Defence Science Journal, Vol. 60, No. 1, January 2010, pp. 39-47


# Motion Segmentation Algorithm using Spectral Framework

Vrinthavani R[*] and M.R. Kaimal[**]

[*]*Center for Airborne Systems, DRDO, Hyderabad*
[**]*Amrita Vishwa Vidyapeetham University, Kollam, Kerala*

**ABSTRACT**

This paper presents a new motion segmentation method using iterative maximum likelihood framework. The method consists of two steps. In the first step, motion regions are detected and motion vectors are computed for these detected regions. In the second step, a similarity matrix is computed from the motion vectors and motion segmentation is done using maximum likelihood method. The method has been tested using real world motion sequences and is found to give very low error rate and it also detects the slow moving objects.

**Keywords:** Spectral framework, motion segmentation, motion vectors, BMA, block matching algorithm

## 1. INTRODUCTION

Motion segmentation is an indispensable step in many vision-based applications. Most of the existing motion segmentation methods do not accurately detect slow moving objects from the video sequences. One of the most popular methods used for motion segmentation is background subtraction. An important drawback of this technique is that it uses the same threshold for every pixel. This way a moving object is likely to disappear in the background when entering a darker (shaded) area in the scene.

Another commonly used method for segmentation is the EM algorithm which is an iterative mechanism. When EM algorithm[1,2] is used each object is represented by a separate Gaussian distribution. The EM algorithm[9] has proved to be cumbersome to use in practice, due to the problems of estimating the parameters of the motion mixture model and of controlling its structure.

Another approach for segmentation and grouping[3,4,5] is graph-spectral[6] methods. These methods all share the feature that they use the eigenvectors of a weighted adjacency matrix to locate salient groupings of objects. At the level of image segmentation, several authors have used algorithms based on the eigen-modes of an affinity matrix to iteratively segment image data. For instance, Sarkar and Boyer[4] proposed a method which uses the leading eigenvector of the affinity matrix, and this locates clusters that maximize the average association. This method is applied to locating line-segment groupings. Perona and Freeman[5] gave a similar method which uses the second largest eigenvector of the affinity matrix. The method of Shi and Malik[3], on the other hand, uses the normalized cut which balances the cut and the association. Clusters are located by performing a recursive bisection using the eigenvector associated with the second smallest eigenvalue of the Laplacian matrix (the degree matrix minus the adjacency matrix), i.e. the Fiedler vector.

Kelly and Hancock[7] have developed an iterative spectral framework for pairwise clustering. They have used maximum likelihood method[6] to detect moving objects by performing pairwise clustering on a set of motion vectors. There are two problems with this approach. First, in order to reduce the motion vector noise, it uses a multi-resolution block matching method to estimate the motion field. Hence computational cost increases. Another drawback is that it is unable to detect slow moving objects in the video sequence. These problems are addressed in this work.

The method proposed in this paper also uses the iterative spectral framework[7]. In order to reduce noise without increasing computational complexity, the motion regions are detected and motion vectors are computed only for these regions using a block matching algorithm. Instead of using only one previous frame to detect motion pixels, the proposed method uses a set of *m* frames to detect the motion region. Hence the slow moving objects are also detected. The proposed segmentation method has two steps. In first step, motion estimation is done by finding the motion region and applying the block matching algorithm (BMA) to obtain the motion vector. In second step, iterative spectral framework[7] is used to cluster the motion regions.

## 2. COMPUTING MOTION VECTOR

Motion vectors are computed using single-resolution BMA using spatial/temporal correlation[9]. This BMA is based on predictive search that reduces computational complexity and provides a reliable performance. The method measures the similarity of motion blocks using spatial correlation and uses predictive search to efficiently compute block correspondences in different frames. BMA assumes that the translational motion from frame to frame is constant.

Received 22 September 2009

The current frame is divided into non overlapping blocks which are then matched with a block in the destination frame by shifting the current block over a predefined neighbourhood of pixels in the destination frame. At each shift, mean squared distances between the gray values of the two blocks are computed. The distance is calculated as follows:

$$D(A,B) = \frac{1}{n}\sum_{i=1}^{n}(A(i) - B(i))^2 \tag{1}$$

where D is the distance, A is the current block, B is the block the reference frame and n is the total number of pixels in the block. The shift which gives the smallest distance is considered as the best match. In order to reduce the computational burden the motion vector of the current block is predicted from that of the neighbour blocks in the temporal or spatial direction. Since the computational complexity is much lower than the optical flow equation and the pel-recursive methods, block matching has been widely adopted as a standard for video coding and hence it provides a good starting point.

