Fast Global Motion Estimation using Iterative Least-Square Estimation Technique

G Sorwar
School of Multimedia and Information Technology,
Southern Cross University
Coffs Harbour NSW 2457, Australia
gsorwar@scu.edu.au

M Murshed and L Dooley
School of Computing and Information Technology
Monash University
Churchill Vic 3842, Australia
{Manzur.Murshed,Laurence.Dooley}@infotech.monash.edu.au

Abstract

Global motion estimation is an important task in a variety of video processing applications, such as coding, segmentation, classification/indexing, and mosaicing. The main difficulty in global motion parameter estimation resides in the disturbances due to the independently moving objects. The *Iterative Least-Square Estimation* (ILSE) technique [1] is commonly used in estimating a four-parameter model of global motion. In this paper, a *Modified ILSE* (MILSE) technique is developed, which is capable of estimating the parameters with any number of macroblocks without considering them in order of rows and columns. The performance of the MILSE algorithm is analyzed and quantitatively and qualitatively compared with the ILSE technique. Experimental results show that the proposed technique is not only computationally fast but also robust to the disturbance caused by independently moving objects.

1. Introduction

Extracting motion parameters from image sequences has been a central theme in the areas of computer vision and image coding. Motion is primarily due to the movement of a camera (pan, zoom), movement of objects in the scene, or movement of both, camera and objects. The former is often referred to as *global* motion and the latter as *local* or object motion. Most motion estimation technique ignores this aspect and makes no distinction between the *global* and *local* motion

However, separating these two classes of motion is significant for image coding and video indexing. For example, if there is no *local* motion in a scene and only the camera is moving, the dynamics of the resulting video sequences can be adequately described by only a few camera operation parameters. In case that there is both *local* and *global* motion presented in the video sequences, object motion (for object based video representation and retrieval) can be obtained by *global* motion compensation. Thus by separating the *local* and *global* motion, it is possible to represent the motion information in a more compact way.

As the *global motion estimation* (GME) procedure depends on parametric models of camera motion and the way the model parameters are estimated, different GME techniques have been reported in the literature based on the diverse motion models. One of the most well known techniques is a two-step method consisting of *local* motion estimation followed by estimation of the *global* motion parameters [1-9] from the motion field calculated by any motion estimation algorithm such as block-matching algorithm (BMA). The main difficulty in estimating *global* motion parameters resides in the existence of independently moving objects which introduce a bias into the estimated parameters. To reduce the impact of localised object motion and detection errors [10] on the determination of *global* motion parameters, different techniques have been proposed for parameter estimation.

For computational efficiency, sufficient accuracy, and simplicity in application, the *Iterative Least-Square Estimation* (ILSE) technique is commonly used to reduce the disturbance of moving objects. Since in general, camera rotation is comparatively much less frequent than zooming and panning, recently Rath and Makur [1] proposed a four-parameter model to calculate the camera pan and zoom parameters using this ILSE technique. In this technique, each of the iterations considers all the rows and columns of macroblocks in a frame of a sequence in order so that the parameter calculation depends on the entire frame.

Global motion generally spreads over the frame uniformly. If only pan is involved, the motion vector is constant for the entire frame, but in the case of zoom, the length of the motion vector is proportional to the distance of the macroblock from the convergence centre, which is generally at the centre of a frame provided there is no panning.

Another assumption regarding *global* motion is that in most video sequences, only a few blocks are occluded by the moving objects and these objects are mostly in, or around, the middle of a frame, but rarely at the edge of the frame [11]. Based on this assumption, Aghbari *et al.* [11] estimated the different types of camera motion using the macroblock motion information at the edge of the frame; thus, for panning motion, all motion vectors at the outer edges will be in the same direction, whereas for zoom-like motion, vectors on opposite sides will be in the opposite directions. Therefore, instead of using the motion vectors of all macroblocks, a few macroblocks, especially at the edge

0-7803-8185-8/03/$17.00 © 2003 IEEE

of the frame, are sufficient to enable calculation of the *global* motion parameters.

To implement this strategy, the ILSE technique [1] has been modified, and is referred to as the *Modified Iterative Least-Square Estimation* (MILSE) technique. In the latter, instead of considering macrobloack in groups of rows and columns, any number of macroblocks can be used for parameter estimation in any order. Our experimental results indicate that the proposed MILSE technique significantly reduces the computational overhead in calculating the parameters compared with the original ILSE technique in [1].

The remainder of this paper is organized as follows. Section 2 describes the proposed MILSE technique for *global* motion estimation. An analysis of the complexity of the MILSE technique is provided in Section 3. Some experimental results are included in Section 4. Section 5 concludes the paper.

