ESUR: A SYSTEM FOR EVENTS DETECTION IN SURVEILLANCE VIDEO*

Yaowei Wang[1, 2], Yonghong Tian[1], Lingyu Duan[1], Zhipeng Hu[1, 3], and Guochen Jia[1]

[1] National Engineering Laboratory for Video Technology, Peking University, Beijing 100871, China
[2] Department of Electronic Engineering, Beijing Institute of Technology, Beijing 100081, China
[3] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
{ywwang, yhtian, lyduan, zphu, gcjia}@jdl.ac.cn

ABSTRACT

In this paper, we present our eSur (Event detection system on SURveillance video) system, which is derived from TRECVID'09 surveillance tasks. Currently, eSur attempts to detect two categories of events: 1) single-actor events (i.e., *PersonRuns* and *ElevatorNoEntry*) irrespective of any interaction between individuals, and 2) pair-activity events (i.e., *PeopleMeet*, *PeopleSplitUp*, and *Embrace*) involves more than one individual. eSur consists of three major stages, i.e., preprocessing, event classification, and post-processing. The preprocessing involves view classification, background subtraction, head-shoulder detection, human body detection and object tracking. Event classification fuses One-vs.-All SVM and rule-based classifiers to identify single-actor and pair-activity events in an ensemble way. To reduce false alarms, we introduce prior knowledge into the post-processing, and in particular, we apply a so-called event merging process over TRECVID dataset. Extensive experiments have been performed over TRECVid'08 and '09 ED data corpus involving in total 144 hours surveillance video of London Gatwick airport. According to the TRECVid-ED formal evaluation, our prototype has yielded fairly promising results over TRECVid'09 dataset, with top Act.DCR of 1.023, 1.025, 1.02, and 0.334 for *PeopleMeet*, *PeopleSplitUp*, *Embrace*, and *ElevatorNoEntry*, respectively.

Index Terms— Surveillance, events detection, TRECVid

1. INTRODUCTION

Video cameras have been widely deployed in surveillance, for instance, more than 5000 cameras were used for the 2009 United States presidential inauguration. Unfortunately, most of the CCTV (closed-circuit television) monitoring systems are not smart enough to autonomously recognize or predict abnormal events or actions. Especially when a large number of cameras are installed, much surveillance video data would be generated, so that it is extremely hard to discover abnormalities by people looking. So many research efforts have been devoted to video analysis technologies in the field of surveillance video. In TRECVid 2008, NIST initiated an Event Detection (TRECVid-ED for short) evaluation campaign, which is committed to evaluate systems that can detect instances of a variety of observable events in the airport surveillance domain. The challenges of TRECVid-ED tasks come from its large-scale corpus, namely, 144 hours videos from five different cameras in a real CCTV system at Gatwick airport of London, which incurs many disturbing factors such as variable illuminations, variable scales, clutter backgrounds and frequent occlusions between objects. This is different from other existing empirical surveillance datasets such as CAVIAR [8] and PETS [9].

Our eSur prototype system [10] is developed to meet the TRECVid-ED requirements of automatically discovering predefined events, i.e, retrospective events task of TRECVid ED 2009 [1]. For the ten events predefined in TRECVid ED 2009, a very good result of *OpposingFlow* (i.e. ACT.DCR is 0.251) was reported in TRECVid 2008. In the events of *CellToEar, Pointing, objectobjectPut and TakePicture*, the detection of body parts (i.e. arms and hands) movement are necessary, whereas most of the actions are too fast (say 3-10 frames) to detect accurately and the body parts are too small or heavily occluded in camera views on TREVid'09 dataset. Therefore, eSur currently focus on the detection of five events (i.e. *PersonRuns*, *ElevatorNoEntry*, *PeopleMeet*, *PeopleSplitUp* and *Embrace*) out of ten predefined events in TRECVid 2009 ED task. The selected five events involve obvious human body motion instead of body parts movement. We classify the five events into two categories: single-actor events (i.e. *PeopleMeet*, *PeopleSpilitUp*, and *Embrace*) and pair-activity events (*PersonRuns* and *ElevatorNoEntry*), and apply different approaches to each event category.

