A Simple Method for Unusual Event Detection using Mahalanobis Metric

Md. Haris Uddin Sharif, Sahin Uyaver, Md. Haidar Sharif

***Abstract*—In this paper, we address a simple but effective approach to detect and locate unusual activities in videos. The approach does not depend on individual subject tracking. It is based on the statistical treatments of the spatiotemporal information (STI) of a set of points of interest within a region of interest over time. The achieved STI is clustered and Mahalanobis distances are calculated. Member and non-member groups, are resulted with three sigma rule, help to estimate cluster distance. Each frame corresponds to either normal or abnormal frame groups. Cluster distances are summed up to get an effective distance, which is then normalized and fitted with polynomial to know its correspondence. If it corresponds to abnormal group, then its region of the highest cluster distance indicates the most visual attended part and so on. Experiments have been conducted on several real video data and the reported results represent performance.**

***Index Terms*—Unusual activity, Cluster distance, Mahalanobis distance, 3-sigma rule**

I. Introduction

Event detection in surveillance videos is an essential task for both private and public places. As large amount of video surveillance data makes it a backbreaking task for people to keep watching and finding interesting events, an automatic surveillance system is firmly requisite for the security management teams. Video event can be defined to be an observable action or change of state in a video stream that would be important for the security management team. Events would vary greatly in duration and would start from two frames to longer duration events that can exceed the bounds of the excerpt. Some events occur frequently e.g., in the airport people are putting objects, getting objects, meeting and discussing, splitting up, and etc. Conversely, some events go off suddenly or unusually e.g., in the airport a person is running, falling on the escalator, going to the forbidden area, and etc. Both types of event detection in video surveillance is an important task for public security and safety in areas e.g., airports, malls, banks, metros, pedestrian subways, stations, city centers, hospitals, hotels, schools, concerts, parking places, sporting events, political events, and mass meetings. In spite of the concerted effort of computer vision research community, intelligent surveillance systems have not yet attained the desirable level of applicability and robustness. This is mainly due to the algorithmic assumptions as well as the huge amount of video data analysis.

There are some works [7, 23, 15, 12, 24, 25, 3, 2, 11, 5, 1, 10, 16] which detect abnormalities mainly in crowd flows. A system for automatically learning motion patterns for anomaly detection and behavior prediction based on a proposed algorithm for robustly tracking multiple objects can be seen in [7]. Authors in [23] detected events which have never occurred or occur so rarely that they are not represented in the clustered activities. The method includes robust tracking, based on probabilistic method for background subtraction.

In this paper, we address a simple method which detects and localizes abnormalities on video frames based on the statistical treatments of the spatiotemporal information (STI) of a set of interest points within a region of interest over time. The obtained STI is clustered and Mahalanobis distances are calculated for each cluster. Those distances are then classified into two groups namely member and non-member based on three sigma rule. Distances in nonmember group of a cluster are collected to represent its distance called cluster distance $\sum_c$. All cluster distances are summed up to get an effective distance for presenting each frame which is then normalized and poly-fitted to obtain a decision whether it falls into normal or abnormal frame groups. If a frame belongs to an abnormal category, then its region which posses the highest cluster distance demonstrates the most visual attended part of the frame and so on. Our primary goal is to introduce a holistic method (e.g., [1], [16], etc.) free from individual subject tracking to detect anomalies in videos. A common aspect of our work and those of [3, 2, 10] is that there is enough perturbation in the optical flow pattern in case of emergency. Our approach would be deemed as a further enhancement of these works with some senses. We profit from a different course by estimating and analyzing of the STI of video frames to detect anomalies directly without tracking subject singly. Thus it is effective for the high density mover scenes as well as low density scenes. It has further advantages: (i) it detects all events in videos where motion variations are important as compared to previous events; (ii) it works all directional flow of movers without imposing a restriction of their numbers in the videos; (iii) it does not expect efficient learning process and training data but would look for a prior cut-off; (iv) it localizes the abnormalities on the video frames in a sorted order based on distance metrics of the clusters; (v) it does not expect low-level change detection algorithms.

The rest of the paper is organized as follows: Section II illustrates the detailed implementation steps of the proposed approach; Section III demonstrates detection abilities of the proposed detector; Section IV gives few clues for further investigation; and Section V makes conclusion.