2. Modified Iterative Least-Square Estimation (MILSE) Technique

Let there be N blocks in a video frame, and assume that the motion vector of a block is the motion vector of the central pixel of that block. Let $(v_x(k), v_y(k))$ be the measured motion vector, according to the DTS algorithm [12-14], of the block k, $k = 0, 1, \ldots, N\text{-}1$, whose central pixel's coordinates are $(s_x(k), s_y(k))$ with respect to the centre of the frame. In this regard, the *global* motion estimation model represented in [1] for camera zoom and pan is as:

$$\begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix} = \begin{bmatrix} a_1 s_x(k) \\ a_3 s_y(k) \end{bmatrix} + \begin{bmatrix} a_2 \\ a_4 \end{bmatrix} \tag{1}$$

where

$$a_1 = z_x \text{ and } a_2 = f_1(p_x, z_x)$$

$$a_3 = z_y \text{ and } a_4 = f_2(p_y, z_y)$$

In the above definition, z_x and z_y are the zoom factors along the x-axis and y-axis respectively, (p_x, p_y) is the pan vector.

Now consider the ILSE algorithm [1], the optimal values for camera parameters (a_1, a_2, a_3, and a_4) are obtained by using the following criteria:

$$\min_{a_1,a_2} \sum_{k=0}^{N-1} \left(v_x(k) - a_1 s_x(k) - a_2 \right)^2 \tag{2}$$

$$\min_{a_3,a_4} \sum_{k=0}^{N-1} \left(v_y(k) - a_3 s_y(k) - a_4 \right)^2 \tag{3}$$

By differentiating with respect to the parameters, and setting the derivatives to zero, the following solution is obtained as:

$$a_1 = \frac{N\sum_{k=0}^{N-1} v_x(k) s_x(k) - \left(\sum_{k=0}^{N-1} v_x(k)\right)\left(\sum_{k=0}^{N-1} s_x(k)\right)}{N\sum_{k=0}^{N-1} s_x^2(k) - \left(\sum_{k=0}^{N-1} s_x(k)\right)^2} \tag{4}$$

$$a_2 = \frac{\left(\sum_{k=0}^{N-1} v_x(k)\right)\left(\sum_{k=0}^{N-1} s_x^2(k)\right) - \left(\sum_{k=0}^{N-1} v_x(k) s_x(k)\right)\left(\sum_{k=0}^{N-1} s_x(k)\right)}{N\sum_{k=0}^{N-1} s_x^2(k) - \left(\sum_{k=0}^{N-1} s_x(k)\right)^2} \tag{5}$$

$$a_3 = \frac{N\sum_{k=0}^{N-1} v_y(k) s_y(k) - \left(\sum_{k=0}^{N-1} v_y(k)\right)\left(\sum_{k=0}^{N-1} s_y(k)\right)}{N\sum_{k=0}^{N-1} s_y^2(k) - \left(\sum_{k=0}^{N-1} s_y(k)\right)^2} \tag{6}$$

$$a_4 = \frac{\left(\sum_{k=0}^{N-1} v_y(k)\right)\left(\sum_{k=0}^{N-1} s_y^2(k)\right) - \left(\sum_{k=0}^{N-1} v_y(k) s_y(k)\right)\left(\sum_{k=0}^{N-1} s_y(k)\right)}{N\sum_{k=0}^{N-1} s_y^2(k) - \left(\sum_{k=0}^{N-1} s_y(k)\right)^2} \tag{7}$$

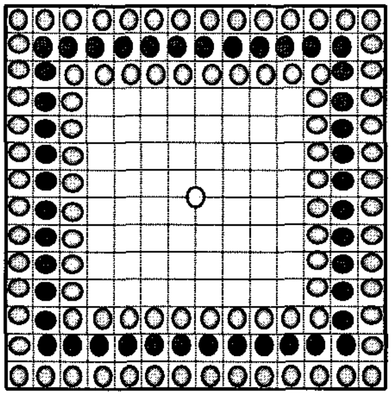
As shown by Rath and Makur [1], to eliminate the influence of the presence of *local* motion, the above procedure is evaluated iteratively, and each iteration eliminates blocks whose motion vectors (estimated by any BMA) do not match with the current *global* motion fields. Matching means that a motion vector lies within a threshold, called the *motion vector matching threshold*, from the corresponding *global* motion vector.

