# Shanghai Jiao Tong University participation in high-level feature extraction and surveillance event detection at TRECVID 2009

Xiaokang Yang, Yi Xu, Rui Zhang,
Erkang Chen, Qing Yan, Bo Xiao, Zhou Yu, Ning Li, Zuo Huang
Cong Zhang, Xiaolin Chen, Anwen Liu, Zhenfei Chu, Kai Guo, Jun Huang
Institute of Image Communication and Information Processing,
Shanghai Jiao Tong University, Shanghai 200240, China

## Abstract

*In this paper, we describe our participation for high-level feature extraction, automatic search and surveillance event detection at TRECVID 2009 evaluation.*

*In high-level feature extraction, we establish a common feature set for all the predefined concepts, including global features and local features extracted from the keyframes. For the concepts related to person activity, space--time interest points are also used. Detection of ROI and Faces is needed for some special concepts, such as playing instrument, female face close-up. Classifiers are trained using these features and linear weighted fusion of the classification results are utilized as the baseline. Specifically, simple average fusion can work pretty well. Further, ASR and IB re-ranking are used to improve the overall performance. We submitted the following six runs:*

- *A_SJTU_ICIP_Lab317_1: Average fusion of classification results with global features and local features used, SVM classifiers are trained on TRECVID2009 development data*
- *A_SJTU_ICIP_Lab317_2: Linear weighted fusion of classification results with global and local features used, SVM classifiers are trained on TRECVID2009 development data*
- *A_SJTU_ICIP_Lab317_3: Max of RUN1 and RUN2, and re-rank on ASR*
- *A_SJTU_ICIP_Lab317_4: Max of RUN1 and RUN2, and re-rank on IB re-ranking*
- *A_SJTU_ICIP_Lab317_5: Based on the result of RUN3, combine ASR and IB re-ranking*
- *A_SJTU_ICIP_Lab317_6: Max of all runs*

*In Event detection, trajectory features obtained from human tracking and optical flow computation, local appearance and shape features are employed in event model training. With regard to particular event detection tasks, several detection rules are tested using HMM models, boosted classifiers, matching and heuristic settings. We provide the detection results of eight event tasks out of 10 required events for performance evaluation.*

- *SJTU_2009_retroED_EVAL09_ENG_s-camera_p-baseline_1: Event detection based on human tracking, motion detection and gesture recognition*

# 1 High-level Feature Extraction

## 1.1 Overview

In TRECVID2009, we explore several novel technologies to help detect high-level concepts. We divide all the 20 concepts into 3 parts, as concepts on object and scene, person action, and face detection. We extract different features to adapt to different concept detection tasks.

There are four main steps in our framework, as shown in Fig. 1:

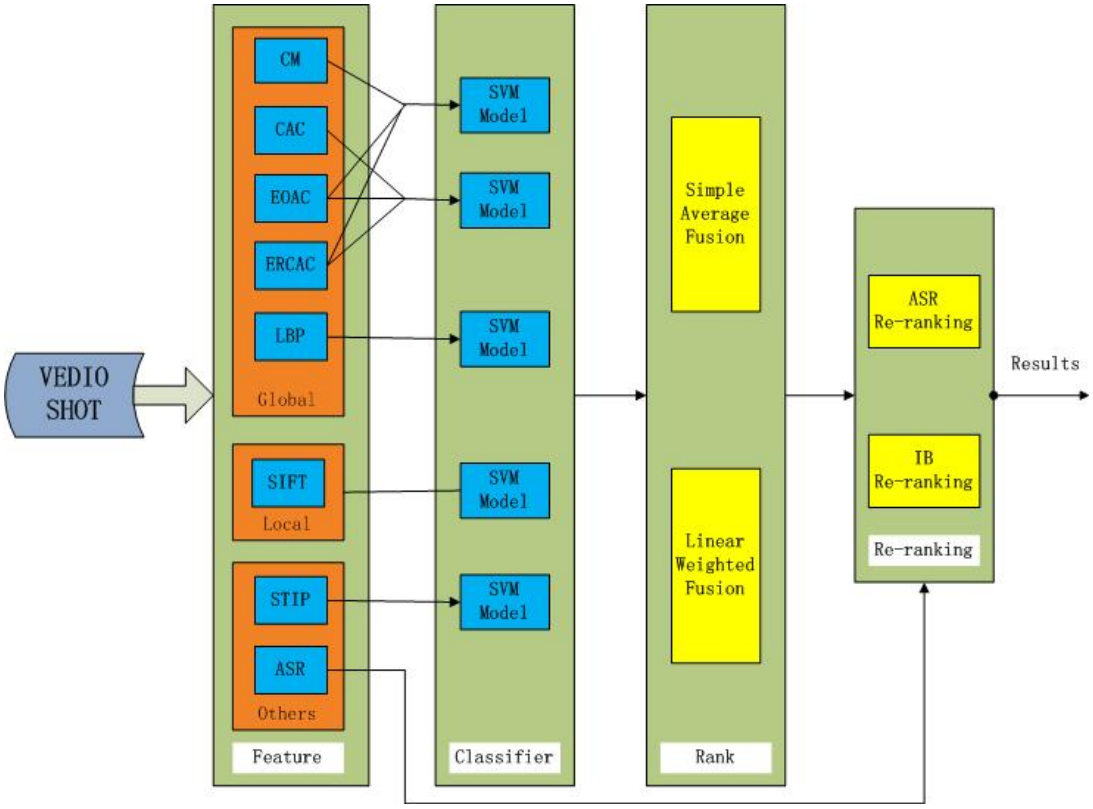



Figure 1 High-level feature extraction framework

- Low level feature extraction: We extract several low level features, including global features, local features and other particular features. As for global features, there are two kinds of color features(CM: Color Moment, CAC: Color Auto-Correlograms), two kinds of complex features(EOAC: Edge Orientation Auto-Correlograms, ERCAC: Edge Region Color Auto-correlogram) and LBP(Local Binary Patterns) features. The local features mainly used is SIFT features, which are described as a bag-of-visual-words (BoWs). In the context of concept detection about person activity, we use Space-Time Interest Points(STIP).
- Model: We adopted Support Vector Machines [1] as our classification method, training the individual SVM classifier for each low-level feature based on valid cross database learning on TRECVID2009 development data.
- Ranking: Simple average fusion and linear weighted fusion are used to combine multiple ranking results obtained using all the trained models.
- Re-ranking: We extracted textual information based on automatic speech recognition

(ASR) and information bottle (IB) principle. By adding the positive textual relevant factor to the previous ranking result, we obtained the re-ranking results.

## 1.2 Low level feature extraction

### 1.2.1 Global feature

We establish five baseline low-level features, out of which 4 types of features had been used in our Trecvid2007 system, including two kinds of color features (CAC, 166 dim; CM, 225 dim, 5*5grids) , one texture features( Local Binary Pattern(LBP), 531 dim, 3*3 grids)), one shape Edge Orientation Auto-Correlograms(EOAC, 144dim). We also propose a novel type of regional feature, which is called edge region color Auto-correlogram (ERCAC, 166 dim). It aims to characterize image using the color and shape features jointly, capturing both color distribution of image and spatial correlation of edge points.

### 1.2.2 Local feature

Besides global features, we also extract local features(i.e. SIFT) from keyframes of the detected shot. We develop SIFT features from integrated Difference of Gaussian (DOG) interest point[10]. Thus, the keyframe can be described as a bag-of-visual-words (BoWs)[2], where k-means is adopted to cluster the local features and each cluster is represented as a visual word. Accordingly, each keyframe is described by a visual dictionary or a vocabulary. SVM can then be used to classify the concept of each shot based on the histogram of the vocabulary,.

The most important issue is how to determine the size of the visual vocabulary, which would greatly influence the performance of BoWs. A smaller vocabulary might not contain the whole content of the keyframe, while larger vocabulary should be a waste of computer performance, and much redundant information is not preferred. We have conducted a large number of experiments over TRECVID2009 development dataset by choosing vocabularies of different sizes. All the detections resulted from different vocabularies are fused to get a stable result.

For some special concepts like traffic-intersection, we use pyramid histogram of word (PHOW) to improve the overall detection performance. PHOW divides the region of interest in keyframe into four parts, and combines the four histograms with the original histogram. Thus, the resulting histogram would contain more spatial information. Good performance is expected on the scene concept.

### 1.2.3 Space-Time Interest Points

For six concepts of human activity, STIP[3] computes locations and descriptors for space-time interest points in video. The detector is the extension of Harris operator in space-time domain. The descriptors HOG (Histograms of Oriented Gradients) are computed for the volumetric video slices around the detected space-time interest points. In the experiments, we directly process the whole video sequences instead of keyframes using the STIP.