## 3. MAXIMUM LIKELIHOOD FRAMEWORK

The 2-D motion vectors for the extracted motion blocks are characterised using a matrix of pairwise similarity weights. Suppose that $\hat{n}_\alpha$ and $\hat{n}_\beta$ are the unit motion vectors for the pixel blocks indexed α and β. The elements of this weight matrix $W_{\alpha,\beta}$ are given by:

$$W_{\alpha,\beta} = \begin{cases} \frac{1}{2}(1 + \hat{n}_\alpha . \hat{n}_\beta) & if \quad \alpha \neq \beta \\ 0 & otherwise \end{cases} \tag{2}$$

The problem of grouping motion blocks into coherent moving objects is treated as that of finding pairwise clusters. The aim in pairwise clustering is to locate the updated set of similarity weights which partition the image into regions of uniform motion. To be more formal, let $V$ denote the index-set of the detected motion blocks in the image and suppose that Ù is the set of pairwise-clusters, i.e. distinct moving objects, to which these blocks are to be assigned. The initial set of clusters are defined by the eigenmodes of the link-weight matrix $W^{(0)}$. Here we follow Sarkar and Boyer[4] which has shown how the positive eigenvectors of the matrix of link-weights can be used to assign objects to perceptual clusters. Using the Rayleigh-Ritz theorem, they observe that the scalar quantity $\underline{x}^T W^{(0)} \underline{x}$ is maximised when $\underline{x}$ is the leading eigenvector of $W^{(0)}$. Moreover, each of the subdominant eigenvectors corresponds to a disjoint pairwise cluster. They confine their attention to the samesign positive eigenvectors (i.e. those whose corresponding eigenvalues are real and positive, and whose components are either all positive or are all negative in sign). If a component of a positive same-sign eigenvector is nonzero, then the corresponding object belongs to the associated cluster of motion blocks. The eigenvalues $\lambda_i$ of $W^{(0)}$ are the solutions of the equation $|W^{(0)} - \lambda I| = 0$ where I is the identity matrix. The corresponding eigenvectors $\underline{x}_{\lambda 1}, \underline{x}_{\lambda 2}, \ldots$ are found by solving the equation $W^{(0)}\underline{x}_\omega = \lambda_\omega \underline{x}_\omega$. Let the set of positive same-sign eigenvectors be represented by $\Omega = \{\omega \mid \lambda_\omega > 0 \wedge [(\underline{x}^*_\omega(i) > 0 \forall i) \vee (\underline{x}^*_\omega(i) < 0 \forall i)]\}$ where $\underline{x}^*_\omega(i)$ is the ith component of the eigenvector indexed ù.

Kelly and Hancock[7] commence a simple model of the cluster formation process based on a series of independent Bernoulli trials. The linkage of each pair of nodes within a cluster is treated as a separate Bernoulli trial. The link-weight for the pair of nodes is treated as the success probability of the trial. The similarity weight $W_{\alpha,\beta}$ is taken as the parameter of the Bernoulli distribution. The probability that the block association are correct is $W_{\alpha,\beta}$ while the probability that it is in error is $1-W_{\alpha,\beta}$. Here they introduce a cluster membership indicator $s_{\alpha\omega}$ which represents the degree of affinity of the object indexed α to the cluster with index ω. The random variable associated with the trial is taken as the product of these cluster indicators for the pair of nodes, i.e. $s_{\alpha\omega}.s_{\beta\omega}$; this indicates whether the two nodes belong to the same cluster. This is unity if both block belong to the same object or cluster and is zero otherwise. Using the property Bernoulli distribution becomes:

$$P(s_{\alpha\omega}, s_{\beta\omega} \mid W_{\alpha,\beta}) = W_{\alpha,\beta}^{s_{\alpha\omega}s_{\beta\omega}}(1 - W_{\alpha,\beta})^{(1 - s_{\alpha\omega}s_{\beta\omega})} \tag{3}$$

This distribution takes on its largest values when either the motion vector similarity weight $W_{\alpha,\beta}$ is unity and $s_{\alpha\omega} = s_{\beta\omega} = 1$ or $W_{\alpha,\beta} = 0$ and $s_{\alpha\omega} = s_{\beta\omega} = 0$.

Using this model a joint likelihood function is developed for the link-weights and the cluster membership indicators. This likelihood function can be used to make both a maximum likelihood re-estimate of the link-weight matrix and a maximum a posteriori probability estimate of the cluster membership indicators. In the case of re-estimating the link-weight matrix, the cluster indicators are treated as data. Applying this model to log-likelihood function for the observed set of motion vector similarity weight, we get

$$L = \sum_{\omega \in \Omega} \sum_{(\alpha,\beta) \in \Phi} \{ s_{\alpha\omega}s_{\beta\omega} \ln(W_{\alpha,\beta}) + (1 - s_{\alpha\omega}s_{\beta\omega}) \ln(1 - W_{\alpha,\beta}) \} \tag{4}$$

The above log-likelihood function can be optimised using EM like process. To maximise the log-likelihood function with respect to the link-weights and the cluster membership indicators we take the derivatives of the expected log-likelihood function with respect to the elements of the link-weight matrix and cluster membership variables. In the E step, the cluster membership probabilities are updated according to the formula

$$s_{\alpha\omega}^{(n+1)} = \frac{\prod_{\beta \in V} \left\{ \frac{W_{\alpha,\beta}^{(n)}}{1 - W_{\alpha,\beta}^{(n)}} \right\}^{s_{\beta\omega}^{(n)}}}{\sum_{\omega \in \Omega} \prod_{\beta \in V} \left\{ \frac{W_{\alpha,\beta}^{(n)}}{1 - W_{\alpha,\beta}^{(n)}} \right\}^{s_{\beta\omega}^{(n)}}} \tag{5}$$

Once the revised cluster membership variables are to the hand, then the M step of the algorithm is applied to update the similarity weight matrix. The updated weights are given by

$$W_{\alpha,\beta}^{(n+1)} = \sum_{\omega\in\Omega} s_{\alpha\omega}^{(n)} s_{\beta\omega}^{(n)} \qquad (6)$$

These two steps are interleaved and iterated to convergence.

