2005

# Filtering of block motion vectors for use in motion-based video indexing and retrieval

Golam Sorwar
*Southern Cross University*

Manzur Murshed

Laurence S. Dooley

## Publication details

Sorwar, G, Murshed, M & Dooley, LS 2005, 'Filtering of block motion vectors for use in motion-based video indexing and retrieval', *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88A, no. 10, pp. 2593-2598.
Published version available from:
http://dx.doi.org/10.1093/ietfec/e88-a.10.2593

PAPER Special Section on Information Theory and Its Applications

# Filtering of Block Motion Vectors for Use in Motion-based Video Indexing and Retrieval

**Golam SORWAR**[†a)], **Manzur MURSHED**[††b)], *and* **Laurence DOOLEY**[††c)], ***Nonmembers***

**SUMMARY** Though block-based motion estimation techniques are primarily designed for video coding applications, they are increasingly being used in other video analysis applications due to their simplicity and ease of implementation. The major drawback associated with these techniques is that noises, in the form of false motion vectors, cannot be avoided while capturing block motion vectors. Similar noises may further be introduced when the technique of global motion compensation is applied to obtain true object motion from video sequences where both the camera and object motions are present. This paper presents a new technique for capturing large number of true object motion vectors by eliminating the false motion vector fields, for use in the application of object motion based video indexing and retrieval applications. Experimental results prove that our proposed technique significantly increases the percentage of retained true object motion vectors while eliminating all false motion vectors for variety of standard and non-standard video sequences.

***key words:*** *true object motion estimation, block-matching algorithm, false motion filtering, video indexing*

## 1. Introduction

With the rapid growth in multimedia and Internet applications, there is a huge amount of video data available, which highlights the need for efficient representation of video information to allow content-based functionality. As motion provides one of the easiest cues to a sequence's temporal dimension [1], it is one of the most important visual features for content-based video representation and is increasingly becoming an essential part of several applications, including content-based video indexing for browsing and retrieval [2], [3], video surveillance systems [4], video object segmentation and tracking [3]. Amongst these different applications, one of the most interesting is using object motion in video indexing for accessing large amounts of multimedia data over the Internet.

Various algorithms [5] have been proposed to index video by dense motion field using *optical flow equation* (OFE) [6] where the apparent velocity and direction of every pixel in the frame has to be computed. Although it is an effective method, it is computationally intensive and very complex. The OFE method also does not cope well with high motion video sequences [7].

To overcome this problem, many recent video processing applications have explored *block-based motion estimation algorithm* (BMA) to estimate true object motion. In [8], BMA techniques were used to estimate the motion for moving object segmentation from the background for object-based video coding and video analysis. As the motion vector information of the macroblocks is available in MPEG coded video streams, an alternative to video representation [2], [3] is provided by extracting this information, thus avoiding time consuming computation of optical flow. However, there are some drawbacks associated with these approaches. The motion vectors in a coded bit stream do not always represent true object motion since motion estimation is performed solely from the coding efficiency point of view, where minimum error matching is the only criterion. Block-based motion estimation also introduces false motion vectors in the form of noise, especially in uniform regions [9]. Moreover, as the block-matching method captures both camera (global) as well as true object motion, the block motion available in the video stream does not directly provide the true object motion. In [10], [11], the authors addressed some of the above mentioned issues by introducing a novel concept of *distance dependent thresholding search* (DTS) algorithm for fast and robust true motion vector estimation for object-based video indexing and coding applications.

To eliminate false vectors, many types of filters have already been proposed and examined for filtering false motion vectors (impulse noises). Among them the median filter and the mean filter are widely used [12], [13]. Though these filters work well for image processing applications such as noise reduction, image enhancement and restoration, they are inefficient for eliminating false motion vectors and retaining true vector. To address this issue, the authors propose a novel filter, named as the *mean accumulative thresholded* (MAT) filter to capture the true object motion vectors. The preliminary version of MAT filter was presented in [14]. In this paper, the MAT filter has also been combined with the DTS algorithm [10], [11] to develop a very useful tool for block-based true object motion estimation. Experimental results prove that MAT filter with DTS

Manuscript received January 1, 2003.
Manuscript revised January 1, 2003.
Final manuscript received January 1, 2003.