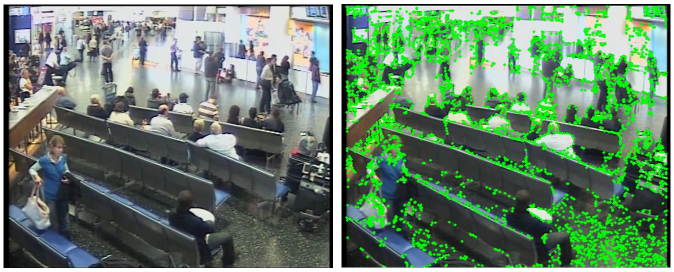

Fig. 1. Left image indicates original video frame, whereas green points in right image represent Harris corners or points of interest.

Fig. 2. Left and right images respectively, represent optical flow before and after suppression of the static points of interest.

II. Implementation Steps

A. Feature selection

Moravec's corner detector [27] is a relatively simple algorithm, but is now commonly considered out-of-date. It is not rotationally invariant. It is susceptible to reporting false corners along edges and at isolated pixels so is sensitive to noise [32]. However, it is computationally efficient. The other way around the Harris corner detector [28] is computationally demanding, but directly addresses many of the limitations of the Moravec corner detector. In our framework, we have considered Harris corner or point of interest detector. The Harris corner detector is a famous point of interest detector due to its strong invariance to rotation, scale, illumination variation, and image noise [33]. It is based on the local auto-correlation function of a signal, where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions.

Like [32], we have considered that in video surveillance scenes, camera positions and lighting conditions admit to get a large number of corner features that can be easily captured and tracked. Fig. 1 demonstrates an example of Harris corner or interest point detector. Fig. 2 depicts suppression of the static points of interest.

B. Estimation of optical flow

The aim of optical flow technique is to compute an approximation to the 2D motion field, a projection of the 3D velocities of surface points onto the imaging surface, from spatiotemporal patterns of images intensity [34], [35]. To estimate optical flow between successive video frames the well known combination of feature selection as introduced by Shi and Tomasi [30] and the algorithm of Lucas and Kanade for feature tracking [29] is used.

For each point of interest which is found in the next frame, we can get its position, direction, and moving distance easily. *How much distance has it moved? Which direction has it moved?*

Suppose that any point of interest i with its position in the current frame $p(x_i, y_i)$ is found in the next frame with its new position $q(x_i, y_i)$; where x_i and y_i be the x and y coordinates, respectively. Now, suppose that the traveled distance d_i and its direction β_i. Then we can calculate d_i and β_i using trigonometry [32]. It is easy to calculate the displacement or Euclidean distance of $p(x_i, y_i)$ and $q(x_i, y_i)$ using Euclidean metric as:

$$d_i = \sqrt{(q_{x_i} - p_{x_i})^2 + (q_{y_i} - p_{y_i})^2}. \tag{1}$$

Direction of motion of the feature i can be calculated using the following trigonometric function:

$$direction = tan^{-1}(\frac{q_{y_i} - p_{y_i}}{q_{x_i} - p_{x_i}}). \tag{2}$$

But there are several potential problems if we wish to calculate motion direction using Eq. 2, for instances [32]:

Firstly: Eq. 2 does not show expected performance for a complete range of angles from 0° to 360°. Only angles between -90° and $+90^\circ$ will be returned, other angles will be (say 180°) out-of-phase. For example, assume that two defined points $(x_1 = 5, y_1 = 5)$ and $(x_2 = -5, y_2 = -5)$. Using Eq. 2, the point $(x_2 = -5, y_2 = -5)$ will produce the same angle as the point $(x_1 = 5, y_1 = 5)$ will do, nonetheless these are not in same quadrant.

Secondly: Points on the vertical axis have $x_i = 0$, hence, if we wish to calculate y_i/x_i we will get ∞ which will generate an exception when calculated on the computer.

To avoid these problems, alike [32], we have used $atan2$ function to calculate the accurate direction of motion β_i of any point of interest i by the help of:

$$\beta_i = atan2(q_{y_i} - p_{y_i}, q_{x_i} - p_{x_i}). \tag{3}$$

C. Algorithm of k-means

Upon static error suppression points of interest, we apply k-means method to get clusters. The geometric clustering method, *k-means*, is a simple and fast method for partitioning data points into k clusters, based on the work done by [31] (so-called Voronoi iteration). It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of points in $\mathbb{R}^d$ (d dimensional real numbers). After being seeded with a set of k centers c_1, c_2, …, c_k in $\mathbb{R}^d$, the algorithm partitions these points into clusters as follows:

1) For each $i \in \{1, \ldots, k\}$, set the cluster $\mathcal{C}_i$ to be the set of points in X that are closer to c_i than they are to c_j for all $j \neq i$.

