Event Detection in Video Using Motion Analysis

Ricardo Castellanos
Florida Atlantic University
777 Glades Rd, SE-413
Boca Raton, Florida, 33431
+1 (561) 755 - 7551
rcastel5@fau.edu

Hari Kalva
Florida Atlantic University
777 Glades Rd, SE-422
Boca Raton, Florida, 33431
+1 (561) 297 - 0511
hari.kalva@fau.edu

Oge Marques
Florida Atlantic University
777 Glades Rd, SE-422
Boca Raton, Florida, 33431
+1 (561) 297 - 3857
omarques@fau.edu

Borko Furht
Florida Atlantic University
777 Glades Rd, SE-422
Boca Raton, Florida, 33431
+1 (561) 297 - 2855
bfurht@fau.edu

ABSTRACT

Digital video is being used widely in a variety of applications such as entertainment, surveillance and security. Large amount of video in surveillance and security requires systems capable of processing video to automatically detect and recognize events to alleviate the load on humans and enable preventive actions when events are detected. The main objective of this work is the analysis of computer vision techniques and algorithms to perform automatic detection of specific events in video sequences. This paper presents a surveillance system based on motion analysis and introduces the idea of event probability zones. Advantages, limitations, capabilities and possible solution alternatives are also discussed. The result is a system capable of detecting events of objects moving in opposing direction in a predefined context or running in the scene; the results showed precision greater than 50% and recall greater than 80%.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *Video*. I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding – *Motion, Video Analysis*. I.4.8 **[Image Processing and Computer Vision**]: Scene Analysis – *Motion*. I.5.4 [**Pattern Recognition**]: Applications – *Computer Vision*.

General Terms

Design, Experimentation, Measurement and Theory

Keywords

Event detection, surveillance, background subtraction, optical flow

1. INTRODUCTION

The huge accumulation of digital data in this new century has become an interesting challenge where storage and processing of such quantities of information are the key factors to satisfy user requirements and expectations. Multimedia data such as video sequences in visual surveillance systems is a very important topic and probably one of the most illustrative examples of this challenge because the large demand for analysis and synthesis that is needed to understand the contents and to determine specific actions based on registered events. Events are phenomena or circumstances that happen at a given place and time which can be identified without ambiguities, for example, a person entering a forbidden zone, a suspicious object abandoned in a public place or a car parking in a garage.

Digital video recording devices are now ubiquitous and pervasive in our daily lives. They are mounted indoors and outdoors everywhere: offices, rooms, halls, banks, hotels, hospitals, casinos, airports, parking lots, buildings, military sites, streets and intersections; some vehicles even have cameras recording passengers and the surroundings of the car. The wide range of potential applications includes: access control in special areas, person-specific identification, crowd flux statistics and congestion analysis, anomaly detection and alarming and interactive surveillance using multiple cameras [5].

Computer vision technologies are intended to perform intelligent tasks with these “digital eyes”, attaching “brains” to the imaging devices and thus, creating a very useful tool used for video surveillance, entertainment/augmented reality applications, autonomous vehicles and driver assistance systems, robotics and smart health care.

Visual surveillance systems, address real-time observation of objects in some environment leading to a description about the activities or interaction of the objects within the environment or among the objects. However, a human operator has either to watch a massive amount of video data in real-time with full attention to detect any anomalies or events, or the video data can only be used as evidence after the abnormal event has occurred, due to the lack of real-time automatic tracking and analysis. An automatic video surveillance system comprises different functional blocks such as foreground segmentation, object detection/tracking, human or object analysis and finally, activity or behavioral analysis [2]. These blocks are implemented using computer vision techniques and algorithms alleviating the load on humans and enabling preventive acts or alarms when a specific event is detected.

This paper presents a system capable to detect specific events automatically in video surveillance applications using an indoor, single and fixed camera, reducing or suppressing human interaction with the system and reporting alerts based on the events detected.

ARTEMIS'10, October 29, 2010, Firenze, Italy.
Copyright 2010 ACM 978-1-4503-0163-3/10/10...$10.00.

