BUPT-MCPRL at TRECVID 2009[*]

Zhicheng Zhao, Yanyun Zhao, Zan Gao, Xiaoming Nan, Mei Mei, Hui Zhang,
Heng Chen, Xu Peng, Yuanbo Chen, Junfang Guo, Anni Cai

Multimedia Communication and Pattern Recognition Labs,
Beijing University of Posts and Telecommunications, Beijing 100876, China
{zhaozc, zyy, annicai}@bupt.edu.cn

Abstract

This paper describes BUPT-MCPRL systems for TRECVID 2009. We performed experiments in automatic search, HLF extraction, copy detection and event detection tasks.

A. Automatic search

A semantic-based video search system was proposed and brief description of submitted 10 runs is shown in Table.1.

Table 1 The performance of 10 runs for automatic search

Run ID	infMAP	Description
F_A_N_BUPT-MCPR1	0.104	HLF-based retrieval and positive WDSS method
F_A_N_BUPT-MCPR2	0.070	Concept-based retrieval and positive WDSS method
F_A_N_BUPT-MCPR3	0.059	Concept-based retrieval and positive and negative WDSS method
F_A_N_BUPT-MCPR4	0.131	Combining concept lexicons of MCPR1 High-Level-Features and MCPR2 search topics and using positive WDSS method
F_A_N_BUPT-MCPR5	0.032	Concept-based retrieval with example bagging method
F_A_N_BUPT-MCPR6	0.024	Visual example-based retrieval
F_A_N_BUPT-MCPR7	0.024	Concept-based retrieval with example weighting method
F_A_N_BUPT-MCPR8	0.016	Re-rank MCPR7 with face score
F_A_N_BUPT-MCPR9	0.009	Fusion MCPR6 and MCPR 7 and re-rank with face score
F_A_N_BUPT-MCPR10	0.048	Fusion with MCPR5, MCPR 6 and MCPR 7

B. High-level feature extraction

In this year, focus of our HLF system was on boosting and fusion of low-level features, the difference of classifiers with cross-validation, and re-ranking of results according to face detection.

Table 2 HLF results and description of BUPT-MCPRL system

HLF Run	infMAP	Description
BUPT-MCPRL_Sys1	0.0313	BUPT-MCPRL_Sys3 is modified by face results.
BUPT-MCPRL_Sys2	0.0487	This run fuses the results of BUPT-MCPRL_Sys3, BUPT-MCPRL _Sys4 and BUPT-MCPRL _Sys5.
BUPT-MCPRL_Sys3	0.03515	This run uses nineteen models for each concept and fuse local features and global features without cross-validation.
BUPT-MCPRL _Sys4	0.02255	This run uses nineteen models for each concept and just fuses local features without cross-validation.
BUPT-MCPRL _Sys5	0.05995	This run uses seven models for each concept, Fuses Local and Color features with cross-validation in the training.

[*] This work was supported by China National Natural Science Foundation under Project 60772114, 90920001.

BUPT-MCPRL_Sys6	0.04835	This run uses seven models for each concept, Fuses Local and Color features without cross-validation in the training.

C. Copy detection

Two different retrieval algorithms based on SURF and SIFT were independently proposed to detect copy videos.

D. Event detection

Our event detection mainly adopted SVM models and rule-based method.

1. Automatic Search

1.1 The Proposed Framework

The proposed semantic-based video system consisting of several main components, including text and visual query pre-processing, visual feature extraction, classification, multimodal fusion and results re-ranking. The framework of our search system is shown in Figure 1.1.

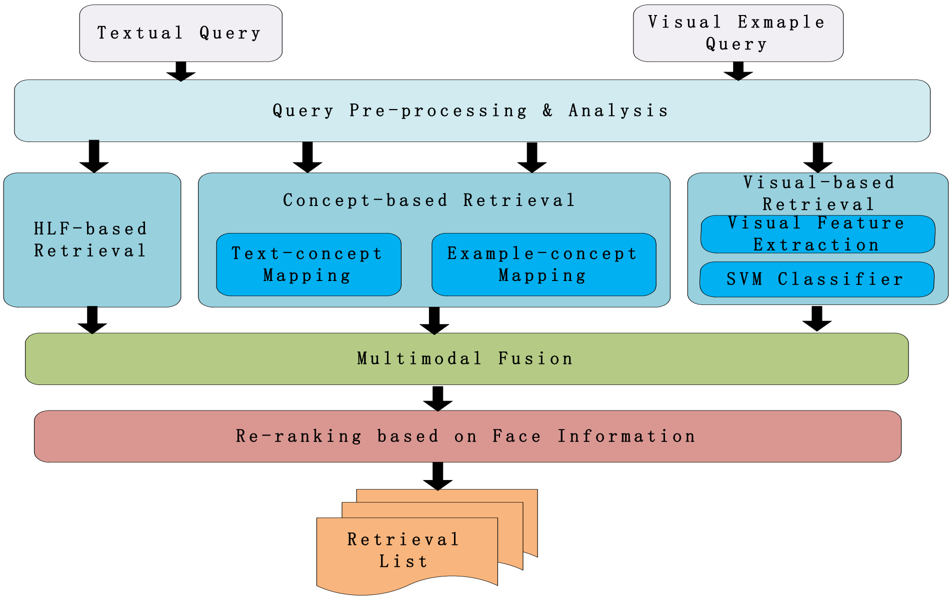


Figure 1.1 The framework of automatic search system

From Table 1, we can find that the MCPR4 achieves the best MAP among our submitted 10 runs. Overall, our contributions are summarized as follows:

