BUPT-MCPRL at TRECVID 2010*

Xin Guo, Yuanbo Chen, Wei Liu, Yuanhui Mao, Han Zhang, Kang Zhou,
Lingxi Wang, Yan Hua, Zhicheng Zhao, Yanyun Zhao, Anni Cai

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

Abstract

In this paper, we describe BUPT-MCPRL systems for TRECVID 2010. Our team participated in five tasks: semantic indexing, known-item search, content-based copy detection, surveillance event detection and instance search. A brief introduction is shown as follows:

A. Known-item search

In this year, we concentrated on the concept-based retrieval, and proposed several methods to improve the searching results. All 4 runs we submitted are described in Table 1.

Table 1 KIS results and description for each run

Run ID	Mean Inverted Rank	Description
F_A_NO_MCPRBUPT1_1	0.294	This run is based on text.
F_A_NO_MCPRBUPT2_2	0.004	This run is based on 86 concepts.
F_A_NO_MCPRBUPT3_3	0.004	This run is based on 86 concepts and several boosting approaches are adopted.
F_A_NO_MCPRBUPT4_4	0.002	This run is based on 86 concepts and co-occurrence matrix is used to expand the scope of concepts selected from one topic.

B. Instance search

An automatic instance search system was proposed for this pilot task. The topics were divided into three categories and different search method was proposed for each of them. We mainly focused on the topics about character and person and got the overall highest infAP for some of them.

C. Semantic indexing

In this task, several visual features were tested and 4 fusion strategies were adopted. 4 runs were submitted and the results are listed in Table 2.

Table 2 SIN results and description for each run

Run ID	InfMap	Description
L_A_MCPRBUPT1_1	0.0295	This run uses all features for each concept and average precision (AP) of each feature is employed as weighted parameter for multimode fusion.
L_A_MCPRBUPT2_2	0.0127	This run uses only one feature with best performance.
L_A_MCPRBUPT3_3	0.02	This run uses best three features and AP is also used just as Run 1.
L_A_MCPRBUPT4_4	0.0312	This run is similar with Run 1, but the weighted parameters are modified (AP, AP^2, AP^4 and AP^8 were tested).

* This work was supported by National Natural Science Foundation of China under Projects 60772114 and 90920001, and by Fundamental Research Funds for the Central Universities.

D. Content-based copy detection

Two approaches for the content-based copy detection task were proposed: one based on SIFT and global feature and another based audio.

E. Surveillance event detection

We focus on 4 events: PersonRuns, Pointing, ObjectPut and Embrace. Firstly our system detects the heads of people from video frames to construct the initial objects of system. Then the system traces the objects and detects new objects from the subsequent frames. Finally, the system extracts the features from these objects and decides if some event occurs based on SVM classifiers and decision rules.

1 Known-item search (KIS)

For the known-item search task, we concentrated on the concept-based retrieval, and proposed several methods to improve the searching result. In our system, two approaches are employed, one of which is based on text and another is concept-based. Both approaches are illuminated in the following part respectively.

1.1 Text-based approach

The proposed automatic text-based search system is consisted of several main components, including textual query analyzing and pre-processing, text-based retrieval, Content-based Retrieval, result fusion and re-ranking. The framework of our KIS system is shown in Figure 1.1.

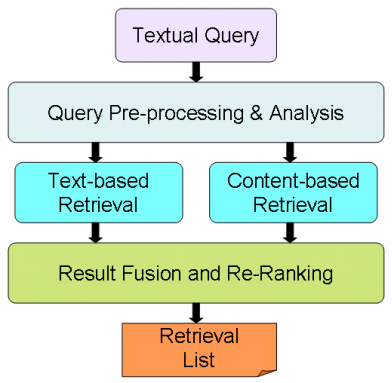

Figure 1.1: The framework of text-based approach

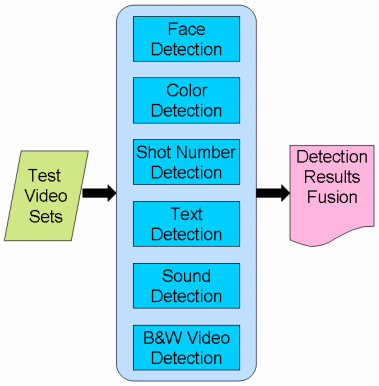

Figure 1.2: Content-based retrieval

1.1.1 Useful Information Selection

By analyzing the textual queries of all KIS topics, we mainly chose some information useful for this search task, including metadata of test videos, the donated ASR (automatic speech recognition) data [3], shot number of each video, information on whether there are faces, particular colors, text, sound, and black and white video clips in videos.