3. Computational Complexity Analysis of MILSE

The computational complexity incurred in *global* motion estimation by the ILSE method depends on two factors: the number of blocks considered in each iteration and the number of iterations required for the convergence achieved. Rath and Makur [1] mention that the convergence usually occurs in less than 5 iterations. Therefore, the computational complexity of the ILSE and MILSE techniques is analysed in terms of the number of blocks used in the first iteration.

If the frame size, for example, is $[N_h, N_v]$ pixels and the block size is N^2 pixels, the total number of blocks in a frame is $\frac{N_h \times N_v}{N^2}$. In a 2-dimentional form this can be represented as $[B_h, B_v]$ where B_h and B_v represent the horizontal and vertical dimension of the blocks respectively. Suppose the total number of operations required for calculating the camera parameters for each block is ξ, then the total number of operations required for the whole frame is $\xi \times (B_h \times B_v)$ for each iteration, which is the computational cost involved in ILSE technique.

Clearly, if a subset of blocks is considered instead of all the blocks of a frame, the number of operations will be fewer, which is the rationale behind the MILSE technique.



◎ Blocks in outer most grid G_0
● Blocks in second outer most grid G_1
○ Blocks in third outer most grid G_2
O Centre of the frame

Fig. 1 An example of all the macroblocks in the three outermost grids of a frame.

If the frame size is $[N_h, N_v] = [352,240]$ and block size is $[N, N] = [16,16]$, the total number of blocks in this frame is $(B_h \times B_v) = 330$. The total number of operations required is ξ×330 when all the blocks of a frame are considered for one iteration in the ILSE technique. Conversely, if only those blocks in the outermost first and second grids, shown as G_0 and G_1 in Fig. 6.2, are considered, the total number of operations involved in the first iteration is ξ×132. If the second and third outermost grids G_1 and G_2 are considered, the total number of operations required is ξ×116. For these two cases, the computational cost is reduced by 60% and 65%, respectively, compared to what is required when all the blocks are considered for parameter estimation in the first iteration. Experimental results

This research confirms that using the outer grid block motion vectors to calculate *global* motion parameter exhibits better performance than using the inner grids which are located around the centre of the frame, since the local object generally exists towards the centre of the frame.

4. Simulation Results

Simulation results for *global* motion parameters (zoom and pan) estimation, in terms of a_1, a_2, a_3, and a_4, are presented using the original ILSE method [1] and the new MILSE method. The simulation was carried out using two standard test video sequences *Table Tennis* and *Flower Garden*. Both sequences have the same frame size 352*240 pixels uniformly quantized to 8-bit per pixel. Throughout the experiments, we use $N = 16$, i.e., each frame is divided into 16×16 pixel blocks.

In the simulation program, two predefined thresholds were used to compare motion vector magnitude and angle. If the difference in magnitude and angle between the original motion vector calculated by the DTS algorithm [12-15] and the calculated current *global* motion using (1) is greater than these predefined thresholds, they are considered mismatched motion vectors and are removed during the next iteration. Tables 1 and 2 show the statistical comparison of camera pan and zoom factors represented by a_1, a_2, a_3, and a_4, calculated by considering the motion vectors for a range of different numbers of macroblocks in given frames.

The first test sequence reflected in both Tables 1 and 2 contains two pairs of images (frames #32 and #33, and frames #33 and #34) of the *Table Tennis* sequence in where the camera zooms out, whilst slightly panning, with the moving objects including ball, bat, and the hand of the player holding the bat. Tables 1 and 2 show that the values of the zoom parameters, a_1 and a_3, were very similar for all cases for both sets of frames, except for the blocks in the outermost grids, G_0 and G_1. This indicates that some noise has been introduced due to boundary artefacts in the outermost grid, introducing a small error. It is also shown that when the blocks of G_1 and G_2 were considered, the panning factors of a_2 and a_4 were smaller than for all other cases. As these images contain almost no panning, the low values of a_2 and a_4 are fully consistent with expectations.

The next test sequence contains two pairs of images (frames #10 and #11, and frames #20 and #21) of the *Flower Garden* sequence where the camera is panning horizontally to the right, and there are no moving objects. Tables 1 and 2 show that only $a_2 \neq 0$, indicating that there was no zooming and vertical panning involved in this sequence. Table 1 also illustrates that the value of a_2 was different for the blocks, G_0 and G_1, compared to all others, due to the aforementioned boundary artefacts.

