Crowd Flow Segmentation based on Motion Vectors in H.264 Compressed Domain

R. Gnana Praveen
Video Analytics Laboratory
Supercomputer Education and Research Centre
Indian Institute of Science
Banglore - 560 012, India.
praveengnan.24@gmail.com

R. Venkatesh Babu
Video Analytics Laboratory
Supercomputer Education and Research Centre
Indian Institute of Science
Banglore - 560 012, India.
venky@serc.iisc.ernet.in

***Abstract*—In this work, we have explored the prospect of segmenting crowd flow in H.264 compressed videos by merely using motion vectors. The motion vectors are extracted by partially decoding the corresponding video sequence in the H.264 compressed domain. The region of interest ie., crowd flow region is extracted and the motion vectors that spans the region of interest is preprocessed and a collective representation of the motion vectors for the entire video is obtained. The obtained motion vectors for the corresponding video is then clustered by using EM algorithm. Finally, the clusters which converges to a single flow are merged together based on the bhattacharya distance measure between the histogram of the of the orientation of the motion vectors at the boundaries of the clusters. We had implemented our proposed approach on the complex crowd flow dataset provided by [1] and compared our results by using Jaccard measure. Since we are performing crowd flow segmentation in the compressed domain using only motion vectors, our proposed approach performs much faster compared to other pixel domain counterparts still retaining better accuracy.**

***Index Terms*—Crowd Flow, Segmentation, Motion Vector Clustering, EM algorithm, H.264 compressed domain, k-means clustering.**

I. Introduction

In the recent years, computer vision algorithms have drawn much attention for the application of video surveillance systems such as detecting anomalous behaviors, instabilities, monitoring, etc. However, these conventional approaches for surveillance systems fails in the case of high density moving objects in the videos due to its complex dynamics. But many real time scenarios involve crowd monitoring for video surveillance applications. So we cannot directly use these traditional approaches for surveillance systems in real time applications. As the density of the moving objects increases, the performance of these conventional approaches gets deteriorated in a rampant way. Hence, there is an imperative and immense need to model and analyze high density crowded videos such as those shown in Fig. 1 in order to avoid catastrophic events such as stampedes, etc. Since the dynamics of the crowded scenes is very complex, modeling and analyzing the crowded scenes pose a significant challenge which drew the attention of several researchers. Several researchers have perceived this problem of crowd modeling from various perspectives. However, since most of the digital data is stored in compressed domain, it will be a tedious process to entirely decode the digital data from the compressed domain. In order to reduce the complexity of the system as well as to increase the speed of the system there is a pivotal need to perceive this problem in compressed domain. Most of the existing conventional approaches for crowd modeling was done in the pixel domain. Our research effort endeavors to tackle this problem of crowd modeling in H.264 compressed domain as it has found a wide number of applications due to its high resolution, low bandwidth usage, reduced storage requirements, faster frame rates and better video quality. In this paper, we have perceived the problem of crowd modeling as crowd flow segmentation. Even though the speed of the system in the compressed domain was expected to be much faster compared to the pixel domain, developing such systems for various applications pose a significant challenge as the information available in the compressed domain is much lower compared to the pixel domain. Motion Vectors available in the compressed domain conveys significant information about the crowd flow. Moreover, as most of the conventional approaches have dealt this problem by considering optical flow, we have tackled this problem of crowd flow segmentation by using only motion vectors which is a coarse approximation of the optical flow.

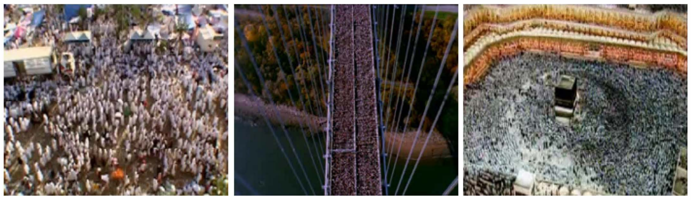

Fig. 1. Example scenarios of crowded scenes

The remainder of the paper is organized as follows. We summarize the related work done for crowd analysis in Section. II. The proposed approach is discussed in detail in Section. III. Section. IV deals with the results and discussions of the proposed approach which is implemented on the database provided by [1]. Finally the conclusion and the scope of future work is outlined in Section. V.

