Shanghai Jiao Tong University participation in high-level feature extraction, automatic search and surveillance event detection at TRECVID 2008

Xiaokang Yang, Rui Zhang, Yi Xu,
Anwen Liu, Jiemin Liu, Zheng Lu, Xiaolin Chen, Erkang Chen,
Qing Yan, Zhaowen Wang, Yanlan Song, Xiaojie Sheng, Bo Xiao,
Zhou Yu, Zhenfei Chu, Hang Su, Jun Huang, Li Song
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 2008 evaluation.

In high-level feature extraction, we use selective attention model to extract visual salient feature which highlights the most visual attractive information of an image. Besides this, we extract 7 low-level features for various modalities as a baseline and use linear weighted fusion of multi-modalities . Results show that simple linear weighted fusion works pretty well. In addition, ASR is useful to improve the performance . We submitted the following six runs:

- *A_SJTU_1: Max of all runs based on different methods, and Re-rank based on ASR*
- *A_SJTU_2: Max of all runs based on different methods*
- *A_SJTU_3: Linear weighted fusion across 7 modalities except Attention model, SVM train on TRECVID2008 development data*
- *A_SJTU_4: Attention model,SVM train on TRECVID2008 development data*
- *B_SJTU_5: SVM train on TRECVID2008 and TRECVID2005 development data based on feature replication*
- *A_SJTU_6: Local Binary Pattern,SVM train on TRECVID2008 development data*

In automatic search, we use text-based search as baseline, combined with concept-based search and query-by-example (QBE). In concept-based search, we just use 20 concepts defined in high-level feature extraction task of TRECIVD 2008 to rank the videos. In QBE, color moment (CM) is used as the feature representation of images, and Euclidean distance of these feature vectors is used to roughly represent the semantic similarity between two images. We submit four runs:

- *F_A_1_SJTU3_3: text baseline by ASR/MT transcripts*
- *F_A_2_SJTU2_2: text baseline with concept-based search*
- *F_A_2_SJTU1_1: text baseline with concept-based search and QBE*
- *F_A_2_SJTU4_4: use concept-based search and QBE without text baseline*

In Event detection, based on trajectory features from human tracking, we test several detection rules, including HMM models, heuristic settings and gesture recognition to detect different events. Motion detection is also used for elevator-related event. We output the results for 5 events out of 10 optional events to be evaluated.

- *SJTU_2008_retroED_EVAL08_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 our framework as shown in Fig. 1, there are four main steps involves:

- Low level feature extraction. We use visual selection model to extract attention feature which embraces luminance, hues and orientations , and will be detailed in Section 1.2. We also implement seven baseline low level features: two color features*(Color Auto-Correlograms(166 dim), Color Moment Grid(225 dim, 5*5grids))* including global and local information, three texture features*(Co-occurrence Texture(96 dim), Wavelet Texture Grid(108 dim, 3*3 grids), Local Binary Pattern(531 dim, 3*3 grids))* also including global and local information, two shape features*(Edge Direction Histogram(73 dim), Edge Auto-Correlograms(144 dim))*.

- Modeling. We adopted Support Vector Machines [2] as our classification method to train the individual SVM classifier for each low-level feature based on valid cross database learning by using TRECVID2008 and TRECVID2005 development data.
- Ranking. Linear weighted fusion is used to combine multiple ranking results based on seven models trained on baseline low-level features. Max single best is another special example of linear weighted fusion method.
- Re-ranking. We extracted textual information based on automatic speech Recognition (ASR). By adding the positive textual relevant factor to the previous ranking result, we obtained the re-ranking results.

