A Novel Filter for Block-Based Object Motion Estimation

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

Abstract

Noises, in the form of false motion vectors, cannot be avoided while capturing block motion vectors using block-based motion estimation techniques. Similar noises are further 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. We observe that the performance of the mean and the median filters in removing false motion vectors, for estimating "true" object motion, is not satisfactory, especially when the size of the object is significantly smaller than the scene. In this paper we introduce a novel filter, named as the Mean-Accumulated-Thresholded (MAT) filter, in order to capture "true" object motion vectors from video sequences with or without the camera motion (zoom and/or pan). Experimental results on representative standard video sequences are included to establish the superiority of our filter compared with the traditional median and mean filters.

1. Introduction

Extracting motion parameters from image sequences has been a central theme in the areas of computer vision and image coding. There are many types of motion estimation algorithm such as pel-recursive [22], block-matching [8], and optical flow based method [7]. In general, block-matching algorithm [8] attracted wider acceptance due to its simplicity, robustness, and lesser hardware complexity which is already adopted by a large number of video coding standards (MPEG-1/2 and H.261/262/263 etc.).

The exhaustive block-matching *full-search* (FS) [8], where each candidate block is searched for the closest match within the entire search region, it generally provides reasonably good performance with the expense of high computational time.

Several fast algorithms have already been proposed to address the above issue. The three-step search algorithm (3SS) [12], the new three-step search (N3SS) [13], the four-step search algorithm (4SS) [17], and the cross-search algorithm [6] are based on the assumption that the block distortion measure increases as the checking points move away from the global minima. But this assumption does not hold true in the real world video sequences [4]. Moreover, search directions of the above algorithms can be ambiguous and therefore, may converge to local minima.

In true motion estimation, where object and/or camera motions are estimated, the FS algorithm tends to pick many "false" motion vectors even when no object motion is present in the search region. This is due to the fact that the distortion of an object in a video frame is proportional to its velocity and therefore, as the length of a motion vector grows so does the block difference error. The FS algorithm is, therefore, modified in our paper [19] by introducing distance dependent *linear threshold* (LT) and *exponential threshold* (ET) named as the Modified Full Search (MFS) algorithm. In this paper we use this MFS algorithm for estimating true block motions.

Block motion is governed by the movement due to the camera (pan and/or zoom) referred as global motion, movement of the objects referred as object motion or "true" motion, or both. Many motion estimation techniques ignore this aspect and make no distinction between the global and the local motion. However, separating these two classes of motions is significant for "true" object motion. In case where both the local and the global motions are present in the video sequences, "true" object motions (i.e., the local motion), necessary for object-based video representation, segmentation, and retrieval, can only be obtained by canceling out the global motion component from the block motion, known as *global motion compensation*.

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 impulse noises (false 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, we must filter out these impulse noises from the scene.

Many types of filters have already been proposed and examined for filtering impulse noises. Among them the *median filter* and the *mean filter* are widely used. While applied to reduce noises in an image, the median filter performs better than the mean filter as the mean filter often blurs the edges [5][21]. The same is not true for filtering out noises from the motion vectors, especially when objects are quite small compare to the size of the scene. In such cases, the median filter tends to remove significant number of "true" object motion vectors along the edge of the objects whereas the mean filter reduces the length of all the motion vectors, including the "true" ones. To address this issue we develop a new filter, named as the *Mean-Accumulative-Thresholded* (MAT) filter, which is successfully applied to a number of representative standard video sequences to capture the "true" object motion vectors.

The remainder of this paper is organized as follows. Section 2 describes the block motion estimation technique used in this paper. The parametric global motion estimation techniques are introduced in Section 3. In Section 4 the general process of estimating local (object) motion, including our proposed MAT filter, is discussed. Some experimental results are included in Section 5. Section 6 concludes the paper.

2. Block Motion Estimation

In [19], 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. To address this issue we modified the FS algorithm (names as the MFS algorithm) by introducing distance dependent thresholds. The MFS algorithm not only avoids capturing a large number of "false" motion vectors but also reduces the search time significantly. In this paper we use the MFS algorithm for estimating true block motions.

3. Global Motion Estimation

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.

3.1. Motion Model

Techniques for *global motion estimation* (GME) have been proposed in [9][18][20]. Most of the GME methods differ in the parametric model to represent the camera motion as well as in the technique to estimate the parameters of the chosen model. Although a complex model results in a better description of the motion, it also leads to a greater difficulty in parameter estimation and higher computational complexity. Conversely, a simple model is sufficient enough to represent the global motion of a small video sequence, especially when the global motion is primarily used for compensating the camera motion from the block motion to obtain "true" object motion.