### 1.2.4 Special feature for some tasks

- ROI feature

For the concepts about human activity, extraction of Region Of Interest(ROI)[4] is needed. In

order to locate people's body parts, an edge-based deformable model is matched to the keyframe. We use the conditional random field(CRF) to obtain such deformable models. Finally, Pyramid Histogram of Oriented Gradients (PHOG) is extracted over ROI as features.

- Face feature

For concept of female human face close-up, we first extracted skin-color image regions and then use a Haar-like feature to detect faces. Only those images in which human face are detected is chosen to be training data. Then we rescale face regions to the same size and extract a LBP feature. Finally, we build a SVM classifier to distinguish male face and female face. It is noted that our female human face detection scheme consists of two steps: human face detection and male/female face discrimination. The weakness is the final result depends a lot on the precision of the face detection.

### 1.3 Re-ranking based on ASR and IB

- Re-ranking based on ASR

ASR is used to improve the rank list by adding the textual information. Through analyzing the ASR information of training data, we extract several most relevant keywords for each high-level feature. For all the shots in the ranked list, additional confidence scores are introduced for each shot by computing the similarity between the current shot and the keyword set.

- Re-ranking based on Information Bottleneck (IB) Principle

IB is implemented in the late fusion of ranking. Our method is inspired by the work of [5] and introduces modification in the step of ranking clusters. Firstly, the feature data is collected for IB clustering and then the posterior is calculated. Dependent on the estimated posteriors, extra confidences are added to the samples in different clusters. Finally a new ranked list is obtained by sorting the modified confidence. This method is applied to an augmented list which contains more than the 2000 samples in final submission.

### 1.4 Early Fusion and Late fusion

- Early Fusion

We propose a structure-based early fusion method to describe image content. Different features are extracted according to different levels of observation, and further concatenated to a single representation. For example, color information and sketch information can be both encoded in such a representation.

To achieve a better retrieval performance, our scheme adopts EOAC to represent contour of image. ERCAC is then taken as a joint distribution of color and shape. Finally, we add the global color features. For global color features, both CM and CAC have good retrieval performance. We combine features in two ways. One is a 535 dimension vector, which concatenated CM, ERCAC and EOAC features. The other is a 476 dimension vector, with CAC instead of CM.

- Late Fusion

We use two kinds of methods to fuse the probability outputs from the SVM models. For run1,

we simply fuse on average the probability trained on global features and local features. For run2, according to the performance of validation experiments on TRECVID2007 development data, linear wighted fusion is used to combine multiple ranking results.

### 1.5 Experimental results

We submitted 6 runs for high-level feature extraction as shown in Table 1, the result with the bold and red fonts shows the best run for each concept. Results show that Run1 is the best result of the six ones, which demonstrates that simple average fusion of baseline low-level features works pretty well. Yet linear weighted fusion based on the validation data, TRECVID2007 development data, don't improve the performance directly. The re-ranking methods, ASR and IB, are also helpful in detecting some concepts although the improvement is very limited. The methods tried in person's activity concept using Space-Time Interest Points and ROI detection, don't obtain a good performance as expected. Furthermore, the application of face detection really helps us in the concept of Female-human-face-closeup, yet still leaving some problems waiting to be solved.

As illustrated in Fig.2, we show our best run for each concept as compared with median and the best performance of all submitted runs.