[†]The author is with the School of Multimedia and IT, Southern Cross University, NSW 2457, Australia.
[††]The authors are with the Gippsland School of Computing and IT, Monash University, Churchill Vic 3842, Australia.
a) E-mail: gsorwar@scu.edu.au
b) E-mail: manzur.murshed@infotech.monash.edu.au
c) E-mail: laurence.dooley@infotech.monash.edu.au

outperformed the well known BMAs such as *full search* (FS) [15], *three step search* (TSS) [16], and *new three step search* (NTSS) [17] algorithms for true object motion estimation.

The remainder of this paper is organized as follows. Section 2 describes the block motion estimation technique used in this paper. In Sect. 3, the process of filtering false motion vector by proposed MAT filter is discussed while the computational complexity associated with MAT filter is analyzed in Sect. 4. Some experimental results are included in Sect. 5 while Sect. 6 concludes the paper.

## 2. Block-based True Object Motion Estimation

In [10], [11], we observed that in true motion estimation, the FS algorithm tends to pick many false motion vectors even when no object motion is present in the search region. Unlike directional fast algorithms such as TSS, NTSS, a non-directional search means it is unlikely that the search will be trapped in either a global minimum or a minimum along any specific direction, especially when the search progresses away from the centre. A global minimum does not always represent the true motion vector, especially if it is far from the search centre, as it may be introduced by a different object or global motion. To address this issue we proposed the DTS algorithm by introducing distance dependent thresholds [10], [11]. This algorithm not only avoids capturing a large number of false motion vectors but also reduces the search time significantly.

In case where both the local (true object) and the global motions are present in the video sequences, true object motions can only be obtained by canceling out the global motion component from the block motion, known as global motion compensation. The global motion estimation and compensation technique described in [18] has been implemented to eliminate the global motion.

Once the global motion is compensated from the estimated block motion, true object motion vectors are clustered in the blocks containing one or more objects. As the block motion estimation cannot be done with complete accuracy due to the limitation of block-based estimation techniques, a number of false (impulse noises) motion vectors are also likely to be introduced after the above processing along with the true object motion vectors. To retain only the true object motion vectors, false vectors need to be filtered out from the scene.

## 3. Filtering the False Motion Vectors

In real world video sequences, most moving objects generally occupy more than one neighbouring macroblock. Based on this characteristic, it can be assumed that true object motion vectors should always occur in a clustered form whereas false motion vectors will tend to appear as impulsive noise.

The simplest strategy is to define a noise tolerance threshold where the decision is based on that threshold value [2]. This approach is flawed however, since the threshold only performs well if the true and false motion vectors have different lengths. In the case where the length of both true and false vectors is equal, this technique does not separate true motion vectors from false ones.

One strategy for removing these false vectors is to consider gradually increasing the length of the true motion vector with a higher ratio compared to the increasing ratio of the length of the false motion vector, so that the ratio of the two different vector lengths is reduced gradually. When, finally, the true motion vector length becomes greater than that of the false one, a predefined threshold can separate the true one from the false one. To achieve this objective, accumulation of the mean vector length, in addition to the original vector length, is utilised. The reasoning behind this is that true motion vectors frequently occur in a clustered form, whereas false vectors tend to occur as isolated impulsive noise. Based on this rationale, the false motion vector elimination process has been formulated in the MAT filter, which is designed explicitly for this application.

The MAT filter has two phases. The first phase is basically an iterative in-place application of the mean filter. However, in this case, a major difference arises in how the in-place values are updated. For each iteration, the mean value is added to, instead of replacing, the existing value as follows:

$$\begin{bmatrix} o_x(k) \\ o_y(k) \end{bmatrix} = \begin{bmatrix} o_x(k) \\ o_y(k) \end{bmatrix} + \begin{bmatrix} \text{mean}_x(k) \\ \text{mean}_y(k) \end{bmatrix} \tag{1}$$

where $(o_x(k), o_y(k))$ represents the $x$ and $y$ components of the motion vector in the current block $k$, which are available after global motion cancellation, and $\text{mean}_x(k)$ and $\text{mean}_y(k)$ are the mean values of the $x$ and $y$ components of the motion vectors, respectively, in the neighbourhood of any kernel considered for the current block, $k$.