The conventional block-matching algorithm assumes that all the pixels in a block have equal displacements, and thus estimates one motion vector for each block. Let there be N blocks in a video frame. Lat us assume that the motion vector of a block is the motion vector of the center pixel of that block. Let $(v_x(k), v_y(k))$ be the measured motion vector, according to our MFS algorithms explained in Section 2, of the block k, $k = 0, 1, \ldots, N-1$, whose center pixel's coordinates are $(s_x(k), s_y(k))$ with respect to the center of the frame.

For global motion estimation, we consider the 4-parameter motion model depicted in [18] with some modification. The generalized 4-parameter motion model for camera zoom and pan is defined as

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

where

$$a_1 = z_x \text{ and } a_2 = f_1(p_x, p_y) \quad 2(a)$$

$$a_3 = z_y \text{ and } a_4 = f_2(p_y, z_y) \quad 2(b)$$

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.

3.2. Motion Parameter Estimation

Now consider the *iterative least-square estimation* algorithm for obtaining the optimal values for camera parameters (a_1, a_2, a_3, a_4) by using the following criteria:

$$\min_{a_1, a_2} \sum_{k=0}^{N-1} \left(v_x(k) - a_1 s_x(k) - a_2 \right)^2 \quad (3)$$

$$\min_{a_3, a_4} \sum_{k=0}^{N-1} \left(v_y(k) - a_3 s_y(k) - a_4 \right)^2 \quad (4)$$

By differentiating with respect to the parameters, and setting the derivatives to zero, we obtain the following solution as shown in (5, 6, 7, 8).

$$a_1 = \frac{N \sum_{k=0}^{N-1} v_x(k) s_x(k) - \left(\sum_{k=0}^{N-1} v_x(k) \right) \left(\sum_{k=0}^{N-1} s_x(k) \right)}{N \sum_{k=0}^{N-1} s_x^2(k) - \left(\sum_{k=0}^{N-1} s_x(k) \right)^2} \quad (5)$$

$$a_2 = \frac{\left(\sum_{k=0}^{N-1} v_x(k)\right)\left(\sum_{k=0}^{N-1} s_x^2(k)\right) - \left(\sum_{k=0}^{N-1} v_x(k)s_x(k)\right)\left(\sum_{k=0}^{N-1} s_x(k)\right)}{N\sum_{k=0}^{N-1} s_x^2(k) - \left(\sum_{k=0}^{N-1} s_x(k)\right)^2} \tag{6}$$

$$a_3 = \frac{N\sum_{k=0}^{N-1} v_y(k)s_y(k) - \left(\sum_{k=0}^{N-1} v_y(k)\right)\left(\sum_{k=0}^{N-1} s_y(k)\right)}{N\sum_{k=0}^{N-1} s_y^2(k) - \left(\sum_{k=0}^{N-1} s_y(k)\right)^2} \tag{7}$$

$$a_4 = \frac{\left(\sum_{k=0}^{N-1} v_y(k)\right)\left(\sum_{k=0}^{N-1} s_y^2(k)\right) - \left(\sum_{k=0}^{N-1} v_y(k)s_y(k)\right)\left(\sum_{k=0}^{N-1} s_y(k)\right)}{N\sum_{k=0}^{N-1} s_y^2(k) - \left(\sum_{k=0}^{N-1} s_y(k)\right)^2} \tag{8}$$

Since all the blocks are taken into consideration, the above estimate will be affected by the presence of the local motion. To eliminate this influence, we use the above procedure iteratively, each time eliminating the blocks whose motion vectors do not match with the so-far-estimated global motion fields. As observed in [18], the iteration converges very quickly in our experiments.

4. Object Motion Estimation

In case where both the local 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*.

Once the global motion parameters for the scene is calculated according to section 3, the "true" object motion vector ($o_x(k)$, $o_y(k)$) of the block k, k = 0, 1, …, N-1, can be calculated as:

$$\begin{bmatrix} o_x(k) \\ o_y(k) \end{bmatrix} = \begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix} - \begin{bmatrix} a_1 s_x(k) \\ a_3 s_y(k) \end{bmatrix} - \begin{bmatrix} a_2 \\ a_4 \end{bmatrix} \tag{9}$$

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 impulse noises 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, we must filter out these impulse noises from the scene.