2. AUTOMATIC VIDEO SURVEILLANCE SYSTEM

The input to a video surveillance system is a video stream coming from a single or multiple cameras. The system analyzes the video content going through each single block separating foreground from background, detecting and tracking objects, and performing a high-level analysis [2]. The high-level analysis provides results such as a scenario being *normal* or *abnormal* and based on this result, the system can report the state of the process to facilitate a human operator to focus on the abnormal scenarios without having to stare at the video trying to find any anomaly. A general framework of a visual surveillance system is presented in [5].

2.1 Environment Modeling

A *sequence* is a set of consecutive frames recorded at the same location. Therefore, group of common elements are shared within this set, and this is what is referred to as *background*. The active construction and updating of the background model is indispensable to visual surveillance so the next blocks in the pipeline depend on the accuracy of this model.

2.2 Motion Segmentation

The objective is to separate foreground from background in the video sequence. Foreground detection is generally easier in the indoor environment because the outdoor environment is more complex, as wavering tree branches, flickering water surfaces, periodic opening and closing of doors are occurring. One of the most generalized methods is the background subtraction based on motion segmentation used when the environment model described before has a relatively static background. Moving regions in an image are detected by taking the difference between the current image and the reference background image in a pixel by pixel approach [6]. Other methods, such as temporal differencing, make use of the pixel-wise differences between two or three consecutive frames in an image sequence to extract the moving regions; this method is very adaptive to dynamic environments but generally does a poor job of extracting all the relevant pixels [5]. Finally, the optical flow based motion segmentation method uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence and can be used to detect independently moving objects even in the presence of camera motion [4]. However, this method is computationally complex and has constraints to be applied in real time without specialized hardware.

2.3 Object Classification

Once the segmentation process has been completed, it is necessary to perform object classification in order to identify the different moving regions for further analysis in the system [9]. This task is necessary to define the moving regions as moving objects with a higher level of knowledge for tracking purposes or behavior observation and analysis. There are two main categories for classifying objects: shape-based and motion-based classification [5].

2.4 Tracking

Using the features extracted for the classified objects and their defined characteristics, it is possible to localize its position along the different sequence of frames. Tracking objects over time typically involves matching them in consecutive images using features such as points, lines or blobs [5,9]. Once the object is tracked, very useful information such as position, velocity, centroid and periodicity becomes available and can be used for further processing and analysis but this is only possible when the object has been tracked for a given period of time.

2.5 Behavior Understanding and Description

To detect the anomaly of a scene, it is necessary to model the behavior of the objects in the frames [5]. This task can be performed using the information gathered in the previous blocks where the object is recognized and classified using specific features like position, blob area, contour, displacement, direction of movement, magnitude of movement, color, etc. The analysis of these features during time can help us to describe the behavior of the objects as well as the interaction with other objects, based on the changes they are experimenting. Once the behavior is identified, it is translated into high level human expressions to be matched with previous defined patterns [5].

2.6 Personal Identification

This block does not apply to all the video surveillance systems, but is included in the general video surveillance framework because its faculty to be used for face recognition purposes. This way, human face and gait is analyzed in order to identify subjects in a non intrusive fashion [2].

With this approach, the video surveillance system can give answers to the questions involving either what? (related to the object detection) or who? (related to the object identification).

3. EVENT DETECTION

The TREC video retrieval evaluation (TRECVid2009) was the motivation for the implementation of a system capable of detecting events in videos[7] reported in this paper. The goal of the TRECVid evaluation is to build and evaluate systems that can detect instances of a variety of observable events in the airport surveillance domain based on video surveillance data collected by the UK Home Office at the London Gatwick International Airport.

There can be a large variety of events to detect in surveillance videos. The number of events depends on different factors which the system must consider according to the design parameters, purpose of detections, camera locations and probability of events occurring in specific locations. This work focuses on the combination of functional blocks using computer vision techniques to detect and identify events based on motion analysis. The "*OpposingFlow*" and "*PersonRuns*" events described in the TRECVid Event Annotation guidelines [12] were selected, because they have common characteristics to be used with the proposed solution such as prolonged motion and object's displacement. Moreover, these two events were common to the Camera1 video subset, sharing the same environmental conditions.