Fig. 3. Clustering of moving interest points by k-means

2) For each $i \in \{1, \ldots, k\}$, set c_i to be the center of mass of all points in $\mathcal{C}_i$: $c_i = \frac{1}{|c_i|} \sum_{x_j \in c_i} x_j$.
3) Repeat steps 1 and 2 until c_i and $\mathcal{C}_i$ no longer change, at which point return the clusters $\mathcal{C}_i$.

If there are two centers equally close to a point in X, we break the tie arbitrarily. If a cluster has no data points at the end of step 2, we eliminate the cluster and continue as before. Fig. 3 illustrates clustering performed by k-means.

D. Analysis of cluster information

On clustering we have estimated the *geometric means* of all x coordinates and y coordinates for each cluster to get single origin. If any cluster c will contain m number of x coordinates, then the number of y coordinates, displacements, and directions will be also m. If x_{g_c} and y_{g_c} symbolize geometric means of all x coordinates and y coordinates of a cluster, respectively. It is easy to define them as:

$$x_{g_c} = \left[\prod_{j=1}^{m} x_j \right]^{\frac{1}{m}} \tag{4}$$

$$y_{g_c} = \left[\prod_{j=1}^{m} y_j \right]^{\frac{1}{m}}. \tag{5}$$

E. Circular mean

Consider four persons starting walking from a fixed location to four different directions, i.e., north-east $60\,^\circ$, north-east $30\,^\circ$, south-east $330\,^\circ$, and south-east $310\,^\circ$. The arithmetic, geometric, and harmonic means of these directions are $182.5\,^\circ$, $116.5\,^\circ$, and $71.1\,^\circ$, respectively. All estimations are clearly in error as depicted in Fig. 4. Since arithmetic, geometric, and harmonic means are ineffective for angles, it is important to find a good estimation method to obtain accurate mean value of the angles. Merely *mean of a series of angles* or *circular mean* is the solution of this problem.

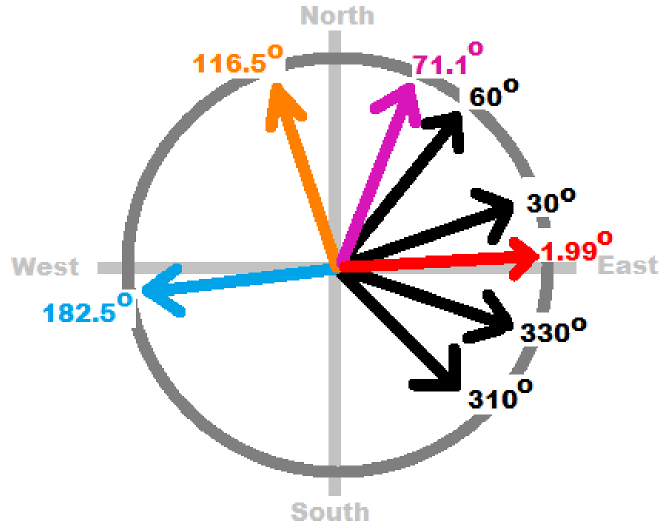

Fig. 4. Arithmetic, geometric, and harmonic means of $60\,^\circ$, $30\,^\circ$, $330\,^\circ$, and $310\,^\circ$ provide inaccurate $182.5\,^\circ$, $116.5\,^\circ$, and $71.1\,^\circ$ means estimation, respectively. Solely circular mean can give correct result as shown by red line.

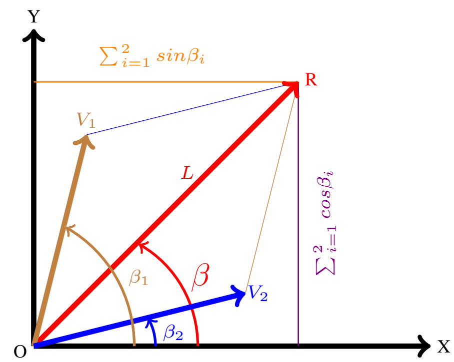

Fig. 5. Analysis of directions of two interest points using trigonometry

Assume that two interest points of common origin O are available which have displacement vectors V_1 and V_2 with directions β_1 and β_2, respectively. Their resultant directional mean β and circular resultant vector length $L = OR$ can be found graphically as shown in Fig. 5. The circular resultant displacement vector length L of these interest points can be defined by using Pythagorean theorem as:

$$L = \sqrt{\left(\sum_{i=1}^{2} sin\beta_i \right)^2 + \left(\sum_{i=1}^{2} cos\beta_i \right)^2}. \tag{6}$$

However the graphical solution becomes extremely inefficient when a large number of directions to be added and also often arises the problem of precision. Yet an elementary trigonometric analysis can solve the problem with high accuracies.