Table 1 Six runs of our high-level feature extraction for each concept

| High-level features | RUN1 | RUN2 | RUN3 | RUN4 | RUN5 | RUN6 |
|---|---|---|---|---|---|---|
| Classroom | 0.063 | **0.064** | **0.064** | 0.05 | 0.063 | 0.05 |
| Chair | **0.033** | 0.032 | **0.033** | 0.02 | **0.033** | 0.023 |
| Infant | 0.009 | 0.017 | 0.017 | **0.027** | 0.009 | 0.013 |
| Traffic-intersection | **0.137** | 0.126 | 0.091 | 0.091 | 0.127 | 0.091 |
| Doorway | 0.091 | 0.091 | 0.091 | 0.072 | **0.093** | 0.072 |
| Airplane-flying | 0.003 | **0.004** | **0.004** | 0.003 | 0.003 | 0.003 |
| Playing-instrument | 0.002 | **0.004** | **0.004** | **0.004** | 0 | **0.004** |
| Bus | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 |
| Playing-sorrer | 0.07 | **0.078** | **0.078** | **0.078** | 0 | 0.005 |
| Cityscape | 0.108 | **0.11** | **0.11** | 0.099 | **0.11** | 0.099 |
| Riding-a-bicycle | 0.004 | **0.005** | **0.005** | **0.005** | 0 | **0.005** |
| Telephone | 0.008 | 0.008 | 0.006 | 0.006 | **0.012** | 0.006 |
| Person-eating | **0.013** | 0.009 | 0.009 | 0.009 | 0 | 0.009 |
| Demonstration _or_Protest | 0.006 | 0.006 | **0.007** | 0.003 | **0.007** | 0.003 |
| Hand | **0.079** | 0.073 | 0.073 | 0.058 | 0.073 | 0.058 |
| People-dancing | 0.005 | **0.006** | **0.006** | **0.006** | 0.001 | **0.006** |
| Nighttime | **0.141** | 0.128 | 0.099 | 0.098 | 0.131 | 0.099 |
| Boat_Ship | 0.122 | **0.128** | 0.109 | 0.111 | 0.088 | 0.109 |
| Female-human-face -closeup | 0.038 | 0.038 | 0.038 | 0.038 | 0.038 | 0.038 |
| Singing | 0.056 | 0.053 | **0.061** | **0.061** | 0.048 | **0.061** |

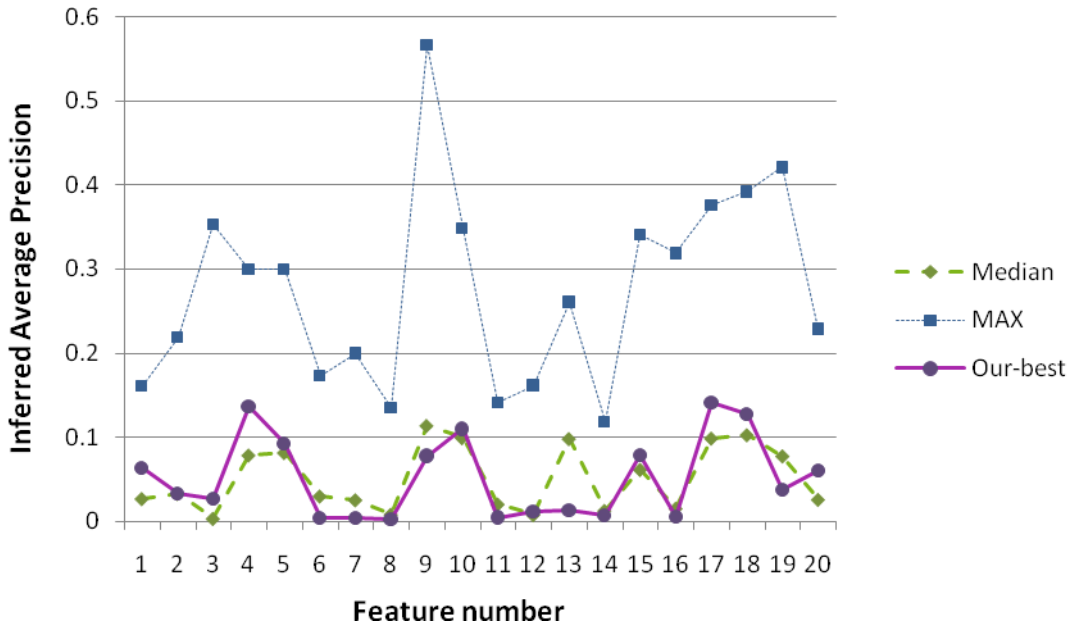



Figure 2 Performance of our best run for each concept vs. median and best performance of all submitted runs