The second phase of the MAT filter is to apply the false motion vector elimination threshold, $T_f$, so that the only motion vectors retainedare those whose lengths are higher than $T_f$. This is mathematically formulated as:

$$\begin{cases} \begin{array}{l} \text{Eliminate the} \\ \text{vector in block } k, \end{array} & \begin{array}{l} \text{if } \sqrt{o_x(k)^2 + o_y(k)^2} \\ \le T_f; \end{array} \\ \begin{array}{l} \text{Retain the} \\ \text{vector in block } k, \end{array} & \text{otherwise.} \end{cases} \tag{2}$$

## 4. Computational Complexity Analysis

The computational complexity of the MAT filter depends on the kernel size used and the number of iterations involved. From Eq. (1), the total number of operations required for the MAT filter is $2\Re$ additions and two divisions for each iteration, where $\Re$ represents the kernel size. If the frame rate $f = 30$ fps, frame size $= [N_h, N_v]$, macroblock size $= [N, N]$, and the number of iterations is $L$, then the total number of operations per second required for the MAT filter is:

$$2(\Re + 1) \times L \times \frac{N_h \times N_v}{N^2} \times f \tag{3}$$

Assume that the block distortion is measured using MAE, which requires three basic operations per pixel. If the frame rate is $f = 30$, frame size $[N_h, N_v] = [352, 240]$, maximum displacement $d = 7$, and macroblock dimension $N = 16$, the number of integer arithmetic operations required for any BMA for motion estimation is bounded between 1.71 billion and 7.6 million per second using integer-pel accuracy [11].

Conversely, if a $3 \times 3$ sized kernel and, at most, 5 iterations are utilised, the total number of operations for the MAT filter is only 0.7 million per second. This overhead is not significant compared to the complexity involved in block motion estimation. These experimental results also prove that this overhead cost can be fully justified by the improvement in capturing a significant number of true motion vectors by elimination of false motion vectors from the motion vector field.

## 5. Experimental Results and Discussion

To evaluate the performance of MAT filter and the combined performance of the MAT filter and DTS algorithm for true object motion estimation, a number of experiments were performed using the standard and non-standard video sequences. The results for Table Tennis and Forman sequences have been included in this paper.

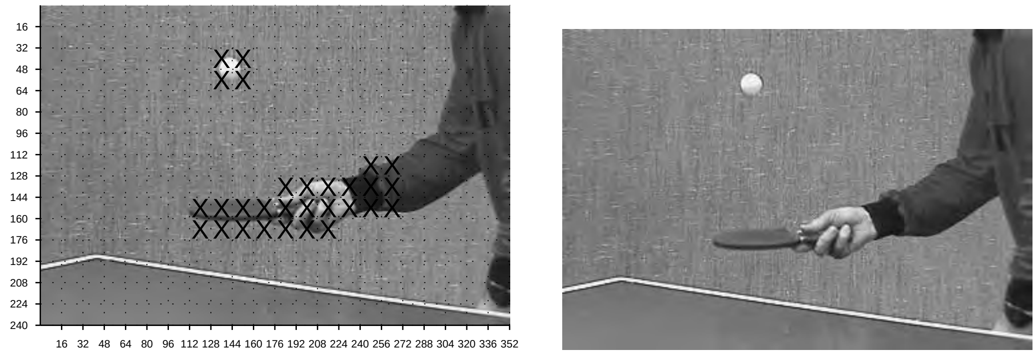
Before examining the performance of the filter, a few key points need to be highlighted in respect to applying the MAT filter.