Consider that in a cluster we have m number of interest points of single origin (x_{g_c}, y_{g_c}) with displacement vectors $V_1, V_2, V_3, \ldots, V_m$ and their respective directions β_1, β_2, β_3, ..., β_m. Then the *tangent of* R also called *mean of a series of angles* or *vector mean* or *circular mean*, symbolized by β_{g_c}, can be formulated by dint of:

$$\beta_{g_c} = \begin{cases} tan^{-1}\frac{\sum_{i=1}^{m} sin\beta_i}{\sum_{i=1}^{m} cos\beta_i} & if\ \sum_{i=1}^{m} sin\beta_i > 0, \sum_{i=1}^{m} cos\beta_i > 0 \\ tan^{-1}\frac{\sum_{i=1}^{m} sin\beta_i}{\sum_{i=1}^{m} cos\beta_i} + 180^\circ & if\ \sum_{i=1}^{m} cos\beta_i < 0 \\ tan^{-1}\frac{\sum_{i=1}^{m} sin\beta_i}{\sum_{i=1}^{m} cos\beta_i} + 360^\circ & if\ \sum_{i=1}^{m} sin\beta_i < 0, \sum_{i=1}^{m} cos\beta_i > 0 \end{cases} \tag{7}$$

and their circular resultant displacement vector length L_{g_c} can be expressed with the help of Pythagorean theorem as:

$$L_{g_c} = \sqrt{\left(\sum_{i=1}^{m} sin\beta_i\right)^2 + \left(\sum_{i=1}^{m} cos\beta_i\right)^2}. \tag{8}$$

Using Eq. 7, we can explicitly express the accurate mean of angles in Fig. 4 which is 1.99°.

F. Events detection

Now, we have information of location (Eq. 4, 5), direction (Eq. 7), and displacement vector (Eq. 8) of each cluster. For each cluster (x_{g_c}, y_{g_c}), β_{g_c}, and L_{g_c} demonstrate its principal components than the individual feature points. Based on estimated direction component β_{g_c} each cluster is named by lower-bound or upper-bound or horizon-bound. A cluster will be called a *lower-bound* if β_{g_c} ranges from 224.98° to 314.98°. A cluster will be called a *upper-bound* if β_{g_c} ranges from 134.98° to 44.98°. A cluster will be called a *horizon-bound* if β_{g_c} limits either from 134.99° to 224.99° or from 44.99° to 314.99°. The lower-bound and upper-bound clusters play a vital role for event detection, while horizon-bound clusters bear no significant information. An ObjectDrop or ObjectPut event can be detected if any cluster on the video frame is proclaimed by lower-bound. An ObjectGet event can be detected if any cluster on the video frame is proclaimed by upper-bound. We draw a circle with center (x_{g_c}, y_{g_c}), direction β_{g_c}, and displacement vector L_{g_c} for each lower-bound and/or upper-bound cluster on the camera view image to demonstrate the delectability of the detector.

III. Experimental Results

A large variety in the appearance of the event types makes the events detection task in the surveillance video streams selected for the TRECVid 2008 extremely difficult. The source data of TRECVid 2008 comprise about 100 hours (10 days * 2 hours per day * 5 cameras) of video streams obtained from Gatwich Airport surveillance video data [16]. A number of events for this task were defined. Since all the videos are taken from surveillance cameras which means the position of the cameras is still and cannot be changed. However, it was not practical for us to analyze 100 hours of video except few hours. The value of k has been considered as 7.

If there is a video event on the real video and the algorithm can detect that event, then this case is defined as a *true positive* or correct detection. If there is no video event on the real video but the algorithm can detect a video event, then this case is defined as a false positive or *false alarm* detection. If there is a video event on the real video but the algorithm cannot detect, then this case is defined as a *false negative* or miss detection.