Many types of filters have already been proposed and examined for filtering impulse noises. Among them the *median filter* and the *mean filter* are widely used [2][5][11][16][21]. The median filter and its variants have already been applied in many applications for noise rejection from block motion vectors [1][10][14][23].

4.1. The Mean Filter

The idea of mean filtering is simply to replace each value with the mean (`average') value of its neighbors, including itself. This has the effect of smoothing values that are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter [24]. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used. Two major characteristics of the mean filter are:

- A single very unrepresentative value can significantly affect the mean value of its neighborhood.
- When the filter neighborhood straddles an edge, the filter will interpolate new values.

4.2. The Median Filter

Like the mean filter, the median filter considers each value in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the value with the mean of neighboring values, it replaces it with the median of those values. Two major characteristics of the median filter are:

- The median is a more robust average than the mean and so a single very unrepresentative value in a neighborhood will not affect the median value significantly.
- Since the median value must actually be one of the values in the neighborhood, the median filter does not create new unrealistic values when the filter straddles an edge.

4.3. The Mean-Accumulated-Thresholded (MAT) Filter

While applied to reduce noises in an image, the median filter performs better than the mean filter as the mean filter often blurs the edges [5][21]. The same is not true for filtering out noises from the motion vectors, especially when objects are quite small compare to the size of the scene. In such cases, the median filter tends to remove significant number of "true" object motion vectors along the edge of the objects. If the length of the "true" object motion vector is of same order of the introduced impulsive noises after the global motion compensation, a single iteration of the mean filtering would fail to remove all the impulsive noises, introduced by the global motion compensation, even after using a threshold value. To address this issue we introduce a new filter, named as the *Mean-Accumulated-Thresholded* (MAT) filter.

The MAT filter has two phases. The first phase of the MAT filter is basically an iterative "in-place" application of the mean filter. But the major difference lies in how the "in-place" values are updated. In each iteration, the mean value is added on top, instead of replacing, the existing value as follows:

$$\begin{bmatrix} o_x(k) \\ o_y(k) \end{bmatrix} = \begin{bmatrix} o_x(k) \\ o_y(k) \end{bmatrix} + \begin{bmatrix} \text{mean}_x(k) \\ \text{mean}_y(k) \end{bmatrix} \qquad (10)$$

where, $\text{mean}_x(k)$ and $\text{mean}_y(k)$ are the mean values, along the *x*-axis and the *y*-axis respectively, in the 3×3 neighborhood kernel for all *k*, *k* = 0, 1, …, *N*-1.

With the mean and the median filters, even after the iterative "in-place" application, the length of the updated motion vectors will never exceed the maximum length of the original vectors in the neighborhood. But the same is not true for the MAT filter. Just after a few iterations (as low as 2), length of the "true" object motion vectors will be increased significantly, compare to the other vectors, including the impulses introduced during the global motion compensation and/or due to the limitations of the block-based motion estimation.

It is, therefore, highly likely that only the "true" object motion will be retained if the vectors, with length higher than a preset threshold, are selected as the last phase of the MAT filter.

5. Experimental Results

This MAT filter has been successfully applied to a number of representative standard video sequences to capture the "true" object motions vectors. Throughout the experiments, we use $M = N = d = 16$, i.e., each frame is divided into 16×16 pixel blocks and the size of the search region is 49×49 pixels, where at most 33^2 search points are used. All experiments are performed on the luminance (Y-component) of the frames.

In Figures 1–3, we present (a) the current frame, (b) the next frame, (c) block motion vectors computed using the MFS algorithm [19], (d) object motion vectors using the median filter of 3×3 kernel, (e) object motion vectors using the mean filter of 3×3 kernel, and (f) object motion vectors using the proposed MAT filter. In all the above-mentioned figures, the MAT filter outperforms the popular median filter, while capturing "true" object motion.

6. Conclusions and Discussion

The median and the mean filters and their variants have been used widely to remove noises from images and to smooth global motion vectors of video sequences. We have observed that the performance of these filters in removing false motion vectors for estimating "true" object motion is not satisfactory, especially when the size of the object is significantly smaller than the scene. In this paper we have introduced a novel filter, named as the Mean-Accumulated-Thresholded (MAT) filter, in order to capture "true" object motion vectors from video sequences with or without the camera motion (zoom and/or pan). Experimental results on representative standard video sequences have been included to establish the superiority of our filter compared with the mean and the median filters.