- The MAT filter has been explicitly designed to eliminate false motion vectors while retaining the true motion vectors. It is not designed for vector field smoothing purposes.
- The MAT filter can be integrated with any existing BMA for true object motion vector capture.
- The number of iterations in the MAT filter depends on the video content and the performance of the search algorithm. These must be selected empirically for optimising the performance of any BMA using the MAT filter.
- The threshold $T_f$ setting in Eq. (2) is empirically derived to maximise the number of retained true object motion vectors while minimising the number of false motion vectors.
- The overall computational complexity of the MAT filter depends on the kernel size used and the number of iterations required in the whole process.
- One assumption in the literature is that motion vectors tend to occur in a clustered form which defines moving objects. For the inherent limitation of the BMA search technique, the performance of MAT filter will probably not be so effective, and capture rates will deteriorate if there is large cluster of false motion vectors with no moving objects. For most real world objects however, this clustering effect is rare, and thus the MAT filter will improve the overall number of false motion vectors eliminated.

Kernel size, representing the size of the neighbourhood to be considered for calculating the mean value, is an important factor in the performance of the MAT filter. Generally, higher kernel sizes are used for heavier smoothing of images or motion vector fields; the purpose of this filter, however, was not for smoothing the motion vectors. Most objects in real world video sequences occur in small clustered forms, where each object contains a few neighbouring macroblocks but not the whole frame. If a larger sized kernel is used, it includes a larger area of the frame, which will eventually reduce the lengths of the true vectors. In this case, in eliminating the false vectors, any value of threshold Tf will remove a higher number of true object motion vectors as well. For this reason, larger kernel sizes perform worse compared to smaller-sized kernels. Again from Eq. (1), it can be seen that the computational complexity for calculating the mean value is directly proportional to the kernel size utilised, so from a computational point of view, the smaller kernel incurs less computational cost compared to a larger kernel. For these reasons, the $3 \times 3$ sized kernel has been considered in this paper for analysis of the performance of the MAT filter.

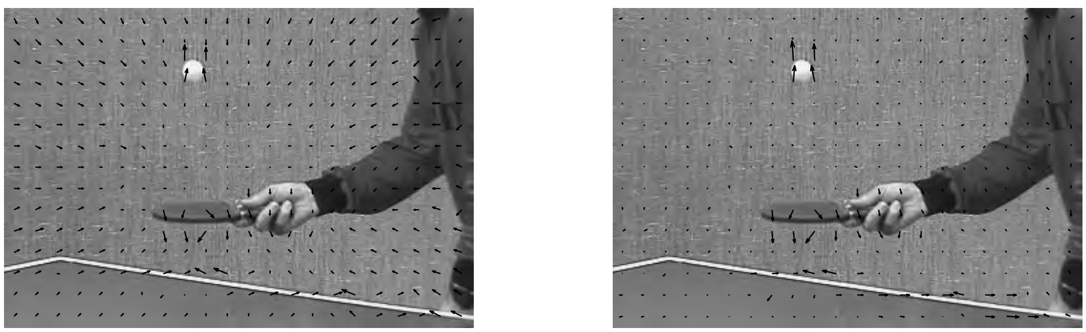
For the Table Tennis sequences, frames #32 and #33, shown in Fig. 1, were considered. Based on a priori knowledge about the moving objects in these frames, moving macroblocks, each having a size of $16 \times 16$ pixels, are identified manually and indicated with the use of an X sign in Fig. 1(a). The *motion vectors* (MVs) of these specific macroblocks are the true vectors, whereas the motion vectors of the remaining blocks are the false one.

The block motion vector calculated by the DTS algorithm is shown in Fig. 2(a), which contains true object motion as well as global motion. Figure 2(b) shows the global motion compensated motion vector needle diagram, from which it is clear that while the




(a) Current frame (b) Reference frame

**Fig. 1** (a) Current frame #32 where ×'s indicate the moving macroblocks and (b) reference frame #33 of the Table Tennis sequence.



(a) Block motion vector captured by the DTS algorithm (b) Object motion vectors after global motion compensation

**Fig. 2** Motion vectors captured by DTS algorithm; (a) before, and (b) after *global* motion compensation.

only moving objects are the ball, bat and the hand holding the bat, spurious (false) motion vectors exist together with the true object motion vectors.