II. Related Work

Research in crowd analysis has drawn much attention of several vision researchers in the past few years. Most of the work in crowd analysis is focussed on crowd detection, tracking of individuals, detection of anomalous behavior, etc. The surveys in [2] and [3] provide a review of computer vision approaches in order to tackle the above problems in crowd analysis. Wu et al. [4] had proposed a method for crowd flow partitioning by considering this problem as a problem of scattered motion field segmentation by assuming the local crowd motion as translational motion field. They had implemented their approach on real life scenes and shown better results in segmenting homogeneous regions in the crowded scenes. Another work was done by Li et al. [5] which extracts the dynamic region of the crowded scenes and the crowd segmentation is performed based on the histogram curve of the angle information of the foreground velocity field. Kuhn et al. [6] had developed a frame work for extracting the motion patterns by combining the optical flow from image processing with lagrangian analysis of time dependent vector fields. They had shown its applicability for crowd analysis like automated detection of abnormal events in the video sequences.

Ali et al. [1] had perceived the problem of crowd flow segmentation from the perspective of fluid dynamics. They had performed the crowd flow segmentation by laying a grid of particles on the video and tracking the trajectories of these particles over the frames of the video. They had also extended their approach for the detection of instabilities such as anomalous behaviors in the crowded scenes. Many researchers have explored this problem in the pixel domain. However, we had explored the prospect of performing crowd flow segmentation in H.264 compressed domain for the first time to the best of our knowledge as it performs much faster compared to pixel domain. In this work, we had performed crowd flow segmentation in H.264 compressed domain by only using motion vectors and benchmarked our approach on the dataset provided by [1].

III. Proposed Approach

In the recent years, H.264 has drawn much importance due to its wide applicability and several algorithms have been developed for a number of real time applications. Our proposed approach uses only motion vectors available in the H.264 compressed domain and mainly has three major steps. 1) Preprocessing of motion vectors 2) Clustering of motion vectors by EM algorithm 3) Merging of coherently moving motion vectors. The overall block diagram of the proposed approach is shown in Fig. 2.

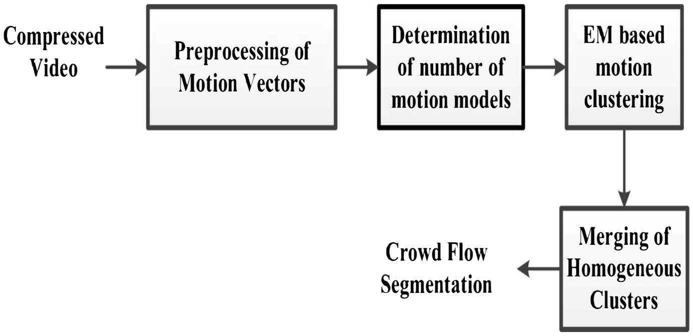
Compressed Video
Preprocessing of Motion Vectors
Determination of number of motion models
EM based motion clustering
Merging of Homogeneous Clusters
Crowd Flow Segmentation

Fig. 2. Overall Block Diagram of the Proposed Approach

A. Preprocessing of Motion Vectors

The first and foremost step for crowd flow segmentation is to extract the dynamic region of the crowded scenes from the background which is nothing but the region spanning the crowd flow in the scene. This plays a crucial role for accurate crowd flow segmentation, where the motion vectors in the static region of the crowded scene is discarded and the motion vectors that represents dynamic region of the scene is retained. In our approach, this is done by using motion vectors as it conveys significant information regarding the crowd flow. Normally, these motion vectors extracted by partially decoding the video sequence in the compressed domain will be very noisy. In order to eliminate the noisy motion vectors that shoots out in the static region of the scene, a motion mask is created by considering the magnitude of the motion vectors. The magnitude of the motion vector of a sub macro block in frame t and location (i, j) can be obtained as in Eqn. 1.

$$M_t(i,j) = \sqrt{u(i,j)^2 + v(i,j)^2} \tag{1}$$

where, u and v represents the horizontal and vertical components of the motion vector of the sub- macroblock at location (i, j). The number of nonzero motion vectors of each sub-macroblock are accumulated temporally over all the frames. Let X be the resultant matrix where, each element represents the number of nonzero motion vectors at the corresponding location of the sub-macroblock over time. Now a motion mask is obtained by discarding the elements in the resultant matrix X which is less than that of the threshold τ, where τ is obtained by taking the average of the elements of the resultant matrix X as shown in Eqn. 2.