## 4. PROPOSED METHOD

The drawback of the block matching scheme is that while the high-resolution field of motion vectors obtained with small block sizes capture fine detail, it is susceptible to noise. At low resolution, i.e., for large block sizes, the field of motion vector is less noisy but the fine structure is lost. Also it will not detect the moving objects with slow motion. In order to remove this drawback a new algorithm for detecting the moving object using maximum likelihood framework is proposed.

The proposed algorithm has two steps: First, motion detection and motion estimation, and second, clustering. Instead of performing BMA to the entire frame, first, the moving objects are detected by taking the difference of maximum intensity values and minimum intensity values for each pixels of a set of frames. Then find the motion vectors of these regions.

### 4.1 Motion Detection

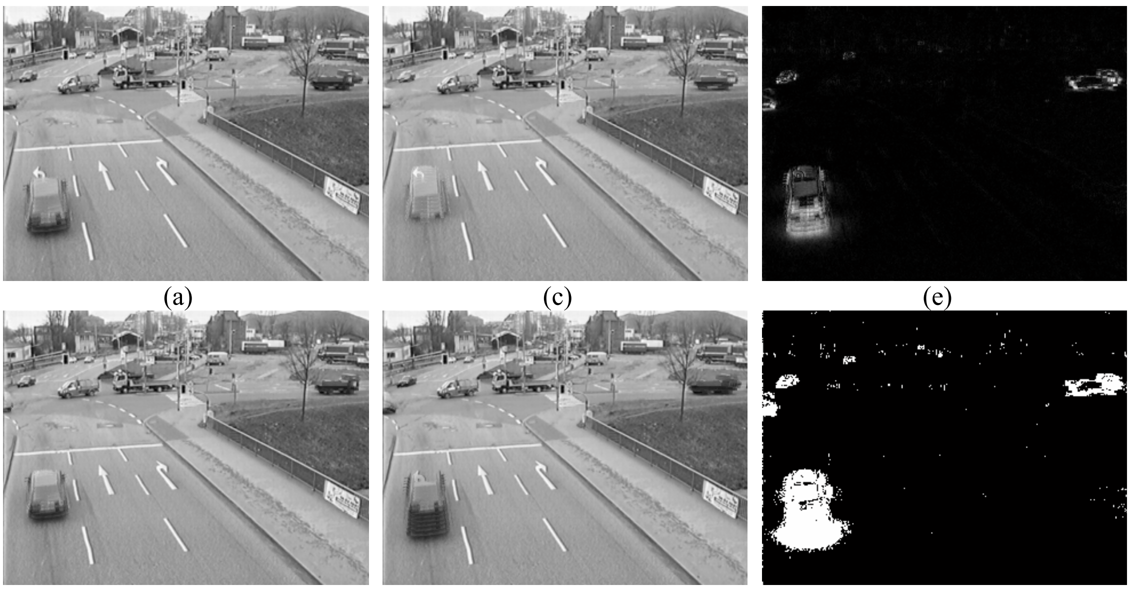
Consider a set of consecutive frames from the static camera video. The range of the pixel values that a particular location ($x$,$y$) can vary significantly if that pixel belongs to moving object. If it belongs to background then the range of the pixel value at ($x$,$y$) in the consecutive set of frames will be small. So this technique can be used to detect the motion in a set of frames.

Let a frame ***t*** in a video sequence be represented as $f(t)$. A pixel at location $(x, y)$ in the frame $t$ be represented as $f(x, y, t)$. Let $MAXP(x, y)$ represents the maximum intensity value of a pixel at location $(x, y)$ in a set of frames $S = \{f(t+i) : i = 0,\ldots, dt\}$ where $t$ is current frame and $dt$ is the number of frames in the set $S$. Similarly $MINP(x, y)$ represents the minimum intensity value at location $(x, y)$ in S. Let the difference between them is denoted as:

$$d(x, y) = MAXP(x, y) - MINP(x, y) \qquad (7)$$

If the object at location, $(x, y)$, is not moving, then $d(x, y)$ will have a very small value. If the object is moving then $d(x, y)$ will greater. Figures 1(a)-(b) shows original frame 1 and 5 of the traffic sequence. MAXP and MINP for the set of frames 1 to 5 is shown in Figs 1(c)-(d). The difference between the MAXP and MINP is shown in Figs 1(e) and 1(f) shows the image after thresholding.

The problem with directly applying the threshold to the difference image is that some noise will be there in the resulting image. If the object is moving very slowly then it is difficult to identify whether the detected pixel is a noise or an object. So if we apply morphological operation to the thresholded image the slow moving object will also disappear. In order to solve this problem we take gradient of difference image along the horizontal and vertical directions. Take the Euclidean distance of the gradient images. Apply threshold to the gradient image. The threshold selected is standard deviation of the gradient image. Modified morphological dilation and erosion operations are then performed on the binary image. For dilation, four connected neighbouring pixels are set to 1 if the current pixel is 1;



(a) (c) (e)

(b) (d) (f)

**Figure 1. (a) original frame 1; (b) frame 5; (c) MAXP image for the set of frames 1 to 5; (d) MINP image; (e) difference image, d; and (f) image after applying the threshold T=10 to the difference image.**

where 1 indicates that the pixel belongs to object mask. For erosion, four connected neighbouring pixels are set to 0; where 0 indicates that the pixel belongs to background. The modified dilation and erosion operations are defined below.

