Fudan University at TRECVID 2008

Xiangyang Xue, Wei Zhang, Yuefei Guo, Hong Lu, Yuejie Zhang, Zichen Sun, Yingbin Zheng, Shile Zhang, Hong Liu, Yuanzheng Song, Jing Zhang*, Xisheng He, Kai Li, Jun Zhou, Yanjie Chen
School of Computer Science, Fudan University, China

1. Introduction

In this notebook paper we describe our participation in the NIST TRECVID 2008 evaluation. We took part in four tasks of benchmark this year, *i.e.*, surveillance event detection, high-level feature extraction, search and content-based copy detection pilot.

For Surveillance Event Detection, we chose 3 required events from the event set:

E05 PersonRuns
E20 OpposingFlow
E21 TakePicture

Motion information is used in the detection of PersonRuns and OpposingFlow, and the key of detection of TakePicture is to find the change of luminance.

For high-level feature extraction, we submitted one run: **FD_BAGGING**. We generated a number of weak learner based on different low-level features and different ratios of negative samples to the positive. A few of them were bagged together after selection by the validation set.

For search, we submitted 6 automatic runs:

FD_IMI_LK: This run is only based on the text from the English ASR/MT output provided by NIST and on the text of the topics.
FD_IMI_ZYB: This run is based on the text search and the visual expand from the text search results.
FD_IMI_HXS: This run is based on the concept mapping method.
FD_IMI_ZJ: This run is based on the fusion of concept mapping and visual search.
FD_IMI_ZW: This run is based on average fusion method.
FD_IMI_SZC: This run is only based on multi-model fusion method.

Focus of our system was on the effective utilization of text, visual features and HLF. Although it is the first time for us to participate in the automatic search task, our experience on interactive search and manual search provided us good knowledge of the state-of-art video retrieval systems and algorithms, from which

*Jing Zhang is with Department of Computer Science and Engineering, East China University of Science and Technology

our system benefited a lot and performed well in the official evaluation.

For content-based copy detection pilot, we submitted 2 runs:

FudanU.v.cdois: Using OIS(Ordinal Intensity Signature) image low-level feature and a lax threshold.
FudanU.v.oisstrict: Using OIS(Ordinal Intensity Signature) image low-level feature and a strict threshold.

In order to design our CD system, we study image features on the impact of a variety of types of video copy detection performance, and present an efficient graph-based approach for video copy detection. It converts the video sequence matching results to a matching results graph, so the problem of video copy detection becomes a problem of finding the longest path in the matching results graph. This graph-based approach gives full consideration to space and time characteristics of video, which not only makes up for the lack of the limited description of global image low-level feature, but also improves the video copy location accuracy. This approach also clusters the video frames to eliminate a large number of redundant, which greatly reduces computation cost in the matching process and enhances the speed of detection.

2. Surveillance Event Detection

2.1. Detection of PersonRuns and OpposingFlow

The reason why the detections of these two events are both introduced in the same chapter is that the motion information calculated from the videos is used in both of them, then the same method is used to remove the isolated noise motion and normalize the motion vectors.

2.1.1 Motion Analysis

Since all the videos are taken from surveillance cameras which means the position of the cameras is still and cannot be changed. As we can see from the videos, there're no movements of the cameras. So, all the motion information extracted from the surveillance videos can be caused only by the activation of people in the videos.

At beginning we once tried to use optical flow to estimate the motion of the videos, but the optical flow estimation is too sensitive to the small noise because of the broadness of the camera view. There are too many people in the videos that small motion noise can be fatal. So we choose a higher level in the Gaussian pyramid of frame image, lower the resolution of the image and separate it into many small blocks and calculated the motion vectors of each block.

The difference between blocks is represented by the sum of absolute value of the difference of pixels at corresponding position. Experiment shows that the method is better than those using histogram to represent the difference between blocks, because some areas in the videos are very similar on color distribution so noise motion information can be caused easily.

Every two adjacent frames in the video generate a motion vector field, but it cannot be used directly to estimate the motion information of the objects in the videos because:

1) Too much noise in the motion vectors field because too many people appeared in the video.
2) The motion vectors are not normalized, because of the angle between the camera and the floor. People near the camera are supposed to generate large motion vectors and people far from the camera cannot generate such motion vectors even when they are running.

An absolute difference image is calculated between two adjacent frames to remove the noise. An absolute difference image is not a binary image, every pixel of it is valued from 0 to 255, the value of a pixel represents the how different the pixels at corresponding position from adjacent frames are. In order to remove those isolated noise motion vectors, a series of erosion and dilation operations are employed. Then the absolute difference image is also separated into blocks the same way as frames. Every pixel's value in a block is added together to get a weighing coefficient α, α represents how much the block is changed from one frame to its next. Isolated noise motion vectors should be removed in the erosion and dilation operations, even if they are not removed, the weighing coefficient α of that block should be small.

Besides the absolute difference image, the other problem is that people near or far from the camera should be fair treated. So another coefficient β is calculated according to the block's vertical coordinate. When block has small vertical coordinate, it's far away from the camera and it's β should be large, and also blocks with large vertical coordinate get smaller β. The i_{th} final motion vector MV_i that can be used to estimate the motion information of the video is calculated from the original motion vector mv_i as follows:

$$MV_i = \alpha_i \cdot \beta_i \cdot mv_i$$

2.1.2 Detection

For the PersonRuns event, after adjusted by the two coefficients, isolated noise has been removed and all the motion vectors can be fair treated. The sum of the size of all the motion vectors between adjacent frames is calculated as the motion strength, only the motion vectors whose size is larger than a threshold T_1 count. After finishing the algorithm on all the frames, we set second threshold T_2, when the motion strength between two adjacent frames is larger than T_2, it's set as 1 and otherwise set as 0. Then, the video are turned into a sequence of number of 1 and 0s. What we need to do is just to find a series of continuous 1s, allowing a few of the motion strength in the series is 0.

For the OpposingFlow event, after checking its definition and the annotation. OpposingFlow can only be detected from these 5 surveillance videos:

LGW_20071123_E1_CAM1.mpeg
LGW_20071130_E1_CAM1.mpeg
LGW_20071130_E2_CAM1.mpeg
LGW_20071206_E1_CAM1.mpeg
LGW_20071207_E1_CAM1.mpeg

The motion information is still used, but we limit the area as a trapezoid covering all the doors. Then the only thing that is different from the detection of PersonRuns event, the motion strength is not the sum of all the motion vectors, what we care about is just those in the trapezoid and opposing to the normal flow direction.