1.1.2 Retrieval Method

Since some queries overlap with the metadata of videos, and are perhaps very easy to tie to the target videos, and the system relying on the video content may be very difficult, the text-based retrieval method with ASR data and metadata was firstly used for each topic query. Through analysis of the textual queries, content-based retrieval method was used to re-rank the result by text-based retrieval.

a. Text-based Retrieval

In text-based retrieval, ASR data and metadata of the test videos were used and the indexing and searching tasks were done by Lucene. The ASR data and metadata were first converted to a stream of plain-text tokens and then indexed. The queries were pre-processed and searched for in the index.

b. Content-based Retrieval

The contents appearing frequently in queries were extracted or detected in the test video sets and then the results were fused. The framework of the content-based retrieval is shown in Figure 1.2. The contents detection processes are described as follows:

- **Face Detection:** A fully automatic face detection system was proposed to find whether or not there are any faces in a given keyframe image and, if present, return the image location and content of each face [5].
- **Color Detection:** We detected colors from test video sets in HSV color space. Color of all parts of human bodies described in textual queries was detected in regions basing on face detection results in image and other color information in queries was detected in the global video keyframe images.
- **Shot Number Detection:** Shot number of each test video was got from the master shot reference.
- **Text Detection:** Text detection system was proposed to find whether there were any texts in the videos and, if present, return video position of the text clip.
- **Sound Detection:** This system was used to check whether there was sound in the videos.
- **B&W Video Detection:** This system was used to check whether the video was a black and white video.

1.1.3 Result Fusion and Re-ranking Strategy

For each textual query, some of the detectors in content-based retrieval system were first used to filter the videos impossible to be the target ones and then the results of these models were fused. The fused result was used to re-rank the result by text-based retrieval system. We used the following re-ranking method: Order of the top 10 results by text-based method is unchanged and the sort results form 11 to 100 are re-ranked by fused content-based result. The performance of this automatic run with metadata is shown in Figure 1.3. The ranked automatic submission results with metadata form all teams according to mean inverted average rank are shown in Figure 1.3. The red bar is from BUPT_MCPRL.

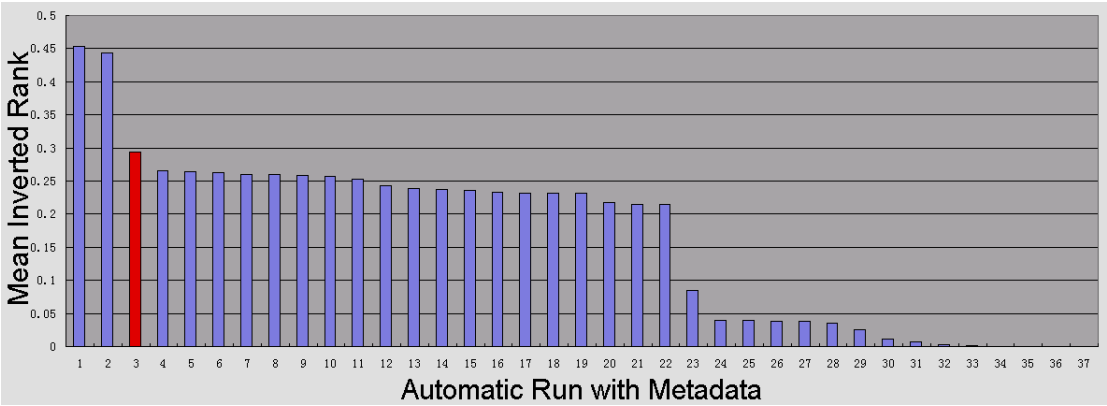

Figure 1.3: The performance of automatic run with metadata

1.2 Concept-based approach

In the concept-based approach, we took three measures to boost the performance of our searching system: black and white video detection (B&W Detection), music and human voice detection (Audio) and motion detection (Motion), see Figure 1.4.

Moreover, in the beginning of our system, no more than 5 key words were selected carefully per topic because of that we thought the "visual cues" supplied by organizer were not reliable.

1.2.1 Concept detectors

In the SIN task, a great deal key frames in the training set were labeled as 130 classes (concepts), and we deleted or merged some classes contained too few key frames. At last, we trained 86 concept detectors totally.

1.2.2 Boosting approaches

- B&W detection: a detector which can detect whether the video is black and white or not.
- Audio: including two detectors, music detector and human voice detector, these can detect whether the background audio contains music or voice; MFCC feature is used in both detectors [12].
- Motion: several statistical features of motion are generated after optical flow detection.

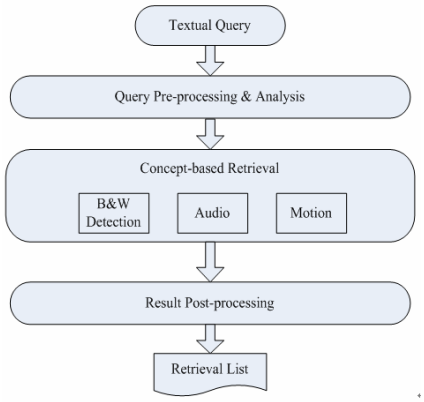

Fig. 1.4: The framework of concept-based approach

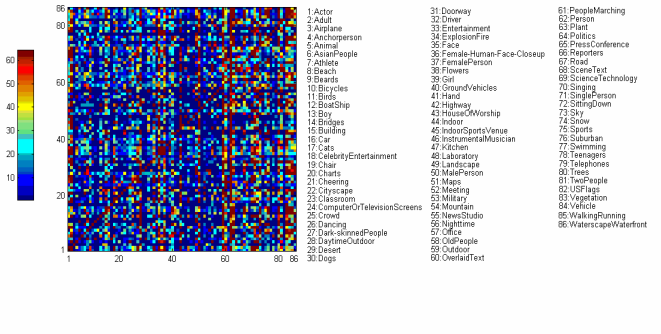

Figure 1.5: Co-occurrence Matrix of 86 Concepts

1.2.3 Co-occurrence Matrix of Concepts

Instead of WordNet Similarity package, co-occurrence matrix was used to expand concepts for per topic, and we thought that the probability of co-existence between concepts is more effective. The co-occurrence matrix of 86 concepts is shown in Figure 1.5.

