Tsinghua University at TRECVID 2005

Jinhui Yuan, Huiyi Wang, Lan Xiao, Dong Wang, Dayong Ding, Yuanyuan Zuo,
Zijian Tong, Xiaobing Liu, Shuping Xu, Wujie Zheng, Xirong Li, Zhangzhang Si,
Jianmin Li, Fuzong Lin, Bo Zhang
State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, P. R. China

Abstract

Our shot boundary determination system consists of three components, including a FOI detector, a generalized CUT detector, and a long gradual transition detector. One support vector machine, taking score vector calculated with graph partition model as input, is used to detect CUT. Long gradual transition is determined by another three support vector machines with multi-resolution score vectors as input. After these detectors make decision successively, the locations of shot boundaries and the corresponding types are obtained. It is found in the experiments on development data that by tuning penalty ratio of loss of misclassifying the positive and the negative samples, it is possible to control the trade-off between precision and recall. 31 runs are generated from the same system with the 4 support vector machines being trained with different parameters. Among them, 10 runs are submitted for evaluation. And the results show that our system is among the best.

In our system for low level feature extraction, some spatial features of motion vectors are proposed to select the motion vectors which describe the camera motion in deed. The four-parameter affine model is used to describe the camera motion, and the ILSE technique is used to calculate the parameters. Then camera motion will be classified into three classes: pan, tilt and zoom with an accurate classification method based on finite-state automata. Our system achieves best results in this task of TRECVID2005.

Our systems for high level feature extraction rely heavily on the visual information. Visual features include Color Auto-Correlograms, Color Coherence Vector, Color Histogram, Color Moment, Edge Histogram and Wavelet Texture. Two different systems using regional and global image features are compared to explore the effectiveness of regional features. In the regional system, keyframes are segmented and regional feature of all the six types mentioned above are extracted. Then support vector machine classifier with Earth Mover Distance (EMD) kernel is built. In the global system, the six types of global feature are extracted for each keyframe directly. Then the classifier ensembles for detecting each concept are formed by using the Relay Boost algorithm. This is followed by a concept context module. We tried mainly two approaches, one based on stacked SVM and the other based on weighted sum of the confidence scores of the related concepts. We then apply time clustered post-filtering to remove false positive shots. Based on these two systems we have our 7 runs. From the results, we find that multi-feature fusion improves over any single modality significantly.

Our automatic video search systems have three basic retrieval models: a text model based on script generated by ASR, an image model based on region-based image matching and a concept model which automatically parses the queries and video shots into concept vectors, and then searches video shots through query-shot similarity computing in concept space. Based on these models, we also develop some combination systems. In the score fusing system, the results are ranked by fusing the scores generated from the basic retrieval models. In the fusing system based on query type, queries are classified into two classes, and then retrieved using different models. Among our 7 submissions, the results show that when searching for general topics, which are always less related to person, combining text and concept models performs better than only using text model.

It is the second time that we participate in TRECVID. In this year, we submitted four tasks for evaluation. Our approaches for these tasks are described in Section 1, Section 2, Section 3, and Section 4. Conclusions are presented in the Section 5.

1. Shot boundary determination

1.1 Overview

Last year we developed a shot boundary detection system and participated in the shot boundary detection evaluation of TRECVID 2004 [thu_notebook04, vcip05]. The evaluation results show that the performance of our system is among the best. However, there is still much room to improve the system. Firstly, the system is a thoroughly rule-based one. It is difficult to select the proper thresholds for various videos. Furthermore, our experiences reveal that even adaptive thresholds can not achieve satisfying results. Secondly, to utilize multiple complementary features, we have to extract five different kinds of features from each video in the system of 2004, which is a rather time-consuming procedure. To boost the efficiency of the system, the feature extraction stage is a bottleneck. In fact, there are some redundancies among different features. Therefore, the number of required features can be decreased. Finally, to reduce the various disturbances such as flashlight and abrupt movement, we incorporated various modules such as post processing and flashlight detector into the system. The various modules, on the one hand, can effectively improve the precision of shot boundary detection; on the other hand, we have to design complex collaboration rules among different modules.