2.2. Detection of TakePicture

After the study of the annotation, we found that all the videos are taken indoor and all the TakePicture event happen with flashlight. So the task is equivalent to the detection of flashlight from the videos(Figure 1).

2.2.1 Flashlight Pattern

Flashlight of a camera usually lasts less than 0.1 second. We extract frames from the surveillance videos every 0.1 second, so the flashlight can only appears in one frame. Then, when a flashlight appears, the flash

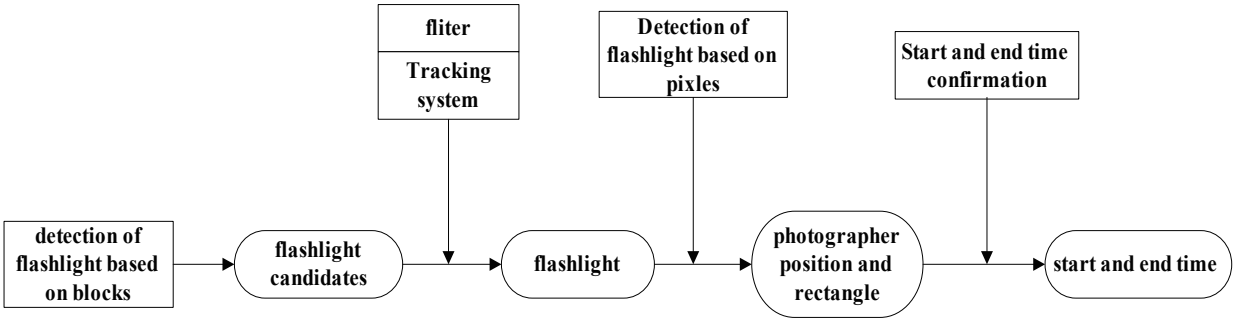

Figure 1. Overview on the framework of detection of TakePicture

area is supposed to be "dark-bright-dark" in continuous 3 frames. So what we have to do first is to find the "dark-bright-dark" pattern from the frame series.

During experiments, we found that if we look after the "dark-bright-dark" pattern based on pixels, it's too sensitive to noise and if based on the whole image it's not sensitive enough to small flash. So as mentioned before, again we separate a frame into small blocks and look for that pattern based on average pixel luminance of each block.

However, only block-separation and based on average pixel luminance are not enough to detect the flashlight, there are many other event could also generate the "dark-bright-dark" pattern. For an instance if a person wearing a pair of dark pants walking on a floor with very light color, the area in the middle of his legs may be mis-detected as a flashlight. However, we can consider a flashlight coming from "nowhere" and cannot be tracked. When a candidate is detected, we track it in the frame series, if it can be detected in a series of frames, this candidate should be dropped, and otherwise we consider it as a flashlight.

2.2.2 Detection

As the definition of the TakePicture event, the earliest time when a person holds a camera in a fixed position prior to activating it and end Time when the earliest time when the camera moves away from a fixed position following the photograph should be confirmed.

When a flashlight is detected, we look for that pattern again based on pixel, and then we can get a binary image that all the pixels that fit the pattern are set to 1 and other pixels sre set to 0. A series of erosion and dilation operations are employed to remove isolated noise pixels and connect the 1-value pixels into contiguous regions. We calculate the average coordinate of those 1-value pixels as the position of the camera. We consider the camera position as the head position of the photographer, and according to the vertical coordinate of that position we can infer a rectangle of an average human body size.

Usually when a person holds a camera in a fixed position, it'll last for a while and the photographer keeps still, so we assume that from the start time to the flashlight, what's in the rectangle won't change a lot. So we use a method which is widely used in face-recognition to find the start time. First we arrange the pixels in the rectangle in order to turn it into a vector, then PCA is used to reduce the dimension. We consider the rectangle in the frame before the flashlight as criterion and we compare the former ones to the criterion, until the difference is larger than a threshold. The same method is used to confirm the end time too.

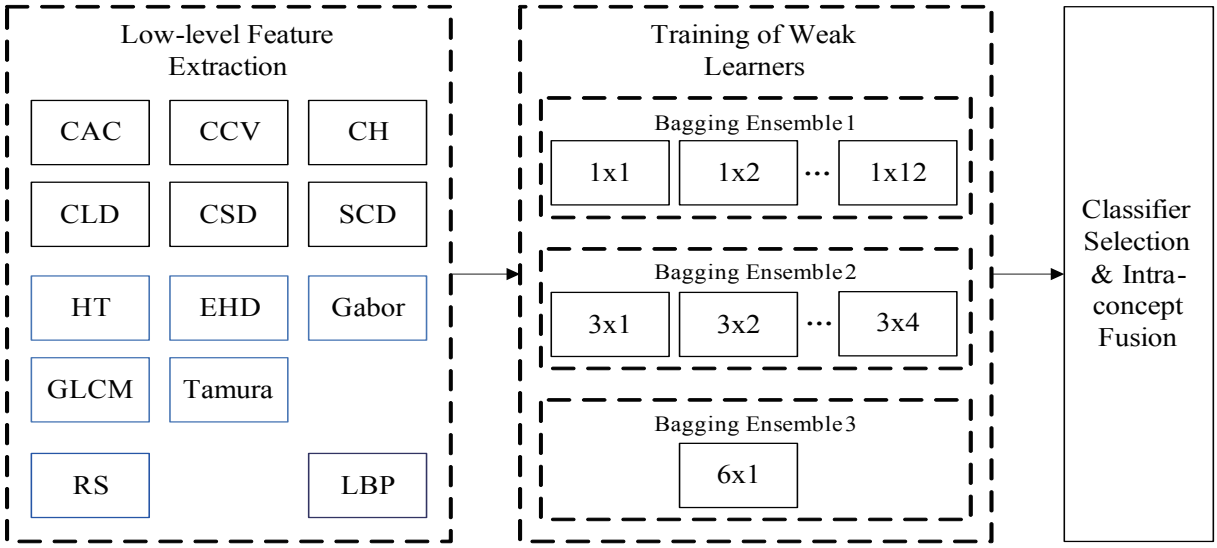

Figure 2. Overview on the framework of high-level feature extraction

3. High-level Feature Extraction

We extract more low-level features for the HFE task this year. To reduce the complexity, all these low-level features are extracted at global scale. However, some of them have characterized the local visual feature, such as EHD (Edge Histogram Descriptor) and LBP (Local Binary Patterns, [9]). Bagging is the main strategy of the learning phase. Two types of weak learners are trained, with different ratios of the negative sample number to the positive. And weak learners of the same type compose an ensemble classifier. At last, we select several optimal classifiers to fuse by a validation set. The whole flowchart is described as Figure 2.