The remainder of the paper is organized as follows. In section 2, we present ESUR system framework. Our events detection approach is described in section 3. Experimental results are given in section 4. Finally, we conclude this paper.

*The work is supported by grants from the Chinese National Natural Science Foundation under contract No. 60973055 and No. 90820003, National Basic Research Program of China under contract No. 2009CB320906, and Fok Ying Dong Education Foundation under contract No. 122008.

2. ESUR SYSTEM FRAMEWORK

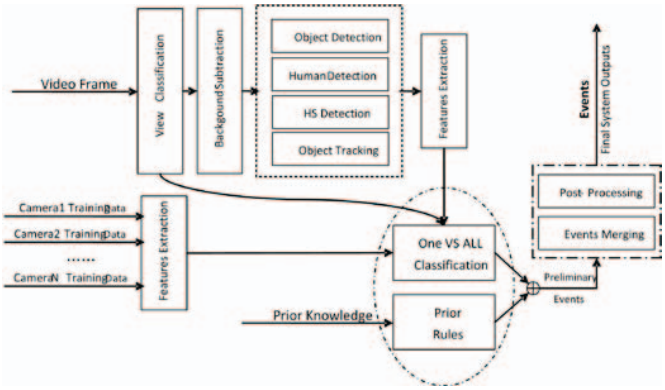

Fig.1 Framework of eSur system (HS means Head-Shoulder)

In our eSur system, design and implementation efforts have been made in three stages, namely, preprocessing, event classification, and post-processing, as illustrated in Fig.1.

The preprocessing stage consists of five elementary steps: view classification, background subtraction, human or object detection, object tracking and visual features extraction. Because several events occur in a specific view (e.g., *ElevatorNoEntry* only happens in those views including elevators), it is helpful to figure out which camera view an input video come from. We can easily identify the view based on its background information. Also, background subtraction can be used to accelerate the detection process. Background models are constructed with a PCA method [3]. Within foreground regions, detection is carried out to find out human head-and-shoulder contour, human body and other important objects (e.g., elevators). We combine human body and head -shoulder detection results to derive the final results of human detection. We use HOG (Histogram of gradient) [4] as features and employ a cascaded framework [5] to improve human detection. Subsequently, object tracking is applied. On the basis of detection results, an online-boosting method [6] [7] is employed to track moving objects. Finally, we extract features such as position, velocity, motion direction and time span for each object. Some high-level features are generated as follows.

$$dist(obj_1, obj_2) = \left\| pos_{obj1} - pos_{obj2} \right\| \quad (1)$$

$$relCode(obj_1, obj_2) = \begin{cases} 0 & if \ \ 0 < |\theta_1 - \theta_2| < \pi/6 \\ 1 & if \ \ 5\pi/6 < |\theta_1 - \theta_2| < \pi \\ 2 & \text{otherwise} \end{cases} \quad (2)$$

$$cotime(obj_1, obj_2) = min(t_{end}(obj_1), t_{end}(obj_2)) - max(t_{start}(obj_1), t_{start}(obj_2)) \quad (3)$$

The distance between two objects is measured by their Euclidean distance, where *pos* is an object's centeroid coordinate. The relativity of two objects' motion directions is described in equation (2), where θ is the angle between an object's motion direction and horizontal axis. In equation (3), we count the co-existing time span of two objects, where t is the frame number of an object entering or exiting a camera view.

In the second stage, we classify events by fusing one vs. all SVM and prior rules. Ryan Rifkin [2] showed that one-vs-all scheme tends to achieve better performance than one-for-all scheme. So we employ One-vs.-All SVM to identify different events. We train a classifier for each of the five selected events. Also we use some prior rules to facilitate decision making in the system. To identify an event, we empirically use a sliding window with 12 consecutive frames.

In the training corpus, we manually label the corresponding objects for each event with bounding boxes. We equally divided the training data to ten subsets, and tuned the classifier parameters by a ten-fold cross-validation.