To address the above problems, we have been focusing on developing an effective, efficient, unified and easy re-implemented shot boundary detection system. In the previous work [pakdd05, acmmm05], we have proposed a unified shot boundary detection framework based on graph partition model. In the proposed framework, graph partition model is used to construct the signal characterizing the content variation. The experiment shows that this method is robust to various abrupt noises like flashlight. Temporal multi-resolution analysis is adopted to unify the methods of detection cuts and gradual transitions. To overcome the drawbacks of threshold decision method, we construct a novel kind of feature and employ support vector machine to classify boundaries and non-boundaries. Extensive experiments have been conducted on TRECVID dataset to verify the effectiveness of the framework. However, several indispensable topics have left open to make the system complete and usable:

(1) Detection of fade out/in effects.
(2) Precisely locate the boundaries of each gradual transition.
(3) In-depth discussion of multi-resolution.
(4) Effective collaboration of separate modules.

In our new integrated system shown in Figure 1, we have approached the above topics, as well as improved the implementation of the original ideas. Three separated modules are introduced and integrated to establish a complete system. FOI detector has been singled out. In CUT detector, the definition of CUT goes beyond the conventional understanding to embrace the short GTs which can be treated as CUT. GT detector employs several innovative ideas in order to enhance both the accuracy of detection and boundary determination.

1.2 FOI detector

In the process of fade out/in effect, the first shot fades out into a sequence of monochrome frames and then the next shot fades in. FOI is virtually one sub-type of GT. However, we decide to separate the detection of FOI in consideration that some FOI involves a very fast graduation from one shot to another, only having no more than three monochrome frames in between, which intuitively looks like a CUT. So we have proposed an innovative method to detect FOI before the detection of CUT, so that a fast FOI would not mingle with CUT.

a. Recognition of consecutive monochrome frames

Step 1. Given a video sequence, extract the RGB color histogram - 16 bins for each channel, 48 bins totally - for each frame.

Step 2. For each frame, compute the summation of the first 4 bins in each channel. If the summation surpasses the threshold T_{MF}, take it as a monochrome frame.

Step 3. For all the frames identified monochrome, track their frame index numbers to form blocks whose frames are all monochrome.

b. FOI boundary detection

Step 1. Given the block collection generated by Stage a., determine its corresponding left boundary by the following process:

(1) For each frames lies before the first frame of the block, compute the summation of the first four bins of each channel into accum1, and then repeat the computation into accum2 for the frame distance away from the previous one.

(2) Repeat (1) until | accum1-accum2| < T_{diff},

(3) The left boundary is defined as the mean index number in between.

Step 2. Determine the right boundary in the similar way, while it is the frame right after the last frame of the block that should be computed.

Step 3. Upon initial process of the boundary detection for every FOI candidate (based on one block), a special sub-module is employed to deal with the overlapping among those candidates, since it is possible that the boundary of some FOI stretches so far as to overlap with another FOI. The module should be able to merge such candidates into a single FOI.

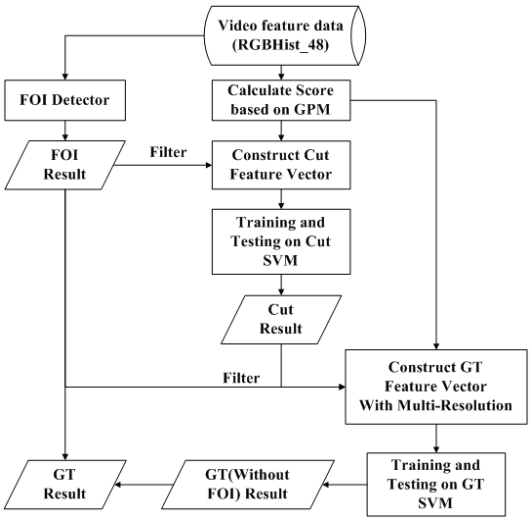

Figure 1. Architecture of Shot Boundary Detector.

1.3 CUT Detector

The detection of CUT is considered much easier. A number of previous approaches could achieve good performance on its detection even with the single-threshold scheme. Nevertheless, most of these solutions are disturbance-sensitive especially by flashlights. Our cut detector based on graph partition model is able to minimize the disturbance. Furthermore, we extend our understanding of CUT and make CUT include short gradual transition which lasts no more than five frames so that our CUT module can also be utilized in a broader sense.

In summary, the working process of our cut detector consists of following steps:

Step1. Given a video data set, constructed a weighted graph G = G(V, E). Treat each frame as a node and link each other by an edge, then calculate the weight of edges to obtain the similarity matrix.