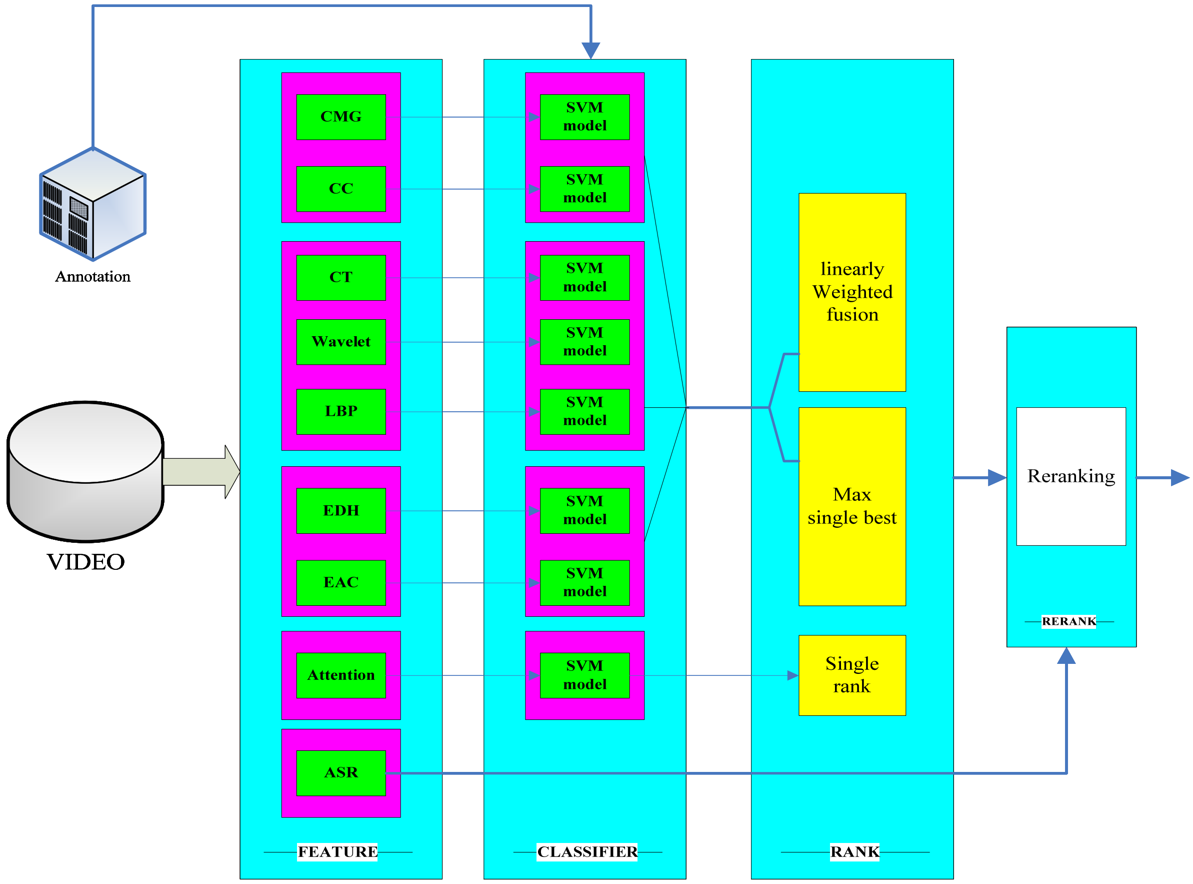


Figure 1. High-level feature extraction framework

1.2 Visual selective model

In the visual selective model for the extraction of attention features shown in Fig. 2, we decompose the input image into a set of channels, by using linear filters tuned to specific stimulus dimensions, such as luminance, red, green, blue and yellow hues or various local orientations. In addition, such decomposition is performed at a number of scales, to allow the model to represent smaller or larger objects in sub-channels:

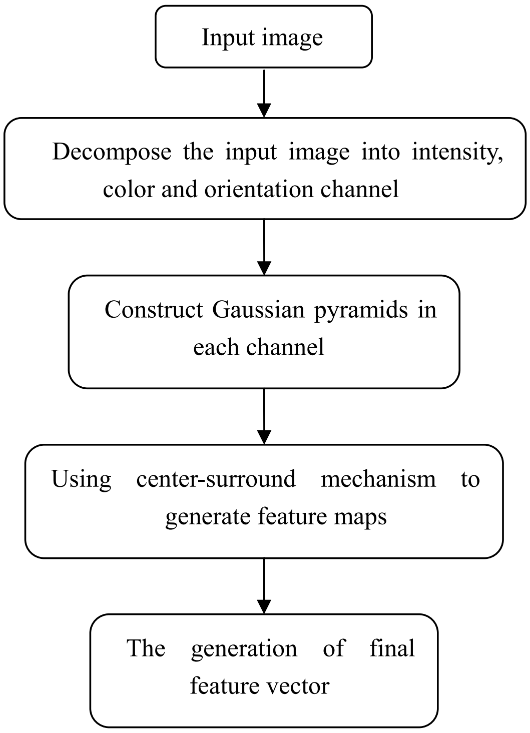

Input image
Decompose the input image into intensity, color and orientation channel
Construct Gaussian pyramids in each channel
Using center-surround mechanism to generate feature maps
The generation of final feature vector

Figure 2. Flowchart of visual selection model

1. Intensity channel:

   If r,g and b are the red, green and blue components of a color image respectively, then the intensity map is computed as:

$$I = \frac{r + g + b}{3} \tag{1}$$

2. Color channel:

$$RG = \frac{r \cdot g}{\max\{r, g, b\}} \tag{2}$$

$$BY = \frac{b \cdot \min\{r, g\}}{\max\{r, g, b\}} \tag{3}$$

   We obtain the pyramid representation of the two channels by filtering and sub-sampling the original input image in each channel and we get Gaussian pyramids of the intensity and color channels. Each feature is computed in a center-surround structure to simulate visual receptive fields. The operation is implemented as difference between a fine and a coarse scale for a given feature. The center corresponds to the pixel at level

$c \in \{2, 3, 4\}$ in the pyramid and the surround corresponds to the pixel at level $s = c + \sigma$, with $\sigma \in \{3, 4\}$. We compute six maps for intensity and color separately.

3. Orientation channel:
   The local orientation information is obtained from I using oriented Gabor Pyramids $O(\theta, \sigma)$ where $\sigma \in [0, 8]$ represents the scale and $\theta \in \{0°, 45°, 90°, 135°\}$ is the preferred orientation. In our implementation scale 1 to scale 4 are chosen. Thus we have 4 maps for each orientation.

Till now we have totally $1 * 6 + 2 * 6 + 4 * 4 = 34$ maps. For each map in intensity, color and orientation channel, we divide it into $4 * 4$ blocks and use the mean value of the block to represent it. And the final attention feature is formed by concatenate the features of all 34 maps and the final feature dimension is $34 * 16 = 544$.

1.3 SVM based on valid cross database learning

We adopt SVM for elementary classification. Besides seven classifiers based on low-level features and another one for visual selection model, we train SVM models based on valid cross database learning by using both TRECVID2008 and TRECVID2005 development data. There are three high-level features ”Bus”, ”Mountain” and ”Boat_ship” for evaluation both in TRECVID2005 and TRECVID2008, and another relevant couples of high-level features ”Airplaane_flying : Airplane”. For these four high-level feature, we add valid positive support vectors of previous database to the current database in order to improve learning positive boundary.

1.4 Re-ranking bases 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, we add additional confident scores for every shot by computing the similarity between the current shot and keywords set.

1.5 Experimental results

We submitted 6 runs for high-level feature extraction as shown in Table 1, the results with the bold fonts show the best run for each concept. Results show that simple linear weighted fusion of baseline low-level features works pretty well. Selecting best modality for every concepts is ineffective unexpectedly. It means no single feature works well in all concepts due to the partial information. Training SVM models based on valid cross database learning from both TRECVID2008 and TRECVID2005 development data can improve the performance by adding the positive learned support vectors. ASR is also helpful in re-ranking step although a marginal improvement of 2.63% is achieved. Our visual selection model performs an average result due to its comprehensive feature of image involving luminance, hue and orientation. The results in Fig. 3 show the best run of ours for each concept *vs.* median and the best performance of all submitted runs.