Previous work related to detection of the aforementioned events on TRECVid uses background modeling techniques to detect moving objects [3, 11, 13, 14, 16]. Some groups performed manual labeling of humans in the training dataset using different shape identifiers [11] (heads, heads and shoulders, faces and whole bodies). Object identification is performed using Haar features and Bayesian filters [10, 15]. Finally, event detection is performed using SVM and rule-based classifiers. Other approaches detect humans using Histogram of Oriented Gradient (HOG) and events are detected based on change detection and human tracking over extracted trajectories [11, 13]. Tracking is improved by Kalman filtering, blob size and speed analysis and trajectory curvature during intervals. Some other approaches describe events as a pattern detected by Gaussian Mixture Models or Hidden Markov Models where the degree of correspondence of the extracted trajectory with the model is expressed by likelihood [4].

The following sections describe the implementation of the proposed solution using background subtraction techniques to perform the background modeling and optical flow as a feature of trajectory generation.

3.1 Implementation

The general block diagram for the proposed solution is represented in Figure 1, which shows the system composed of five blocks. The input video is pre-processed to decode it in frames and determine the selection of relevant information which will make the event detection task easier to achieve. The foreground/background estimation block and the optical flow block receive the pre-processed information. The first separates the background from the foreground; and the second evaluates the optical flow for all the pixels contained in the pre-processed image. After that, the optical flow information is selected and classified in the next block based on the foreground contents and the conditions designed to flag the selected event. Finally, the data is post-processed to determine whether or not the event is occurring, so that the user can visualize the results obtained.

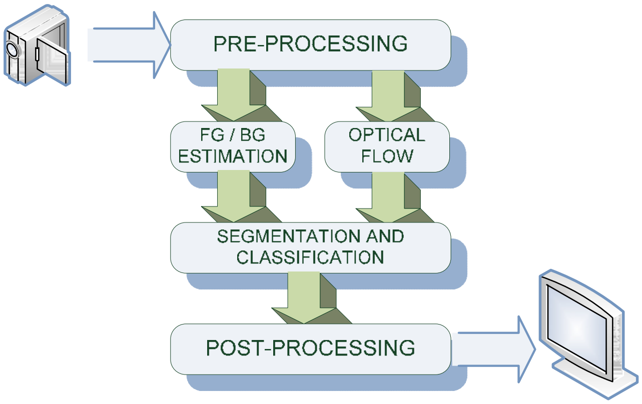

Figure 1. Block diagram

3.2 Pre-processing

The pre-processing stage of the system uses an event probability model to set the likelihood of events in different zones of an image. There are certain zones in the video which have a higher probability of certain events occurring compared to others. The two events selected for detection are related to the motion of objects, so it is natural to analyze the motion in specific areas where movements are expected and discard the remaining ones. The selected region where motion is expected is called the region of interest (ROI). In the case of *OpposingFlow*, the ROI is manually selected to cover only the area of the doors because that's the only region of the video where the event is defined [12], in the case of *PersonRuns*, the ROI discards the areas where static objects (desks, boxes, etc) may block the detection of objects in motion in the segmentation stage..

3.3 Foreground/Background Estimation

The FG/BG estimation block receives the grayscale information contained in the ROI to separate the foreground from the background for further analysis of information contained in the foreground. The output consists of binary frames where zero (0) corresponds to background pixels and one (1) represents foreground pixels. The Approximate Median method was used as a medium complexity approach. It is easy to implement and is more robust than the Frame Difference method. It offers performance near what we can achieve with higher-complexity methods, therefore is less sensitive to noise.

3.4 Optical Flow

The Optical flow block also receives the grayscale information contained in the ROI to evaluate the optical flow for each pixel in every frame. The output is expressed as a complex number where the real part represents the optical flow value in the x axis and the imaginary part represents the optical flow value in the y axis. The Lucas-Kanade algorithm was used to perform the optical flow calculation because of its acceptable quality and low computational complexity.

3.5 Segmentation and Classification

Using the information provided by the FG/BG estimation block, the segmentation and classification block performs the segmentation of moving objects (blobs) identified in the foreground. The blobs are labeled to be properly identified and counted in every frame (Figure 2). Also, properties, such as area, centroid and bounding box, are measured for each blob for further processing purposes.