- Proper use of Weight Distribution based on Semantic Similarity (WDSS) method. The WDSS strategy aims to select the concepts with the most semantic similarity by parsing the lexicon of text query and the lexicon of visual concepts.
- Evaluation of a large number of visual descriptors. We have explored various low-level visual features at different granularities and employed a boosting feature selection method to select the most effective descriptors.
- Exploring different methods for making use of visual examples. Training classifiers with visual clips, bagging with example scores and weighting with selected concepts, all of these are applied in our system.
- Comparison retrieval strategies based on textual query and that based on visual query. Among the submitted results, the former 4 runs focus on text-concept mapping strategy, while the later 6 runs attempt to make use of topic examples. From Table 1.1, we find the text-concept mapping method performs much better than visual-based retrieval.

1.2 Building the Dataset

To conquer the lack of prior knowledge, we built two concept lexicons, the one of High-Level-Features from TRECVID over the past 3 years and the other of 48 topics from TRECVID 2008 search task. The ground truth of all those concept sets was manually annotated on the Sound and Vision development data (tv7.sv.devel, tv7.sv.test), which was divided into 3 partitions, 60% of the annotated shots as training data, 20% as validation data, as well as the rest 20% as fusion data.

1.3 Feature Selection

Since no unique visual feature can represent all information contained in a video, and no given visual feature is effective for all concepts, we extracted a great deal of features at local, regional and global levels, which can be divided into four categories: key-point features, texture features, edge features and color features. At the same time, in order to save computation time and improve effectiveness, a feature selection scheme based on boosting method was employed. At last, 11 low-level features with better performance were selected in our system, which were listed in the following table [1, 4, 6, 7, 8, 9].

Table 3 Selected low-level visual features

Features	Description
SIFT + SURF Global	Concatenate sift-visual-keywords and surf-visual-keywords with global partition
SIFT + SURF Region 1*3	Concatenate sift-visual-keywords and surf-visual-keywords with 1*3 regional partition
SIFT + SURF Region 2*2	Concatenate sift-visual-keywords and surf-visual-keywords with 2*2 regional partition
SIFT + SURF Region 3*1	Concatenate sift-visual-keywords and surf-visual-keywords with 3*1 regional partition
Gabor Wavelet	3-scale and 6-direction Gabor feature with 3*3 regional partition
Local Binary Pattern	256 dims histogram of each LBP code with global partition
Edge Directional Histogram (EDH)	145 dims histogram by concatenating global and regional EDH
R_HoG	Histogram of Oriented Gradient with rectangle block
HSV_Correlogram	Color Auto Correlogram feature with global partition
RGB_BlkHist	RGB color histogram with 3*3 regional partition
Block RGB Moment	RGB color moment with 5*5 regional partition

1.4 Visual Example-based Search

In the visual example-based retrieval, these selected visual features were extracted to describe the images of topic examples. For each feature, a SVM classifier was built for each topic. The positive samples used to train the classifier were the topic examples and the negative samples were randomly sampled from negative training dataset without repetition. We used radius basis function as the kernel function and determined the parameters in a coarse-to-fine searching method by 3-fold cross-validation at the training stage. However, because of the limited number of topic examples, MAP of the example-based retrieval just reaches 0.024 (MCPR6), which needs to be further improved. This point is consistent with conclusion in [3, 5, 10].

1.5 Concept-based Search

Concept-based retrieval has played a crucial role to improve the performance in automatic video retrieval systems [11, 12]. The key problem for this module is how to use the limited number of pre-trained concept classifiers to satisfy various user’s queries. Two sets of concept lexicons were adopted in our system. The first one was the set of HLF concepts of TRECVID over the past 3 years and the second one consists of 48 search topics of TRECVID 2008. In addition, we have explored text-concept mapping strategy and example-concept mapping strategy respectively.

As for text-concept mapping method, we select semantic similar concepts according to the Weight Distribution based on Semantic Similarity (WDSS) mapping method [2, 3 and 11]. First of all, query expansion with synonyms, acronyms, and stop word removal were implemented in pre- processing stage. Then, by virtue of WordNet we could calculate concept similarity to query words, and after a similarity clustering, an unfixed number of similar concepts were selected, with normalized similarity score as the concept weight. In addition, the negative WDSS method was also introduced in our system. For example, the relevant shots of topic 290 “Find shots of one or more ships or boats, in the water” should not contain concepts like “office” and “kitchen”, so that the weights of these concepts were set to negative to avoid their appearances.