It is worth mentioning that although the MAT filter increases the length of the original object motion vectors significantly, it should not cause any problem as long as these vectors are not used for video coding. In case we are interested in capturing object motion vectors of "normal" length, it can easily be achieved by normalizing the MAT filtered vectors.

Although in our definition, the MAT filter uses the mean filter of 3×3 kernel, any other kernel size can also be used without loosing any generality. No study is done on the optimal kernel size to be used with the MAT filter. In future, we also like to explore whether different optimal kernel sizes exist for different video sequences with objects of different velocity.

7. References

[1] Avrithis Y.S., Doulamis N.D., Doulamis A.D. and Kollias S.D., "Efficient content representation in MPEG video databases," Proceedings. IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 91-5, 1998.

[2] Brownrig D.R.K., "The weighted median filter", Comm. of the ACM, vol. 27, no. 8, pp. 807-18, 1984.

[3] Chieh-Min Fan and Namazi N.M., "Simultaneous motion estimation and filtering of image sequences," IEEE Trans. of Image Processing, vol. 8, no. 12, pp. 1788-95, 1999.

[4] Chow H.-K. and Liou M.L., "Genetic motion search algorithm for video compression," IEEE Trans. on Circuits and Systems for Video Technology, vol. 3, pp. (s): 440–445, 1993.

[5] Davies E., Machine Vision: Theory, Algorithms and Practicalities, Academic Press, 1990.

[6] Ghanbari M, "The cross-search algorithm for motion estimation (image coding)," IEEE Trans. on Comm., vol. 38, pp. 950-953, 1990.

[7] Horn K.P. and Schunck B.G., "Determining Optical flow," Artificial Intelligence, Vol. 17, pp. 185-203, 1981.

[8] Jain J.R. and Jain A.K., "Displacement measurement and its application in inter frame image coding," IEEE Trans. Comm., vol. COM-29, pp. 1799-1808, 1984.

[9] Jozawa H., Kamikura K., Sagata A., Kotera H. and Watanabe H., "Two-stage motion compensation using adaptive global MC and local affine MC," IEEE Trans.

on Circuits & Systems for Video Technology, vol. 7, no. 1, pp. 75-85, 1997.

[10] Kim J.-G., Chang H. S., Kim J., and Kim H.M. “Efficient camera motion characterization for MPEG video indexing,” IEEE International Conference on Multimedia and Expo. ICME2000.

[11] Kim J.-S. and Park H.W., “Adaptive 3D median filtering for restoration of an image sequence corrupted by impulse noise,” Signal Processing: Image Comm., vol. 16, no. 7, pp. 657-68, 2001.

[12] Koga T., Iinuma K., Hirano A., Iijima Y. and Ishiguro T., “Motion-compensated inter frame coding for videoconferencing,” IEEE National Telecommunications Conference, vol. 4, pp. G5. 1-5, 1981.

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

[14] Milanese R., Deguillaume F. and Jacot-Descombes A., “Efficient segmentation and camera motion indexing of compressed video,” Real-Time Imaging 1999.

[15] Nam J.-Y., Seo J.-S., Kwak J.-S., Lee M.-H. and Yeong H.H., “New fast-search algorithm for block matching motion estimation using temporal and spatial correlation of motion vector,” IEEE Trans. on Consumer Electronics, vol. 46, pp. 934-942, 2000.

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

[17] Po L.-M. and Ma W.-C., “Novel four-step search algorithm for fast block motion estimation,” IEEE Trans. on Circuits & Systems for Video Technology, vol. 6, pp. 313-317, 1996.

[18] Rath G.B. and Makur A., “Iterative least squares and compression based estimations for a four-parameter linear global motion model and global motion compensation,” IEEE Trans. on Circuits & Systems for Video Technology, vol. 9, no. 7, pp. 1075-99, 1999.

[19] G. Sorwar, M. Murshed, and L. Dooley, "Fast Block-based True Motion Estimation using Adaptive Distance Dependent Thresholds in the Full-Search Algorithm, Submitted in Pattern recognition Letters, 2001.