To remove the false motion vectors after each iteration, a range of values for $T_f$, starting from an empirically selected low value, is gradually increased until all the false motion vectors are removed. The length of a motion vector increases with the number of iterations, so when the length of a motion vector is high, the range for the threshold, $T_f$, in eliminating false vectors, is also correspondingly high. The step size for incrementing $T_f$ was empirically selected. The percentage of true and false motion vectors after thresholding at different values of $T_f$ was then calculated.

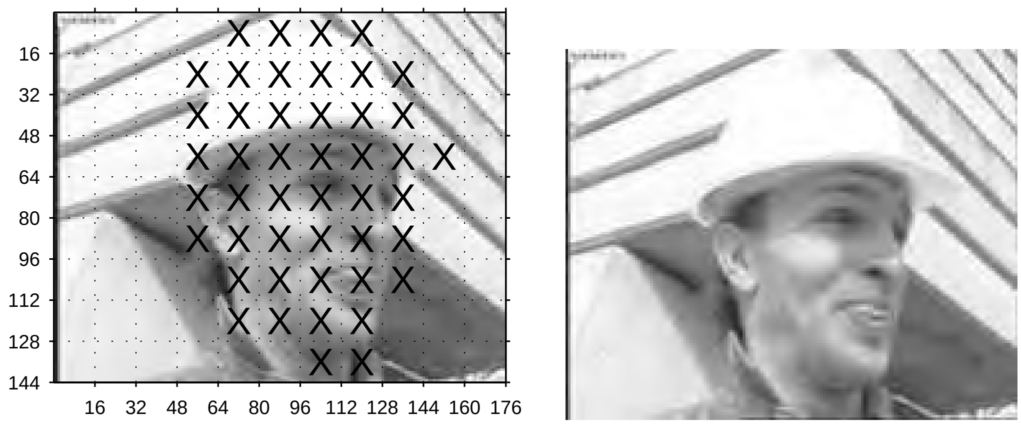
The experimental results for this test sequence are given in Table 1. To analyse the effect of the MAT filter, the experimental results from 0 to 5 iterations are shown in this table. It is important to clarify that the number of iterations used is not optimal, but based on the experimental results, it was found that the performance of all BMAs examined did not change significantly after four iterations. The increased percentage of true motion vectors captured for FS, NTSS, TSS and DTS was 10%, 6.7%, 13.3%, and 17%, respectively, from iteration 3 to 4, whereas it was only 6.3%, 3.3%, 6.7%, and 0% from iteration 4 to 5. For this reason, only those results from 0 to 5 iterations have been included.

It is clear that the percentage of true motion vectors captured by any BMA significantly increases with the number of iterations to the point at which the percentage of false motion vectors is zero. For example, the true motion vectors captured by the DTS algorithm increased from 13.3% to 80% using the MAT filter over 5 iterations. A similar trend was also found for the NTSS, TSS, and FTS algorithms. This demonstrates that when the MAT filter is used, the percentage of true object motion vectors captured significantly increases, while all false motion vectors are eliminated from the motion vector field.

It can be concluded that the DTS algorithm in conjunction with the MAT filter significantly outperformed the FS, TSS, and NTSS algorithms by capturing 36.7%, 50%, and 20% more true motion vectors respectively, with all false motion vectors being eliminated. It can also be concluded that the MAT filter improved not only the performance of the DTS algorithm, but also the performance of the FS, TSS, and NTSS algorithms by capturing 30.3%, 30%, and 60% more true motion vectors, respectively.