On the other hand, we also attempt ways for example-concept mapping strategy [3, 10]. The first method is acquiring concept weight by classifying topic examples to obtain the possibility vector. By using this vector, the concept results for each shot were linearly weighted, and finally the retrieval list was calculated. Additionally, the second strategy is a bagging strategy with example scores, in which the classifiers were trained by example score vectors and the retrieval list was computed by classifying each shot by using these detectors. From the results table, we find that the MAP of bagging strategy outperforms that of direct weighting method by 33.3%.

1.6 Multimodal Fusion and Re-ranking Strategy

In our system, the linear average precision weight fusion strategy was applied for intra concept fusion while the max fusion strategy was employed as a method for multimodal fusion. The average precision could be acquired with the fusion dataset.

This year, we try to use face detection information for each shot as a basis for re-ranking. For those topics relevant with human, face scores calculated by face number and size were added to the original result list. But maybe due to the unreasonable strategy, or because of the low precision of our face detection algorithm, the performance of re-ranking is not satisfactory.

1.7 Experiments and Discussions

This year, we submitted 10 automatic search (type A) runs for search task and the performances are shown in Figure 1.2. Because of the unreliable face detection, MCPR8 and MCPR9 with re-ranking had our lowest MAPs of 0.009 and 0.016. As for the visual example-concept mapping strategy, the weighting method reached 0.024, while the bagging strategy improved to 0.032. With limited number of positive samples, MCPR7 using detectors trained by topic examples achieved 0.024, and MCPR10 the max fusion of all example-based retrieval had a MAP of 0.048.

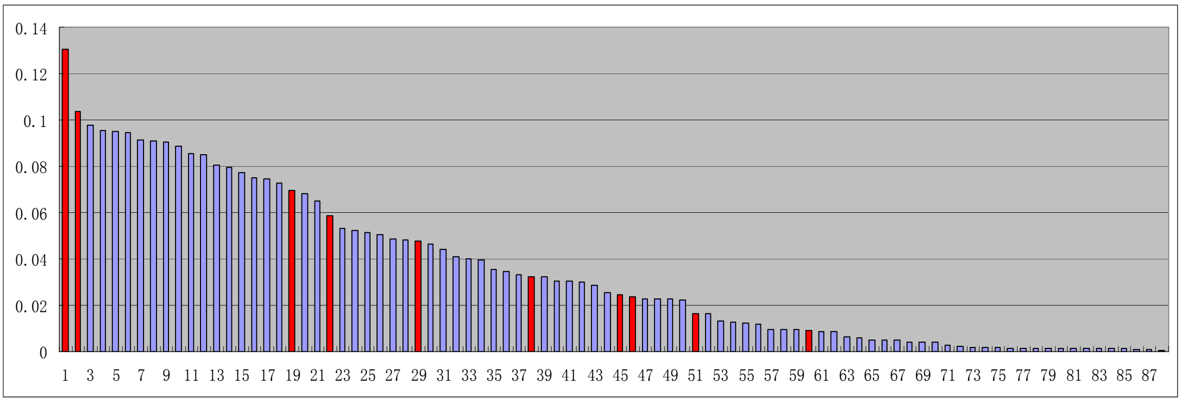


Figure 1.2 The performance of 10 submitted runs for automatic search. The red bars are from BUPT_MCPRL

For the text-concept mapping method, the MAP was greatly improved than the example-based retrieval. MCPR2 (0.070) jus with positive concept weight distribution was better than MCPR3 (0.059) with both pos- and negative concept weights. Using the same positive weight distribution method, MCPR1 with 60 HLF concept detectors reached 0.104, advanced than that of MCPR2 by 48.5%. From this comparison, we may deduce that the larger number of concepts in lexicons and the more accurate description for each concept would result in the better retrieval performance. At last, combining lexicons of both High-Level-Features and search topics, MCPR4 achieved the best MAP of 0.131, ranked as No. 1 among all the runs, which is shown in Figure 1.3

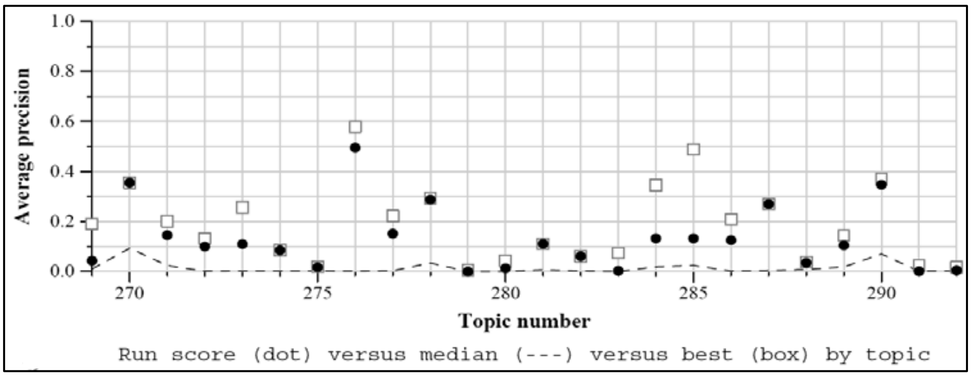


Figure 1.3 The performance of BUPT_MCPR4

2. High-level Feature Extraction

In this year, focus of our HLF system was on boosting and fusion of global features, local features and interest points, the difference of classifiers with cross-validation and re-ranking of results according to face detection. The results indicated that the performance of submitted 5 runs was not as good as last year.