So far, the performance of MILSE and ILSE has been analysed in which consecutive pairs of frames have been considered for *global* motion estimation. Generally, the pan and zoom factors in a video sequence changes proportionally to the distance between the current and the reference frames. To analyse the effectiveness of the MILSE technique, a number of experiments were also conducted based on non-consecutive (*skipping*) frame

Table 1: Statistical comparison of camera pan (a_2 and a_4) and zoom (a_1 and a_3) factors in relation to the different numbers of macroblocks considered

Test sequences	a_1 (zoom)	a_2 (pan)	a_3 (zoom)	a_4(pan)	No. of blocks considered	
Table Tennis (Frames #32 and #33)	-0.02	-0.33	-0.02	0.24	All blocks	ILSE
	-0.02	-0.40	-0.02	-0.14	Blocks in G_0 and G_1	MILSE
	-0.02	-0.27	-0.02	0.14	Blocks in G_1 and G_2	
	-0.02	-0.47	-0.02	-0.16	Blocks in G_1, G_2, and G_3	
	-0.02	-0.43	-0.02	0.05	Blocks in G_1, G_2, G_3, and G_4	
	-0.02	-0.41	-0.01	0.04	Blocks in G_1, G_2, G_3, G_4, and G_5	
	-0.02	-0.33	-0.01	0.12	Blocks in G_1, G_2, G_3, G_4, G_5, and G_6	
Flower Garden (Frames #10 and #11)	0.00	-2.00	0.00	0.00	All blocks	ILSE
	0.00	-2.51	0.00	0.00	Blocks in G_0 and G_1	MILSE
	0.00	-2.00	0.00	0.00	Blocks in G_1 and G_2	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, and G_3	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, G_3, and G_4	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, G_3, G_4, and G_5	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, G_3, G_4, G_5, and G_6	

Table 2: Statistical comparison of camera pan (a_2 and a_4) and zoom (a_1 and a_3) factors in relation to the different numbers of macroblocks considered

Test sequences	a_1 (zoom)	a_2 (pan)	a_3 (zoom)	a_4(pan)	No. of blocks considered	
Table Tennis (Frames #33 and #34)	-0.02	-0.12	-0.02	0.22	All blocks	ILSE
	-0.02	-0.38	-0.02	0.00	Blocks in G_0 and G_1	MILSE
	-0.02	-0.10	-0.02	0.03	Blocks in G_1 and G_2	
	-0.02	-0.22	-0.02	-0.02	Blocks in G_1, G_2, and G_3	
	-0.02	-0.18	-0.01	0.12	Blocks in G_1, G_2, G_3, and G_4	
	-0.02	-0.18	-0.01	0.29	Blocks in G_1, G_2, G_3, G_4, and G_5	
	-0.02	-0.07	-0.01	0.31	Blocks in G_1, G_2, G_3, G_4, G_5, and G_6	
Flower Garden (Frames #20 and #21)	0.00	-2.00	0.00	0.00	All blocks	ILSE
	0.00	-2.00	0.00	0.00	Blocks in G_0 and G_1	MILSE
	0.00	-2.00	0.00	0.00	Blocks in G_1 and G_2	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, and G_3	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, G_3, and G_4	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, G_3, G_4, and G_5	
	0.00	-2.00	0.00	0.00	Blocks in G_1, G_2, G_3, G_4, G_5, and G_6	

Table 3: Statistical comparison of camera pan (a_2 and a_4) and zoom (a_1 and a_3) factors in relation to the different numbers of macroblocks considered

Test sequences	a_1 (zoom)	a_2 (pan)	a_3 (zoom)	a_4(pan)	No. of blocks considered	
Table Tennis (Frames #32 and #34)	-0.03	-0.32	-0.03	-0.27	All blocks	ILSE
	-0.03	-0.38	-0.03	-0.09	Blocks in G_0 and G_1	MILSE
	-0.03	-0.57	-0.03	-0.08	Blocks in G_1 and G_2	
	-0.03	-0.44	-0.03	-0.02	Blocks in G_1, G_2, and G_3	
	0.03	0.40	-0.03	0.17	Blocks in G_1, G_2, G_3, and G_4	
	-0.03	-0.31	-0.03	0.27	Blocks in G_1, G_2, G_3, G_4, and G_5	
Flower Garden (Frames #10 and #12)	0.00	2.90	0.00	0.00	All blocks	ILSE
	0.00	3.40	0.00	0.00	Blocks in G_0 and G_1	MILSE
	0.00	3.00	0.00	0.00	Blocks in G_1 and G_2	
	0.00	3.00	0.00	0.00	Blocks in G_1, G_2, and G_3	
	0.00	3.00	0.00	0.00	Blocks in G_1, G_2, G_3, and G_4	
	0.00	3.00	0.00	0.00	Blocks in G_1, G_2, G_3, G_4, and G_5	
	0.00	3.00	0.00	0.00	Blocks in G_1, G_2, G_3, G_4, G_5, and G_6	

Table 3 shows the simulation results when *skipping* one and two frames of the *Table Tennis* and *Flower Garden* video sequences. It is interesting to note that the zoom and pan factors gradually increase when the distance between the current and the reference frames increases. It can also be observed that the parameter values were similar in all cases except when blocks in G_0 and G_1 were considered, again indicating the effect of boundary artefacts.