## 2 Surveillance Event Detection

### 2.1 Overview

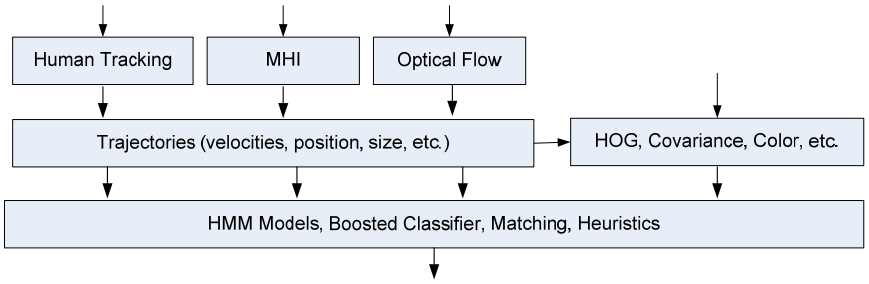



Figure 3 Event detection framework

Fig. 3 shows the framework of our event detection system. Because the characteristics and challenges of the required events are different from each other, the detection methods for the submitted events are loosely related. Some of the events, e.g., PeopleMeet and PeopleSplitUp, are strongly related to the trajectory information of relevant people in these events. Thus human detection and multi-person tracking [11] are performed to extract trajectory features, such as position, scale and velocity of each person at the scene. Sometimes, trajectory features of interest points other than human bodies may be more robust, taking the occlusions in the scene into account. Therefore, optical flow algorithm [8] is used to calculate motions of interest points in local regions. Motion history image [10] features is also adopted for motion detection. For detecting events like Embrace and Pointing, apparently, appearance and shape information are

very useful. We extract several features, including HOG, Covariance matrix and color histogram, and train detectors based on these features. As for TakePicture event, flashing analysis is performed. Section 2.2 explains how each submitted event is detected.

## 2.2 Event Detection

### 2.2.1 Human Detection and Tracking

The task of human detection and tracking is divided into four steps: foreground segmentation, head top detection, human detection and multi-object tracking, which is largely the same as our previous algorithm in TRECVid 2008 participation [12]. We adopt the algorithm in [6] for human detection, and use an optical flow guided particle filtering method that is capable of robustly tracking human in challenging conditions.

### 2.2.2 PeopleMeet and PeopleSplitUp

In each frame, at first we segment and track the motion object and get their trajectories. Then, we calculate the distance between every couple of objects as feature. After that, an HMM model is used to find the possible couple of objects that meet with each other or split from each other (Fig. 4). The likelihood derived from the HMM model is compared with a threshold, which validates an event if the likelihood is above the threshold.

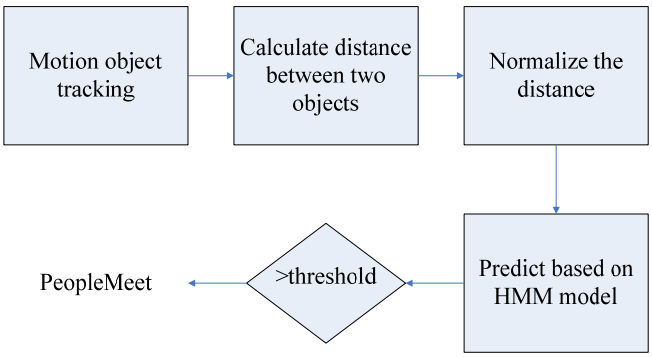



Figure 4 Detection of PeopleMeet and PeopleSplitUp

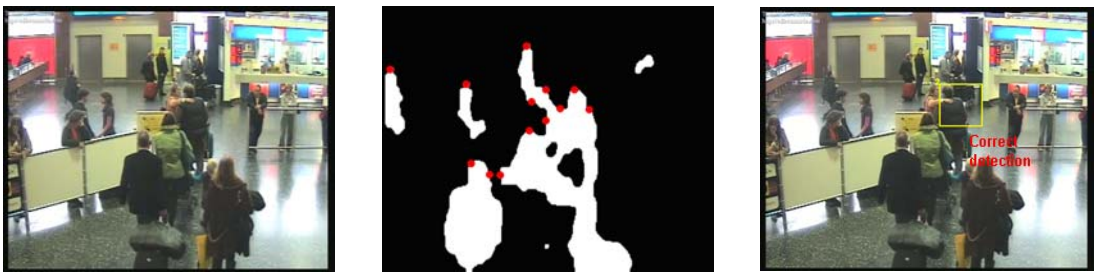
### 2.2.3 Embrace

Embrace detection can be divided into two steps: detection and tracking validation, taking into account of the appearance similarity and lasting time of the event.

First, we use Histogram of Oriented Gradient (HOG) features and a Cascade of Adaboost classifier [6] to train and detect possible Embrace regions. After foreground and background separation, the contour of foreground is analyzed to find convex points, each corresponding to a possible upper region of human body. For each region, if there are sufficient foreground pixels within it, the classifier is applied to detect possible Embrace (Fig. 5).

After a possible region is detected, a new tracking process is started. To obtain reliable human trajectories for Embrace events, we propose an optical flow guided particle filtering method that is capable of robustly tracking human in challenging conditions. In every subsequent frame, the HOG-based classifier is imposed at the resulting region of tracking process to further

validate whether it is Embrace or not. If less than 10 track results is judged as Embrace in 20 successive frames, it is considered that Embrace has finished and the tracking process is terminated.