2.1 Feature Representation

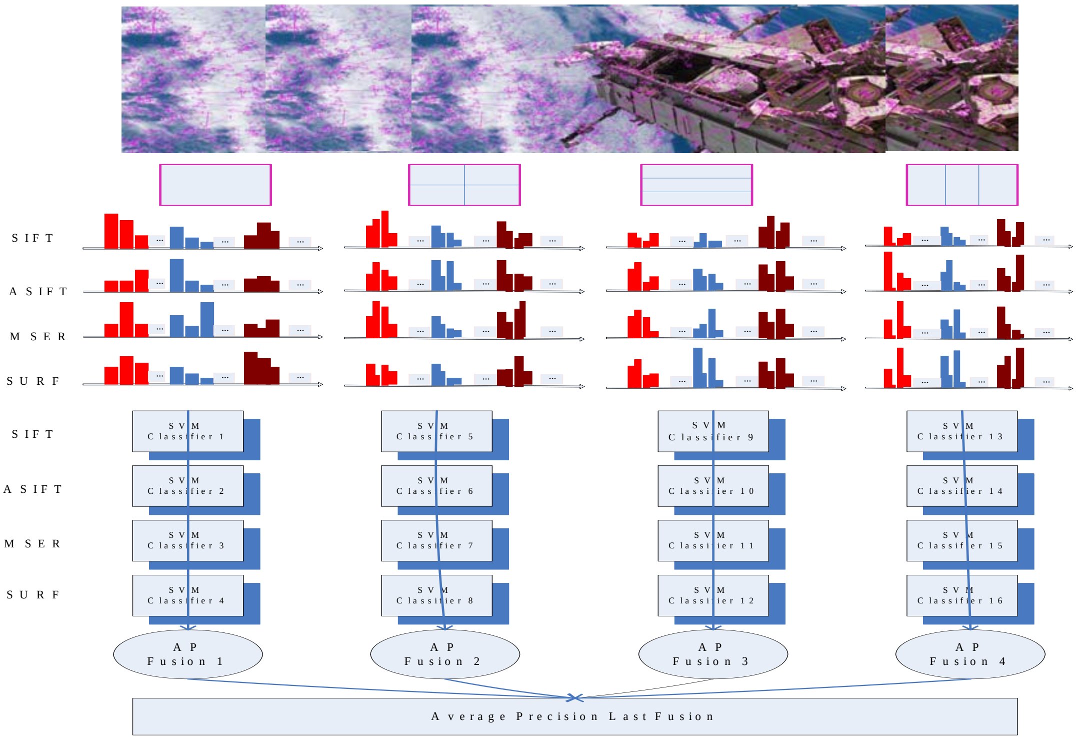


Figure 2.1 BoW representation of features

Some features such as SIFT[12], MSER-SIFT [13], ASIFT [14], SURF[15] were used in our system. Keypoints were extracted from salient image patches and were clustered into visual vocabulary in 500 words. In order to obtain spatial information, we divided images into different blocks and built the visual-word feature from each block. Finally, a late fusion approach was use to fuse these local features and the processing is shown in Figure 2.1. Except local features, some global features such as Gabor wavelet, Local binary pattern (LBP), HSV Color correlogram, RGB color moment were used.

2.2 System Framework

The framework of our system is shown in Figure 2.2. The tv7.sv.devel dataset was used to train classifiers, and tv7.sv.test dataset for validation and self-testing. Before SVM training, adaboost algorithm was first used to select feature vectors. Linear fusion was performed to produce final result. In addition, face detection was also considered to re-rank the results for different runs.

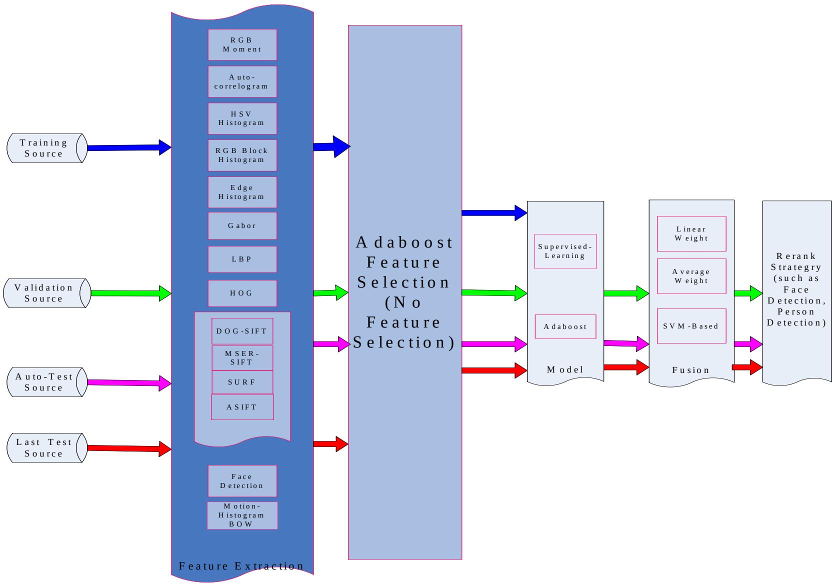


Figure 2.2 The system framework of HLF extraction

2.3. Experiments and Discussions

6 runs were submitted for evaluation, and corresponding performance and description are shown in Table 2 and Figure 2.3. According to Figure 2.2, we can see that Sys4 achieves the best infMAP, 0.487, among our submitted 6 runs. Though the infMAP is not satisfactory, we find some significative conclusions:

- During the training, multifold cross-validation is important for performance improvement. For example, the MAP of Sys5 which applied cross-validation is better than Sys6 23.99%.
- Selection and fusion of global feature is crucial for HLF extraction. For instance, performance of Sys3 outperforms Sys4 by 55.87%.
- Spatial information consumes a great deal computing resources, however it only achieves a limited improvement far less we expect.
- Bad face detection would worsen re-ranking results, such as Sys2.
- Keypoint feature is effective for detection of rigid objects, but it is also sensitive for some concepts which involved “people”.

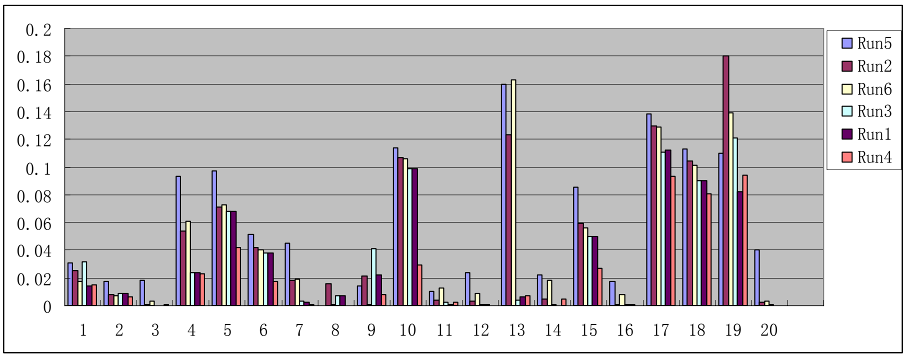


Figure 2.3 Average Precision of each concept for each run.

3. Copy Detection

In this paper we describe our two methods and evaluation results for copy detection. The first system framework based on SIFT is shown in Figure 3.1. It mainly consists of three parts: feature extraction, inverted file generation, step of scoring and voting strategy.

3.1.1 Approach Based on SIFT

SIFT is chosen as the local feature. Taking into account the computational complexity, visual features are extracted only in keyframes. In first strategy, one frame per second was extracted as the keyframe. The traditional matching method between points described by SIFT is to calculate the ratio of the most nearest distance and the second nearest distance among all point pairs from two images, and two points are matched if this ratio is below a pre-defined threshold. While enough points between the images matched, it is said that the two images matched. This method is accurate but the matching speed for large-scale searching is unacceptable. So a visual vocabulary is built for the need of searching in a large database. The visual vocabulary is generated by clustering the 4000000 descriptors which is extracted from the database videos using the K-means algorithm.

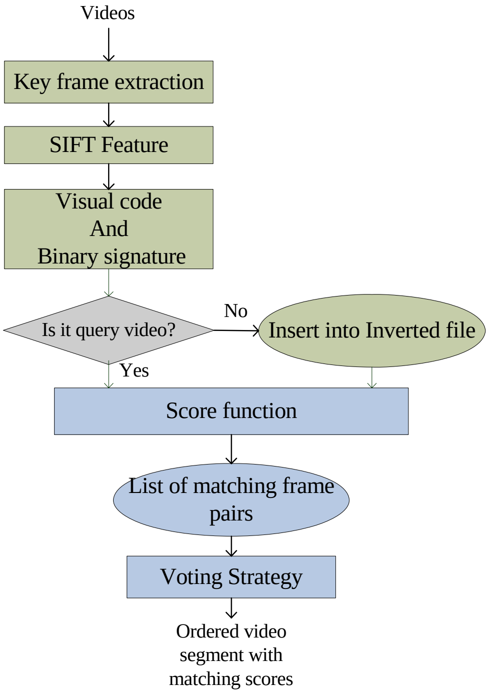


Figure 3.1 Framework of video copy detection

3.1.2. Filtering

After generate the codebook, descriptors are assigned to their closest codes both for query video and database video. As a result, a 128-dimensional descriptor is quantized into a 1-dimensional code index. There may be several point pairs that are not really matched but having the same visual code. In the following steps, we will try to filter out these mismatched points. Here, the ordinary signature of an interest point is proposed. The neighborhood of an interest point is divided into 4x4 sub-blocks, and the ordinary signature is with 16 dimensions. Refer to [16], a weighted similarity score (WOSS) of two points could be obtained.

Refining the score is as the further step because many wrong matches are still existed. And the filtering results are seen in Figure 3.2, where (a) shows point pairs with the same code, (b) is the result filtered from (a) by using OS, (c) is the result filtered from (b) by using quantized angle, and (d) is the final matching result filtered from (c) by using quantized log-scale. And we can say the results are significantly improved.