1.3 Experiments and Discussions

This year, we submitted 4 automatic searching runs, one is based on text, and the others is based on concepts. The results and description of all 4 runs is listed in Table 2.

Compared with other 3 concept-based runs, the first text-based run gained the highest Mean Inverted Rank, and there is no doubt that text-based method has better performance than visual-based in the area of searching technology currently.

Although the Mean Inverted Rank of run 3 is equal to that of run 2, there is some improvement in run 3 indeed, see Table 3.

We receive worse result in run 4 used co-occurrence matrix than run 2, and we believe that the reason is all concepts were referred in our system, and 3~5 concepts would be better.

Table 3: The rank of videos found correctly in run 2 and 3

Topic	F_A_NO_MCPR BUPT2_2 rank	F_A_NO_MCPR BUPT2_3 rank	Topic	F_A_NO_MCPRB UPT2_2 rank	F_A_NO_MCPRB UPT2_3 rank
12	16	16	137	16	16
60	84	55	145	X	44
63	X	92	155	99	99
76	71	X*	157	81	81
81	49	29	178	78	38
95	X	92	216	38	38
105	33	12	223	57	57
111	15	15	229	X	9
122	8	8	241	63	63
128	X	58	266	31	31
132	2	2	273	22	22

X -- Not found

X* -- The audio of Video 3477 was not extracted perfectly with the FFMpeg package.

2 Instance Search

The proposed automatic instance search system is consisted of several main components, including visual query pre-processing, face detection, visual features extraction from the region of interest (ROI) of the frames, retrieval by Euclidean distance based on visual features, multimodal fusion and results re-ranking. The framework of our INS system is shown in Figure 2.1.

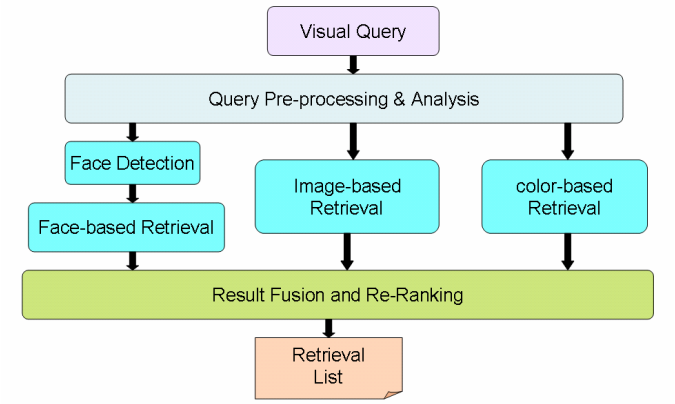

Figure 2.1: The framework of automatic instance search system

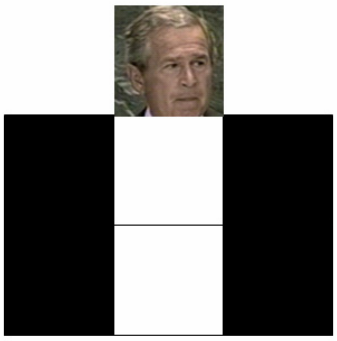

Figure 2.2: ROI by face information

2.1 Feature Selection

Since no unique visual feature can represent all information contained in a keyframe, and no given visual feature is effective for all topics, we extracted several visual features at regional and global levels[6, 7, 8], the details of which are listed in Table 4.

By analyzing the visual queries of all topics, we proposed three different search methods: face-based retrieval, keyframe image-based retrieval, body color-based retrieval. We mainly focused on the face-based retrieval.

Table 4: Selected visual features

Features	Description
HSV Histogram	HSV color histogram with 3*3 regional partition
Gabor Wavelet	3-scale and 6-direction Gabor feature with 3*3 regional partition
Edge Directional Histogram	145 dims histogram by concatenating global and regional EDH
Local Binary Pattern	256 dims histogram of each LBP code with global partition
HSV_Correlogram	Color Auto Correlogram feature with global partition
Black and White Information	share of black and white pixels

2.2 Face-based Retrieval

Before this retrieval method, a fully automatic face detection system was proposed to finding whether or not there are any faces in a given keyframe image and, if present, returning the image location and content of each face [5]. Each face region in image was expanded at different scales to make sure the hair and the whole head was included in the new face regions. Four visual features including HSV Histogram, Gabor, EDH and LBP were extracted from the new face regions. The same strategy was used in some of visual query examples of the INS topics. The similarity between the query and each video clip keyframe was computed by Euclidean distance based on visual features of the face region. Topics searched by this retrieval method are: 9001~9012 and 9014.

2.3 Image-based Retrieval

In the image-based retrieval, HSV_Correlogram and LBP features were extracted at image global level and the similarity was computed by Euclidean distance [9]. Topics searched by this retrieval method are: 9013, 9014 and 9016~9022.