High-level features	MAX+ASR	MAX	Baseline	Attention	Cross_database	Single_LBP
Classroom	**0.071**	0.027	0.027	0.013	0.027	0.001
Bridge	**0.005**	0.003	0.003	0.002	0.003	0
Emergency_Vehicle	**0.006**	**0.006**	0.003	0.002	0.003	0
Dog	0.002	0.002	**0.234**	**0.234**	**0.234**	0.002
Kitchen	0.006	0.004	0.004	**0.014**	0.004	0
Airplane_flying	0.019	0.011	**0.039**	0.07	0.007	0.02
Two people	0.066	0.066	0.066	**0.068**	0.066	0.003
Bus	0.004	0.004	0.005	0.002	**0.006**	0
Driver	0.069	**0.107**	**0.107**	0.057	**0.107**	0.002
Cityscape	**0.101**	**0.101**	**0.101**	0.071	**0.101**	0.016
Harbor	**0.008**	0.006	0.006	0.004	0.006	0.001
Telephone	**0.058**	0.043	0.043	0.031	0.043	0
Street	0.151	**0.152**	**0.152**	0.112	**0.152**	0.025
Demonstration_or_protest	0.012	0.012	**0.033**	0.012	**0.033**	0.01
hand	0.181	**0.195**	**0.195**	0.107	**0.195**	0.001
Mountain	0.04	0.026	0.026	0.041	**0.048**	0
Nighttime	**0.089**	**0.089**	**0.089**	0.084	**0.089**	0.002
Boat_Ship	0.008	0.004	0.103	0.046	**0.128**	0.004
Flower	**0.116**	0.109	0.109	0.014	0.109	0.005
Singing	0.043	**0.061**	**0.061**	0.029	**0.061**	0
MAP	0.05275	0.0514	0.0703	0.05085	**0.0711**	0.00415

Table 1. six runs of our high-level feature extraction for each concepts

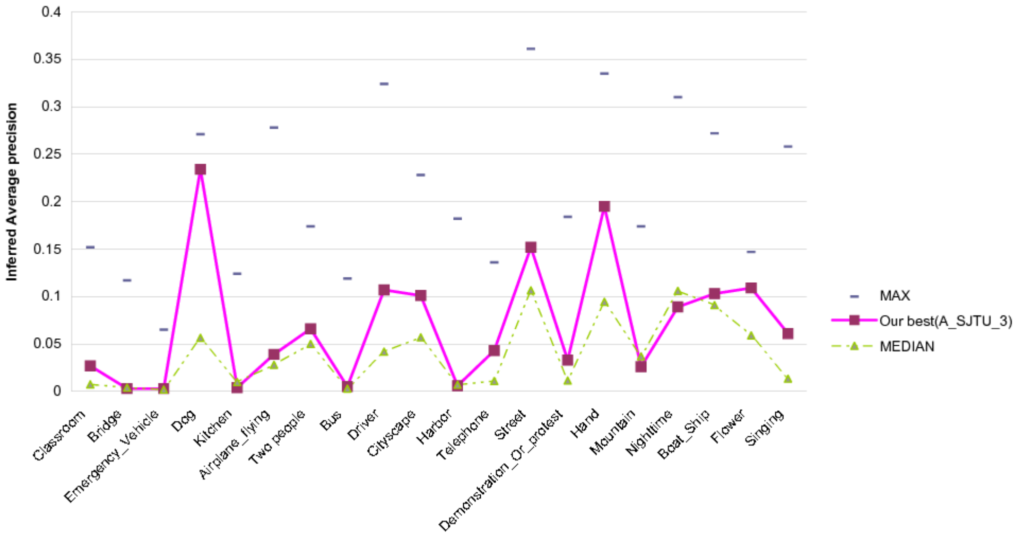


Figure 3. Performance of our best submitted run for each concept *vs.* median and best performance of all submitted runs

2 Automatic Search

2.1 Text Baseline

In text-based search, only ASR/MT transcripts given by NIST are used. These transcripts are first separated by the time points. Accordingly,videos are also divided into shots. Afterwards, the ASR/MT texts corresponding to shots are retrieved by text-based search method. Our text search engine is based on BM25 [6]. Because noun is more important in video, we give the highest weights to nouns in query. And keyword expansion is done automatically through a small self-established word list.