**Table 1** Performance comparison of the DTS and MAT filters in capturing true object motion vectors for the Table Tennis sequence.

| BMA | FS | | | NTSS | | | TSS | | | DTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # of iteration | True MV% | False MV% | $T_f$ | True MV% | False MV% | $T_f$ | True MV% | False MV% | $T_f$ | True MV% | False MV% | $T_f$ |
| 0 | 16.7 | 0.3 | 5.0 | 6.7 | 0.7 | 8.0 | 3.3 | 14.3 | 8.0 | 16.7 | 0.3 | 5.0 |
| | 13.3 | 0.0 | 6.0 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | 9.0 | 13.3 | 0.0 | 6.0 |
| 1 | 13.3 | 0.3 | 8.0 | 16.7 | 0.3 | 9.0 | 10.0 | 5.7 | 9.0 | 46.7 | 1.7 | 5.0 |
| | 13.3 | 0.0 | 9.0 | 13.3 | 0.0 | 10.0 | 6.7 | 3.3 | 10.0 | 40.0 | 0.0 | 6.0 |
| 2 | 26.7 | 0.3 | 12.0 | 23.3 | 0.3 | 12.0 | 13.3 | 6.3 | 13.0 | 56.7 | 0.3 | 8.0 |
| | 16.7 | 0.0 | 13.0 | 16.7 | 0.0 | 13.0 | 10.0 | 3.7 | 14.0 | 46.7 | 0.0 | 9.0 |
| 3 | 26.7 | 0.3 | 19.0 | 56.7 | 0.7 | 15.0 | 16.7 | 4.7 | 21.0 | 70.0 | 0.3 | 12.0 |
| | 26.7 | 0.0 | 19.5 | 50.0 | 0.0 | 16.5 | 10.0 | 1.7 | 22.5 | 63.3 | 0.0 | 13.0 |
| 4 | 40.0 | 0.3 | 23.5 | 60.0 | 0.3 | 24.0 | 26.7 | 1.3 | 42.0 | 80.0 | 1.3 | 18.0 |
| | 36.7 | 0.0 | 24.0 | 56.7 | 0.0 | 26.0 | 23.3 | 0.7 | 44.0 | 80.0 | 0.0 | 20.0 |
| 5 | 43.3 | 0.3 | 60.0 | 66.7 | 0.3 | 42.0 | 36.7 | 1.0 | 64.0 | 80.0 | 0.7 | 37.5 |
| | 43.3 | 0.0 | 62.0 | 60.0 | 0.0 | 45.0 | 30.0 | 0.7 | 65.0 | 80.0 | 0.0 | 40.0 |




(a) Current frame (b) Reference frame

**Fig. 3** (a) Current frame #8 where ×'s indicate the moving macroblocks and (b) reference frame #9 of the Foreman sequence.

Table 2 is also obtained utilising the same procedure explained above for Foreman video sequence, where the frame pairs #8 and #9 were considered shown in Fig. 3. The block motion vectors captured by different algorithm are shown in Fig. 4. Table 2 shows that by using any threshold value without the MAT filter, no algorithm could eliminate the false motion vectors without also eliminating the true motion vectors. The percentage of true motion vectors captured is significantly increased when the MAT filter is combined with a BMA. For example, the true motion vectors captured by the FS algorithm increased from 0% to 77% using the MAT filter over 5 iterations. A similar trend was found for the NTSS, TSS, and DTS algorithms. These results, again, endorse the effectiveness of the MAT filter in eliminating false vectors while retaining the true object motion vectors.

## 6. Conclusions

In this paper, we have presented a very promising technique for efficient filtering of some unreliable motion vectors (false) captured by any block-based motion estimation techniques. Our experimental results have clearly proved that our proposed *mean accumulated threshold* (MAT) filter very elegantly removes the false vector while the true vector remains. It is also proved that MAT filter in conjunction with DTS algorithm provides a very useful tool for block based true object motion estimation.

The effectiveness of the MAT filter in false motion vector elimination in improving the performance of the DTS algorithm in capturing true object motion has opened a potential new research direction in block-based true object motion estimation. Future research could extend to optimising the performance of the MAT filter by analysing the impact of video content on different design parameters.

**References**

[1] S. Jeannin and A. Divakaran, MPEG-7 Visual Motion Descriptors, IEEE Trans. Circuits & Systems for Video Tech., vol. 11, pp. 720–724, 2001.