3.1. Low-level Feature Extraction

Several low-level features are defined in MPEG-7, including CLD (Color Layout Descriptor), CSD (Color Structure Descriptor), SCD (Scalable Color Descriptor), HT (Homogenous Texture), EHD (Edge Histogram Descriptor), and RS (Region Shape). Some are reported in other TRECVID teams in the earlier years, including CAC (Color Auto-Correlogram), CCV (Color Coherence Vector, [10]), and LBP (Local Binary Patterns, [9]). The rest are some common features: CH (Color Histogram), GLCM (Gray-Level Cooccurrence Matrix), Tamura and Gabor.

3.2. Learning

At first, we divide the whole training set into two halves, one for training and the other for fusion. For each low-level feature, 17 classifiers are trained. For 12 of them, the training set have as many negative samples as the positive. The training sets of 4 of them have negative instances 3 times of the positive, and the rest has got 6 times. Thus there are 3 types of learners and every two types of learners compose an ensemble, whose member has the same weight. Such strategy is adopted to avoid the unbalance problem during the learning phase.

Each learner is trained by SVM and the kernel is RBF. 5-folder cross-validation is taken for optimal

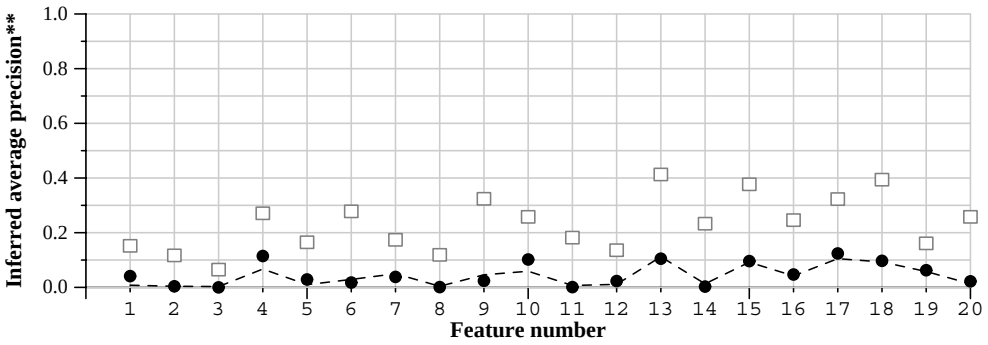

Figure 3. High-level feature extraction result curve

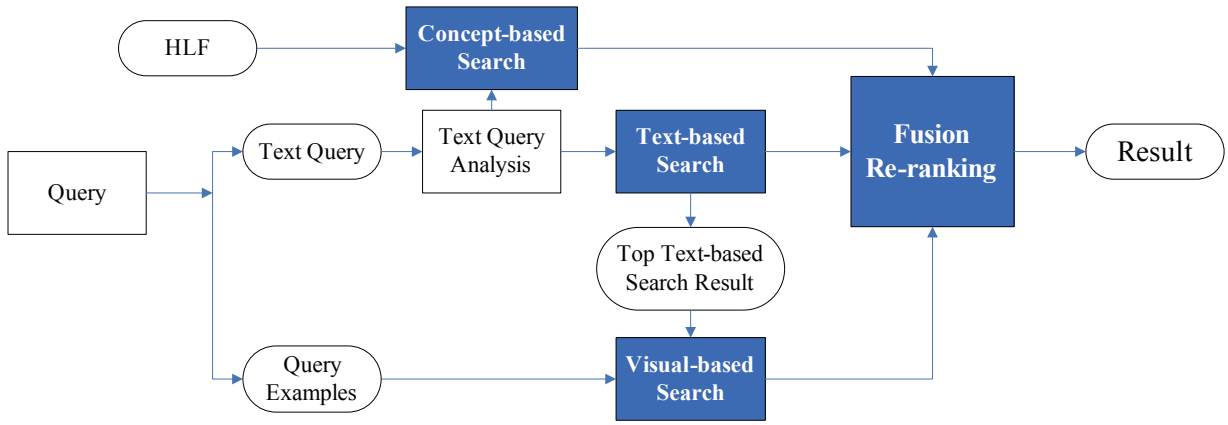

Figure 4. Overview of the automatic search

parameter pair (C, γ). We use the average precision as performance evaluation rather than the accuracy.

3.3. Fusion

In the earlier experiments, we found that the fusion of too many classifiers leads to worse performance, so we use a validation set to select only a few classifiers to fuse (about 3-5 for each concept). First we select the best classifier for every low-level feature and then select the top few classifiers of different low-level features to fuse, and their weights are the same for we found that adjusting the weights in the fusion improves the final performance little in the earlier experiments. Our experiment result is shown in Figure 3.

4. Search

This year, we participated in the automatic search and submitted 6 automatic runs. Focus of our system was on the effective utilization of text, visual features and HLF. The framework of automatic search is shown in Figure 4.

4.1. Text-based Search

The text retrieval module of our system is of fundamental importance to the overall performance. It consisted three components including indexing sub-module, query processing sub-module and searching sub-module.

The official texts correspondent to each video were Dutch words produced by an ASR (automatic sound recognition) algorithm and their English translations [3]. The Dutch words were recorded along with their speakers's code and the time stamps of their starting and ending time. And the translated English sentences were aligned to their speakers. With these information and the time of the shots, we computed the corresponding relation of speakers and shots. Then we built an inverse index table. In the table the terms were the English words and the documents' ids were the shots. We built a list of all the morphological of the words appeared in the translated sentences to condense the index. This list was also used to process the queries later.