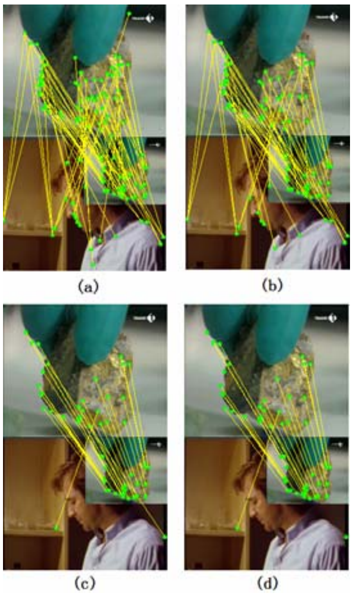


Figure 3.2 Process of filtering points

3.1.3 Voting Strategy

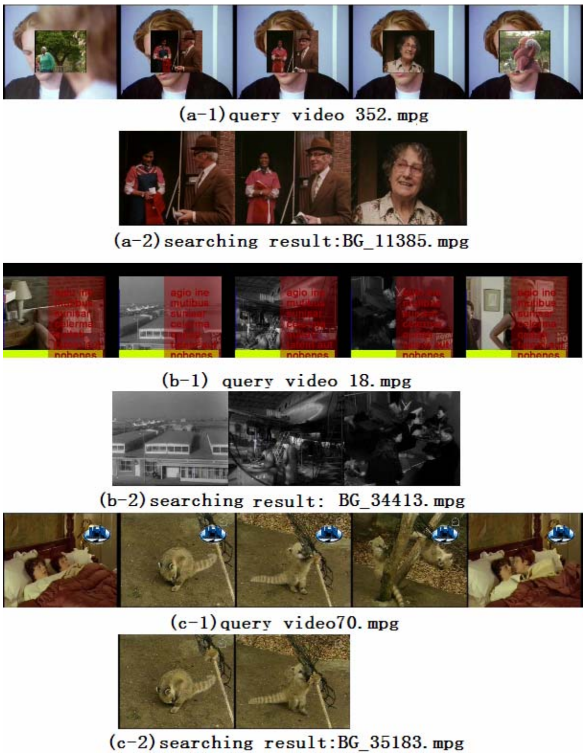
Until the step of voting strategy, we have obtained a set of matched frame pairs between the query video and database video. Because of the consistence in temporal order, each frame pair in the same matching segment of query video and database video should have the same difference $\delta_{frameNo_i - frameNo_j}$ in frame numbers.

Suppose $(q_1, d_1), (q_2, d_2), ..., (q_n, d_n)$ are n matched frame pairs between a query video and a database video, their difference in frame numbers can be calculated as $\delta_i = d_i - q_i = a, \quad i = 1,..,n$, where a is a constant. Let fs_i be the matching score of fame pair δ_i and let $fs = \min(fs_i)$. A score sequence s is projected to accumulate scores like this:

$$s[i] = \begin{cases} 0, & i = 0 \\ s[i-1] + fs_i, 0 < i \le n \, and \ fs_i \ge threshold \\ s[i-1] - fs, \ 0 < i \le n \, and \ fs_i < threshold \end{cases} \quad (1)$$

Where $s[i]$ is the accumulate score of frame pair matching scores from 0 to i.. Then, we find the maximum value in s (let it be $s[k], 1 \le k \le n$), and go back from this point until the value of $s[i]$ reduced to zero (let it be $s[w], 1 \le w \le k$). We can say that the segment $(q_{w+1}, ..., q_k)$ matches to the segment $(d_{w+1}, ..., d_k)$, and their matching score is $s[k]$. Find all matching segment, order them by the matching scores, and the shortlist of matching segments is generated for each video in database.

Different thresholds in the filtering steps and the voting step are chosen for our two runs. And some experimental results which were tested on tv8.sv.test are shown in Figure 3.3.


(a-1)query video 352.mpg

(a-2)searching result:BG_11385.mpg

(b-1) query video 18.mpg

(b-2)searching result: BG_34413.mpg

(c-1)query video70.mpg

(c-2)searching result:BG_35183.mpg

Figure 3.3 Examples of copy detection

3.2.1 Approach based on SURF

In this approach, a fixed number of keyframes per time unit were extracted for query and testing datasets: one keyframe per 40 frames for the query and one keyframe per 50 frames for testing dataset. And then, SURF feature[3] was extracted and visual words with size of 2000 was generated. During the course of testing, Hamming embedding was used to improve the preciseness. If the corresponding points are mapped in the same visual word, and the Hamming distance is calculated, then the score between two points is given: higher scores to smaller Hamming distances [17]. Figure 3.4 shows the flowchart of this algorithm.