[2] S. Y. Kim and Y. M. Ro, Fast content-based MPEG video indexing using object motion, Proc. IEEE Region 10 Conf., Cheju Island, South Korea, vol. 2, pp. 1506–1509, 1999.

(a) FS (b) TSS

(a) NTSS (b) DTS

**Fig. 4** Motion vector obtained from all four search algorithms applied to the frame pair shown in Fig. 3 for the Foreman sequence.

**Table 2** Performance comparison of the DTS and MAT filters in capturing true object motion vectors for the Foreman sequence.

| BMA | FS | | | NTSS | | | TSS | | | DTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # of iteration | True MV% | False MV% | $T_f$ | True MV% | False MV% | $T_f$ | True MV% | False MV% | $T_f$ | True MV% | False MV% | $T_f$ |
| 0 | 4.6 | 13.2 | 3.0 | 2.2 | 1.9 | 6.0 | 2.2 | 3.8 | 5.6 | 4.3 | 3.8 | 2.6 |
| | 0.0 | 11.3 | 3.2 | 0.0 | 1.9 | 6.2 | 0.0 | 3.8 | 5.8 | 0 | 1.9 | 2.8 |
| 1 | 4.4 | 11.3 | 4.0 | 2.2 | 1.9 | 7.0 | 2.2 | 3.8 | 5.6 | 13.0 | 1.9 | 3.0 |
| | 0.0 | 11.3 | 4.2 | 2.2 | 0.0 | 7.2 | 0.0 | 3.8 | 5.8 | 10.9 | 0.0 | 3.2 |
| 2 | 4.4 | 1.9 | 7.4 | 13.0 | 1.9 | 7.0 | 2.2 | 1.9 | 8.6 | 43.5 | 1.9 | 3.6 |
| | 0.0 | 1.9 | 7.6 | 10.9 | 0.0 | 7.2 | 0.0 | 1.9 | 8.8 | 39.1 | 0.0 | 3.8 |
| 3 | 37.0 | 1.9 | 10.0 | 58.7 | 1.9 | 8.5 | 2.2 | 1.9 | 15.0 | 71.7 | 1.9 | 4.8 |
| | 32.6 | 0.0 | 10.2 | 56.5 | 0.0 | 8.8 | 0.0 | 1.9 | 15.5 | 69.6 | 0.0 | 5.0 |
| 4 | 73.9 | 1.9 | 10.5 | 78.3 | 1.9 | 9.6 | 13.0 | 1.9 | 25.0 | 78.3 | 1.9 | 7.6 |
| | 73.9 | 0.0 | 11.0 | 76.1 | 0.0 | 9.9 | 6.5 | 0.0 | 26.0 | 78.3 | 0.0 | 7.8 |
| 5 | 76.1 | 1.9 | 17.5 | 80.4 | 1.89 | 16.0 | 43.5 | 1.9 | 33.0 | 80.4 | 1.9 | 13.0 |
| | 77.0 | 0.0 | 18.0 | 78.3 | 0.00 | 16.5 | 41.3 | 0.0 | 34.5 | 80.4 | 0.0 | 13.2 |

[3] H. Zen, T. Hasegawa, and S. Ozawa, Moving object detection from MPEG coded picture, Proc. ICIP'99, Kobe, Japan, vol. 4, pp. 25–29, 1999.

[4] E. Sahouria and A. Zakhor, Motion Indexing of video, Proc. ICIP'97, Santa Barbara, CA, USA, vol. 2, pp. 526–529, 1997.

[5] E. Ardizzone and M. L. Cascia, Video indexing using optical flow field, Proc. ICIP'96, Lausanne, Switzerland, vol. 3, pp. 831–834, 1996.

[6] K. P. Horn and B. G. Schunck, Determining Optical flow, AI, vol. 17, pp. 185–203, 1981.

[7] F. Dufaux and F. Moscheni, Motion estimation techniques for digital TV: a review and a new contribution, Proc. IEEE, vol. 83, pp. 858–876, 1995.

[8] S. Ji and H. W. Park, Moving-object segmentation with adaptive sprite for DCT-based video coder, Proc. ICIP'01, Greece, vol. 3, pp. 566–569, 2001.