For the preliminary results after the second stage contain many false alarms, two post-processing processes are adopted to decrease the false alarms. A so-called "Events Merging" process deals with those events occurring in an overlapping time span. For example, sometimes we cannot distinguish person from other moving objects with the human detection algorithms. When a person is running with a suitcase, our system will probably detect two *PersonRuns* events at the same time. To remove such false alarms, we merge these concurrent similar events by the involved objects' spatio-temporal relationships. The other post-processing processes are to address several other forms of false alarms, which may apply different rules. For example, at the end of a *PeopleMeet* event, two persons should not move.

3. EVENTS DETECTION

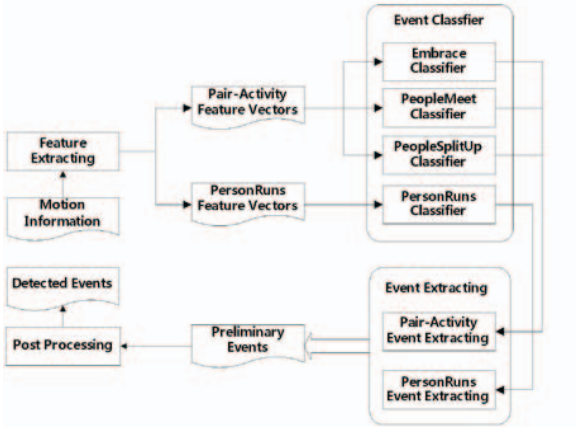

Fig.2 Flow diagram for detecting four events of PeopleMeet, PeopleSplitUp, Embrace and PersonRun

According to the TRECVid ED evaluation plan 2009 [1], an event is identified by its interval with a start frame number and an end frame number. We first attempt to detect key frames that could signify the happening of an event. Then,

preliminary results are refined by searching forward or backward from such key frames. According to TRECVID events definition, there is no visually significant beginning for "*Embrace*" and "*PeopleMeet*" events, but the pair-persons are very close to each other at the end of the event. In other words, key frames actually happen at the end of the event. In contrast, key frames would be located at the beginning of "*PeopleSplitUp*" events.

3.1. Pair-activity Events

Pair-activity events involve the interaction of at least two persons. This kind of event detection is addressed as a classification problem. We first treat the events of *PeopleMeet*, *PeopleSplitUp* and *Embrace* as one category and employ One-vs.-All SVM to classify them from the others. Each kind of three events is identified by object motion patterns.

Given two detected peoples, their distance, coexisting duration and motion direction's correlation are combined to form a feature vector, which are generated in a sliding window of twelve consecutive frames. One-vs.-All SVM is trained to classify these three events.

To distinguish "*Embrace*" or "*PeopleMeet*", we apply a backward search to locate the beginning of an event. In contrast, forward search is used to detect "*PeopleSplitUp*". Finally, we refine the results with post-processing. A set of heuristic rules are used. For instance, if two peoples' distance at the end of an event is greater than a threshold for "*PeopleMeet*" and "*Embrace*", or their distance at the beginning of an event is greater than a threshold for "*PeopleSplitUp*", a preliminary detection would be considered as a false alarm.

3.2. Single-actor Events

Speed and direction of movements are key characteristics of "*PersonRuns*". It is observed that a running person have a larger velocity than others, and the motion direction would not change dramatically. According to the feature statistics, we make use of the constraints of object position and motion direction. A SVM classifier is trained to identify *PersonRuns*. In the camera setting of TRECVid dataset, running people always move from left-bottom to top in the view of camera one. So, by the post-processing we may remove many false alarms caused by tracking drifting.

