Camera Based Motion Tracking by Camera Images

Camera Based Motion Tracking by Camera Images

Tran Lan Anh Nguyen*, Jeong-Hwan Kim**, Guee-Sang Lee***

↲

Abstract In this paper, a motion tracking method for built-in camera mobile phones is proposed based on global motion estimation. By using Three-Step Search algorithm, block based motion vectors are first extracted from each of two adjacent frames. Motion vector outliers are then removed by a rejection cascade algorithm. Finally, a perspective model is applied to estimate the global motion vector and perform the movement of mobile phone. The simulated experimental results from real recording videos show that the proposed method generates acceptable motion estimation by performing a smooth trajectory under most conditions of movement in a reasonable processing time.

↲

Keywords: global motion estimation, perspective model, motion tracking, mobile devices

*First author : Dept. of Electronics and Computer Eng., email : ntlanh@hotmail.com
**Co-author : Dept. of Electronics and Computer Eng., e-mail: disturb000@naver.com
***Corresponding author : Dept. of Electronics and Computer Eng.; e-mail: gslee@chonnam.ac.kr

1. Introduction

Nowadays built-in camera mobile devices such as PDA, mobile phones, etc., together with the internet, become more and more popular to our society. A trend of making applications which can run in mobile platform is spreading. At the same time, mobile devices have already been equipped sensors which help detect its movement.

To simplify the structure of mobile devices, many researches for developing interface using only built-in camera to trace their motion is being done. A lot of ideas to answer how to understand and use the information obtained from camera images are improved. There are many motion estimation algorithms proposed in the literature [1,2,3,4,5]. Specifically, in [1], the method detects movements based on feature extraction of a sequence of images and through a geometric estimation model. After that, a Hidden Markov Model is applied to classify the type of motion through the given information.

All above methods had a high accuracy. However, they also took much time since they used many arithmetic operations and complex process. They are not suitable for mobile devices which do not have enough strength to provide results in real time as we expect. Thus our method is proposed to satisfy the requests in mobile environment.

In this paper, we consider that the object we want to focus on should be appeared completely in the first frame of the video. Our goal is to apply GME technique with eight-parameter perspective model and gradient descend method to minimize the prediction error and find the trace of mobile phone motion. Before that, local motion estimation is done by Three-Step Search algorithm followed by outlier rejection cascade method to collect a reliable motion vector (MV) field. Thus, the method is more robust to outliers as well as its accuracy is increased.

2. Proposed method

In this paper, our method is based on the structure Kim et al proposed to apply to mobile environment [10]. However, we replaced his final step by applying a rejection cascade algorithm to remove outliers and a general 8-parameter motion model to estimate global MVs. This displacement lets our method estimate smoother not only translation as Kim did but also other movements of mobile phones such as rotation, zoom, pan, etc. . So, our suggested method for solving motion tracking problem is divided into five main steps: 1. Color space conversion, 2. Grid sampling, 3. Motion vector extraction, 4. Outlier rejection, 5. Global motion estimation.

2.1 Color space conversion

In Android OS mobile devices, camera uses YCbCr color model to result a sequence of images. So we need to convert the intensity component of images or their Y component to gray scale to reduce computational complexity in YCbCr color space.

2.2 Grid sampling

To speed up the motion vector extraction process, reduce the computation complexity, and especially decrease the error probability of motion estimate while objects can move separately from camera movement as noise in a screen, the input images are divided into several smaller blocks called Macro Blocks [5,10]. In our experiment, to satisfy all above criteria, the block size is chosen to be 80x80 pixels.

2.3 Motion vector extraction

For each of two adjacent frames, we use the Three-Step Search algorithm (TSS) [6,10] Kim et al applied [11]. In his paper, to choose the best-matching block in the reference frame I' 'towards blocks in the current frame I, Mean Square Error (MSE) criterion is used as

$$MSE(dx, dy) = \frac{1}{MN} \sum_{m=x}^{x+M-1} \sum_{n=y}^{y+N-1} [I(m,n) - I'(m+dx, n+dy)]^2 \quad ----(1)$$

Then the MV of current block is the location of its center point in which its MSE value is the smallest.

$$\overrightarrow{MV} = (MV_x, MV_y) = \arg\min_{(dx,dy)\in\mathbb{R}^2} MSE(dx, dy) \quad ----(2)$$

2.4 Motion vector outlier rejection

Some unexpected sources often affect the quality of camera images in real environment. This reason causes MV field unreliable as well as obtain poor GME. So outliers should be found and rejected before being passed to the estimation process. Here the outlier rejection cascade algorithm [7] is chosen to remove MV outliers for its low complexity and good result.

2.5 Global motion estimation

This is the major change we would like to suggest. Its goal is to collect global MVs in a sequence of images to trace the movement of mobile phones through a built-in camera. This replacement helps estimate very complex models which combine many types of motions. We will thus discuss this step clearly as a separate part.