(a) Original Image (b) Contour Analysis Result (c) HOG Detection Result

Figure 5 Detection of possible Embrace regions

### 2.2.4 OpposingFlow

For OpposingFlow, it's easily noticed that one main difference between opposing flow and normal flow is the motion velocity. Thus, Harris corners points at sub-pixel positions are detected around the door region, and then Pyramid Lucas-Kanade optical flow [8] method is performed to find the velocities of these points. Based on the positions and velocities of these points, we search for local regions which have sufficient points that satisfy heuristics rules (Fig. 6). Such process is done for each frame, allowing us to find the starting and ending time of OpposingFlow events.

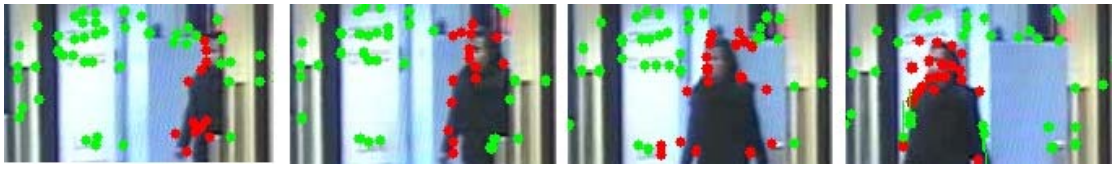


Figure 6 Opposing flow detection. Red dots are corner points with velocities pointing at left-bottom direction.

### 2.2.5 Pointing

The general framework for pointing detection is illustrated below:

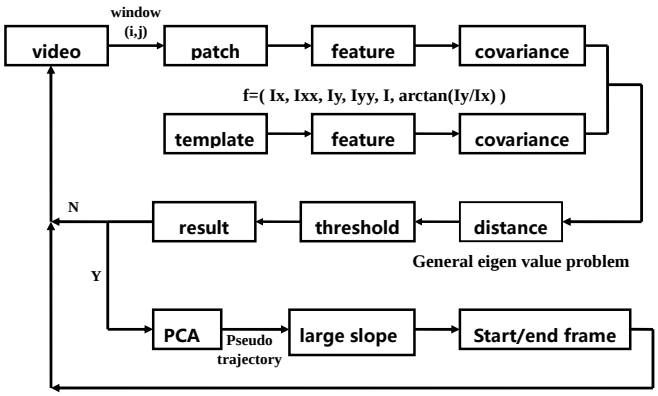



Figure 7 Detection framework of pointing event

In each frame, at first a window is used to extract patches and then covariance features [9] based on the first order and second order gradients are computed. After we have extracted feature vectors, template matching method is used to find the possible region of characteristic gestures that mostly imply a possible pointing event by moving the window in the whole frame.

Once we fix the possible region that pointing may happen, then PCA is applied to achieve the so-called pseudo-trajectory. Then we just find the most varied part (large slope) in a short time span to determine the start time and the end time.

### 2.2.6 TakePicture

It's not easy to detect TakePicture Events since most of the time the person's hands and camera occupy only a small amount of area in the scene, meanwhile, variations of the person's posture make the detection even harder. Hence we only focus on the events that happen with flashlight during the process of taking a picture. We improve the flash detection algorithm in [13] by adding a self-adaptive way of updating the average number of flash-light pixels. After closing examining training samples, we come into the conclusion that the flashlight only lasts in one frame, representing the regular pattern of "dark-bright-dark" related to the former, current and latter frame in brightness. Moreover, the increase and drop of the intensity are almost the same. In our approach, we transform this pattern into another one, considering the number of pixels which have an intensity change greater than certain level. In this case, firstly, we obtain frame difference images, after which the pattern is obviously transformed to "small-big-big-small", demonstrating the trend of the number of pixels with a notable change in intensity (Fig. 8).

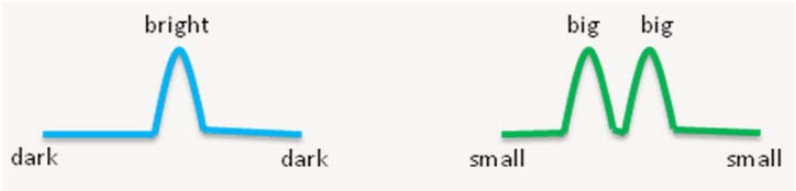



Figure 8 The specific pattern of the numbers of pixels with intensity change when TakePicture Event happens. The left one is the pattern in normal frame, and the right one is the pattern in frame differencing

Figure 9 In continuous three frames, flashlight is detected in a region (shown as a green rectangle).