$$\tau = \frac{1}{M.N} \sum_{i,j} X(i,j) \tag{2}$$

where, M and N represents the number of rows and columns respectively in X. This is done by forming a binary mask R where 1 represents the dynamic region i.e., crowded portion of the region and 0 represents the static region of the video as shown in Eqn. 3.

$$R\left(i,j\right) = \begin{cases} 0 & \text{if } X\left(i,j\right) < \tau \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

where, (i, j) represents the location of the sub-macroblock. In order to retain as many significant motion vectors as possible in the dynamic region of the video, the obtained motion mask is still refined by extending the edges and neglecting the noisy motion vectors outside the region of interest by performing two subsequent morphological operations - closing and opening. The motion vectors that lie in the motion mask is considered for collective representation of motion vectors for the video. The horizontal and vertical components are obtained by taking median along the respective components. This collective representation of the motion vectors in the dynamic region of the crowded scene is considered for further processing.

B. Clustering of Motion Vectors by EM algorithm

Once the collective representation of motion vectors for the entire video is obtained, the motion vectors are clustered using EM algorithm. EM algorithm is widely used for many estimation problems in statistics. Since it has also been used for motion segmentation [7], [8] we have explored its applicability for crowd flow segmentation by clustering the motion vectors. Given the number of motion models and the initial motion hypothesis in terms of model parameters, EM algorithm iterates alternatively between two simple independent stages (E and M steps) until convergence to estimate the model parameters and cluster the data. In our approach, we have assumed the motion flow to be translational model which requires only two components for each model.

The E Step Given the number of motion models K and the corresponding initial motion hypothesis expressed in terms of translational parameter vectors $\{a_1, a_2, a_3, ...a_k\}$, this step estimates the probabilities of each sub-macroblock to be associated with a motion model. In this step, the initial parameters of the translational model are given as the cluster centers obtained by k-means clustering for faster convergence and the number of motion models is also obtained from k-means clustering as described in Section. III-C. Let $v_x(i,j)$ and $v_y(i,j)$ be the horizontal and vertical components of the motion vector respectively at location (i,j). Each translational model is characterized by the corresponding model parameter vector a_k which has two components as shown in Eqn. 4

$$a_k{}^T = [u_k \quad v_k] \tag{4}$$

Let $p = [i \quad j]^T$ be the vector representing the position of the sub-macroblock in the image plane. Now the squared residual $R_k{}^2$ is obtained for each sub-macroblock at the position p for all the motion models as shown in Eqn. 5

$$R_k{}^2(p) = (a_k - v(p))^2 \tag{5}$$

This residual error signifies the error between the actual motion $v(p)$ and predicted motion a_k at the location p. Now the probability for the sub-macroblock to be associated with k^{th} model are obtained from the squared residual $R_k{}^2$. Let $L_k(p)$ be the probability for the motion vector at the location p to be associated with k^{th} class which is derived from Bayes rule as shown in Eqn. 6

$$L_k(p) = \frac{\exp(-R_k{}^2(p)/2\sigma^2)}{\sum\limits_{j=1}^{k} \exp(-R_j{}^2(p)/2\sigma^2)} \tag{6}$$

where, free parameter σ^2 controls the fidelity of the affine model fit to the dense motion vectors. Then the data point is associated to the motion model which has the maximum probability among all the motion models. Once these weights are obtained, they are given to the M step.

The M Step In this step, the obtained weights in the previous step are used to estimate the model parameters, which is done by weighted least square estimation. Now given the classification of the data points associated with the motion models, the model parameters are estimated. The estimated motion parameters a_k of class k is a solution to the following Eqn. 7.

$$M_k a_k = B_k \tag{7}$$

where

$$M_k = \begin{pmatrix} \sum_{p \in Z} w_k(p) & 0 \\ 0 & \sum_{p \in Z} w_k(p) \end{pmatrix} \tag{8}$$

and

$$B_k = \begin{pmatrix} \sum_{p \in Z} w_k(p) v_x(p) \\ \sum_{p \in Z} w_k(p) v_y(p) \end{pmatrix} \tag{9}$$

where Z represents the entire image plane in the compressed domain. The estimated model parameters for class k are given by

$$a_k = M_k{}^{-1} B_k \tag{10}$$

After a few iterations between E and M steps, the actual model parameters are obtained by reducing the residuals and the data point is associated to a class k using the final model parameters. The motion vector v_p at the location p is assigned to k^{th} model if the likelihood function of p belonging to class k is greater than that of class m where m belongs to all motion models except for $m = k$ which is demonstrated in Eqn. 11.