Now it is time to feed something to the system. In the TRECVID tasks, queries are sentences describing objects or events that may appear in the video. First, the words are analyzed morphologically. The verbs and nouns were changed into the forms of dictionary headwords. Second, all the notional words were expanded based on the WordNet [7]. A query word was expanded by its hypernyms, hyponyms and synonyms. The three types had different expansion weights which were smaller than one. The effect of various weighting parameters was observed through experiments. In fact there was no absolutely correct choices. But at least we could make the conclusion that the order of weights from the biggest to the smallest were hyponyms, synonyms and hypernyms.

For the search, we employed a document weighting method which was a linear combination of the Okapi weighting score [11]:

$$\sum_{t \in Q,D} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

and the pivoted weighting score [2]:

$$\sum_{t \in Q,D} \frac{1 + \ln(1 + \ln(tf))}{(1-s) + s\frac{dl}{avdl}} \cdot \ln \frac{N+1}{df}$$

According to the searching experiments, the pivoted weighting score playing 63% role in the combined score would lead to relatively more desirable results.

We have also tried to cluster the shots into fewer sets because many of them included too few words. Two methods were used: the k nearest neighbor clustering and the hierarchical clustering. Through tuning the parameters, the two methods produced quite similar results. We observed that most shots that were clustered together were also visually similar. But it was no better than clustering results using only visual features.

4.2. Visual-based Search

Visual-based search system relies on the query of image and video(key-frame) examples from the given topics. This year we extract more visual features, including Color Auto-Correlogram(CAC), Local Binary Patterns(LBP, [9]), *etc*., which were introduced in High-Level Feature Extraction (3.1). Those features were tested in the training dataset for each query topic, and we find HSV Color Histogram(CH), Gabor and Edge Histogram(EH) perform better than others. Query-by-example (QBE) method was used after generating the feature vectors. We rank the key-frames from the test dataset according to the Euclidean distances between

their feature vectors and those of the query images, and fuse the results returned using different features by linear combination for its simplicity. Moreover, since results from text-based search may contain positive infomation, we apply the QBE method to expand the text-based search (namely *TextEx* as will be referred later), which considers top results from text-based search as the query topic examples of the visual-based search engine.

4.3. Concept-based Search

Motivated by the recent study of using ontology to refine the video concept detection, we intend to improve our detection results by exploiting the pairwise correlation of concepts. Our work is similar to Zha *et al*. [17], with a simple modification which improves the performance.

Assuming n concepts which is selected for constructing our ontology, let $\mathbf{M}$ denotes the $n \times n$ correlation matrix with the entry m_{ij} defined as the mutual information of concept C_i and C_j, which is

$$m_{ij} = \sum_{y_i, y_j} P(y_i, y_j) \log \frac{P(y_i, y_j)}{P(y_i)P(y_j)} \tag{1}$$

where $y_i \in \{+1, -1\}$ and $y_j \in \{+1, -1\}$ are the labels of C_i and C_j for a sample, respectively. $P(y_i)$, $P(y_j)$ and $P(y_i, y_j)$ can be computed from the ground truth.

To utilize the correlation matrix, a propagation strategy proposed in [17] is described as follows. Let $\mathbf{P_0}$ denotes the $n \times k$ confidence matrix with its entry p_{ij} representing the confidence score that C_i appears in the video shot S_j. Here, k is the number of shots. These scores can be refined by the following equation

$$\mathbf{P_t} = (1 - \alpha)\mathbf{P_0} + \alpha \sum_{l=1}^{t} \mathbf{M}^l \mathbf{P_0} \tag{2}$$

The weight factor $\alpha (0 \leq \alpha \leq 1)$ in (2) defines the relative contribution to a concept from its initial scores and its related concepts. In other words, $\mathbf{P} = (1 - \alpha)\mathbf{P_{init}} + \alpha \mathbf{P_{rel}}$. Furthermore, to avoid the self-reinforcement problem, the diagonal entries of M are set to be 0. The parameter t in (2) is specified in [17] as $t = \infty$. To facilitate the convergence, they introduce a degradation factor β and normalize $\mathbf{M}$ by $\mathbf{D}^{-1}\mathbf{M}$, where D is a diagonal matrix with $d_{ii} = \sum_{j=1}^{n} m_{ij}$. As a result, equation (2) can be rewritten as

$$\mathbf{P}_\infty = (1 - \alpha)\mathbf{P_0} + \alpha \lim_{t \to \infty} \sum_{l=1}^{t} (\beta \mathbf{M})^l \mathbf{P_0} \tag{3}$$

Then the refined score is obtained as follows

$$\mathbf{P}_\infty = [(1 - 2\alpha) + \alpha(\mathbf{I} - \beta \mathbf{M})^{-1}]\mathbf{P_0} \tag{4}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix.

In our method, we set $t = 1$ in (2), which means the propagation is only carried out one time. Accordingly, we rewrite equation (2) as

$$\mathbf{P_1} = (1 - \alpha)\mathbf{P_0} + \alpha \mathbf{M} \mathbf{P_0} \tag{5}$$

The fusion of all the concepts in a topic adopts the relation algebra expression which is set manually. Assuming there are N concepts which will be joined in a topic, and the refined confidence value of every concept is $p_1 \ldots p_N$, then the confidence value of topic t for a shot S is computed by

$$p(t|S) = \frac{\lambda}{N} \sum_{i=1}^{N} p_i - \frac{\mu}{2} \max_{i,j=1}^{N} \{|p_i - p_j|\} \tag{6}$$

where λ, μ are parameters($0 < \lambda, \mu \leq 1$). Then this confidence value is used to rank the shot list for every topic.

Our ontology instance includes 64 LSCOM Semantic Visual Concepts [8] manually selected in advance, among which 14 concepts are adopted in TRECVID 2008 concept detection task [12]. These 14 concepts are "Boat_Ship", "Bridges", "Bus", "Cityscape", "Classroom", "Dogs", "Driver", "Flowers", "Harbors", "Kitchen", "Nighttime", "Singing", "Streets", "Telephones". We name them as VALID14. The correlation matrix M was computed from the ground truth for for 374 LSCOM Semantic Visual Concepts over the TRECVID2005 development set, released in the pool of Columbia374 [16]. And the initial score matrix $\mathbf{P_0}$ was obtained from the baseline detectors for the above 374 concepts over the TRECVID2008 development set. The baseline detectors for both the TRECVID2008 development and test set are released as a fusion of Columbia374 and VIREO374 [4]. We compared the above two approaches to the baseline on VALID14. When (2) is used to compute $\mathbf{P}_\infty$, the parameter α is set to three different values, *e.g.*, $0.15, 0.50, 0.85$, and β set to 0.99, as in [17]. As to $\mathbf{P_1}$ which is calculated from (5), α is set to $0.5, 1.0$. To evaluate the performance, we use Average Precision (AP) to compare the above approaches on each concept. Furthermore, the mean average precision (MAP) is calculated by averaging the AP on all the 14 concepts in TRECV15.