After the flashlight is detected, we locate the region of the flashlight and mark it with a rectangle (Fig. 9). Due to the reason that in most cases the person's hands are hard to see even by

eyes, so we didn't detect the hand motion.

### 2.2.7 ElevatorNoEntry

For ElevatorNoEntry events, a Gaussian background model is used to separate the foreground (Fig. 10), and Motion History Images [10] are extracted. When the door of an elevator moves, its linear edges can be detected after motion detection. Thus, with MHI, we can perform straight line detection using Hough transform. In this way we could check if the elevator is opening or closing. When the door is opening, human detection is performed and all the detected human bodies are saved in a database in terms of both appearance and color histogram. When the door is closing, once again we detect humans and match previously saved ones through appearances and color histogram.

Figure 10 Foreground image, silhouette image, and motion history image (MHI).

### 2.2.8 PersonRuns

Running event detection is based on optical flow tracklets. After foreground detection, feature points are extracted within the foreground region every a certain number of frames. We use Harris corners points that are robust for sparse optical flow tracking. Pyramid Lucas-Kanade method is performed to track these points over a number of frames, resulting in hundreds of tracklets (Fig. 11), one for each feature point. These tracklets are classified into running-related and unrelated categories, based on their location and velocity. A running event is detected if the number of running-related tracklets over a local region is larger than an empirical threshold.

Figure 11 Optical flow tracklets

## 2.3 Experimental Results

For the event detection task, we submit one experiment, outputting the results for 8 events,

"TakePicture", "PersonRuns", "OpposingFlow", "Pointing", "ElevatorNoEntry", "PeopleMeet", "Embrace", and "PeopleSplitUp" from the 10 candidate events for performance evaluatation. The development data consists of the 2008 Event Detection Training and Testing sets. The evaluation dataset is the UK Home Office Scientific Development Branch's (HOSDB) i-LIDS MCTTR dataset. Table 2 shows the detailed scoring analysis report of our detection system, including the true reference results, the number of false alarms and miss detections, and Normalized Detection Cost Rate (NDCR) measure, etc. The Detection Error Tradeoff (DET) Curves corresponding to those results, which plots a series of event-averaged missed detection probabilities and false alarm rates that are a function of a detection threshold, Θ, are given in Fig. 12.

Table 2 Event Detection Scoring Analysis Report

| Analysis Report | #Ref | #Sys | #CorDet | #FA | #Miss | Act. RFA | Act. PMiss | Act. DCR | Min RFA | Min PMiss | Min DCR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TakePicture | 12 | 2 | 0 | 1 | 12 | 0.066 | 1 | 1 | 0.131 | 1 | 1.001 |
| PersonRuns | 107 | 2217 | 19 | 1228 | 88 | 80.539 | 0.822 | 1.225 | 21.447 | 0.981 | 1.089 |
| OpposingFlow | 1 | 6 | 1 | 5 | 0 | 0.328 | 0 | 0.002 | 0 | 0 | 0 |
| Pointing | 1063 | 1858 | 12 | 225 | 1051 | 14.757 | 0.989 | 1.062 | 2.886 | 0.998 | 1.012 |
| ElevatorNoEntry | 3 | 28 | 2 | 26 | 1 | 1.705 | 0.333 | 0.342 | 1.64 | 0.333 | 0.342 |
| PeopleMeet | 449 | 19739 | 108 | 7706 | 341 | 505.404 | 0.759 | 3.287 | 1.443 | 0.996 | 1.003 |
| Embrace | 175 | 14189 | 64 | 1919 | 111 | 125.859 | 0.634 | 1.264 | 0.328 | 0.994 | 0.996 |
| PeopleSplitUp | 187 | 22877 | 66 | 11690 | 121 | 766.697 | 0.647 | 4.481 | 1.705 | 0.984 | 0.993 |