From the above analysis, it can be observed that *global* motion parameter estimation does not require a consideration of all blocks of a frame. It is also shown that if only those blocks in the second and third outermost grids are considered, then this provides better results compared to others, as well as avoiding boundary artefacts. Consequently, the proposed MILSE technique can be shown to improve computational efficiency by 65% compared with the first iteration of the ILSE technique described in [1].

5. Conclusions

A *Modified Iterative Least-square Estimation* (MILSE) technique has been proposed in this paper to approximate a four-parameter model of global motion. This technique is flexible enough for use with any number of blocks in a frame in any order. Since in general, camera rotation is comparatively much less frequent than zooming and panning, six-parameter model has not been considered. Experimental results have shown that the proposed MILSE technique has a similar performance compared to the traditional ILSE technique [1] while reducing computational cost almost 65% in camera parameter estimation.

Topics for future research include investigating the use of *global* motion parameters using MILSE technique for object motion estimation for video indexing and for object based image coding applications.

6. References

[1] G. B. Rath and A. Makur, "Iterative least squares and compression based estimations for a four-parameter linear global motion model and global motion compensation," IEEE Transactions on Circuits & Systems for Video Technology, Vol. 9, pp. 1075-1099, 1999.

[2] Y. T. Tse and R. L. Baker, "Global zoom/pan estimation and compensation for video compression," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'91), vol. 4, pp. 2725-2728, 1991.

[3] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, and H. Watanabe, "Two-stage motion compensation using adaptive global MC and local affine MC," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 7, pp. 75-85, 1997.

[4] M. Etoh and T. Ankei, "Parametrized block correaltion2D parametric motion estimation for global motion compensation and video mosaicing," Proceedings of IEICE TR PRMU97, 1997.

[5] C. -T. Hsu and Y. -C. Tsan, "Mosaics of video sequences with moving objects," Proceedings of International Conference on Image Processing (ICIP'01), Thessaloniki, Greece, Vol.2, pp. 387-390, 2001.

[6] Y. -P. Tan, S. R. Kulkarni, and P. J. Ramadge, "A new method for camera motion parameter estimation," Proceedings of International Conference on Image Processing (ICIP'95), Los Alamitos, CA, USA, Vol. 1, pp. 406-409, 1995.

[7] R. Wang and T. Huang, "Fast camera motion analysis in MPEG domain," Proceedings of International Conference on Image Processing (ICIP'99), Kobe, Japan, Vol. 3, pp. 691-694, 1999.

[8] T. Vlachos, "Simple method for estimation of global motion parameters using sparse translational motion vector fields," Electronics Letters, Vol. 34, pp. 60-62, 1998.

[9] D. Wang and L. Wang, "Fast and robust algorithm for global motion estimation," Proceedings of International Conference of SPIE-International Society of Optical Engineering, San Jose, CA, USA, Vol. 3024, pp. 1144-1151, 1997.

[10] K. Kamikura and H. Watanabe, "Global motion compensation in video coding," Electronics & Communications in Japan, Vol. 78, pp. 91-102, 1995.

[11] Z. Aghbari, K. Kaneko, and A. Makinouchi, "A motion-location based indexing method for retrieving MPEG videos,"Proceedings of Ninth International Workshop on Database and Expert Systems Applications, Vienna, Austria, pp. 102-107, 1998.

[12] G. Sorwar, M. Murshed, and L. Dooley, "Distance dependent thresholding search for fast motion estimation in real world video coding application," Proceedings of IEEE Asia-Pacific Conference on Circuits and System (APCCAS'02), Kartika Plaza Hotel, Bali, Indonesia, ol. 2, pp. 519-524, 2002.

[13] G. Sorwar, M. Murshed, and L. Dooley, "Modified full-search block-based motion estimation algorithm with distance dependent thresholds," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02), Orlando, Florida, USA, Vol. 4, pp. 4189-4189, 2002.

[14] G. Sorwar, M. Murshed, and L. Dooley, "Fast block-based true motion estimation using distance dependent thresholds (DTS)," Proceedings of the 6th International Conference on Signal Processing (ICSP'02), Beijing, China, Vol. 2, pp. 937-940, 2002.