For dilation, if $g(x,y)=1$, then

$$\begin{aligned} Pixel(x,y) &= Pixel(x+1,y) = Pixel(x,y+1) \\ &= Pixel(x-1,y) = Pixel(x,y-1) = 1 \end{aligned} \quad (8)$$

For erosion, if $g(x,y)=0$, then

$$\begin{aligned} Pixel(x,y) &= Pixel(x+1,y) = Pixel(x,y+1) \\ &= Pixel(x-1,y) = Pixel(x,y-1) = 0 \end{aligned} \quad (9)$$

The operations are performed in the order of erosion followed by dilation. When erosion is performed isolated noisy pixels will be removed, and then when dilation is performed the object pixels will be highlighted. Figure 2(a) shows the gradient image of the difference image shown in Fig. 1(e). Figure 2(b) shows the motion detected binary image after applying the morphological operation. Figure 2(c) corresponds to motion detected image

The parameter that affects the motion detection is the number of frames in the set. If the number of frames used in the set is less then small motions will not be detected. If the set contains large number of frames then noise will also detected as motion. Figure 3(c) shows motion detection with five frames in the set. Figure 4(c) shows the motion detected images with ten frames in the set. The difference between the two figures is shown in Figure 4(c) and it exhibits more noise. This is due to the fact that , small noises in the frame may appear as moving when the number of frames is large.

### 4.2 Motion Estimation

After detecting the motion pixels, motion is to be estimated for these pixels. For estimating the motion, the block matching algorithm discussed in section II is used.

(a) (b) (c)

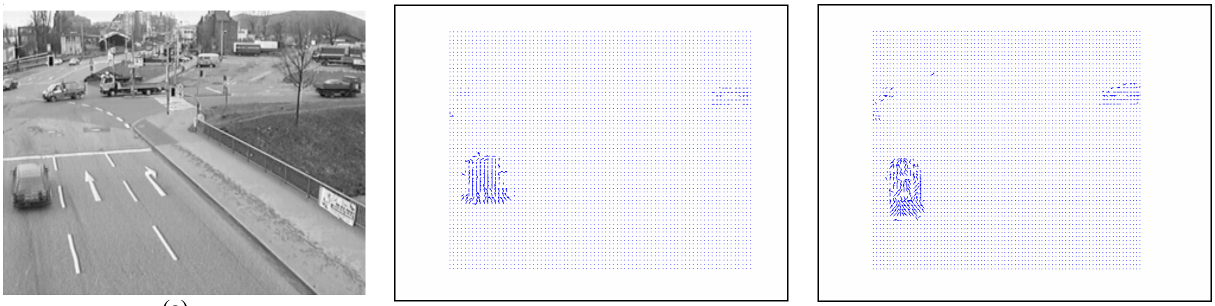
**Figure 2. (a) Gradient image of the difference image shown in figure 1(e); (b) Binary image after applying the modified morphological operations; (d) corresponding motion detected image.**

(a) (b) (c)

**Figure 3. Motion detection with five frames in the set *S*. (a) first frame of the set, frame 26; (b) last frame of the set, frame 30; (c) motion detected image.**

(a) (b) (c)

**Figure 4. Motion detection with ten frames in the set *S*. (a) first frame of the set, frame 20; (b) last frame of the set, frame 30; (c) motion detected image.**



(a) (b) (c)

**Figure 5. (a) Frame 10 of the traffic seqence; (b) motion map with original frames; and (c) motion map with motion detected image.**

The only difference is that instead of applying the algorithm to the whole frame, it is applied only to the motion regions, thereby reducing the computational complexity to great extent. Figure 5(b) shows the motion map obtained by passing the original frames to BMA. Figure 5(c) shows the motion map obtained by passing the motion detected image to BMA. In both the images the block size is 4×4 and the threshold used is 5.

### 4.3 Clustering using Spectral Framework

The spectral framework described by Robles-Kelly & Hancock[7] is used in this step. The 2-D velocity vectors for the extracted motion blocks are then represented using a matrix of pairwise- similarity weights $W$. The link weight matrix is calculated using the Eqn (2). The clustering is done by maximizing the log likelihood function described in the section III with respect to weight matrix and the cluster membership indicator. This is done using method that closely resembles EM. In the E step, cluster membership probabilities are updated and in the M step, similarity weight matrix is updated.

The same sign eigenvectors are extracted from the current link-weight matrix $W$. These are then used to compute the cluster-membership matrix $S$ using the equation

$$\hat{s}_{i\omega} = \frac{\left|\underline{x}^*_\omega(i)\right|}{\sum_{i\in V}\left|\underline{x}^*_\omega(i)\right|} \tag{10}$$

The number of same sign eigenvectors determines the number of clusters for the current iteration. Using the cluster-membership matrix $S$, link-weight matrix $W$, is updated. This is done as follows. For each cluster, compute the link-weight matrix $\hat{W}_\omega = s_\omega s_\omega^T$. Perform an eigen-decomposition on each cluster link-weight matrix to extract the non-zero eigenvalue $\lambda^*_\omega$ and the corresponding eigenvector $\phi^*_\omega$. Since the matrix $\hat{W}_\omega$ is rank one since it is defined as the product of two vectors, the computation of the first eigenvector can be regarded for computational purposes as a normalisation of the vectors $s_\omega$.