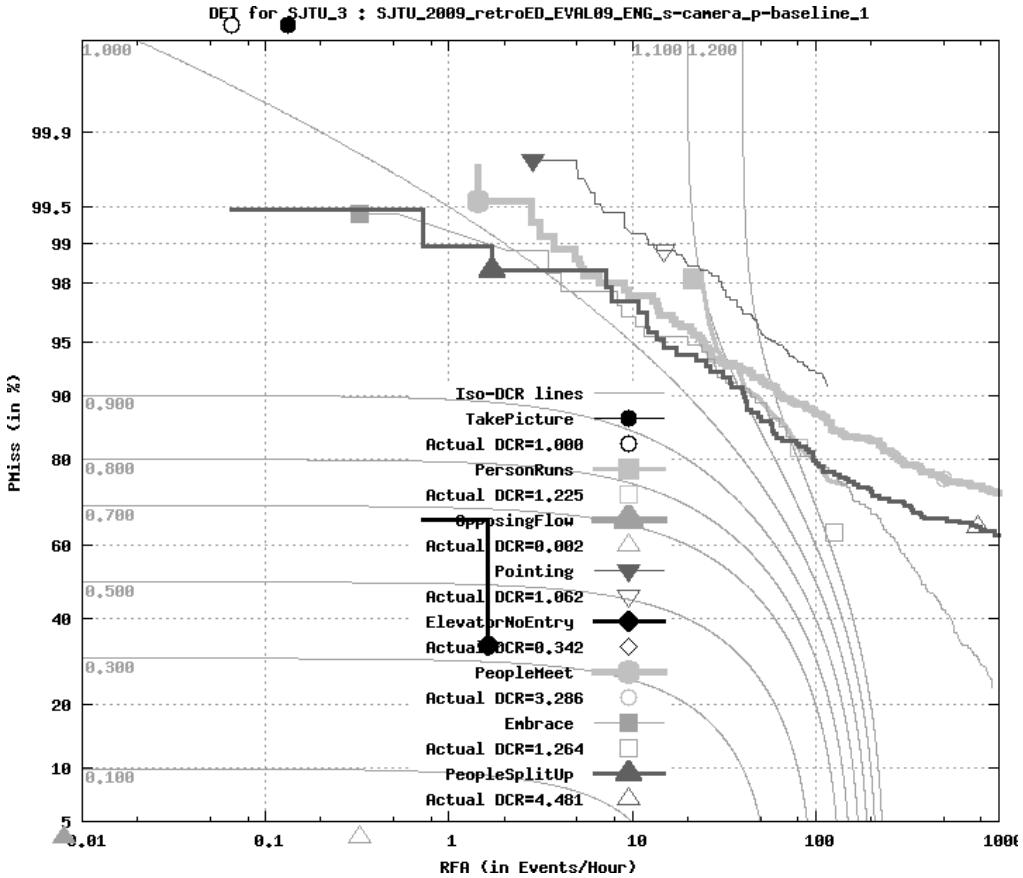


Figure 12 Detection Error Tradeoff (DET) Curves

## Acknowledgement

This work was supported in part by Research Fund for the Doctoral Program of Higher Education of China (200802481006), NSFC (60828001, 60902073, 60932006), NCET-06-0409, and the 111 Project.

## References

[1] Chih-Chung Chang and Chinh-Jen Lin. LIBSVM: a library for support vector machines,2001

[2] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. IEEE Comp. Vis. Patt. Recog. 2005.

[3] I van Laptev, Marcin Marszałek, Cordelia Schmid and Benjamin Rozenfeld. Learning Realistic Human Actions from Movies, In Proc. CVPR'08.

[4] D.Ramanan. Learning to parse images of articulated bodies. In NIPS, 2006

[5] W. Hsu, L. Kennedy, and S.F. Chang. Video search reranking via information bottleneck principle. In ACM Multimedia, Santa Babara, CA, USA, 2006.

[6] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histogram of oriented gradients. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 1491-1498, 2006.

[7] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. European Workshop on Advanced Video-based Surveillance Systems, 2001.

[8] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker. Intel Corporation Microprocessor Research Labs, 2000.

[9] Tuzel, O.; Porikli, F.; Meer, P.. Region Covariance: A Fast Descriptor for Detection and Classification. European Conference on Computer Vision (ECCV), May 2006

[10] Gary R. Bradski and James Davis. Motion segmentation and pose recognition with motion history gradients. In IEEE Workshop on Applications of Computer Vision, pages 238–244, 2000.

[11] Zhaowen Wang, Xiaokang Yang, Yi Xu, and Songyu Yu. Camshift guided particle filter for visual tracking. Pattern Recognition Letters, pp. 407–413, 2009.

[12] http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/sjtu.pdf

[13] Yarlagadda P, Demirkus M, Garg K and Guler S.. IntuVision Event Detection System For Trecvid 2008. IntuVision, Inc.2008