$$v_p \in k^{th} \text{ class if } L_k(p) > L_m(p) \tag{11}$$

where

$$m \in [1 \ldots K] \quad \forall \quad m \neq k \tag{12}$$

C. Estimation of Number of Motion Models

The estimation of number of motion models plays an important role in accurate crowd flow segmentation as it decides the homogeneous segments to cluster together. In our approach, we have estimated the number of motion models based on k means clustering as proposed in [9]. All the motion vectors are divided into 8×8 blocks and the parameters of the translational models for each block is obtained. The obtained translational model parameters of each block are clustered using k-means iteratively by increasing the number of cluster centers from 1 onwards and the MSE is observed. Since the clustering may converge to local minima, the k-means clustering algorithm was performed multiple number of times by considering various randomly chosen cluster centers and the minimum value of MSE is picked. The number of classes where MSE falls less than the threshold η is considered to be the number of motion models, which is given as input to the EM algorithm. Typically, the value of η is chosen to be 5 to 15 percent of the maximum error. For faster convergence, the cluster centers corresponding to the number of motion models are given as the model parameters to the EM algorithm for clustering of motion vectors.

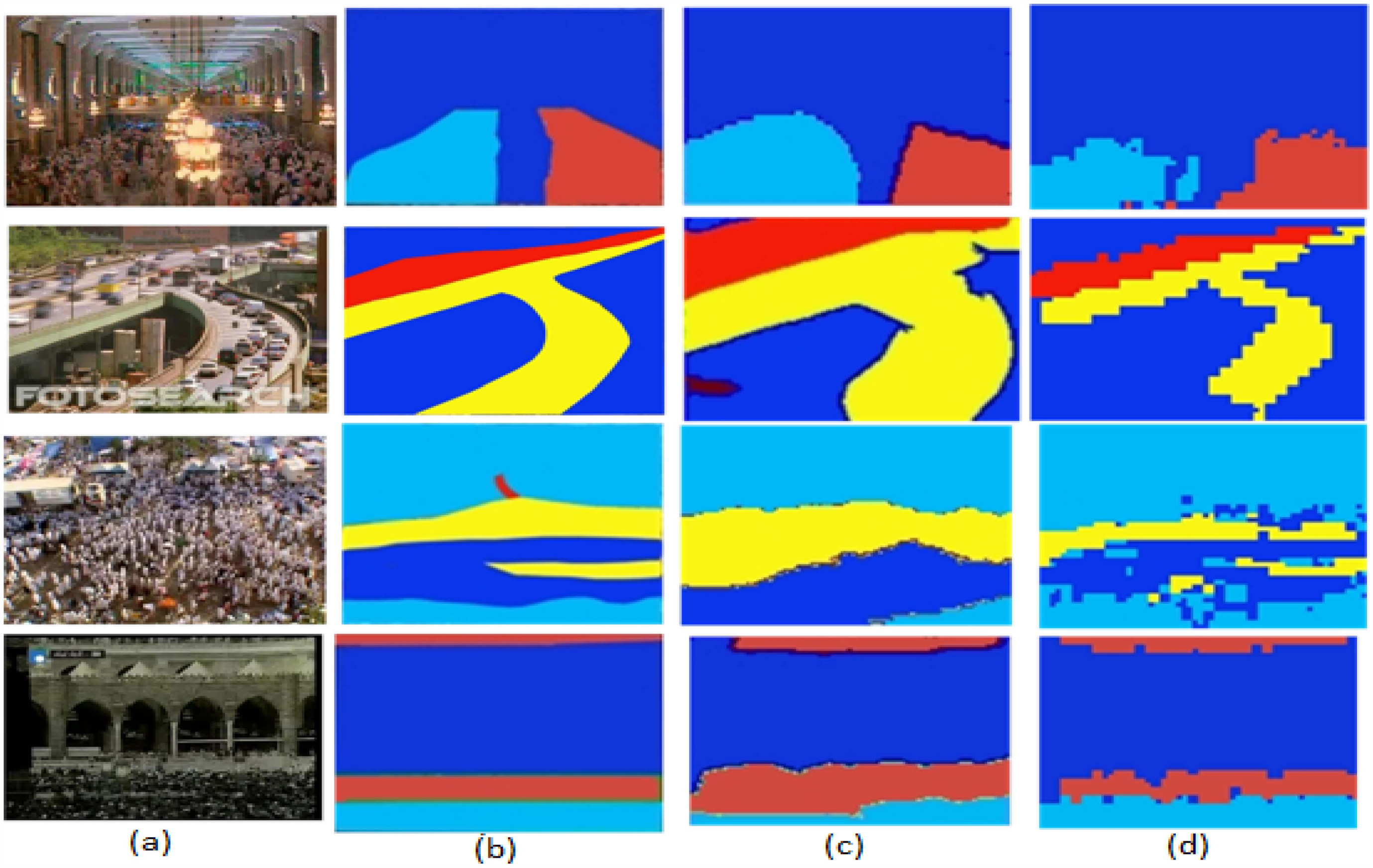

Fig. 3. Results of some of the videos in the database (a) Video Sequence (b) Ground Truth (c) Ali et al's result (d) Proposed Approach

D. Merging of Coherently Moving Motion Vectors

After performing the flow segmentation using EM algorithm, some of the homogeneous clusters which converges to same direction may remain as separate segments. So these clusters which converges to the same direction has to be merged together to perform accurate crowd flow segmentation. We have achieved this task by examining the coherency of the orientation of the sub-macroblocks at the boundaries of the clusters. The Normalized Histogram of the Orientation of the Motion Vectors of the sub-macroblocks at the boundaries of the clusters was obtained and the similarity measure between the distributions was examined for coherency. In this paper, we refer Normalized Histogram of the Orientation of the Motion Vectors of the sub-macroblocks at the boundaries of the clusters as $(NHOMV)$. We had used the metric of Bhattacharya coefficient as similarity measure as it is widely used to measure the similarity between two distributions as shown in Eqn. 13. Let $r(x)$ and $s(x)$ be the histogram of orientations of the sub-macroblocks at the boundaries of two clusters, then the Bhattacharya coefficient (ρ) for the two distributions $r(x)$ and $s(x)$ is given by

$$\rho(r,s) = \sum_{x \in D} \sqrt{r(x)s(x)} \qquad (13)$$