In practice, the link-weight matrix may be noisy and hence the cluster structure may be subject to error. In an attempt to overcome this problem, the updated link-weight matrix is refined with a view to improve its block structure. The aim here is to suppress structure which is not associated with the principal modes of the matrix. This is done by applying the following equation

$$W^* = \sum \frac{\lambda^*_\omega}{|\Omega|} \phi^*_\omega (\phi^*_\omega)^T \tag{11}$$

The link-weight matrix is then used to update cluster-membership matrix. An updated matrix of cluster membership variable $\hat{S}$ is computed by applying the following equation to the revised link-weight matrix $W^*$.

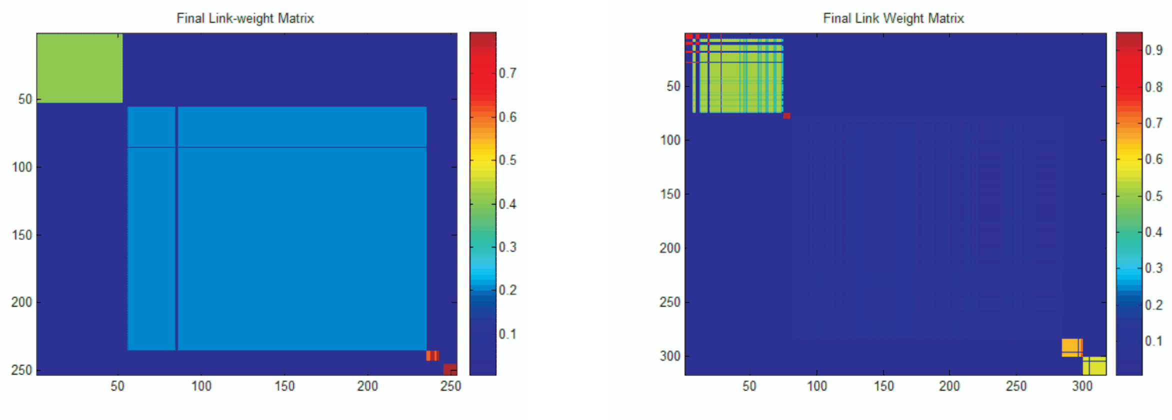
$$\hat{s}_{i\omega} = \frac{\prod_{j\in V}\left\{W_{ij}/(1-W_{ij})\right\}^{s_{j\omega}}}{\sum_{i\in V}\prod_{j\in V}\left\{W_{ij}/(1-W_{ij})\right\}^{s_{j\omega}}} \tag{12}$$

The updated cluster membership matrix $\hat{S}$ is used to compute the updated link-weight matrix $\hat{W} = \frac{1}{|\Omega|} SS^T$. Once the updated link weight matrix is in hand, it is then again used to compute same sign eigenvectors and the whole process is repeated till convergence.

## 5. RESULTS AND DISCUSSION

The algorithm is tested using video sequences with known ground truth. Figure 6(a) shows the final link weight matrix obtained for the 40th frame with the original method. Here four block structures are visible. Each block structure represents the one cluster. Figure 6(b) shows the final link weight matrix obtained for the 40th frame with the improved method. Here six block structures are visible.

Figure 7 shows the result of the motion segmentation algorithm with the traffic sequence. First row shows the original frames 10, 20 and 40. Second row shows the ground truth of these frames. Third row shows the motion map obtained with the original method. Fourth row shows the result of the motion segmentation algorithm using the original method. Here for the 10th frame three motion objects are detected, for 20th frame five motion objects are detected and for 40th frame four motion objects are detected. Fifth row shows the output of the motion detection step of the improved algorithm. Sixth row shows the motion map obtained with improved algorithm. Motion map is obtained using the motion detected images. Seventh row shows the result