2.2 Concept-based Search

Since we take part in high level feature extraction task this year, we use our 20 concepts detection results to rank. These 20 concepts are defined in high level feature extraction task of TRECVID 2008. Because the amount of the detected concepts is quite few, some queries cannot be well described by concepts combination and even have no predefined concepts involved. So the scarcity of detected concepts results in the decline of the search results.

2.3 Query-by-Example (QBE)

We select Color Moment (CM) as the feature representation of images. Thus, all the key-frames of the videos in test collection and positive examples are represented by their CM feature vectors. Euclidean distance is calculated between these feature vectors to roughly represent the semantic similarity between two images.

2.4 Experimental Results

We submitted 4 runs of automatic search totally, which are the different combination of three methods abovementioned. MAP of each run is shown in Table 2. To our surprise, the run F_A_2_SJTU4_4 is better than F_A_2_SJTU1_1. Text-based search has almost no even negative use. From the MAP of the run F_A_1_SJTU3_3, we can also find text based search hardly works to the topics of this year. By comparing the run F_A_2_SJTU2_2 and F_A_2_SJTU1_1, we can find that QBE does improve the search results. But our method of QBE this year is quite rough, it only improves the search results of several topics.

RUNS	**MAP**
F_A_1_SJTU3_3	0.0062
F_A_2_SJTU2_2	0.0186
F_A_2_SJTU1_1	0.0211
F_A_2_SJTU4_4	0.0215

Table 2. Four runs of our automatic search and their MAP

3 Surveillance Event Detection

3.1 Approach Overview

Fig. 4 shows the framework of our event detection system. For most of the required events in TRECVID 2008 are strongly related to the trajectory information of relevant people in these events, human detection and multi-person tracking [8] constitute the first main part of our system, after which trajectory features, such as position, scale and velocity of each person at the scene, are extracted. Hidden Markov Model (HMM) or heuristics are then employed to make decision. For the special event concerning elevators, apart from the above, motion detection [1] is performed to get the status of elevator doors. For the event ”Pointing”, apparently gesture information is very important. Based on tracking of people at the scene, a Haar-like object detector [7] is used for gesture recognition along the trajectories.