2.4 Body Color-based Retrieval

Body color-based retrieval was proposed specifically for the topic 9015. In this method, face detection result was got for each video clip keyframe. By the face information of image we got the body part of human as shown in Figure 2.2: the white and black part below the face is the ROI (region of interest). The ratio of width between the white rectangle and the face part is 1:1 and the ratio of height is 2:1, the two black parts next to the white rectangle have the same size as the white one. The percentages of the black and white pixels in the ROI were calculated, and by them the result was determined.

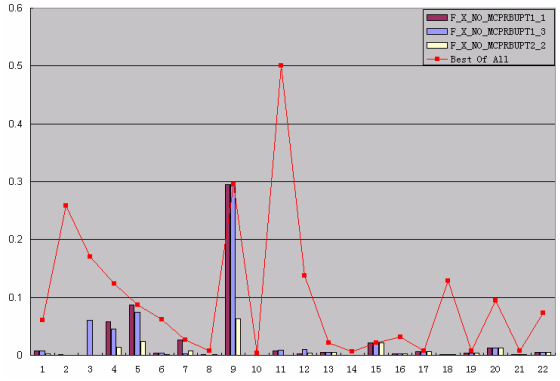

Figure 2.3: 3 runs for automatic instance search

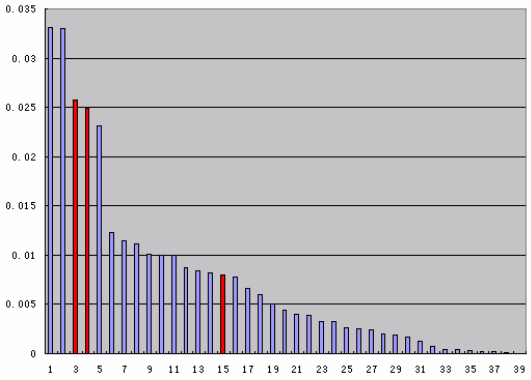

Figure 2.4: instance search task submission

2.5 Result Fusion and Re-ranking Strategy

In the face-based retrieval, results by different visual features are fused and ranked. The same fusing and ranking strategy was first used in image-based retrieval, then the retrieval results were re-ranked by the information whether or not the topic examples are closely related with human face. The percentage results of the black and white pixels in the ROI were used to rank in body color-based retrieval.

2.6 Experiments and Discussions

In this task, we submitted 3 automatic runs and the performances are shown in Figure 2.3.

F_X_NO_MCPRBUPT1_1: Only image examples given by the INS topics were used and for each topic one or more image examples was chosen. In the face-based retrieval, results by different visual features are first fused and then ranked. This run obtained a mean infAP of 0.025, with the overall highest infAP for 4 topics: 9005, 9007, 9009, and 9015.

F_X_NO_MCPRBUPT2_2: In the face-based retrieval, each result by visual feature was first ranked and then fused and the results tend to lag behind those in F_X_NO_MCPRBUPT1_1. This run only achieved a mean infAP of 0.008.

F_X_NO_MCPRBUPT1_3: In the face-based retrieval, only one image was chosen for each topic and for some topics (9002, 9003, 9011, 9012, 9014), one web image was used as the topic example. For other topics the same strategies are used as in F_X_NO_MCPRBUPT1_1. This run obtained a mean infAP of 0.026, which is better than F_X_NO_MCPRBUPT1_1. Compared with the results in F_X_NO_MCPRBUPT1_1, result of topic 9003 is obviously improved, while results for other topics are not very good overall.

The ranked automatic submission results form all teams according to mean inferred average precision are shown in Figure 2.4. The red bar is from BUPT_MCPRL.

3 Semantic indexing (SIN)

In the semantic indexing task, we focused on the visual feature extraction: 12 features were examined and several fusion strategies were adopted.

3.1 System Framework

Our visual based semantic indexing system consists of three components: feature extraction, classification and fusion, which is shown in Figure 3.1.

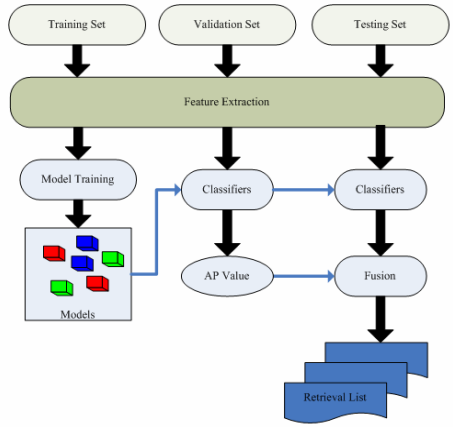

Fig 3.1: The framework of semantic indexing system

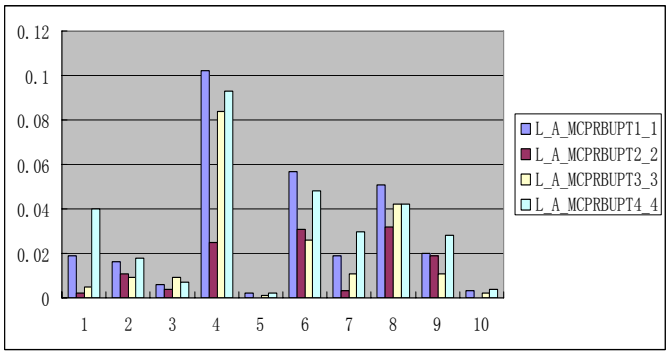

Figure 3.2: Average Precision of each concept for each run

The frames labeled by active participants were divided into training set and validation set, the former was used to train models and the latter was for performance comparison of models. First of all, we trained a series of models on the training set for each concept, and then average precision (AP) was generated on the validation set, which was an evaluation criterion for models trained before and linear weighted parameter in the fusion step later. The testing set was processed as before and the final SIN result was generated with three fusion methods.