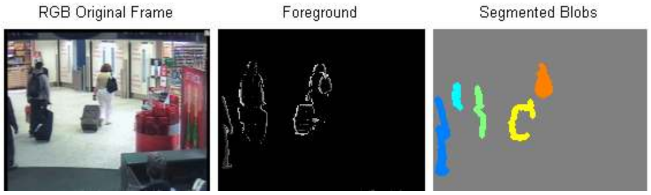

Figure 2. Segmentation

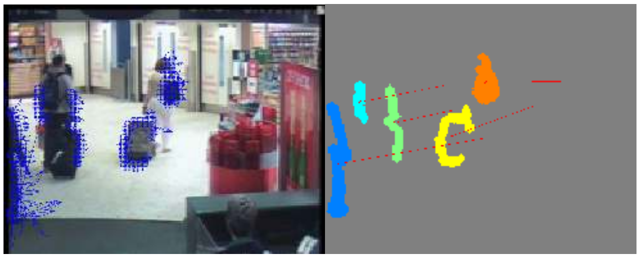

Figure 3. Optical flow

Every blob has to be analyzed to determine the speed and direction of its trajectory. To achieve this task, every different blob will serve as a mask to calculate the average vector of the optical flow vectors within the blob area (Figure 3, left); the result is a unique vector for every blob whose magnitude represents how fast the blob is moving in the frame and the angle represents its direction of displacement (Figure 3, right).

At this point, every blob has information that needs to be classified according to the event detection definition in order to resolve whether the blob is a candidate matching the criteria or not. To accomplish this goal, the angle of the blob motion vector has to be checked to determine if the value is inside the interval in which the *OpposingFlow* event is defined in the case of *OpposingFlow* event detection (Figure 4, shaded area, green). In the case of *PersonRuns* event detection, the magnitude of the blob motion vector has to be checked to determine if the value exceeds a certain threshold (Figure 4, dashed line, red), showing higher motion activity which is a direct consequence of running events. Blob motion vectors outside these conditions are dismissed and the ones matching the conditions are clustered for a further process to determine the existence of the desired event.

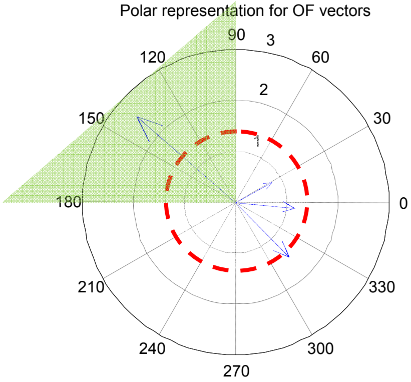

Figure 4. Blob motion vector selection

3.6 Post-processing

The last step in the system corresponds to the post-processing block which receives the clustered information from the previous block and creates a histogram with the number of optical flow vectors per frame (Figure 5, top). A convolution operation is performed between this data and a small constant window to act as a low pass filter to increase the confidence in detection under the assumption that when the event is happening, it is supposed to last during several contiguous frames (Figure 5, middle, blue line). Then, the convolution results are compared to a threshold, which is calculated empirically due to the depth perception according to the camera position, to determine the candidate frames where the event is happening, creating a new histogram where values equal to one (1) correspond to candidate frames and values equal to zero (0) correspond to non-candidate frames (Figure 5, middle, green).

Due to some possible discontinuities in the threshold stage, the presence of occlusions, noise and other factors in the classification of candidates, it may happen that the new histogram has non-continuous values for a detected event as well as isolated presence of candidate frames, which can make the final decision inaccurate leading to false event detections. To solve these issues, the new histogram is differentiated so we can extract the start frame and the end frame for the candidate events (Figure 5, bottom, blue). The starting point of a detected event is identified with a positive one (+1) and the ending point of the event is identified with a negative one (-1).

The final decision is made by analyzing the distances between start-end points as well as end-start points (*Min_dist* variable in Figures 7, 8, 9 and 10). Distance between start-end points is performed in order to discard detected events which fail to comply with the minimum duration required to tag the event. On the other hand, distance between end-start points is intended to identify cases where the event should be continuous, but it has small discontinuities leading to tag different instances of the same event detected.

Finally, the system is able to specify the start frame and end frame ef the detection where the event has been identified (Figure 5, bottom, red) as a result of suppression of intermediate detections as explained in the aforementioned step. This information is useful to perform some statistical analysis according to event annotations based on the ground truth. Moreover, the data is used to show the tagged event during playback, so as to act as an interface with the end user of the video surveillance system.