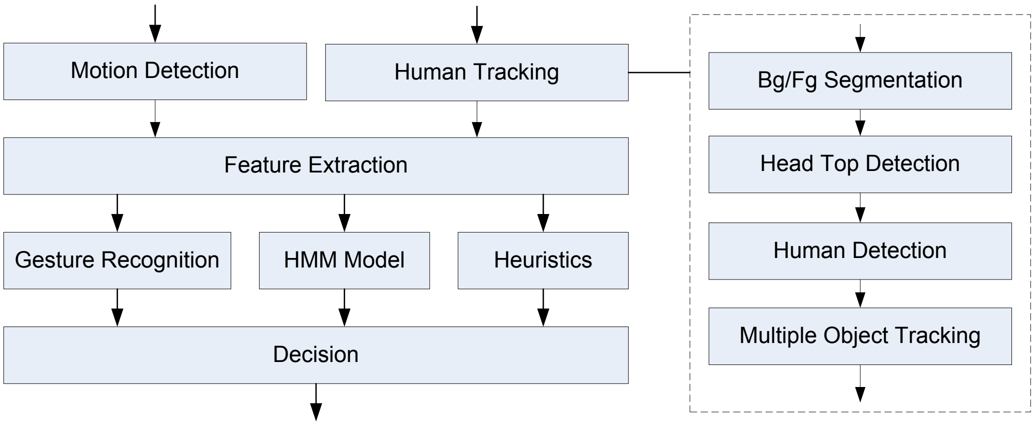


Figure 4. Event detection framework

3.2 Feature Extraction

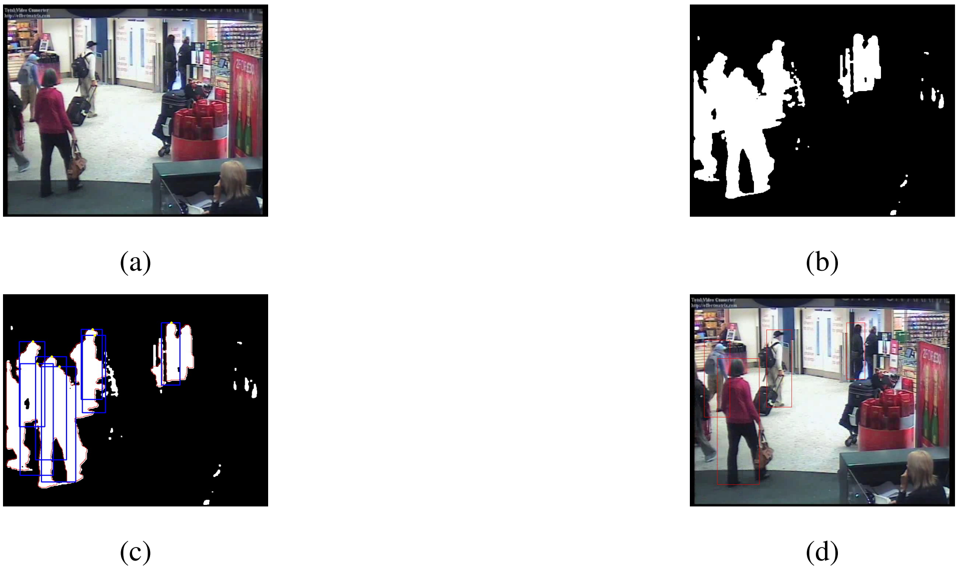
3.2.1 Human Tracking

The modules in the right (dashed) block of Fig. 4 show four steps of human tracking of our approach: background/foreground segmentation, head top detection, human detection and multi-object tracking.

Prior to the final tracking, human should be detected from the scene. Simple head top detection and specific human detector can both do the job. The latter, though powerful, is costly compared to simple foreground segmentation and head top detection. On the other hand, due to the noise in video sequences, head tops detected may be false positive or correspond to the same person. We propose to combine these two methods, i.e., human detection is only performed in the local regions located by the detected head tops. Moreover, human detection is only needed when a person first appears at the scene. Therefore, if a head top is near the one of a tracked person, then no human detection is needed for that region. In this way, a tradeoff between detection accuracy and computational efficiency is achieved.

For segmentation of moving regions, we adopted the robust method [4] which incorporate shadow detector, for shadows at the indoor surveillance scene of TRECVID 2008 video. It models the color of each pixel in the image as a Gaussian mixture model. After that certain morphological processing are applied to get a smooth foreground regions (Fig. 5(b)).

A faster version similar to [9] is used to find the head top of every person from the moving foreground. Because the surveillance cameras are mounted several meters above the ground, the human heads are less likely to be occluded. First the highest point is found for each region in the vertical direction along the boundary with a range defined by the average size of a human head assuming an average height. Then we extract the contour the region



(a)

(b)

(c)

(d)

Figure 5. (a) Original image. (b) Foreground segmentation. (c) Human detected by head tops. (d) Human detected finally.

and use the first derivative statistics of the contour data to rule out some false points, given the fact that the shape of human head-shoulder part is distinctive. Points that are too close to each other are merged. Fig. 5(c) shows the human detected by head tops.