3. Global motion estimation using a perspective model

After extracting from two adjacent frames and removing outliers, these (inlier) MVs are applied to a perspective model to estimate the global MV. The goal is to minimize the sum of squared differences over all corresponding pairs of local MVs between blocks in the current frame I and the reference frame I'. All following knowledge is referred to and briefed on [8,9].

3.1 Perspective model

It is also called an 8-parameter model described by a vector of parameters $\mathbf{m} = [m_0, \ldots, m_7]$. Given (x, y) and (x', y') which are coordinates in the current and the reference frame respectively, this model has the transformation form in both two directions as

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1}$$

$$y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1}$$

---- (3)

The estimation becomes to fit the transform coefficients, or eight parameters of the model. With each input block-based MV (MV_{x_i}, MV_{y_i}), the matching squared error is defined in two directions as

$$E = \sum_i \left(e_{x_i}^2 + e_{y_i}^2\right) = \sum_i \left(\left(MV_{x_i} - x'_i + x_i\right)^2 + \left(MV_{y_i} - y'_i + y_i\right)^2\right)$$

----(4)

where (x_i, y_i) is the center point of i^{th} block in frames and (x'_i, y'_i) is the estimated point after estimating 8 parameters.

3.2 Newton-Raphson method

This is a specific case of Gradient descent method for solving a sum of squared error function. Since the error function E is dependent on the nonlinear m, we have to apply the Newton-Raphson method which uses an iterative procedure to compute these global motion parameters [12] as below

$$\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + \Delta\mathbf{m}$$

----(5)

where $\mathbf{m}^{(t+1)}$ and $\mathbf{m}^{(t)}$ represent vector $\mathbf{m}$ at t^{th} and $t+1^{th}$ iteration and $\Delta\mathbf{m} = \mathbf{H}^{-1}\mathbf{b}$. The gradient vector $\mathbf{b}$ and the Hessian matrix $\mathbf{H}$ are calculated by getting the first and second derivative of the error function with respect to all parameters respectively. The calculation of matrix $\mathbf{H}$ and vector $\mathbf{b}$ can be found in [8,9] in detail.

This procedure repeats to update the value of $\mathbf{m}$ consecutively. It stops when the update term $\Delta\mathbf{m}$ is less than a given threshold or after a given number of iterative steps.

3.3 Initialization

To begin an iterative procedure, an initial value must be chosen. This is also an important step to ensure that the estimation can be converged. For this reason, the minimum value of error function can be get if the starting point is near the global rather than other local minima. In this paper, we refer to the method of initialization in [8]. The initial estimate of eight parameters is calculated based on a translational model by the average of input motion vectors

$$m_0 = m_3 = 1, \qquad m_1 = m_4 = m_6 = m_7 = 0$$
$$m_2 = \frac{1}{N}\sum_{i=1}^{N} MV_{x_i}, \qquad m_5 = \frac{1}{N}\sum_{i=1}^{N} MV_{y_i}$$

----(6)

where N is the number of MVs.

4. Simulation results

This section reports simulation results in the evaluation of our proposed mobile phone motion tracking method. To implement the proposed content, a PC of Core™ 2 6700 2.67GHz and 2GB RAM is used to run Matlab R2009a. Simulations have been carried out by using real recording videos from a digital camera. For rejection cascade and GME algorithms, we used the source code in [13] to achieve our research results.

To begin the trace, a center point of the screen is chosen to be the starting point of camera motion. Resulting figure is the first image of the sequence with a red trajectory drawn to perform the estimation. Figure 1 results the movement trajectory of an estimation of a combination of zoom and translation given by 134 consecutive camera images of the size 640x480.

Fig.1. Motion tracking performance of simultaneous camera zoom and translation

Moreover, we show that our method can estimate other complex camera motions given by a sequence of 89 images of the size 640x480 in two following figures. In figure 2, the movement of camera is a circle. In figure 3, the camera is moved in a rectangular way with a combination of translation and tilt.

Fig.2. Motion tracking performance of circular movement of camera

To catch the best tracking results, outlier rejection step is really needed. In table 1, we compare the performance of resulting figures to show the efficiency of GME after applying the rejection algorithm. In both two cases of frame size (320x240 and 640x480), the global MVs are always estimated more satisfactorily and smoother if we remove MV outliers in advance. Moreover, in most cases, the trace of images of size 640x480 results better than the other.

Fig.3. Motion tracking performance of simultaneous camera translation and tilt

In table 2, we give a comparison of the speed of our method and the method proposed in [7,8]. A sequence of 74 consecutive camera images whose size is 640x480 or 320x240 is input. The result table shows that the method in [7,8] is quite time-consuming. Otherwise, our proposed method decreases processing time considerably which relies on the simplification in MV extraction step as well as is suitable for mobile environment.

Finally, a comparison of results of two methods is showed in table 3. The input is a sequence of images which has size of 320x240. One method in the left column is proposed by our paper. The other in the right column is the method written in [7]. It represents that our method has both more exact and smoother results than [7] does.