Our experiment result has shown that, $\mathbf{P_1}$ outperforms $\mathbf{P}_\infty$ with all parameters defined above, partly because of the save of the computation of matrix inverse, which leads to the confusion of the results when the matrix grows larger, *e.g.*, 64×64. We note that $\mathbf{P_1}$ with $\alpha = 0.5$ gains 2.5% improvement to the baseline, and achieves better result than that when $\alpha = 1.0$. So we adopt $\mathbf{P_1}$ with α equals 0.5 as our final method to complete our experiments on the the TRECVID2008 test set.

For each topic, we select a list of concepts which are related to it. Topic confidences for each shot are computed from (6), where the both parameters λ and μ are empirically set to be 1. Finally, we rank the shot list for each topic.

4.4. Fusion and Re-ranking

Fusion

Multimodal fusion method has been proved to be of great help in our previous trecvid [15] and videolympics [13] approaches, which treats each module of a specific multimedia modality as an atom search engine and combines their output to achieve results with the refinement through multimodal information. The process could also further improve the fusion model iteratively by taking in user feedback and could reach a state with notable performance [6]. However, in automatic search this kind of user data is unavailable, which limits the final fusion model to be simple.

This year we use linear multimodal fusion for all our 6 automatic search submissions(The run with pure text etc. could be regarded as a regression of the fusion with weights only on text search engine). The linear combination schemes were chosen according to the contribution of each search engine. And measurement outputs were used instead of rankings to make up for the possible accuracy loss brought by the decrease of refine iterations. Details of the overall weights distribution are illustrated in Figure 5. Note that the weights vary slightly among different topics according to the adjustments of our query analysis engine.

We could see from the chart that the concept-based search (4.3) plays an important role in our final results as the pre-trained concept detectors contain the richest multimodal information. The visual-based search(4.2) is also helpful when combining with the output of pure text-based search(4.1) (as labeled by *TextEx*). As will be described later in the Evaluation Results section(4.5), our methods with multimodal fusion performed well among all the automatic search submissions.

Re-ranking

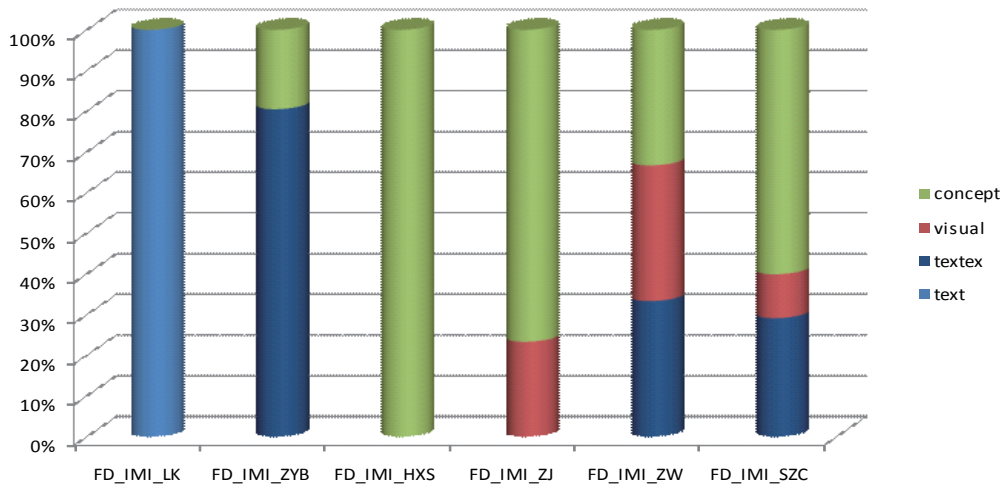

Figure 5. Linear multimodal fusion overall weights distribution

The refinement of the fusion model utilizes the feedback of the initial retrieval results, and is crucial to the performance of combining multimodal information. Since user interaction is disabled in automatic search, we used KTF and QETF framework in our previous work [15] [13] to extract positive information from the output automatically by selecting the shots with top confidences, generating semantically distinct clusters, finding relevant shots of the clustering centroids using different search engines, and finally combining the newly retrieved results with the original ones using linear fusion. The iteration could go several times, but in our system we only used it for one loop to avoid potential noises. The *TextEx* results described in the previous section and Figure 5 is an example of our re-ranking scheme, and its overall performance improved for over 30% against that of the original in our experiments.

4.5. Evaluation Results

This year we submitted 6 automatic runs for evaluation,

FD_IMI_LK: based only on the text from the English ASR/MT output and on the text of the topics.
FD_IMI_ZYB: based on the text search and the visual expand from the text search results.
FD_IMI_HXS: based on the concept mapping method.
FD_IMI_ZJ: based on average fusion method.
FD_IMI_ZW: based on average fusion method.
FD_IMI_SZC: based on multimodal fusion method.

Although it is the first time for us to participate in the automatic search task, our past experience on interactive search [15] and manual search [14] provided us plenty of experience and knowledge about video retrieval systems and algorithms, from which our system benefited a lot and performed well in the official evaluation. Figure 6, 7, 8 illustrated our evaluation results.

From Figure 6 we could see that our best run ranked 10 of all the 82 automatic search submissions, and 4 runs ranked in the top 20s. Large portions of information from concept-based search are used in these runs, of which the run FD_IMI_HXS even using pure concept info resulted quite well with the rank of 12.