Such head top detection is very efficient, however, it could result in some false positive and redundant points, which can be further eliminated by human detection (Fig. 5(d)). We use the detector presented by Dalal & Triggs [3]. It uses a dense grid of Histogram of Oriented Gradients (HoG), computed over blocks of size 16×16 pixels to represent a detection window. A linear SVM is applied on this representation to classify humans.

The scene of TRECVID 2008 surveillance video presents several challenges for multi-person tracking, such as affine transformation, illumination changes, occlusion, noise, and so on. Our tracking algorithm is based on our previous work, CamShift guided particle filter (CAMSGPF) [8] in which two basic algorithms - CamShift and particle filter - work cooperatively and benefit from each other, so that the overall performance is improved and some redundancy in algorithms can be removed. CAMSGPF can track objects robustly in various environments, and is much faster than the existing methods.

3.2.2 Motion Detection

When the doors of an elevator moves, their linear edges can be detected after motion detection. Thus, we extract Motion History Image (MHI) [1] from the video, and perform straight line detection using Hough transform. In this way we could check if the elevator is opening or closing.

3.3 Decision Making

3.3.1 HMM Model

For the events in which people interact, the changes of trajectories features, such as distance and relative velocity between relevant people, have common characteristics. We define hidden states and use HMM models [5] to detect "PeopleMeet", taking as observations the trajectories features during the time when two people co-appear at the scene. The start frame and the end frame of the event corresponds to the temporal information of hidden states.

For the event "ElevatorNoEntry", after human detection and motion detection, the status of elevator doors are closed, opening, open, or closing, and the status of human are no human waiting, static human, moving human. These are combined to form the observations of the HMM model.

3.3.2 Heuristics

After obtaining trajectories of human, we use heuristics to detect the events "PersonRuns" and "OpposingFlow". Apparently, a running person has a higher speed for some sustaining time, and the moving direction is the most obvious indicator for a person moving through a door opposite to the normal flow of traffic. We use the several statistics of the velocity to detect running, including the magnitude, the sustaining time when it is above the some threshold, the mean and the variance of velocity in that period of time. For "OpposingFlow", we make decision by judging whether the starting point of trajectory is beside the door and whether its direction is opposite to the normal flow. Thresholding is used to the start and end frame of these events.

3.3.3 Gesture Recognition

To detect "Pointing", static gesture recognition is employed along the trajectory of each person. First canny edge detector is applied to the tracked regions. Then a Haar-like object detector [7] using edge features is used to recognize the gesture of pointing. To overcome the scale problem, the same process is repeated at different scales. When the frame of the most probable gesture is detected, the start frame and end frame of the event are found by thresholding the score of recognition score.

3.4 Experimental results

For the event detection task, we submit one experiment, outputting the results for 5 events, “PeopleMeet”, “PersonRuns”, “OpposingFlow”, “ElevatorNoEntry” and “Pointing” from the 10 candidate events to be evaluated. The training and testing data consists of surveillance footage from 5 camera views in the same airport over a period of 10 days, and different camera views present different levels of challenges. From the above discussion, our system largely depends on the performance of human tracking which faces the difficulties of affine transformation, illumination changes, occlusion and noise. Human tracking in camera view 1, 3 and 4 is more stable than in camera view 2 and 5, for there are more crowded people and more occlusion in the latter two views. Gesture recognition also faces challenges because of different view angles and scales.

Acknowledge

This work was supported in part by 863 (2006AA01Z124), NSFC (60502034, 60702044, 60828001), Shanghai Postdoctoral Foundation (06R214138) and the 111 Project.

References

[1] 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.

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

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, pages 886–893, 2005.

[4] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. In *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems.*

[5] Nuria Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:831–843, 2000.

[6] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In *TREC*, 1994.

[7] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.

[8] Zhaowen Wang, Xiaokang Yang, Yi Xu, and Songyu Yu. Camshift guided particle filter for visual tracking. In *IEEE Workshop on Signal Processing Systems*, pages 301–306, 2007.

[9] Tao Zhao and Ram Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1208–1221, 2004.