Fig. 6 shows four sample frames of an ObjectDrop event detected by our proposed detector. Something from a hand of a baby suddenly fall off on the floor. The in-flight stuff has been detected as true positive ObjectDrop event. Object centered at green circle and falling direction with red arrow have been drawn by the algorithm. Fig. 7 demonstrates four sample frames of an ObjectPut event detected by the detector. A person is putting a hand bag from one location to another. That event has been detected as true positive ObjectPut event. Fig. 8 represents four sample frames of another ObjectPut event detected by our proposed detector. A person is putting a hand-phone from one place to another. The event has been detected as true positive ObjectPut event. Fig. 9 shows four sample frames of an ObjectPut event detected by the detector. Someone is going to sit on the bench in the waiting area. But this event has been detected as false positive event. Fig. 10 shows four sample frames of an ObjectGet event detected by the detector. A person is getting some stuff from the floor and placing some upper location. The event has been detected as true positive ObjectGet event.

Nonetheless, our proposed event detector cannot detect accurately event e.g., Fig. 11 and 12. These type of events occur with partial occlusion normally. However, up to to this point, we can conclude that some results represent the effectiveness of the proposed event detector, while the rests show the degree of the difficulty of the problem at hand. TRECVid 2008 surveillance event detection task is a big challenge to test the applicability of such detector in a real world setting. Yet, we have obtained much practical experiences which will help to propose better algorithms in future.

Challenges which make circumscribe the performance of our event detector include but not limited to: (i) a wide variety in the appearance of event types with different view angles; (ii) divergent degrees of imperfect occlusion; (iii) complicated interactions among people; and (iv) shadow and fluctuation. Besides these, video events have taken place significantly far distance from the camera and hence the considerable amount of motion components were insufficient to analyze over the obtained optical flow information.

These extremely challenging reasons also directly reflected on the detectors proposed by [19], [23], [25], and [26]. Among those detectors, the best ObjectPut results obtained by the detector of Orhan et al. [23]. Their proposed detector used 1944 ObjectPut events and detected 573 events. This record limited the performance of their detector by about 29.5% only, which is still far behind for real applications.

IV. Future Work Directions

To detect events from the TRECVid 2008 are extremely difficult tasks. To solve the existing challenges of TRECVid 2008, it would take some decades for computer vision research community. Future direction would be proposing new efficient algorithms citing the solution and/or minimization of the exiting limitations.

Notwithstanding our short term targets cover detection of events like OpposingFlow along with the performance im-

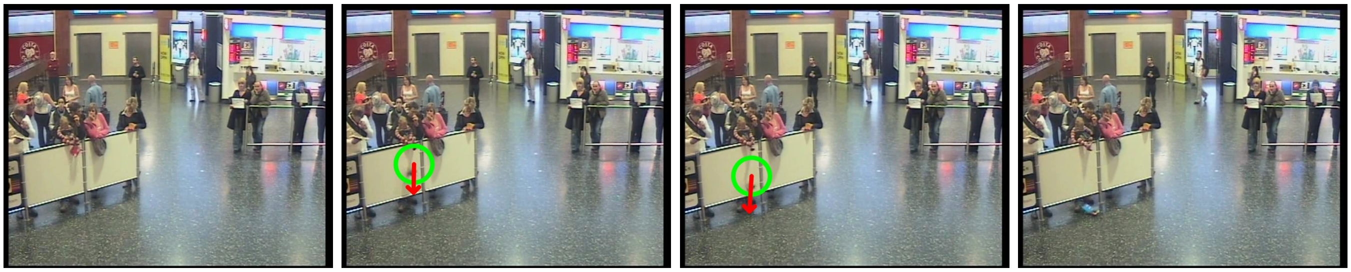

Fig. 6. True positive ObjectDrop: Something is dropping from the hand of a baby which has been detected by the detector.

Fig. 7. True positive ObjectPut: A person is putting some stuff from one place to another which has been detected by the detector.