3.2 Feature Extraction

We extracted 12 low-level visual features from the frames labeled by active participants: RGB_Moment, RGB_BlkHist, HSV_CorHist, Gabor, Gabor_Sort, HoG_Blk, LBP, EDH, SIFT, SURF, HoG_Edge and CSIFT [1, 2, 6, 7, 8]. At last, 10 features with better performance were selected in our system, which are listed in Table 5.

3.3 Multimode Fusion

For each concept, three fusion strategies were employed, one with all features, one with best three features, and the other one with best feature. For each concept, the final retrieve list was based on the probability that were generated by SVM classifiers [10] and linear weighted with APs.

3.4 Experiments and Discussions

We submitted 4 runs this year, and the training type of which were type "A" [4]. The description and performance of all runs are shown in Table 1 and Figure 3.2 respectively.

From the Table 1, we find that the fourth run worked very well, but it is meaningless to compare this run with other 3 runs for the reason that 4^{th} one had been modified manually. However, it is obvious that semantic indexing with artificial factors has better performance than that automatic.

The performance of run 1 is greatly improved than that of run 2, as the result of run 1 referred more features. The MAP of run 1 is very close to run 3, so we deduce that single feature can not meet the demand for semantic indexing, 3 ~ 5 would be suitable.

Table 5 Selected low-level features

Features	Description
RGB_Moment	RGB color moment feature
RGB_BlkHist	RGB color histogram with 3*3 regional partition
HSV_Correlogram	HSV color correlogram feature
Gabor	Statistical features of 3-scale and 6-direction Gabor Transform with 3*3 regional partition
LBP	Local Binary Pattern
Edge Directional Histogram (EDH)	Histogram of global and regional EDH
SIFT	SIFT feature and BoW method with 1000 visual words
SURF	SURF feature and BoW method with 1000 visual words
HoG_Edge	HoG feature at the edge and BoW method with 1000 visual words
CSIFT	C-SIFT feature and BoW method with 1000 visual words

4 Copy Detection

In the following sections, we describe our two framework and evaluation results for copy detection. The first part is about the

video copy detection system based on SIFT and Global Feature. Audio copy detection method is shown in part two, and we'll see the result and analysis in part three.

4.1 Video Copy Detection Based On SIFT

4.1.1 Framework

SIFT is chosen as the local feature. To make the computation efficient, visual features are extracted only in key-frames. In this year, one frame per second was extracted as the key-frame. The matching method is to find SIFT points which are most likely matching the original feature points in the query.

Since the number of the records in database is large than hundred of million, we can not travel all the data. So a visual vocabulary is built for the need of searching in a large database. After the filtering step using the vocabulary, the references are voted by the left points. The framework of our system is shown in Figure 4.1.

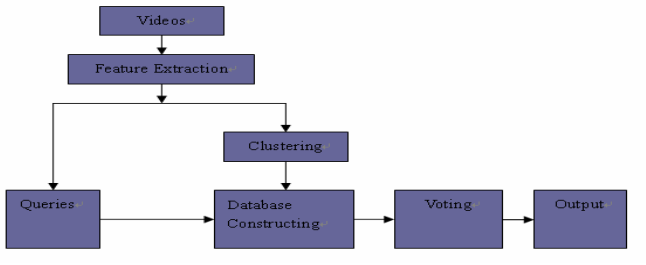

Figure 4.1: Framework of video copy detection system

4.1.2 Vocabulary Generating

The 50000-scale visual vocabulary is generated by clustering the 10,000,000 descriptors which is extracted from the 256-objects image database the K-means algorithm. Different to the traditional training method, because of the large scale of the vocabulary, we trained the codebook via a multiple computation plat named MPI [11]. 40 CPUs were set up to carry the computation process. There was one main node calculating and updating, and the other just assigned the training samples to their closest centers and return the center id to the main node. The structure is shown in Figure 4.2.

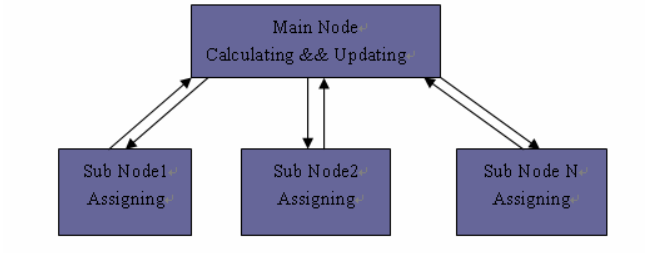

Figure 4.2: Structure of vocabulary generating system

4.1.3 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 limit the influence of these

mismatched points. Here, the signature of an interest point is proposed. We compare the values of the descriptors between the points pairs, and the signature with 128 bits is generated, after that, a weighted similarity score (WOSS) of two points could be obtained. This score will be used in the next voting step if it's above a pre-defined threshold.