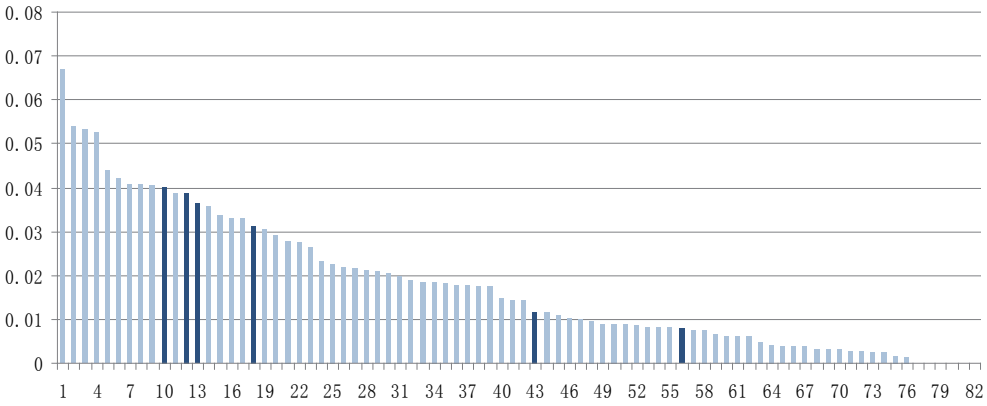

Figure 6. Overall performance of our submissions (darkened lines) against others

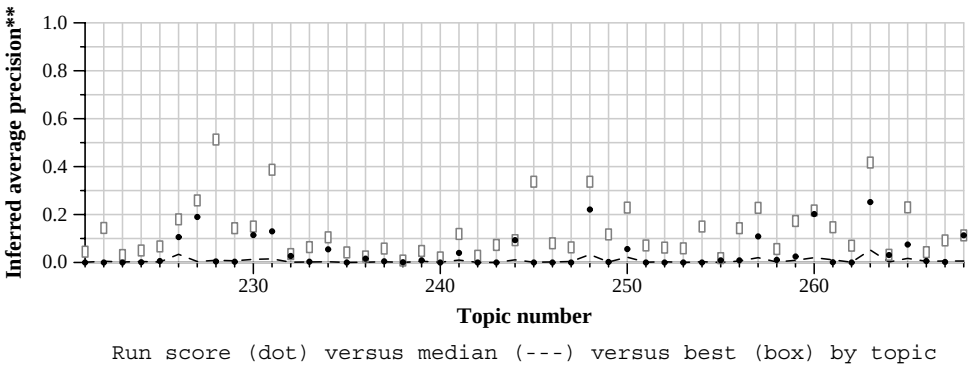

Figure 7. Comparison of our best run with others

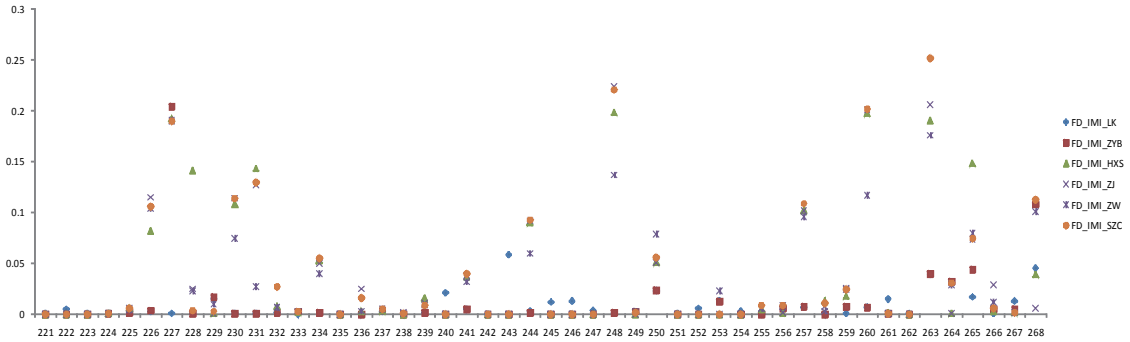

Figure 8. Comparison within our 6 submissions

Visual and textual retrieval have also provided positive samples, however their contribution was limited since only few latent visual and textual features could be expressed through the input topic with simple text and image example queries, which indicates that our future research should focus on concept-based retrieval and retrieval that could most effectively extract the relevant visual and textual information from given shot examples.

Figure 7 and Figure 8 compare the performance of our runs and others with regard to each topic in the evaluation. Generally, automatic search still face great challenges when dealing with most of the topics. But for some topics like "226 - people and lots of plants", "227 - big face", "248 - outdoor crowds", "260 - airplane exterior", "263 - people walking upstairs", etc. which showed good performance in our best run and the overall best run, there does exist effective algorithms for the retrieval, and mostly due to the concept detectors and ontology training. And for topics like "228 - typed paper", "231 - map", "245 - microscope", etc., our system's not performing well while the best run having a high score reminds us of that our study on ontology training and concept detectors still need a long way to go.

Note there are also topics like "240 - books", "243 - microscope", "246 - kitchen", etc. that pure textual search has the best performance, and topics like "227 - big face", "229 - people and water", etc. that fusion using only textual and visual information outperforms others(Figure 8). Since in the automatic search we lack user feedback and could not adjust the fusion model iteratively to the best condition, we had to sacrifice some of the positive samples for the sake of overall performance. However, this could be walked around through further query analysis and modification of the weights using pre-trained topic model, which will also be an important task in our future study.

5. Content-based Copy Detection Pilot

5.1. Some Approaches about CBCD (Content Based video Cope Detection)

Hampapur *et al*. [2] perform a comparison of video matching techniques using different features extracted from each frame of the reference and test clips: motion direction, ordinal intensity and color histogram. Then, the generated signature is applied to the reference clip by using different types of metrics (convolution for motion direction based signature, $L1$ distance for ordinal matching based signature and histogram intersection for color histogram based signature).

Julien Law-To *et al*. [5] propose a concept of trajectories of points along the video sequence. This method builds trajectories of points in videos for video content indexing. This method takes advantage of the trajectories for indexing the spatial-temporal contents of videos. It has two advantages, first, the redundancy of the local description along the trajectory can be efficiently summarized with a reduced loss of information and second, the trajectory properties will allow enriching the local description with a spatial, dynamic and temporal behavior of this point. Analyzing the obtained trajectories allows to highlight trends of behaviors and then to assign a label of behavior to a local descriptor. This method has well effect for Picture in Picture Copy Type.