where, D represents the entire set of bins (orientation) in the histogram. If the measure of Bhattacharya coefficient (ρ) for a pair of distributions is greater than the threshold (ξ), then the two clusters are merged together. In order to determine the threshold (ξ), $(NHOMV)$ of the two clusters are obtained, where area over the distribution becomes equal to one. Ideally, if both the $(NHOMV)$ of the two clusters are exactly same, then Bhattacharya coefficient (ρ) for the corresponding two clusters becomes one since it is nothing but the summation over the area of the distribution. Empirically, if most of the orientations of the two clusters are in the same direction then the threshold (ξ) was found to be 0.5.

IV. Results and Discussions

We had implemented our proposed approach on the dataset provided by Ali et al. [1]. For each video, the motion vectors are extracted by partially decoding the corresponding video sequence in the H.264 compressed domain, preprocessed and clustered using the EM algorithm described in Section. III-B. Even though the motion vectors obtained in H.264 compressed domain are of different sizes, they are replicated at the sub macro block level i.e.,4×4 block in order to get the standard matrix for all the frames. The number of motion models for the EM algorithm is obtained by k-means clustering as described in Section. III-C. The convergence of the EM algorithm depends on the number of motion models, geometric shape of real clusters and initial values of model parameters. In order to reduce the computational complexity and increase the speed of convergence the initial values for the model parameters is given as the cluster centers obtained from k-means clustering. Empirically, the EM algorithm was found to converge within 3 to 4 iterations. The value of σ^2 used for calculating the weights is kept at 0.01. This parameter σ^2 controls the error i.e., distortion in the flow segmentation. After obtaining the flow segmentation, some of the homogeneous segments may remain as two or three separate segments. These over segmented regions are clustered together as described in Section. III-D. Our proposed approach performs much faster with better accuracy for the complex and challenging dataset which has huge variation of dynamics. The results for some of the videos in the dataset are shown in the Fig. 3. The accuracy of the proposed approach and that of [1] was compared with reference to the ground truth in terms of Jaccard measure as shown in Eqn. 14. The Jaccard measure obtained for the video sequences are shown in Table I in the same order as that of the Fig. 3. Let A be the ground truth for the crowd flow segmentation of a video sequence in the dataset and B be the crowd flow segmentation obtained by the proposed approach or Ali et al [1] approach of the corresponding video sequence, then the Jaccard (J) measure for A and B is given by