4.1.4 Global Feature

Since the amount of global feature points generated are not so large, we do not have to apply any index structure to it, we just match the query and reference pairs instead, and the same to SIFT detection framework, a voting step is used to get the final result.

a. Local Binary Pattern

In the conventional LBP approach, the image pixels are first labeled as a binary class by thresholding the difference between the center pixel and its neighbors using the step function u(x) (i.e. u(x) = 1 when x⩾0 and u(x) = 0 otherwise). The concatenation of the neighboring labels is then used as a unique descriptor for each pattern. Figure 4.3 gives a simple example. The patterns are uniform if the transitions between "0" and "1" are less than or equal to two. For example, 01100000 and 11011111 are uniform patterns. The histogram of the uniform patterns in the whole image is used as the feature vector. It has been proven to be effective for both face recognition and facial expression recognition applications.

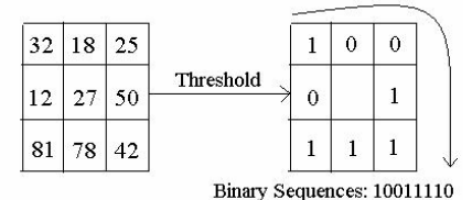

Figure 4.3: An example of LBP feature

b. Histograms of Oriented Gradients

The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. In practice this is implemented by dividing the image window into small spatial regions (*cells*), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form the representation. For better invariance to illumination, shadowing, *etc*., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram (energy) over somewhat larger spatial regions (*blocks*) and using the results to normalize all of the cells in the block. We will refer to the normalized descriptor blocks as *Histogram of Oriented Gradient (HOG)* descriptors.

c. HSV Correlograms

HSV color space is supposed to provide better correspondence with human visual perception of color (dis)similarities than RGB color space, for example. We explore different quantization of the HSV color space and try to make the correlogram more sensitive to changes in color content and less sensitive to illumination, by quantizing the hue component more precisely than the value component.

Hence we are trying to quantify to which extent HSV correlograms are able to discriminate semantic categories of images.

4.1.5 Voting

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 in frame numbers.

Suppose (q_1, r_1), $(q_2, r_2) \ldots (q_n, r_n)$ are n matched frame pairs between a query video and a database video, their difference in frame numbers can be calculated as $d_i - q_i = a$, $i = 1,2,\ldots n$ where a is a constant. Let fs be the matching score of fame pair and let $fs = \min(fs_i)$. A score sequence s is projected to accumulate scores like this:

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

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 $(q_{w+1}, \ldots, q_k)$ matches to the segment $(d_{w+1}, \ldots, d_k)$, 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 four runs.

4.2 Audio Copy Detection

4.2.1 Framework

An effective copy detection system usually includes two different aspects: the detection precision and the runtime. As a result, in this paper, we emphasize the content description and the indexing scheme. An overview of the proposed framework is shown in Fig.4.4, including low-level feature extraction, indexing and matching schemes and parallel computing.

Two main processes can further describe the framework. Audio fingerprints from the queries and test audios are first extracted and stored into feature files. And then the fingerprints from test set are organized into a database through a certain structure, at the same time, an index file is also generated for the next searching step. All the work is finished off-line.

After that, the searching process starts. The fingerprints of queries first are obtained from feature files. Then a matching and voting scheme is used to search queries in database.

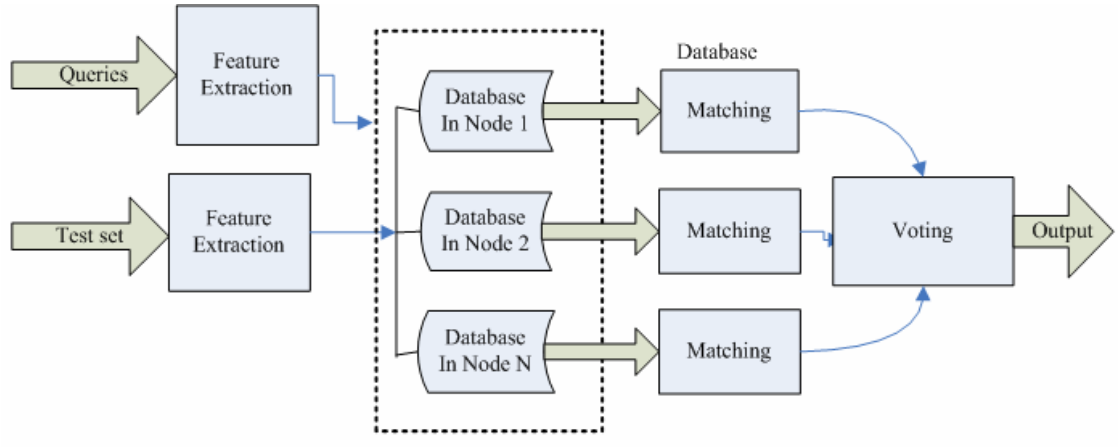

Figure 4.4: Framework of the audio copy detection system

4.2.2 Feature Extraction

As shown in Figure 4.5, the input X, digital PCM signals of audio clips which are sampled and quantized from analog signal, is lowpass filtered to 4 KHz and divided into windows of 1024 PCM samples with 512 samples frame advance in order to eliminate the influence of noise which mainly distribute in higher frequency bands. Here, in order to filter signals in time domain, a FIR lowpass filter named Equiripple is applied, and corresponding coefficients of the filter can be calculated. Additionally, since we have to match between the queries and tests in section or in other words under frames, a pre-emphasis of 0.97 is applied and then multiplied by a Hamming window before computing the Fourier transform.