[9] A. Yoneyama, Y. Nakajima, H. Yanagihara, and M. Sugano, Moving object detection and identification from MPEG coded data, Proc. ICIP'99, 1999.

[10] G. Sorwar, M. Murshed, and L. Dooley, Distance dependent thresholding search for fast motion estimation in real world video coding application, Proc. APCCAS'02, vol. 2, pp. 519–524, 2002.

[11] G. Sorwar, M. Murshed, and L. Dooley, Fast block-based true motion estimation using distance dependent thresh-

olds, J. of Research and Practice in Inf. Tech., vol. 36, no. 1, pp. 83–95, February 2004.

[12] D. R. K. Brownrig, The weighted median filter, Commun. ACM, vol. 27, no. 8, pp. 807–18, 1984.

[13] Pitas and Venetsanopoulos A., Nonlinear digital filters, Kluwer, Dodrecht, 1990.

[14] G. Sorwar, M. Murshed, and L. Dooley, A novel filter for block based object motion estimation, Proc. DICTA'02, Melbourne, Australia, pp. 201–206, 2002.

[15] J. R. Jain and A. K. Jain, Displacement measurement and its application in inter frame image coding, IEEE Trans. on Commun., vol. 29, pp. 1799–1808, 1984.

[16] T. Koga, K. Linuma, A. Hirano, Y. Iijima, and T. Ishiguro, Motion-compensated inter frame coding for videoconferencing, Proc. NTC, New Orleans, LA, USA, pp. G5.3.1–G.5.3.5, 1981.

[17] R. Li, B. Zeng, and M. L. Liou, A new three-step search algorithm for block motion estimation, IEEE Trans. on Circuits & Systems for Video Tech., vol. 4, pp. 438–442, 1994.

[18] G. Sorwar, M. Murshed, and L. Dooley, Fast Global Motion Estimation using Iterative Least-Square Estimation Technique, Proc. ICICS-PCM 2003, Singapore, 2003.

**Golam Sorwar** received B.Sc.(Hons) degree in Electrical and Electronic Engineering in 1994 from Bangladesh University of Engineering and Technology (BUET), Bangladesh, M.Sc. degree in Electrical, Electronic and Systems Eng. in 1998 from National University of Malaysia, Malaysia, and Ph.D. degree in Information Technology from Monash University, Australia in 2003. He is currently a lecturer at School of Multimedia and Information Technology at Southern Cross University, Australia, where his major research interests are in the fields of image/video coding, indexing, retrieval, motion estimation, shot detection, multimedia communication, and artificial intelligence. He is a member of IEB, ACS and IEEE.

**Manzur Murshed** received his B.Sc.Engg. (Hons.) degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh, in 1994 and Ph.D. degree in Computer Science from the Australian National University, Australia, in 1999. He is currently the Director of Research and a Senior Lecturer at Gippsland School of Computing and Information Technology at Monash University, Australia, where his major research interests are in the fields of multimedia signal processing and communications, parallel and distributed computing, simulations, and multilingual systems development. He has published more than 85 peer-reviewed journal articles, book chapters, and conference papers. He is the recipient of numerous academic awards including the University Gold Medal by BUET. He is a member of IEEE.

**Laurence Dooley** received his B.Sc.(Hons), M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Wales, Swansea in 1981, 1983 and 1987 respectively. Since 1999, he has been Professor of Multimedia Technology in the Gippsland School of Information Technology, Monash University, Australia, where his main research interests are multimedia signal processing, mobile communications, sensor networks and bioinformatics. He has published over 125 international scientific peer-reviewed journals, book chapters and conference papers. He is also Director of the Monash Regional Centre for Information and Communications Technology, which has created a Technology Transfer Gateway (TTG) to stimulate the innovation cycle of regional small business and provide a bridge to commercialisation of new emerging technologies for its industry partners. The TTG project is partly funded by an AusIndustry /Innovation Access/ grant. Professor Dooley is a Senior Member of the IEEE, a Chartered Engineer (C.Eng), and a corporate member of the BCS.