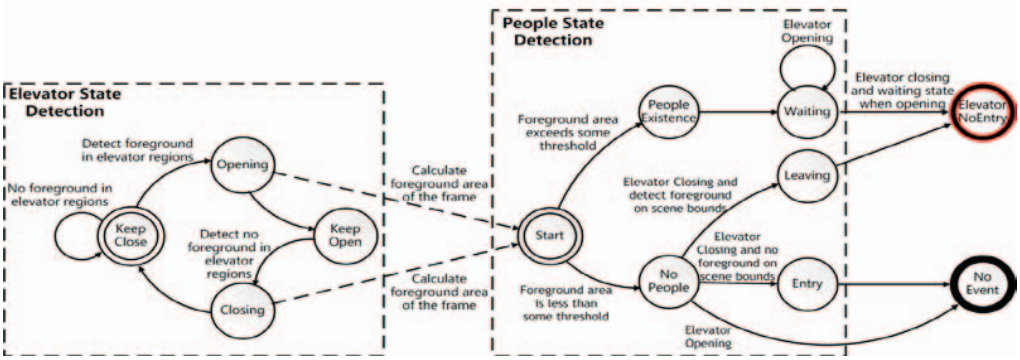

Fig.3 An automaton for detecting ElevatorNoEntry

ElevatorNoEntry is defined as "elevator doors open with a person waiting in front of them, but the person does not get in before the doors close". As illustrated in Fig.3, we introduce an automaton to model the detection process of *ElevatorNoEntry*. As there is no elevator in the view of cameras one, two and five over TRECVid'09, we execute the automaton in the views of camera 3 or camera 4.

As elevators' position are fixed, our system can easily locate elevators. When an elevator door is closed, the elevator region is labeled as background. And when the door is moving, the elevator region is detected as foreground. Thus, we can identify each elevator's states (open or closed) by using background subtraction.

The foreground area is related to the number of persons in front of an elevator. We detect the *ElevatorNoEntry* event according to the elevators' states and the size of foreground area. To distinguish whether the number of persons around an elevator is changed, we simply compute the ratio of detected foreground regions before and after an elevator open-and-close action. When the ratio is smaller than a threshold, it is probable that some people has entered the elevator, and the frame interval is labeled as a potential event of *ElevatorEntry*. Furthermore, we have to determine how and where people disappear, namely, either entering the elevator or leaving the scene through the view's boundary. The ratio change of foreground area of the frame's boundary region is used to check whether any one leaves.

4. EXPERIMENTAL RESULTS

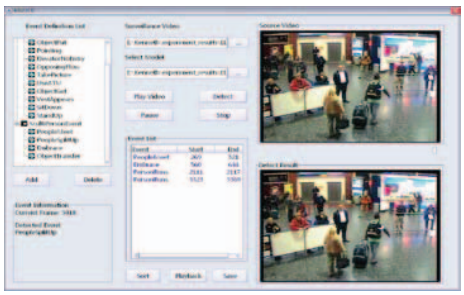

Fig.4 the interface of eSur

The interface of eSur prototype system is shown in Fig.4. User can flexibly load existing event model files. People who participate in the event are marked with a red rectangle in the "Result window" (at the right-bottom). It is easy to visually validate whether a detection is correct. Each detection result, which marked with a start and an end frame ,is listed in the middle of the interface.

Table 1 Comparison between the reported best results of TRECVid 2008 and our best results of TRECVid 2009

Event	Our Best	Best 2008	Imp.
PeopleMeet	**1.023**	1.337	-0.314
PeopleSplitUp	**1.025**	4.856	-3.831
Embrace	**1.020**	1.271	-0.251
ElevatorNoEntry	**0.334**	N/A	-
PersonRuns	1.068	0.989	+0.079

Table 2 Comparison between the reported best results and our best results on TRECVid 2008 corpus

Event	Our Best	Best 2008	Imp.
PeopleMeet	**1.245**	1.337	-0.092
PeopleSplitUp	**1.976**	4.856	-2.880
Embrace	**1.208**	1.271	-0.063
ElevatorNoEntry	**0.130**	N/A	-
PersonRuns	1.249	0.989	+0.260

According to the TRECVid-ED formal evaluation, our system has achieved promising results over TRECVid'09 dataset, with top Act.DCR [1] of 1.023, 1.025, 1.02, and 0.334 for *PeopleMeet*, *PeopleSplitUp*, *Embrace*, and *ElevatorNoEntry*, respectively, as listed in Table 1. Act.DCR is defined by the following equation.

$$NDCR = P_{Miss} + Beta \times R_{FA} = \frac{N_{Miss}}{N_{\text{targ}}} + \frac{Cost_{FA}}{Cost_{Miss} \times R_{\text{Target}}} \times \frac{N_{FA}}{T_{source}}$$

Where N_{miss} is the missed number of an event, $N_{t\text{arg}}$ is the system outputs number of an event, N_{FA} is the number of false alarms, T_{source} is the frame number of an input video, and $Cost_{FA} = 1$, $Cost_{miss} = 10$, $R_{t\text{arg}et} = 20$. A smaller Act.DCR means better performance.