$$Mel(f) = 2595\log(1+\frac{f}{700}) \tag{2}$$

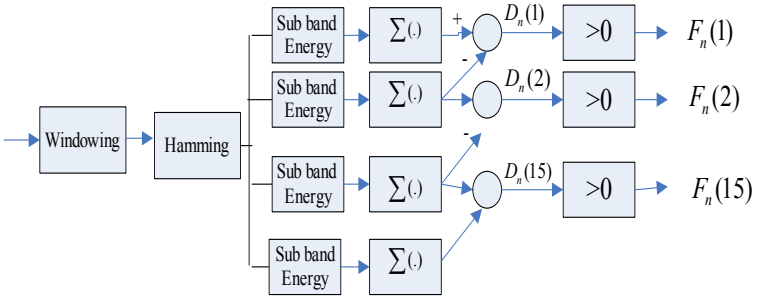

Figure 4.5: Flow chart of feature extraction

After transforming audio signal from time domain to spectrum, we divide the spectrum between 300Hz and 3000Hz into 16 sub-bands by using Mel-scale. The conversion between Mel-frequency and natural frequency is defined as follows:

$$F(n,m) = \begin{cases} 1, EB(n,m) - EB(n,m+1) > 0 \\ 0, EB(n,m) - EB(n,m+1) \le 0 \end{cases} \tag{3}$$

A triangular window is then used to compute energy in each band. The energy differences between the sub-bands are employed to compute the fingerprint. Suppose EB (n, m) represents the energy value of the nth frame at the m^{th} sub-band, then the m^{th} bit F (n, m) of 15-bit fingerprint is given.

In our framework, a 15-bit fingerprint is generated from consecutive sub-band and the consecutive frame difference is used. We extract 15 bits fingerprint from each frame. It is because 15 bits more robust to bandwidth limitations and extraneous speech addition. In addition, we can obtain more frequent repetition of the fingerprints even for the transformed audio. Since a 15-bit EDF fingerprint can be represented just by a Hash value, we call it energy difference fingerprint.

4.2.3 Matching Scheme

Every query has a sequence of Hash values in a feature file. For every query file, a voting table is created. This voting table holds a vote that is calculated by counting the number of the equal time differences between the matching points of query and reference data. For example, for the 3rd and 5th Hash values in the query, the corresponding matching in the reference file are the 10th and 12th Hash values. Accordingly the voting table records 2 for the difference 7. And if there are multiple matches with the same difference of 7, the voting value will be increased. And then, the sequential matching is carried out within the reference data to locate the query. Meanwhile, the voting table also keeps the first and the last indices of the corresponding difference value to determine the query. The voting function V is given in (4), which is defined to calculate the value obtained for the time differences between the query and the reference file.

$$V(\tau) = \sum_{\{r,q\} \in R} \delta(\tau - |r - q|) \tag{4}$$

In Equation (4), q and r are the time indices of the matching locations of the query and reference fingerprints respectively, and τ is the difference between the time indices.

The similarity for every difference value τ is calculated by dividing V (τ) by the difference of the first and last time index of the corresponding difference in seconds. The point with the highest similarity gives the most similar area for the reference and query data. In other words, similarity is calculated as the number of exact matches per second. After all the similarities of test set being calculated, they are sorted to get a list, and the ones whose scores are higher than a certain threshold will be output into the final result.

4.3 Result and Feature Work

From the CBCD experiments result of TRECVID 2010, we found that:

- The performance of 50000-scale codebook is better than the 2000-scale codebook used before.
- Audio method and global feature well complement the SIFT framework.
- Time location is a defect in the algorithm.

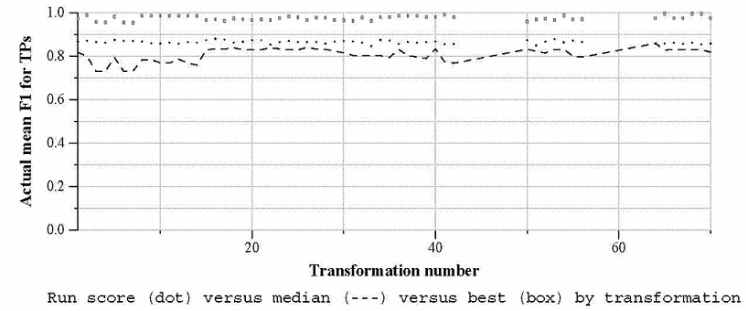

Figure 4.6: CBCD results

5 Event Detection

We focus on 4 events: PersonRuns, Pointing, ObjectPut and Embrace. The system framework is described as Figure 5.1. Firstly the system detects the heads of people from video frames to construct the initial objects of system. Then the system traces the objects and detects new objects from the subsequent frames. Finally, the system extracts the features from these objects and decides if some event occurs based on SVM classifiers and decision rules. The details of four events detection are presented as follow.