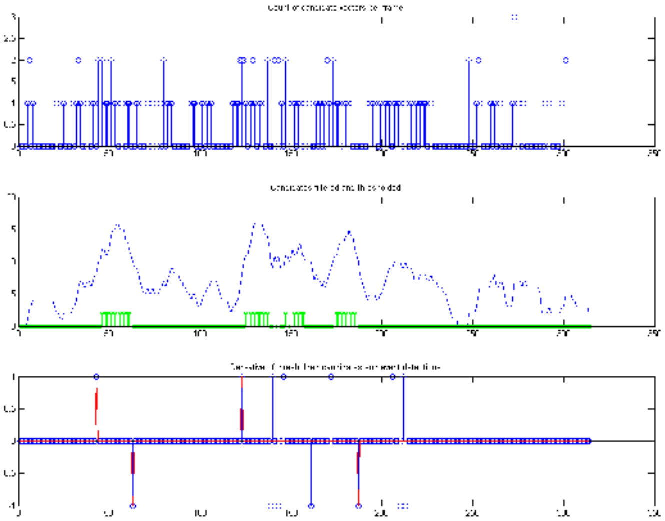

Figure 5. Histogram with optical flow vectors (top), optical flow vectors filtered and thresholded (middle), derivative of thresholded data showing start points and end points (bottom).

4. EXPERIMENTS AND RESULTS

The video clips used to test the program were extracted from the video database provided by TRECVid [7], where *OpposingFlow* and *PersonRuns* events have the same environmental conditions (Camera1). Every clip was extracted according to the ground-truth of detected events assuring the existence of only one event per clip. Additionally, every clip contains 100 frames before the event starts and another 100 frames after the event finishes. 33 video clips were used for *OpposingFlow* and 25 for *PersonRuns.*

The classification of detections (true and false detections) is calculated based on the information given by the ground truth. As every video clip was extracted in order to contain only one single event, and the video clips have extra frames before and after the event, it is expected to have the true detections during the window when the event is happening according to the ground truth annotation. Therefore, the false detections are going to take place when the detection is outside the event annotation, in other words, when there is a detection that belongs to the extra frames in the video.

Figure 6 shows an example where the system performed three different detections. Detection 1 is identified as a false detection, although it has some frames inside the event annotation window, most of the frames are outside. Detection 2 is identified as a true detection because it is entirely inside the event annotation window. Finally, detection 3 is identified as a false detection because it is completely outside the event annotation window, in other words, it belongs to the extra frames in the video where there are not expected detections.

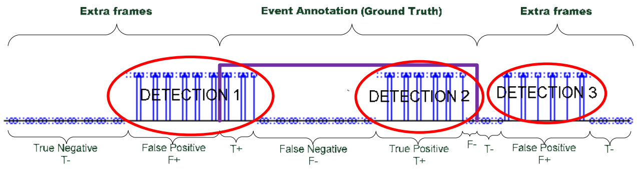

Figure 6. Classification of detections

Table 1. Detections

Event	False Detections	True Detections
OpposingFlow	15	27
PersonRuns	18	18

Results show that the number of true detections is acceptable for event detection in *OpposingFlow* and *PersonRuns*, having a precision of 64.28% at recall of 81.8% and precision of 50% at recall of 81.8% respectively. It is very important to mention that these values depend on the method used to analyze the performance of the system (video clips with only one event) and they will decrease substantially when adapting the system to process an entire video with several frames, as the original videos provided by TRECVid, because there is a high probability to have many more false detections in longer videos. The aforementioned information is supported by a test performed in a different set of 60 video clips, extracted randomly from the TRECVid video dataset where each video clip had 301 frames; the result was 26 false detections.

Results from TRECVid 2008 reported precision values between 1.85% - 7.5% at recall values between 75% - 81% for OpposingFlow events, and precision values between 1.9% - 5.9% at recall values between 26% - 45% for PersonRuns events [3, 7, 8, 10, 15]. The difference between some of the values compared to TRECVid is because TRECVid reports include all the results from the five different cameras in the dataset and this work used only videos from one camera, moreover, the small video clips used have reduced number of false detections compared with a system analyzing larger videos.