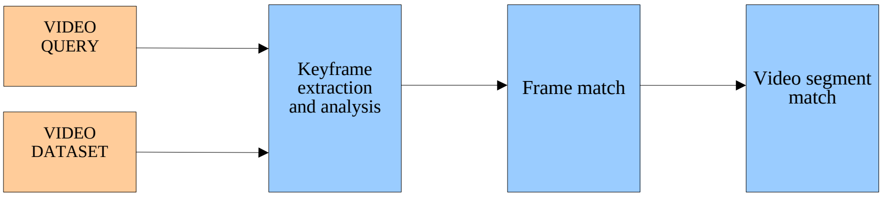


Figure 3.4 Flowchart of CBCD system based on SURF

3.2.2 Frame match and video segment match

Each couple of points are given a score [19] in the comparing keyframe through codebook and Hamming distances. We add all these scores and normalize them. If the sum is bigger than a threshold, they are matching.

Smith Waterman algorithm [18] was used to compute the similarity of two video segments. This algorithm adopts iterative method to compare the similarity of all possible scores, and then go back through the method of dynamic programming to find the optimal similarity comparison. Two steps were taken. Firstly, frame scores are used to construct score-matrix by SW algorithm and N score-matrix is obtained for each video query. N represents the number of videos in the video dataset. Secondly, in each score-matrix, the highest score is found and utilized to normalize the score-matrix. Thirdly, best matching path is determined by dynamic programming.

3.3 Experiments and Discussions

From the CBCD experiments of two approaches in TRECVID 2009, we found that:

- The performance of SIFT-based is slight better than SURF-based, Figure 3.5 shows the results.
- Features selection is still an open question and lots of miss detections appear.
- Video indexing should be reconstructed to speed up video search.
- Machine learning should be introduced.

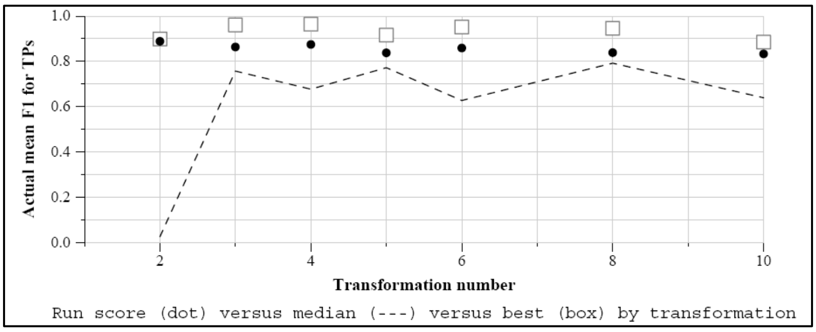


Figure 3.5 CBCD results

4. Event Detection

Challenges of event detection mainly on several-fold: the semantic gap between events and obtained motion patterns, the diversity of the same event shot from different viewpoint, and different degrees of occlusions and accidents, different event has different appearance. Thus, we independently detected 5 events according to event shape and motion patterns and submitted for evaluation this year.

4.1 System Framework

The system framework mainly depends on learning multiple one-against-all SVM classifiers for foreground object and experiential rules.

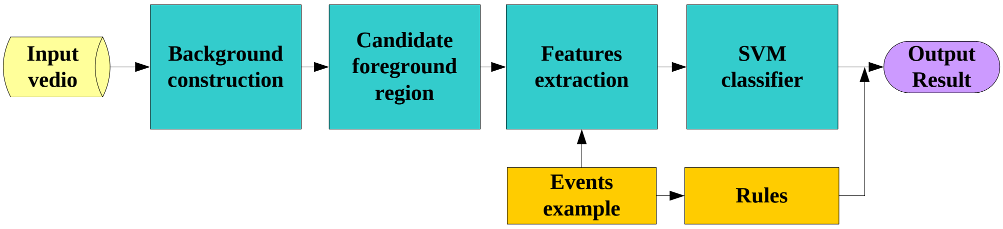


Figure 4.1 System framework of event detection

4.2 Detection method for Single Event

1. Person Runs

To detect this event, we focus on the special gesture detection of moving persons, as the gesture of a running person is quite different from other objects or walking ones. HOG feature is extracted to encode the action patterns, which used to train the model. For each region, we divide it into small cells of a fixed number. Then the gradients are calculated, and the angles of the gradients are classified into some sets. So each cell has some features, which respectively labels the distribution proportion of the angles. Main steps are as follows:

Step 1: Background construction. A dynamical refresh method is used to get the initial background.

Step 2: Candidate foreground rectangle. By subtracting the background value from the pixel value of the current frame, some candidate foreground regions could be obtained.

Step 3: HOG features. Morphological approaches are first applied, and then HOG is extracted.

Step 4: SVM classifier. SVM is applied to train one-against-all classifiers to determine whether the candidate foreground region is a moving person or not., and then SVM is reused to determine whether it represents a running object or not.

2. Pointing

Pointing event is a special gesture detection, which includes two-fold: person and action of pointing. The algorithm is as follows:

Step1: Frame difference. Frame difference was calculated to get the moving edge of each object, and then the movement of each small block's center of gravity. If the value of movement is bigger than a threshold, we determine that this area is a moving area.

Step2: Foreground rectangle. Some morphological approaches were done to smooth the moving area, and rectangles with proper size were labeled.

Step3: Color detection. According to the train data, we get the probability U/V component of fingers and hands, which can eliminate some wrong candidate foreground rectangles.

Step4: HOG features. HOG feature of each candidate region was calculated.

