumulated angle (d) estimated rotational center of x-axis (e) estimated rotational center of y-axis

4. CONCLUSION

A simple algorithm for estimating the rotational motion from consecutive images was proposed. The algorithm computes the rotational center and angle by least square method and motion compensation is done by bilinear interpolation. The test results for the real and synthetic images show that the algorithm estimates the rotational angle accurately and the peak signal-to-noise-ratio(PSNR) of the compensated image is high compared with previous algorithm.

We expect the proposed algorithm is applicable to the DIS of sight system for roll axis. For implementation for sight system, extensive tests will be done in embedded environment. And for more general application we will study advanced algorithm which can be working in translation motion and rotational motion disturbance.

REFERENCES

[1] K. Uomori, A. Morimura, and H. Ishii, “Electronic image stabilization system for video cameras and VCRs,” *Journal of the Society of Motion Picture Television Engineers*, Vol. 101, pp. 66-75, 1992.

[2] J. K. Pail, Y. C. Park, and D. W. Kim, “An adaptive motion decision system for digital image stabilizer based on edge pattern matching,” *IEEE Transactions on Consumer Electronics*, Vol. 38, pp. 607-615, 1992.

[3] S. J. Ko, S. H. Lee, and K. H. Lee, “Digital image stabilizing algorithms based on bit-plane matching,” *IEEE Transactions on Consumer Electronics*, Vol. 44, pp. 617-622, 1998.

[4] S. J. Ko, S. H. Lee, S. W. Jeon, and E. S. Kang, “Fast digital image stabilizer based on gray-coded bit-plane matching,” *IEEE Transactions on Consumer Electronics* Vol. 45, pp. 598-603, 1999.

[5] S. Ertürk , and T. J. Dennis, “Image sequence stabilization based on DFT filtering,” *IEE Proceedings on Image Vision and Signal Processing*, Vol. 127, pp. 95-102, 2000.

[6] A. Burt, and P. Anandan, “Image stabilization by registration to a reference mosaic,” *Proceedings of ARPA Image Understanding Workshop*, pp. 425-434, 1994.

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

[8] J. Y. Chang, W. F. Hu, M. H. Cheng and G. S. Chang, “Digital image translational and rotational motion stabilization using optical flow technique.” *IEEE Transactions on Consumer Electronics*, Vol. 48, No. 1, pp.108-115, 2002.

[9] C. N and R. Cellappa. “Evaluation of image stabilization algorithms,” *Proceedings of IEEE International Conference on*, Vol. 5, pp. 2789-2792, 1998

Table 1. Simulation result for Synthetic image

rotational angle			rotation center			PSNR		
Real value	proposed algorithm	chang's algorithm	Real value	proposed algorithm	Chang’s algorithm	UB value	proposed algorithm	chang's algorithm
1degree (clockwise)	0.9870 (-0.0130)	1.0687 (0.0687)	x=200 y=256	x=201 y=260	x=144 y=210	46.4304	**38.6848**	**24.4241**
2degree (clockwise)	2.0159 (0.0159)	1.9213 (-0.0787)	x=200 y=256	x=199 y=254	x=239 y=254	45.5657	**36.3186**	**24.3867**
2degree (counter clockwise)	-2.0139 (0.0139)	-1.8364 (-0.1636)	x=200 y=256	x=202 y=254	x=254 y=232	44.9182	**35.7404**	**22.8311**

motion vector : searching area= 7, block size = 11
center search for chang's algorithm : searching area = 55, block size = 77