N. Guil *et al*. [1] divides the query video in clusters and extracts a representative key frame for each cluster. Features from these key frames constitute the signature of the video. Then, it performs a dense comparison between the signature of the query video and every frame of the target video using relaxed distance constraints to speed-up the search process. This approach can cope with sequences with different resolution, frame rate and bit rate.

Based on the above-mentioned methods, we propose our approaches, these approaches will be described in section 5.2, Figure 9 presents the whole video copy detection framework.

5.2. Our Approaches

5.2.1 Video Sequence Representations

In video copy detection, much more data has to be processed than in image copy detection, images (video frames) feature selection becomes a key point to develop a specific approach to video comparison. Usu-

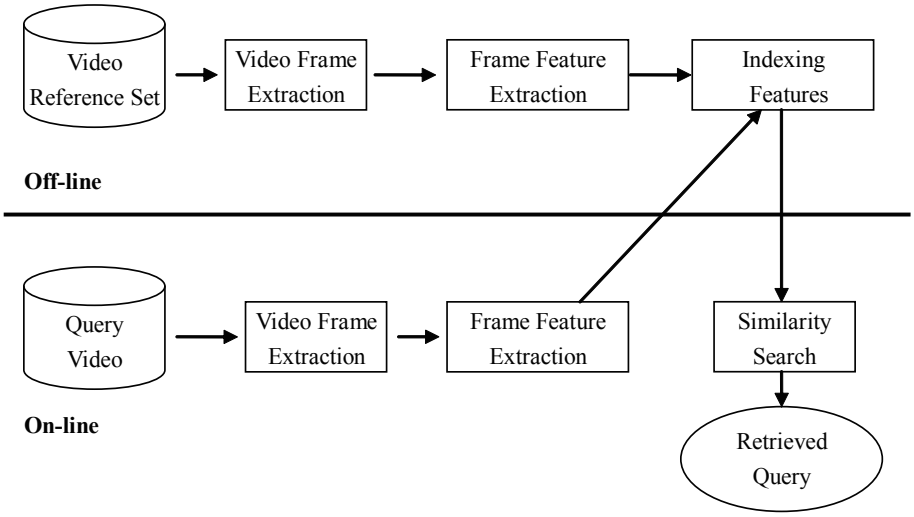

Figure 9. Video copy detection framework

ally, the features employed are simple, distinctive and easy to compute. Hampapur *et al*. [2] examined several sequence-matching methods based on the motion, ordinal, and color features, and reported that the ordinal signature achieves the best video copy detection performance. We also compare several image low-level features for CBCD and also find the ordinal signature having better performance for specific cope type(especially for *Change of gamma*).

In addition, in large scale video copy detection, the computational costs are also an important criterion for measuring the comparison method. Commonly, sparse comparison methods require less computational resources during the comparison process. On the other side, dense comparison approaches are more robust. To get a trade-off between computational costs and detection precision, we cluster the consecutive video frames in advance(Figure 10 presents clustering results). It is different to the common shots segmentation because video information has a strong temporal redundancy, and our aim only makes the consecutive similar video frames become a cluster and reduces the computational costs. Furthermore, clustering the similar video frames brings an advantage, namely, it can cope with video sequences with different resolution, frame rate and bit rate [1].

5.2.2 Video Sequence Matching

In order to accurately locate the copy video in reference video data, we transform it into finding a longest-path in matching results graph (using Dijkstra algorithm). We define the similarity of two video clips as follows:

$$0 \leq i < j < k \leq M$$

$$0 \leq u < v < w < x < y < z \leq N$$

Note: query video q has M clusters, reference video r has N clusters.

$$0 \leq i < j < k \leq M$$

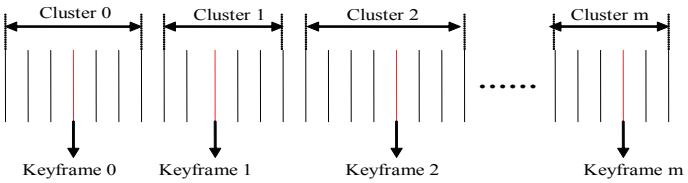

Figure 10. Clustering the video frames

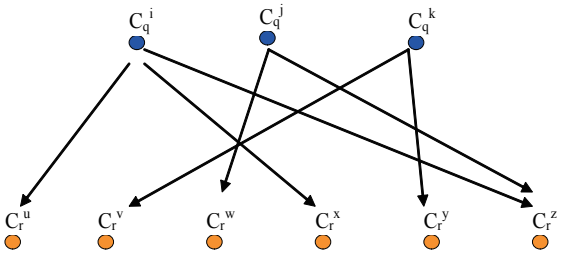

Figure 11. The matching results ($sim(q, r) > T$)

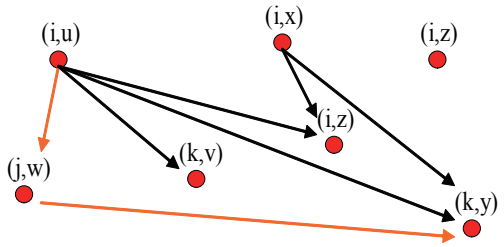

Figure 12. Constructing matching results graph (directed acycling graph) and finding a longest-path in matching results graph

$$0 \leq u < v < w < x < y < z \leq N.$$

In Figure 11, each directed edge means $sim(C_q^m, C_r^n) > T$ (T is a threshold).
In Figure 12, the path $[(i, u) \rightarrow (j, w) \rightarrow (k, y)]$ is the longest-path, we select this path represents the matching of two clips, the similarity of two video clips is:

$$sim(q, r) = \frac{\sum_0^m sim(C_q^i, C_r^j)}{m} \log(1 + m)$$

If the similarity of two video clips is more than the threshold, the copy is detected.

5.3. Experiment and Evaluation

We submitted 2 runs in this year. The detailed performance is show in Figure 13. The evaluation result shows that our system needs to be improved in many aspects. A most crucial limitation of our system is that our system can not detect the picture in picture transformation, so the detection performance of our system is very poor at 2 transformations (picture in picture, combination) in all 10.

Acknowledgements

This work was supported in part by MoE Research Project(104075), NSFC Project(60873178, 60875003), and MSRA Young Faculty Innovation Fund.