TABLE 1

MOTION TRACKING PERFORMANCE OF DIFFERENT SIZE OF FRAMES

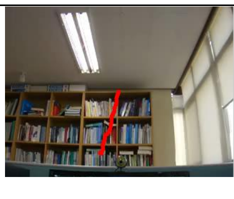
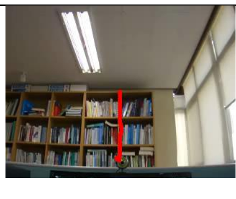
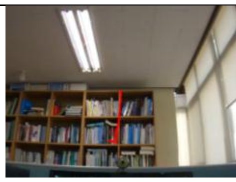
	Our method without outlier cascade rejection step	Our method with outlier cascade rejection step
320 x 240		
640 x 480		

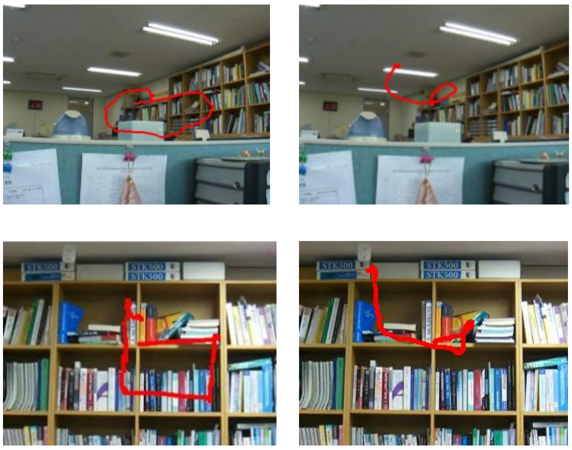
TABLE 2

COMPARISON OF PROCESSING TIME

	Frame size	Fps	Nof	Block size	Processing time (seconds)
Our method	640 x 480	29	74	80	47.4109
	320 x 240	29	74	80	12.7707
Propos-ed method in [7,8]	640 x 480	29	74	80	469.77242
	320 x 240	29	74	80	82.525178

TABLE 3

COMPARISION BETWEEN TWO METHODS



5. Conclusion

In this paper, eight-parameter model based global motion estimation is used to estimate mobile phone motion tracking in a sequence of camera images. The movement of mobile phones is determined automatically. Since this model can estimate camera movement which is independent of types of motions, the results are achieved acceptably. Additionally, the results are also obtained more smoothly. The experiment indicates that the proposed motion tracking method is effective for any size of camera images in reasonable processing time.

REFERENCES

[1] Mark Barnard, Jari Hannuksela, Pekka Sangi and Janne Heikkil, “A vision based motion interface for mobile phones”, Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)

[2] Jane Hwang, Gerard J. Kim and Namkyu Kim, "Camera-based Relative Motion Tracking for Hand-held Virtual Reality", in NICOGRAPH International 2006, Seoul, South Korea, June 2-3. 2006.

[3] Jingtao Wang, Shumin Zhai, John Canny, "Camera Phone Based Motion Sensing: Interaction Techniques, Applications and Performance Study", pp. 101-110, UIST'06, October 15-18, 2006

[4] Antonio Haro, Koichi Mori, Tolga Capin, Stephen Wilkinson, “Mobile Camera-based User Interaction”, Computer Vision in Human-Computer Interaction, 2005

[5] J. Wang, and J. Canny, "TinyMotion: Camera Phone Based Interaction Methods," In Extended Abstracts on Human Factors in Computing Systems(CHI06'), pp.339-344, 2006.

[6] L. Tao, Y. Suying, S. Zaifeng and G. Peng, "An Improved three-step search algorithm with Zero Detection and vector filter for motion estimation" International Conference on Computer Science and Software Engineering, pp.967-970, 2008.

[7] Yue-Meng Chen and Ivan V. Bajíe, “Motion Vector Outlier Rejection Cascade for Global Motion Estimation”, IEEE Signal processing letters, Vol. 17, No. 2, February 2010.

[8] Yeping Su, Ming-Ting Sun and Vincent Hsu, “Global motion estimation from coarsely sampled motion vector field and the applications”, IEEE transactions on circuits and system for video technology, Vol. 15, No. 2, February 2005.

[9] Frédéric Dufaux and Janusz Konrad, “Efficient, Robust and Fast global motion estimation for video coding”, IEEE transactions on image processing, Vol. 9, No. 3, March 2000.

[10] Jeong-Hwan Kim, Kyeong-Mi Kim, Chan-Hun Jeong, Guee-Sang Lee, and Tran Lan Anh, “Camera based motion estimation for mobile devices”, JCICT 2010.

[11] P. Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation," Kluwer Academic Publishers, 1999.

[12] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, “Numerical Recipes in C: The Art of Scientific Computing”, Cambridge Univ. Press, Cambridge, 1988.

[13] http://www.sfu.ca/~ibajic/software.html