Recall and precision for *OpossingFlow* (Figures 7-8) are evaluated using different values for the threshold *k* and the distance between events *min_dist* when performing the filtering stage. It is noticeable the recall is dependent on the threshold and the precision is dependent on the minimum distance. As the value for the threshold is incremented, the recall tends to decrease because the system is dropping detections below the threshold. On the other hand, as the minimum distance increases, the precision tends to increase but there is not a strong dependence with the value used for thresholding.

Recall and precision for *PersonRuns* (Figures 9-10) are evaluated using different values for the threshold *magnitude* used when evaluating the resultant optical flow vector, and the distance between events *min_dist* used when performing the filtering stage.

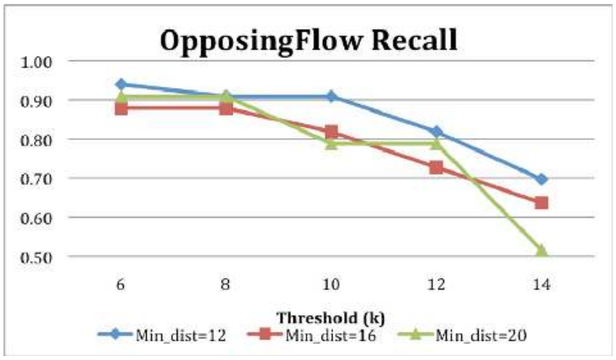

Figure 7. Recall for OpposingFlow event

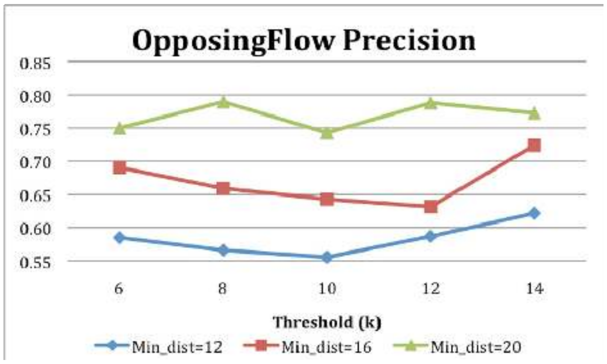

Figure 8. Precision for OpposingFlow event

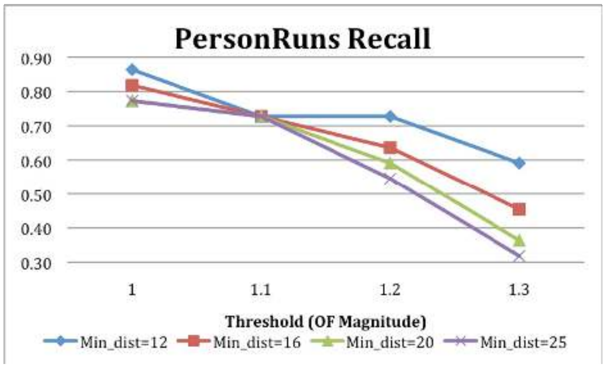

Figure 9. Recall for PersonRuns event

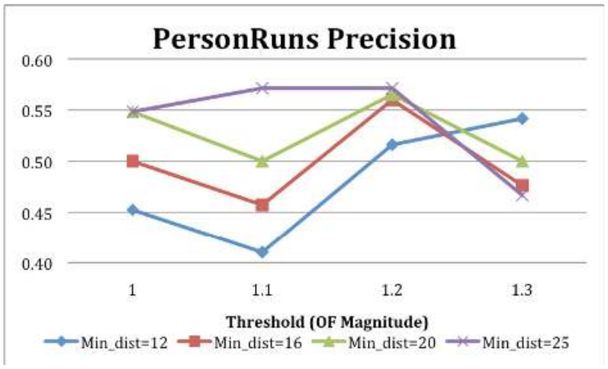

Figure 10. Precision for PersonRuns event

5. CONCLUSION

The detection of OpposingFlow and PersonRuns events was tested and analyzed as an approach to detect events based on motion analysis. Use of grayscale images was useful for simplicity in the implementation and definition of zones where events should occur proved to be effective to discard undesired detections. After the analysis of the background subtraction and optical flow techniques, Approximate Median and Lucas-Kanade were chosen respectively. The segmentation of blobs gave shapes far from ideal representation of semantic objects, but they were useful to identify and determine the magnitude and direction of objects in motion using the correspondent optical flow vectors within to the blob's area. Magnitude of blob motion vectors was the key to determine objects running, and the angle of blob motion vectors was the key to determine objects moving in specific directions. Detected blobs matching the criteria were counted for every frame and then filtered to determine whether the event was or not present. Results from a test performed on a set of videos give a precision of 64.28% at 81.8% of recall for OpposingFlow events and a precision of 50% at 81.8% of recall for PersonRuns events. TRECVid 2008 participants reported precision values between 1.85% - 7.5% at recall values between 75% - 81% for OpposingFlow events, and precision values between 1.9% - 5.9% at recall values between 26% - 45% for PersonRuns events.

6. REFERENCES