Step5: SVM classification. SVM classifier is applied to determine the final results.

3. Elevator No Entry

To detect this event, we focus on the status detection of the elevators, while detecting the persons by the elevator. The algorithm is as follows:

Step1: Elevator opened detection. By monitoring the elevator, if the elevator opens, we record the start time, meanwhile with the persons' status when they are standing by the elevator.

Step2: Elevator closed detection. To monitor the elevator, if the elevator closes, we also record the time, with the persons who still stand by the elevator.

Step3: Action detection. To compare the persons at the two different moments, if someone appears at both the two moments, we think that he is still there with elevator no entry.

4. Take Picture

To detect this event, we focus on the change detection of the intensity between two adjacent frames. We monitor the real-time intensity difference of the adjacent frames, and record the value as value 2. Then we compare value 2 with value1, if the different value is bigger than a threshold, we think maybe some flash appears. If the adjacent 3 frames' value 2 is all bigger enough than value1, we think a Take Picture event occurs.

The algorithm is followed:

Step1: Normal intensity difference. Get the average intensity of the first part of frames of the video, which is set to be the normal intensity difference value as value1.

Step2: Monitoring real-time frame difference.

Step3: Action detection.

5. Opposing Flow

To detect this event, we focus on the trajectory detection of moving persons. The algorithm is as follows:

Step1: Foreground detection. Background model is constructed to get the foreground.

Step2: Following person's trajectory. Foreground object is followed to get the trajectory of the moving.

Step3: Trajectory analysis. We analyze the person's moving trajectory to determine whether he is walking opposing others.

Step4: Action detection. If a person moves ahead of the different direction, we think he is in an opposing flow.

4.3 Experiments

Evaluation of above 5 events is not satisfactory, our system still exists many missing and fault detections.

References

[1] C.G.M. Snoek, I. Everts, J.C. van Gemert et al, “The MediaMill TRECVID 2008 Semantic Video Search Engine”, In: Proceedings of TRECVID 2008 Workshop.

[2] Tat-Seng Chua, Shi-Yong Neo, Yan-Tao Zheng et al, “TRECVID 2007 Search Tasks by NUS-ICT”, In: Proceedings of TRECVID 2007 Workshop.

[3] Yingyu Liang Liang, Xiaobing Liu, Zhikun Wang et al, “THU and ICRC at TRECVID 2008”, In: Proceedings of TRECVID 2008 Workshop.

[4] Shih-Fu Chang, Junfeng He, Yu-Gang Jiang et al, “Columbia University/VIREO-CityU/IRIT TRECVID2008 High-Level Feature Extraction and Interactive Video Search”, In: Proceedings of TRECVID 2008 Workshop.

[5] Xiangyang Xue, Hui Yu, Hong Lu et al, “Fudan University at TRECVID 2008,” In: Proceedings of TRECVID 2008 Workshop.

[6] David G. Lowe, “Object Recognition from Local Scale-Invariant Features”, Proc. of the International Conference on Computer Vision, Corfu, September, 1999.

[7] D. Lowe, “Distinctive image features from scale-invariant key points,” Int. Journal on Computer Vision, vol.60(2), pp:91-110, 2004.

[8] Gao Zan, Nan Xiaoming, Liu Tao et al, “A new framework for high-level feature extraction”, Industrial Electronics and Applications, pp2118-2122, 2009

[9] Xiaoming Nan, Zhicheng Zhao, Anni Cai et al, “A Novel Framework for Semantic-based Video Retrieval”, ICIS 2009.

[10] Juan Cao, Yong-Dong Zhang, Bai-Lan Feng et al, “TRECVID 2008 Search Task by MCG-ICT-CAS”, In: Proceedings of TRECVID 2008 Workshop.

[11] Apostol Natsev, Alexander Haubold et al, “Semantic Concept-Based Query Expansion and Re-ranking for Multimedia Retrieval”, In: Proceedings of MM’ 07.

[12] D. Lowe. Distinctive image features from scale-invariant key points. Int. Journal on Computer Vision, 60(2):91-110, 2004.

[13] J. Matas, O. Chum, M. Urban, T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions”, BMVC, 2002.

[14] J.M. Morel and G.Yu, ASIFT: A New Framework for Fully Affine Invariant Image Comparison, SIAM Journal on Imaging Sciences, vol. 2, issue 2, 2009.

[15] H. Bay, A. Ess, T. Tuytelaars, L. van Gool, "Speeded-up Robust Features (SURF)", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008.

[16] D. Bhat and S. Nayar. “Ordinal measures for image correspondence,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(4):415–423, 1998.

[17] Herve Jegou, Matthijs Douze, and Cordelia Schmid. “Hamming embedding and weak geometric consistency for large scale image search” European conference on computer vision, 2008

[18] Needleman S, Wunsch C. “A general method applicable to the search for similarities in the amino acid sequences of two proteins. Journal of Molecular Biology, 1970, 48(3):443-453.

[19] Matthijs Douze, Adrien Gaidon, Herve Jegou, “INRIA-LEAR’s video copy detection system” In TRECVID Workshop, November, 2008.