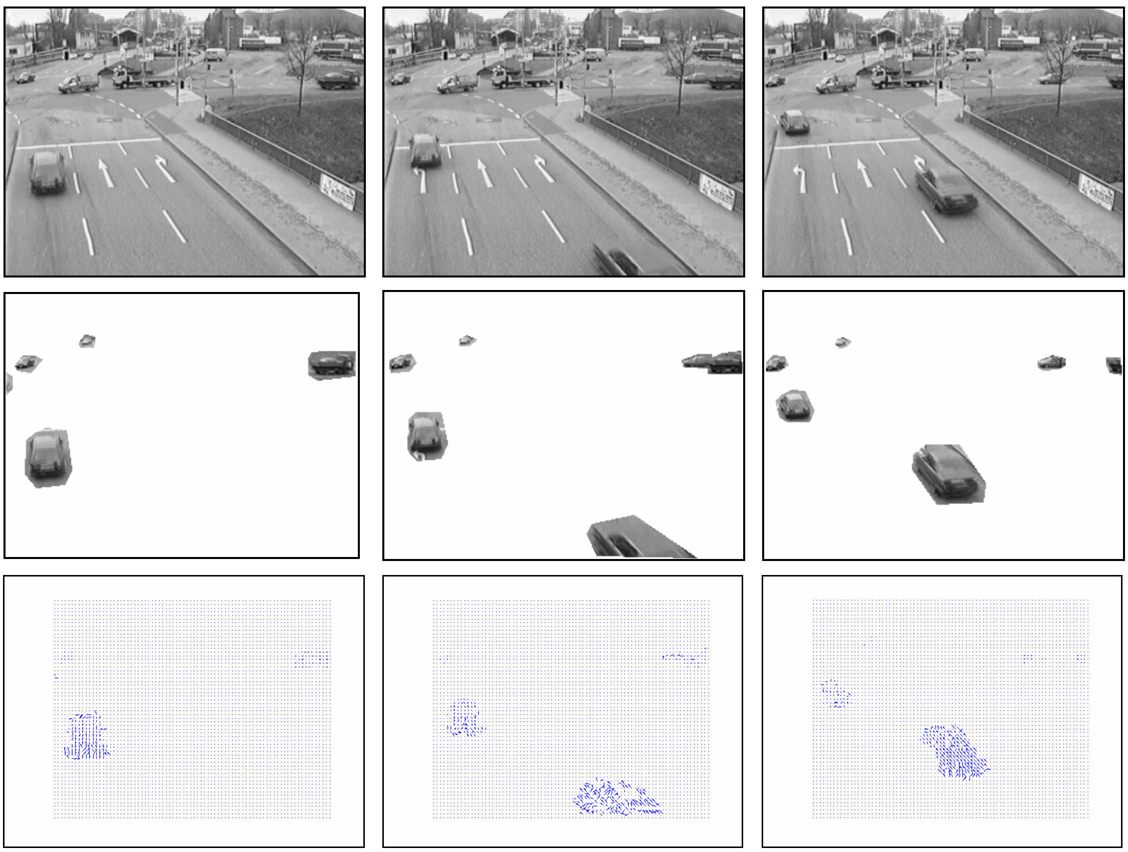



(a) (b)

**Figure 6. (a) Final link weight matrix obtained after applying the original method; and (b) final link weight matrix obtained after applying the improved motion segmentation algorithm.**

of the motion segmentation algorithm using the improved method. Here for the 10th frame five motion objects are detected, for 20th frame six motion objects are detected and for 40th frame six motion objects are detected. Here the number of frames used in the motion detection step is five. The block size used is 4×4. The algorithm converged in an average of 3 iterations.

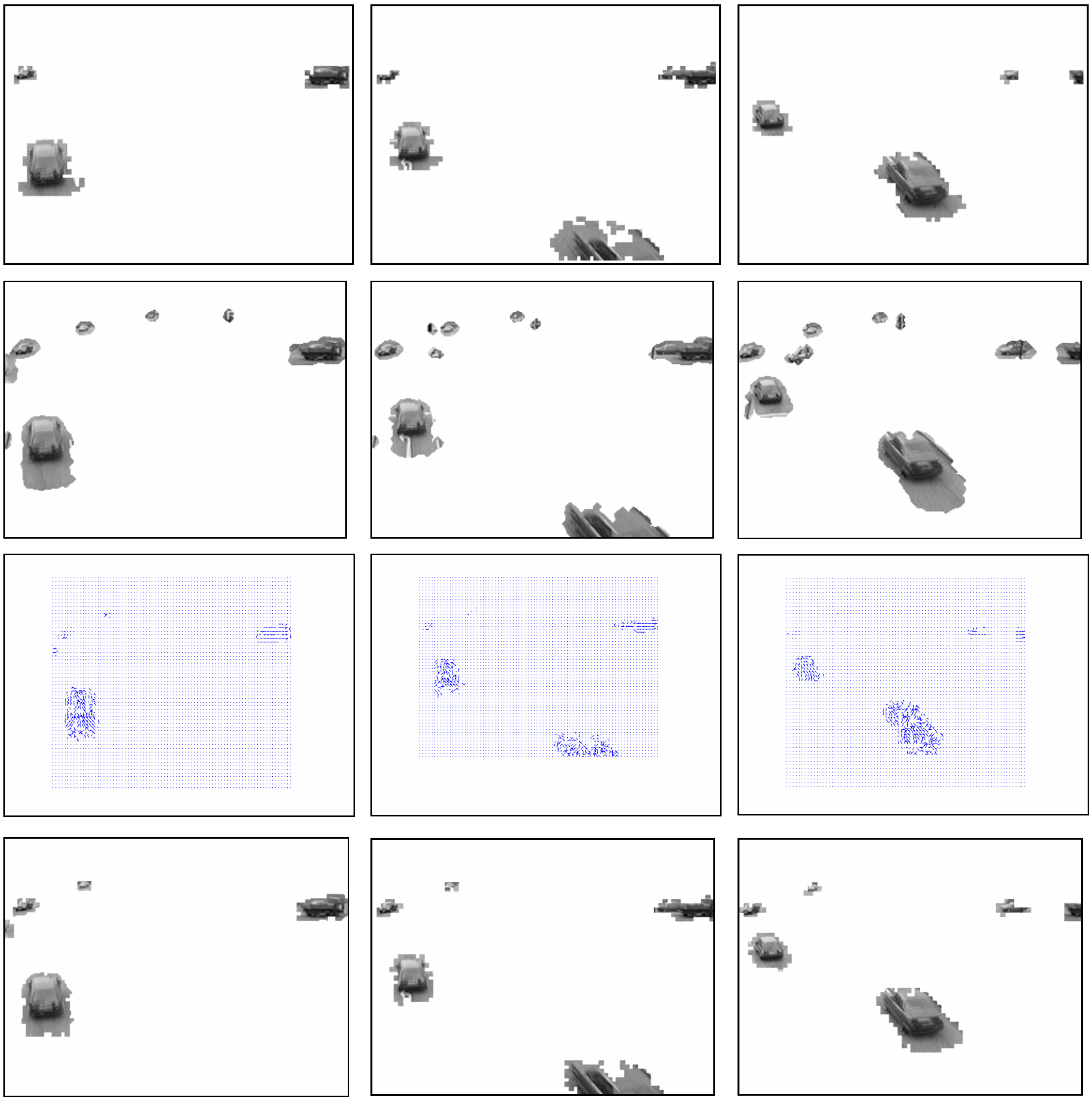
Figure 8 shows the result of the motion segmentation algorithm with the taxi sequence. First row shows the original frames 10, 20 and 30. Second row shows the ground truth of these frames. For all the three frames the number of moving objects is four. The four objects are: the left car, the middle car, the right vehicle and the pedestrian. Third row shows the motion map obtained with the original method. Fourth row shows the result of the motion segmentation algorithm using the original method. Here for the 10th frame four motion objects are detected, for 20th frame three motion objects are detected and for 30th frame three motion objects are detected. Fifth row shows the motion map obtained with improved algorithm. Motion map is obtained using the motion detected images. Sixth row shows the result of the motion segmentation algorithm