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{14}$$

The values of the Jaccard measure (J) for some of the videos shown in the Table I confirms that the proposed approach performs better than that of Ali et al. [1]. For instance, if we consider the second sequence in Fig. 3 (traffic video), one can clearly observe that the proposed approach is able to segment the crowd flow better than that of [1]. This is due to the fact that the proposed approach meticulously retains the dynamic region of the crowded sequence and discards the static region whereas Ali et al. [1] performs the flow segmentation at the global level by compromising the precision of the dynamic region in the crowded sequence. Similarly, the proposed approach is able to perform better crowd flow segmentation for other videos in the dataset [1]. For the complex crowd flow (third sequence in the Fig. 3), where the people are alternatively moving in opposite directions, the proposed approach was found to capture the diverse variation of the complex flow better than that of the results provided in [1]. This is due to the fact that EM algorithm clusters the motion vectors by considering the orientation of each motion vector. Even though motion vectors with the same flow are separated by motion vectors of another flow, our proposed approach is able to capture the variation in the diverse flow whereas, the algorithm proposed in [1] considers only the dominant flow of the region since it is global approach.

TABLE I
JACCARD INDEX SIMILARITY MEASURE FOR [1] AND OUR PROPOSED APPROACH FOR SOME OF THE VIDEOS IN THE DATABASE WITH REFERENCE TO GROUND TRUTH

Video Sequences	Jaccard Similarity Measure	
	Ali et al [1]	Proposed
Sequence 1	0.6717	0.6814
Sequence 2	0.2827	0.4674
Sequence 3	0.4129	0.4500
Sequence 4	0.5676	0.6212

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an approach for the crowd flow segmentation by using only motion vectors in H.264 compressed domain. The obtained motion vectors are clustered using EM algorithm and then refined by comparing the orientation flow of the macro blocks at the boundaries of the clusters. The proposed approach was found to be much faster since it is being performed in compressed domain and the quantitative measures indicate better performance of the proposed approach. As we are performing the flow segmentation at the sub-macro block level, our proposed approach may fail to perform precise flow segmentation as that of the optical flow as it is done at the pixel level. The clustering can be further extended to improve its performance to handle low velocity motion as that of the mecca sequence.

ACKNOWLEDGMENT

This work was supported by CARS (CARS-25) project from CAIR, DRDO, Govt. of India.

REFERENCES

[1] S. Ali and M. Shah, "A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.

[2] J. C. S. Jacques Junior, S. Raupp Musse, and C. R. Jung, "Crowd analysis using computer vision techniques," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 66 – 77, 2010.

[3] B. Zhan, D. Monekosso, P. Remagnino, S. Velastin, and L.-Q. Xu, "Crowd analysis: a survey," *Machine Vision and Applications*, vol. 19, no. 5 - 6, pp. 345 – 357, 2008.

[4] S. Wu and H. S. Wong, "Crowd motion partitioning in a scattered motion field," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 5, pp. 1443 – 1454, 2012.

[5] W. Li, J.-H. Ruan, and H.-A. Zhao, "Crowd movement segmentation using velocity field histogram curve," in *International Conference on Wavelet Analysis and Pattern Recognition*, 2012, pp. 191 – 195.

[6] A. Kuhn, T. Senst, I. Keller, T. Sikora, and H. Theisel, "A lagrangian frame work for video analytics," in *IEEE International Workshop on Multimedia Signal Processing*, 2012, pp. 387 – 392.

[7] Y. Weiss, "Motion segmentation using em - a short tutorial," 1997.

[8] R. Babu, K. Ramakrishnan, and S. Srinivasan, "Video object segmentation: a compressed domain approach," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 4, pp. 462–474, 2004.

[9] R. V. Babu and K. R. Ramakrishnan, "Sprite generation from mpeg video using motion information," *International Journal of Image and Graphics*, vol. 4, no. 2, pp. 263–280, 2004.