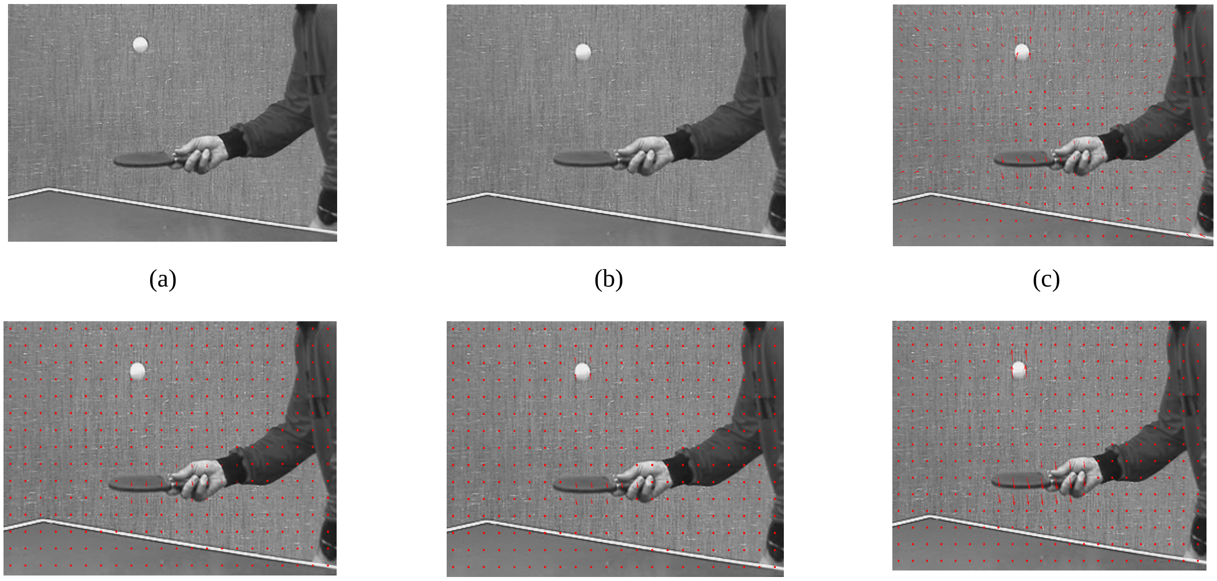
[20] Tse Y.T. and Baker R.L., “Global zoom/pan estimation and compensation for video compression,” International Conference on Acoustics, Speech and Signal Processing, vol. 4. pp. 2725-8, 1991.

[21] Vernon D., Machine Vision, Prentice-Hall, 1991.

[22] Walker D.R. and Rao K.R., “Improved pel-recursive motion compensation,” IEEE Trans. Comm., vol. COM-32, pp. 1128-1134, 1984.

[23] Zhong D. and Shih-Fu, “Video Object Model and Segmentation for Content Based Video Indexing," ISCAS'97, vol. 2, pp. 1492-1495, 1997.

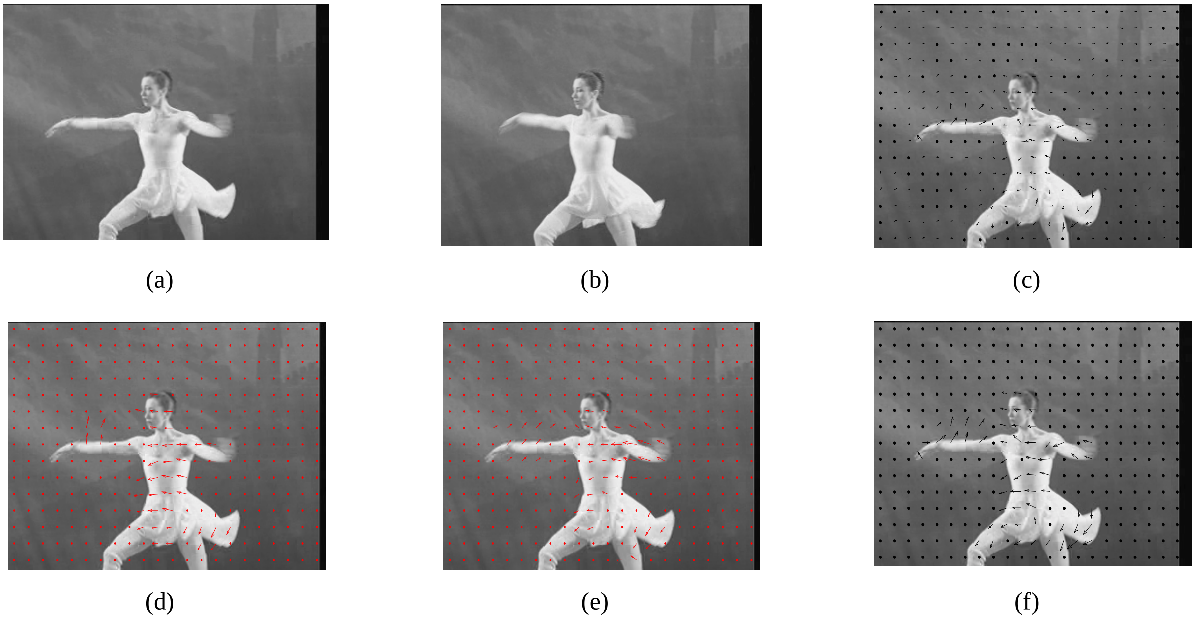
[24] Boyle R. and Thomas R., Computer Vision: A First Course, Blackwell Scientific Publications, pp. 32-34, 1988.