References

[1] N. Guil, J. M. Gonzlez-Linares, J. R. Czar, and E. L. Zapata. A clustering technique for video copy detection. In *IbPRIA (1)*, volume 4477 of *Lecture Notes in Computer Science*, pages 451–458. Springer, 2007.

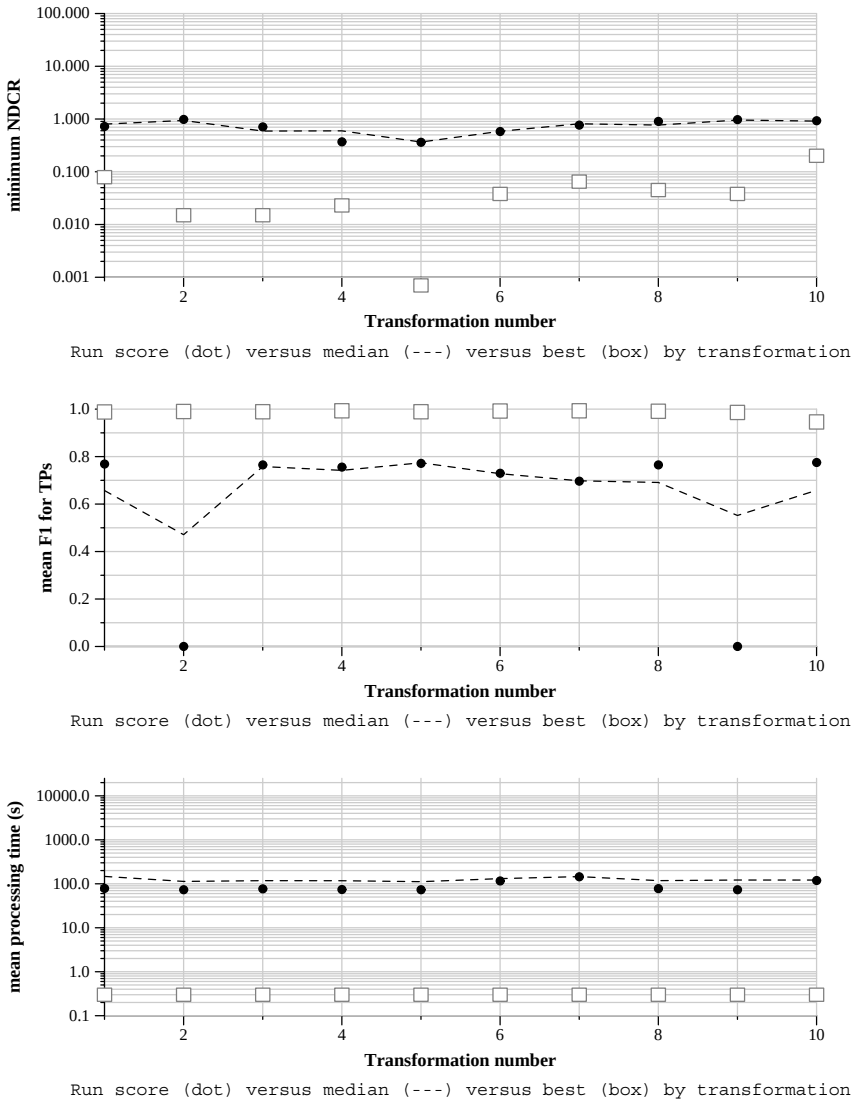

Figure 13. Copy detection evaluation result curve

[2] A. Hampapur and R. Bolle. Comparison of sequence matching techniques for video copy detection. *Conf. on Storage and Retrieval for Media Databases*, pages 194–201, 2002.

[3] M. Huijbregts, R. Ordelman, and F. de Jong. Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proceedings of the second international conference on Semantics And digital Media Technologies (SAMT)*, Lecture Notes in Computer Science, Berlin, December 2007. Springer Verlag.

[4] Y.-G. Jiang, A. Yanagawa, S.-F. Chang, and C.-W. Ngo. Cu-vireo374: Fusing columbia374 and vireo374 for large scale semantic concept detection. In *Columbia University ADVENT Technical Report 223-2008-1*, 2008.

[5] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Robust voting algorithm based on labels

of behavior for video copy detection. *ACM Multimedia*, pages 835–844, 2006.

[6] O. Melnik, Y. Vardi, and C.-H. Zhang. Mixed group ranks: preference and confidence in classifier combination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):973–981, Aug. 2004.

[7] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[8] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3):86–91, 2006.

[9] T. Ojala, M. P. Inen, S. Member, and T. M. A. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.

[10] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. *In Proceedings of ACM Multimedia*, pages 65–73, 1996.

[11] S. W. S. E. Robertson and M. Beaulieu. Okapi at trec–7: automatic ad hoc, filtering, vlc and filtering tracks. *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, page 253C264, July 1999.

[12] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[13] Z. Sun, Y. Song, Y. Zheng, H. Yu, C. Jin, H. Lu, and X. Xue. Fudan university: hierarchical video retrieval with adaptive multi-modal fusion. In *CIVR '08: Proceedings of the 2008 international conference on Content-based Image and Video Retrieval*, pages 549–550, New York, NY, USA, 2008. ACM.

[14] X. Xue, H. Lu, H. Yu, S. Zhang, B. Li, J. Zhang, J. Ma, B. Su, and Y. Guo. Fudan university at trecvid 2006. In *NIST TRECVID Workshop*, 2006.

[15] X. Xue, H. Yu, H. Lu, Y. Guo, Y. Zhang, S. Zhang, B. Li, B. Su, Y. Zheng, W. Zhou, L. Cen, J. Zhang, Y. Jiang, J. Qi, J. Lu, Q. Diao, Z. Shi, and Z. Sun. Fudan university at trecvid 2007. In *NIST TRECVID Workshop*, 2007.

[16] A. Yanagawa, S.-F. Chang, L. Kennedy, and W. Hsu. Columbia universitys baseline detectors for 374 lscom semantic visual concepts. In *Columbia University ADVENT Technical Report 222-2006-8*, 2007.

[17] Z.-J. Zha, T. Mei, Z. Wang, and X.-S. Hua. Building a comprehensive ontology to refine video concept detection. In *MIR '07: Proceedings of the international workshop on Workshop on Multimedia Information Retrieval*, pages 227–236, New York, NY, USA, 2007. ACM.