(a)

(b)

(c)



(d)

(e)

(f)

(g)

**Figure 7. (a) Original frames (10, 20 and 40) of the traffic sequence; (b) Ground truth of the corresponding frames; (c) corresponding motion map with the original method; (d) Result of the motion segmentation algorithm with the original method. (e) Motion detected images using the improved algorithm. (f) Motion map detected using the improved algorithm; and (g) Result of the improved motion segmentation algorithm.**

using the improved method. Here for all the three frames motion objects detected is four.

The complexity of the proposed algorithm is reduced because of the motion detection step. This step will detect the motion and only the detected motion pixels are further passed to the next step. In this way the noise will be eliminated as well as the number of pixels passed to the next step is also decreased compared to the original method. The motion detection step has the complexity of *O(n)* where n is the number of pixels in a frame.

Table 1 shows the quantitative analysis of the result. The table lists the number of objects detected, the percentage of correctly classified pixel in the total number of object pixels in the ground truth (true positive rate), percentage of pixels wrongly detected

**Table 1. Performance numbers**

| | Traffic sequence | | | Taxi sequence | | |
|---|---|---|---|---|---|---|
| **Frame number** | **10** | **20** | **40** | **10** | **20** | **30** |
| No of Moving objects in the ground truth | 5 | 6 | 6 | 4 | 4 | 4 |
| **Original method** | | | | | | |
| No of objects detected | 3 | 5 | 4 | 4 | 3 | 3 |
| True positive rate | 78.25 | 78.62 | 80.32 | 67.25 | 50.14 | 50.93 |
| False negative rate | 0.6 | 1.66 | 1.27 | 1.33 | 1.20 | 1.63 |
| Percentage of correct classification | 98.40 | 97.08 | 97.71 | 95.64 | 91.87 | 91.33 |
| **Improved method** | | | | | | |
| No of objects detected | 5 | 6 | 6 | 4 | 4 | 4 |
| True positive rate | 90.09 | 81.49 | 89.42 | 78.59 | 63.4 | 64.88 |
| False negative rate | 0.95 | 1.09 | 1.90 | 1.18 | 1.23 | 2.30 |
| Percentage of correct classification | 98.64 | 97.79 | 97.62 | 96.88 | 93.73 | 92.79 |