(a) (b) (c)

(d) (e) (f)

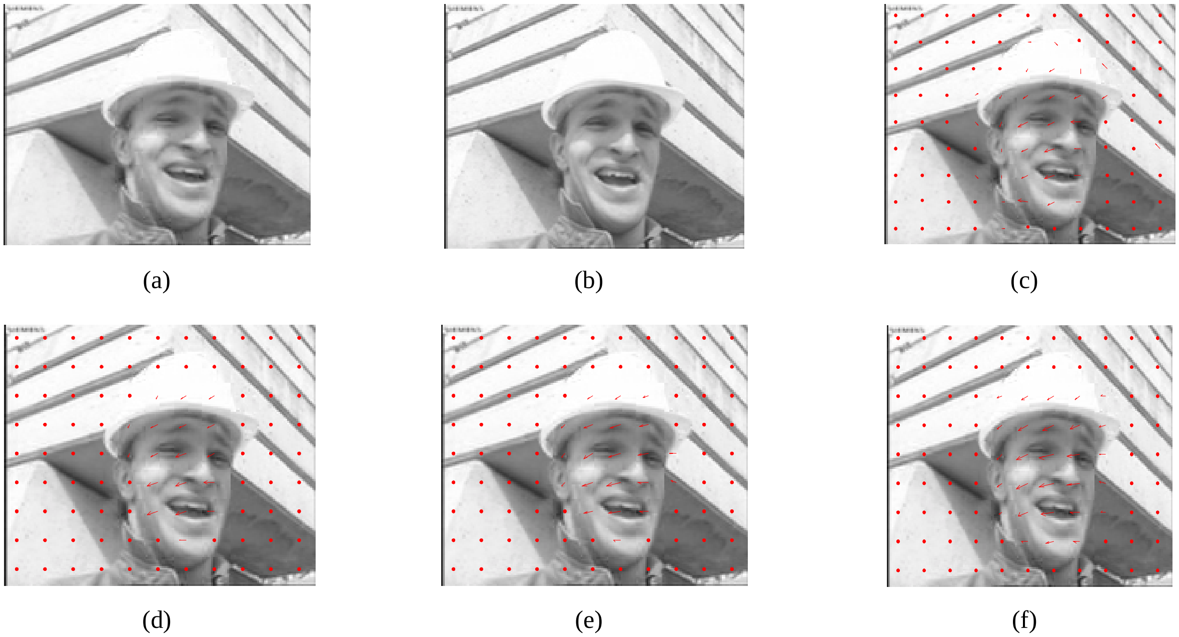
Figure 1: (a) Current frame (frame #32 of “Tennis”); (b) Next frame, (frame #33 of the same video sequence); (c) Block motion vectors computed using the LT algorithm [19]; (d) Object motion vectors using the median filter of 3×3 kernel; (e) Object motion vectors using the mean filter of 3×3 kernel; (f) Object motion vectors using the MAT filter.



(a) (b) (c)

(d) (e) (f)

Figure 2: (a) Current frame (frame #99 of “Ballet”); (b) Next frame, (frame #100 of the same video sequence); (c) Block motion vectors computed using the LT algorithm [19]; (d) Object motion vectors using the median filter of 3×3 kernel; (e) Object motion vectors using the mean filter of 3×3 kernel; (f) Object motion vectors using the MAT filter.



(a) (b) (c)

(d) (e) (f)

Figure 3: (a) Current frame (frame #15 of “Foreman”); (b) Next frame, (frame #16 of the same video sequence); (c) Block motion vectors computed using the LT algorithm [19]; (d) Object motion vectors using the median filter of 3×3 kernel; (e) Object motion vectors using the mean filter of 3×3 kernel; (f) Object motion vectors using the MAT filter.