[1] Barron, J., Fleet, D. and Beauchemin, S. 1994. "Performance of optical flow techniques," *Int. J. Comput.Vis.*, vol. 12, no. 1, (1994), 42–77.

[2] Chen, T. P., Haussecker, H., Bovyrin, A., Belenov, R., Rodyushkin, K. and Kuranov, A. 2005. "Computer Vision Workload Analysis: Case Study of Video Surveillance Systems", Intel Technology Journal, 9(12), (May 2005).

[3] Chmelar, P., Beran, V. and Herout, A. 2008. "Brno University of Technology at TRECVid 2008". TRECVid 2008.

[4] Dikmen, M., Ning, H. and Dennis J. 2008. "Surveillance Event Detection", TRECVid 2008

[5] Hu, W., Tan, T., Wang, L. and Maybank, S. 2004. "Survey on Visual Surveillance of Object Motion and Behaviors", *IEEE Transactions on systems and cybernetics*, Vol. 34, No. 3, (August 2004), 334-353.

[6] Piccardi, M. 2004. "Background subtraction techniques: a review," Systems, Man and Cybernetics, 2004 IEEE International Conference on , vol.4, (October 2004), 3099-3104.

[7] Smeaton, A. F., Over, P., and Kraaij, W. 2006. Evaluation campaigns and TRECVid. In Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval (Santa Barbara, California, USA, October 26 - 27, 2006). MIR '06. ACM Press, New York, NY, 321-330. http://doi.acm.org/10.1145/1178677.1178722.

[8] Stergiou, A., Pnevmatikakis, A. and Polymenakos, L. 2008. "Detecting Single-Actor Events in Video Streams for TRECVid 2008", Athens Information Technology and Aalborg University, CTiF, TRECVid 2008.

[9] Sun, H. Z., Feng, T. and Tan, T. N. 2000. "Robust extraction of moving objects from image sequences," in Proc. Asian Conf. Computer Vision, Taiwan, R.O.C., (2000), 961– 964.

[10] Taj, M., Daniyal, F. and Cavallaro, A. 2008. "Event analysis on TRECVid 2008 LondonGatwick dataset". Queen Mary, University of London. TRECVid 2008.

[11] Tang, S., Li, J., Li, M. and Xie, C. 2008. "TRECVID 2008 Participation by MCG-ICT-CAS", Chinese Academy of Sciences. TRECVid 2008.

[12] "TRECVid 2009 Event Annotation Guidelines". Version 1.0. June 2, 2009.DOI= http://itl.nist.gov/iad/mig/tests/trecvid/2009/doc/TRECVid09_Event_Annotation_Guidelines_v1.0.pdf

[13] Wilkins, P., Kelly, P. and Ó Conaire, C. 2008. "Dublin City University at TRECVID 2008", Dublin City University. TRECVid 2008.

[14] Xue, X., Zhang, W., Guo, Y. and Lu, H. 2008. "Fudan University at TRECVID 2008". Fudan University. TRECVid 2008.

[15] Yarlagadda, P., Garg, K. and Guler, S. 2008. "Intuvision Event Detection System for TRECVID 2008". IntuVision, Inc. TRECVid 2008.

[16] Yokoi, K., Nakai, H. and Sato, T. 2008. "Toshiba at TRECVID 2008: Surveillance Event Detection Task". Corporate Research and Development Center, TOSHIBA Corporation. TRECVid 2008.