Figure 5.1: The diagram of human action detection

5.1 PersonRuns Detection

The method of PersonRuns Detection is constructed of several parts: human head detection, human head region tracing, trajectory analysis, optical flow calculation and PersonRuns decision. The main steps are as follows:

Step 1: Human head detection. We firstly find the possible points of head top by the gradient of video frames, then obtain a

region of interesting (ROI) from each point and extract the HSV feature and histogram of gradient (HOG) feature, finally decide if the ROI is human head by the SVM classifiers. So an object list of head region is formed for detection system. Each object is described by its HSV feature and HOG feature.

Step 2: Human head tracing. In the subsequent frames, the system detects new human head region and matches the feature of these regions with the objects in the list. For the matching object, system replaces its features by the new one detected from current frame. For the mismatching object, the system searches the matching region around the prediction position to find the matching object. So the system can trace the object from one frame to another and get the trajectory of objects.

Step 3: Trajectory analysis. The information of speed, distance, acceleration and linearity can be obtained by trajectory analysis. The decision score can be calculated by fusing these information.

Step 4: Optical flow calculation. The optical flow of each frame can be calculated by the method presented in [13]. The system detects the regions which are high coherence in optical flow orientation and their averages of optical flow amplitude are greater than the threshold selected by experience. Same as step 3, we can calculate the decision score by optical flow amplitude average of the region detected.

Step 5: PersonRuns decision. Fusion of scores from step 3 and step 4 can decide whether an event of person runs occurs.

5.2 Pointing and Embrace Detection

Pointing and Embrace events detection is described as Figure 5.2. For each candidate region after head detection, we calculate HOG descriptors based on raw images, motion edge histogram images (MEHI) [14] image cubes and gray image cubes. Then we train model files for Pointing and Embrace events depend on one-against-all SVM classifiers. On the test part, for each candidate region, the classification scores of three classifiers are combined linearly. If the combined confidence is larger than a threshold T, this frame is regarded as positive.

5.3 ObjectPut Detection

For ObjectPut event, we calculate optical flow though whole images, if there is down direction flows, we say that the ObjectPut event happened. As for the down flow is too small to get even when the ObjectPut event happened, this feature is not very useful, and we may think about other features to get better scores.

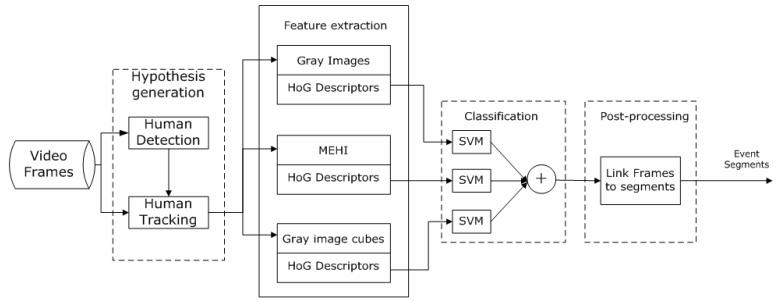

Figure 5.2: The diagram of Pointing and Embrace events detection

5.4 Conclusions

There are some problems in our SED algorithm needed to solve. The main one is to increase the precision of human head detection, and it's the key for improving the performance of SED system. The second one is to research the robust features to describe the human motion in events.

References

[1] Koen E. A. van de Sande, Theo Gevers and Cees G. M. Snoek, Evaluating Color Descriptors for Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (in press), 2010.

[2] David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

[3] J.L. Gauvain, L. Lamel, and G. Adda. “The LIMSI Broadcast News Transcription System”. Speech Communication, 37(1-2):89-108, 2002.

[4] A. F. Smeaton, P. Over, and W. Kraaij. “Evaluation campaigns and TRECVid”. In Proc. ACM SIGMM International Workshop on Multimedia Information Retrieval, pages 321–330, 2006.

[5] Hao Ji, Fei Su, Geng Du, “Multiple Face Tracking Based on Joint Kernel Density Estimation and Robust Feature Descriptors”, IC-NIDC2009.

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

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

[8] Zhicheng Zhao, Yanyun Zhao, Zan Gao, Xiaoming Nan et al, “BUPT-MCPRL at TRECVID 2009”, In: Proceedings of TRECVID 2009 Workshop.

[9] Online available: http://mmc.sice.bupt.cn/Project.php.

[10] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines, Version 3.0 2010. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.

[11] Du Zhihui, Li Sanli, Chen Yu, Liu Peng, “High performance computing parallel programming- MPI parallel programming “.

[12] Zheng Zhen，Wang Huaqing, “Improved MFCC Mel cepstral algorithm in Speech feature extraction”, Computer Engineering and Applications 44(22) 2008

[13] Andrés Bruhn, Joachim Weickert, Christian Feddern, etc.“Real-Time Optic Flow Computation with Variational Methods”. Lecture Notes in Computer Science, 2003, Volume 2756/2003, 222-229.

[14] Ming Yang, Shuiwang Ji, Wei Xu, etc. “Detecting Human Actions in Surveillance Videos”, In: Proceedings of TRECVID 2009 Workshop.