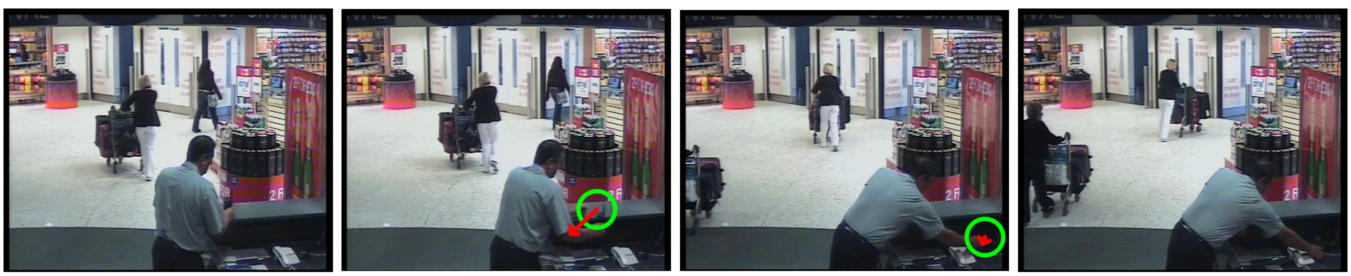

Fig. 8. True positive ObjectPut: A person is putting a cell phone on the self detected by the detector.

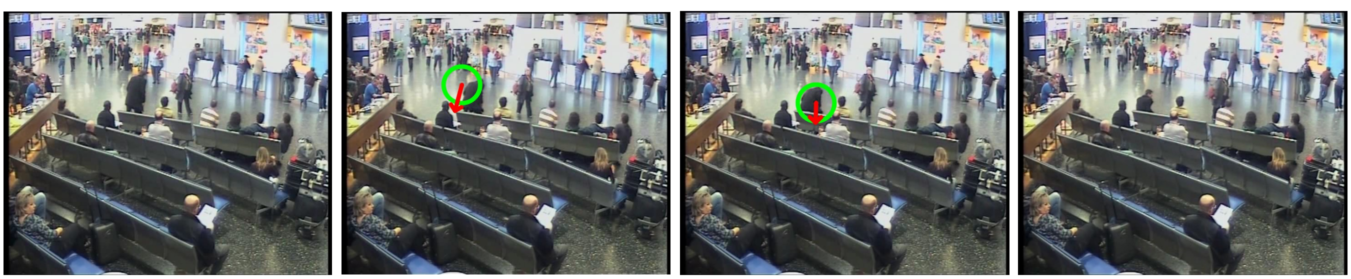

Fig. 9. False positive ObjectPut: A person is going to sit on a bench has been detected as ObjectPut event.

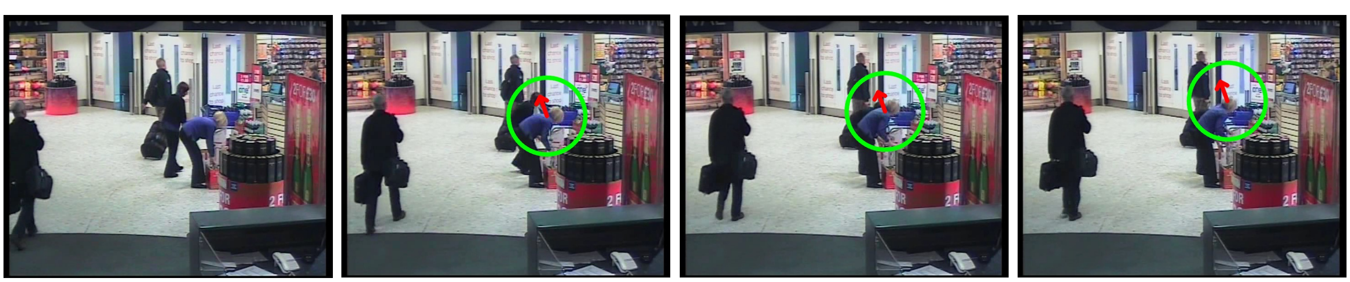

Fig. 10. True positive ObjectGet: A person is getting some stuff from the floor which has been detected as ObjectGet event.

Fig. 11. False negative ObjectPut: One person getting some stuffs but cannot be detected.

Fig. 12. False negative ObjectPut and ObjectGet: One person is putting some stuff while another is getting something but cannot be detected.

provement of the proposed event detector. Our current event detector cannot detect OpposingFlow video event. But we can detect that event by using the unused horizon-bound cluster.

On the other hand, our long term targets would be detecting various events e.g., PeopleMeet, PeopleSplit, Embrace, TakePicture, CellToEar, Pointing, etc. by proposing better algorithms concerning some degree of minimization of the exiting limitations. In this aspect, the comprehended practical experiences will assist substantially.