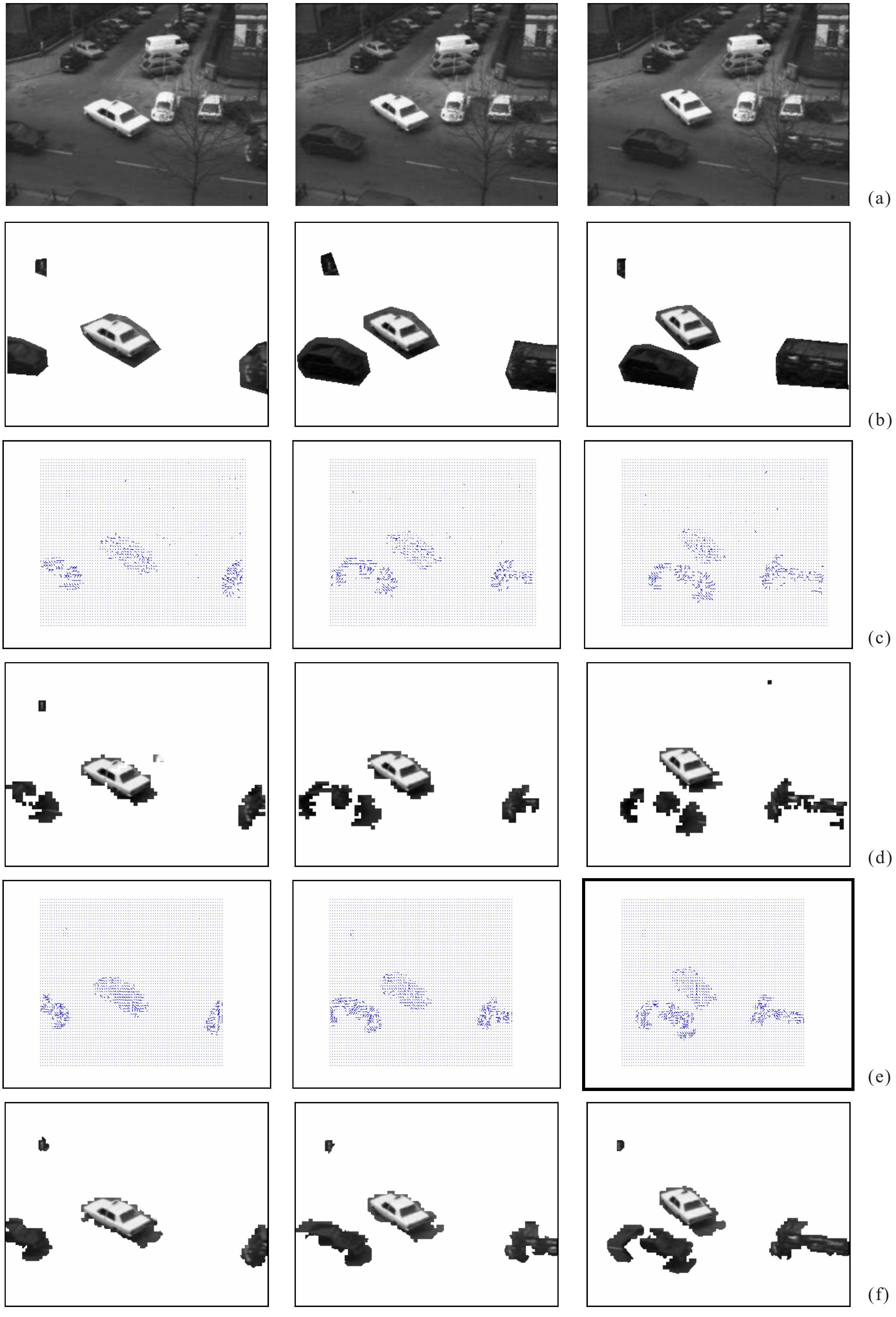


**Figure 8. (a) Original frames (10, 20 and 30) of the taxi sequence; (b) Ground truth of the corresponding frames; (c) corresponding motion map with the original method; (d) Result of the motion segmentation algorithm with the original method. (e) Motion map detected using the improved algorithm; and (f) Result of the improved motion segmentation algorithm.**

as object pixels to the total number of pixels in the background in the ground truth (false negative rate) and the total number of pixels classified correctly in the whole image (percentage of correct classification) comparison between the two algorithms. The percentage of correct classification is almost same using both methods but the true positive rate is much higher in the improved method compared to the original method. And also the number of moving objects detected is greater in the improved method compared to the original method.

## 6. CONCLUSIONS

An improved iterative spectral framework using maximum likelihood for motion segmentation is presented. The proposed algorithm first detects the motion using a set of frames and uses the motion detected image for computing the motion vector. The advantage of this step is that it can detect moving objects with very slow motion and also reduces the time complexity in computing the motion vector.

Using this motion vector the pairwise similarity matrix of motion blocks was computed. This matrix to compute the cluster membership probabilities. Under the assumption that similarity weights follow a Bernoulli distribution, a log likelihood function was used to update the similarity matrix and the cluster membership probabilities.

## ACKNOWLEDGEMENT

The work was carried out when the authors were with University of Kerala, Trivandrum, India.

## REFERENCES

1. Jepson, A.D.; MacLean, W.J. & Frecker, R.C. Recovery of ego-motion and segmentation of independent object motion using the EM Algorithm. *In* Proceedings of the British Machine Vision Conference, 175-84, 1994.
2. Adelson, E.H. & Weiss, Y. A unified mixture framework for motion segmentation: Incorporating spatial coherence. *In* Proceedings of the IEEE Computer Vision and Pattern Recognition, 1996. pp. 321-26.
3. Shi, J. & Malik, J. Normalized cuts and image segmentations. *In* Proceedings of the IEEE Computer Vision and Pattern Recognition, 1997. pp.731-37.
4. Sarkar, S. & Boyer, K.L. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Comp. Vision Image Understanding*, 1998, **71**(1), 110-36.
5. Perona, P. & Freeman, W.T. Factorization approach to grouping. *In* Proceedings of the ECCV, 1998. pp. 655-70.
6. Chung, Fan R.K. Spectral graph theory. American Mathematical Society, 1997.
7. Robles-Kelly, A. & Hancock, E.R. A probabilistic spectral framework for grouping and segmentation. *Pattern Recognition,* 2004, **37**(7), 1387-405.
8. Robles-Kelly, Antonio & Hancock, Edwin R. Maximum likelihood motion segmentation using eigendecomposition. *In* Proceedings of the IEEE Image Analysis and Processing, 2001. pp. 63-68.
9. Shyn, J.S.; Hsieh, C.H.; Lu, P.C. & Lu, E.H. Motion estimation algorithm using inter-block correlation. *IEE Electron. Lett.,* 1990, **26**(5), 276–77.