As listed in Table 1, our best results of *PeopleMeet*, *PeopleSplitUp*, *Embrace*, and *ElevatorNoEntry* outperform the reported best results of TRECVid 2008. And in table 2, it is indicated that our best results of four events are promising over TRECVid 2008. For *PersonRun*, the 2009's reported best ACT.DCR is 0.971, and our lower performance is mainly due to a bit serious tracking drifts.

There are several open problems yet. Although we have greatly reduced the false alarms by post-processing, a considerable number of correct detections are also removed. Overall, the precision and recall rates at the system level are too low. So we have to seek a tradeoff between reducing false alarms and improving recall. In addition, the detection precision is to be improved too. Another problem is heavy computation. Six computers (four 8-core, one 16-core and one 4-core workstations) run about one week to detect and track objects on the 144 hours corpus.

Also, there are still many false alarms and missing events. "*Embrace*" would be wrongly detected when two meeting people are occluded by each other. "*PeopleMeet*" would be mistakenly labeled when one person walks into the view and stand behind someone. On the other hand, the event would be missed when two meeting people can't be detected correctly (for people occlusion or near the boundary of a view). In practice, when other people pass by, a "*PeopleSplitUp*" would be detected by mistake. For "*PersonRun*", most of the children's runnings for fun are missed, and our prototype is deficient in distinguishing "run" from "fast walk".

5. CONCLUSION

We have reported our design and implementation efforts on TRECVid ED task. This benchmarking activity has revealed the research and practice challenges of robustly detecting events in real-world surveillance video. Comparatively, the promising results have validated our prototype system.

Although our reported ACT.DCR is better, the low overall precision limits the application of vision based approaches in large-scale surveillance datasets. Basically, our successful experiences in TRECVid'09 show our system framework is feasible to some extent. However, to reach desirable performance, we have extensive research and engineering work to do in terms of algorithms and system design.

REFERENCES

[1] National Institute of Standards and Technology (NIST), "TRECVID2009EvaluationforSurveillanceEventDetection," http://www.nist.gov/speech/tests/trecvid/2009/, 2009.

[2] Ryan Rifkin and Aldebaro Klautau, "In Defense of One-Vs-All Classification", *Journal of Machine Learning Research*, pp.101-141, January, 2004

[3] Oliver, N.M., Rosario, B., Pentland, A.P., "A Bayesian computer vision system for modeling human interactions", *IEEE Transactions PAMI*, 22(8), pp.831- 843, 2000.

[4] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, Shai Avidan, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients". *CVPR* (2): pp. 1491-1498, 2006

[5] Wu B., Nevatia R. Detection and Tracking of Multiple, "Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors", *IJCV (75)*, No. 2, pp. 247-266, 2007.

[6] H. Grabner, T.T. Nguyen, B. Gruber, H. Bischof, "On-line boosting-based car detection from arial images", *ISPRS Journal of Photogrammetry & Remote Sencing*, 63(3), pp.382-396, 2007

[7] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses", *In ECCV'08,* volume 2, pp. 788–801, 2008

[8] INRIA, "CAVIAR: Context Aware Vision using Image-based Active Recognition", http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[9] Desurmont X., etc., "Performance evaluation of frequent events detection systems," *Proc. 9th IEEE Int'l. Workshop Performance Evaluation of Tracking and Surveillance*, 2006.

[10] Zhipeng Hu, Guangnan Ye, Guochen Jia ,et al. PKU@ TRECVID2009: Single-Actor and Pair-Activity Event Detection in Surveillance Video. http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/ pku-idm.pdf.