V. Conclusion

We have presented principally a detection based approach to detect video events, which is heavily based on optical flow techniques. Optical flow techniques track low level information like points of interest. The tracked interest points are grouped into several clusters using *k-means* algorithm. To find the principle components of each cluster, geometric means of locations and circular means of direction and displacement of the interest points of each cluster are estimated. Based on those components each cluster is defined either lower-bound or horizon-bound or upper-bound cluster. Lower-bound clusters are used for ObjectDrop and ObjectPut video events, while upper-bound clusters are considered for potential ObjectGet video events. The detection results of ObjectDrop, ObjectPut and ObjectGet at TRECVid 2008 in real videos have been demonstrated. Some results substantiate the competence of the proposed event detector, while the remainders give the degree of the difficulty of the problem at hand. The achieved practical experiences will help to propose imperious algorithms in future. Consequently, further investigation in future would bring about superior results.

Acknowledgment

Authors would like to thank TRECVid 2008 for accessing the surveillance video streams.

References

[1] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001.

[2] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *International Conference on Image Processing (ICIP)*, pages 900–903, 2002.

[3] T. Serre, and L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Computer Vision and Pattern Recognition (CVPR)*, pages 994–1000, 2005.

[4] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *International Conference on Computer Vision (ICCV)*, pages 84–93, 2001.

[5] R. Hamid et al. Unsupervised Activity Discovery and Characterization From Event-Streams. In *Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 251–258, 2005.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[7] S. Bileschi and L. Wolf. Image representations beyond histograms of gradients: The role of Gestalt descriptors. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[8] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Computer Vision and Pattern Recognition (CVPR)*, pages 819-826, 2004.

[9] D. Zhang, D. G. Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 611-618, 2005.

[10] E. L. Andrade, S. Blunsden, and R. B. Fisher. Hidden markov models for optical flow analysis in crowds. In *International Conference on Pattern Recognition (ICPR)*, pages 460-463, 2006.

[11] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1-6, 2007.

[12] N. Ihaddadene and C. Djeraba. Real-time crowd motion analysis. In *International Conference on Pattern Recognition (ICPR)*, pages 460-463, 2008.

[13] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):893-908, 2008.

[14] M. H. Sharif and C. Djeraba. Exceptional motion frames detection by means of spatiotemporal region of interest features. In *International Conference on Image Processing (ICIP)*, pages 981-984, 2009.

[15] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition (CVPR)*, pages 935-942, 2009.

[16] TRECVid. Benchmark data of TRECVid. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008. 4

[17] X. Xue and al. Fudan University at TRECVid. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008.

[18] J. Guo et al. MCG-ICT-CAS at TRECVid. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008.

[19] A. Hauptmann et al. Informedia at TRECVid : Exploring New Frontiers. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008. 4

[20] K. Yokoi and H. Nakai and T. Sato. Toshiba at TRECVid : Surveillance Event Detection Task. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008.

[21] Y. Kawai, and M. Takahashi, and M. Sano, and M. Fujii. NHK STRL at TRECVid : High-Level Feature Extraction and Surveillance Event Detection. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008.

[22] S. Hao et al. Tokyo Institute of Technology at TRECVid. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008.

[23] O. B. Orhan et al. University of Central Florida at TRECVid : Content Based Copy Detection and Surveillance Event Detection. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008. 4

[24] P. Wilkins et al. Dublin City University at TRECVid. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008.

[25] P. Chmelar et al. Brno University of Technology at TRECVid. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008. 4

[26] M. Dikmen et al. IFP-UIUC-NEC at TRECVid. In *Surveillance Event Detection Pilot. http://www-nlpir.nist.gov/projects/trecvid*, 2008. 4

[27] H. P. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. In *Technical Report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University* & *doct. diss., Stanf. University*, 1980. 2

[28] C. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, 1988. 2

[29] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981. 2

[30] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994. 2

[31] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982. 2

[32] M. H. Sharif, N. Ihaddadene, and C. Djeraba. Finding and indexing of eccentric events in video emanates. *Journal of Multimedia*, 5(1):22–35, 2010. 2

[33] C. Schmid, and R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000. 2

[34] B. K. P. Horn. Robot Vision. In *MIT Press, Cambridge*, 1986. 2

[35] A. Verri and T. A. Poggio. Against Quantitative Optical Flow. In *International Conference on Computer Vision (ICCV)*, pages 171–180, 1987. 2