Step2. Calculate scores of N-1 possible cuts. The cut with local minimum score is the CUT boundary candidate. Note that there is no frame intersection between the candidate set and the FOI set.

Step3. Select the local minimum found in step 2 as a central point and extend fore-and-aft corresponding neighborhood scores within a radius of N/2 in order to construct the feature vector.

Step4. Training and testing the set of feature vectors to build the cut model by SVM to get the CUT results.

1.3 Gradual Transition Detector

Up to now, gradual transition detection remains difficult. This year, our gradual transition detector is developed based on Yuan's work, but

as it is mentioned before, that model is not completed. For one thing, one could only get the minimum point of GT in Yuan's original model whose performance still needs to be improved. The determination of the boundary of GT is an unconsummated problem. Moreover, the multi-resolution approach which enlarges the disturbance of motion still needs to be discussed. Through discussion and experiment, we present a new multi-distance graph partition algorithm for GT detection and have got satisfying result.

The first question relates to how to find out proper boundary of GT. We propose two approaches and finally choose the single-threshold scheme. From the evaluation result of recall and precision on the data set of TRECVID 2003 and 2004, the single-threshold scheme is effective because of GT's gradualness. Furthermore, to avoid undesirable segmentation in one gradual transition, we incorporate the overlapping segments after boundary determination.

In regards to the approach to deal with the disturbance of motion, it should be noted that the multi-resolution approach cannot eliminate disturbance of motion and may even enlarge it, thus further studies are still needed on how to utilize multi-resolution in an effective manners. However, it inspires us to attempt other schemes. We propose a multi-distance approach to in the hope of solving minimizing the disturbance and achieve remarkable performance when results by different distance are incorporated. Unfortunately, we have to point out this multi-distance approach would not solve the forenamed problem thoroughly; it is to detect more long gradual transitions so as to reduce the effect caused by motion, and compensate for the missed gradual transition through the model constructed by different distance.

In summary, the working process of our GT detector consists of the following steps:
Step1. Given a video data set, constructed a weighted graph G = G(V, E). Treat each frame as a node and link each other by an edge, then calculate the weight of edges to obtain the similarity matrix.
Step2. Calculate scores of N-1 possible cuts. Note that there is no intersection among the candidate set, the FOI set and the Cut set.
Step3. Use fixed threshold to detect the boundary of gradual transition. Then select the minimum score in each boundary to be the central point and extend fore-and-aft corresponding neighborhood scores within a radius of $N*\Delta/2$ to construct the feature vector. Note that Δ is the sampling rate of the frames, and $\Delta \in \{1, 2, ...\}$.
Step4. Employ SVM to construct different model with different Δ. For instance, we can get three groups of result when Δ = 1, 2, 3, and select the union of them to get final gradual transition result.

1.4 Experiment and Evaluation

The system is developed based on the collections of 2003 and 2004 SBD task. The performance of this system is testified by TRECVID 2005 SBD. LibSVM is adopted to train the model of shot boundary classification [libsvm]. RBF kernel is used in the SVM model, and the best parameter settings, including kernel parameter *g* and penalty parameter *C*, are determined after a cross-validation process. For each run, there are four models are trained, they are one CUT model and three GT models based on different resolution feature and represented by svm_{cut}, svm_{gt1}, svm_{gt3}, svm_{gt5} respectively. By tuning the ratio of penalty weights of positive and negative examples, we can effectively control the precision vs. recall of each model's output. With different penalty ratios, 31 runs are yielded and ten are submitted for evaluation. The evaluation result is depicted in Table 1. Compared to other participants' systems, our system is one of the best. Let w_x denote the ratio between the penalty of misclassifying the positive examples and that of misclassifying the negative ones. The different settings for w_x are summarized in Table 2. As the table shows, with the w_{cut} increasing from 1 to 15, the recall of cuts in "thu01" increases from 0.929 to 0.959 of "thu05". It is effective to tune of penalty ratio to adjust the precision vs. recall of shot boundary detection.

Table 1: Evaluation results of the ten submissions (Ranked by F-measure)

	All Transitions				Cuts				Graduals				Gradual Frame Accuracy			
	Sys	Rcl	Prc	F#	Sys	Rcl	Prc	F#	Sys	Rcl	Prc	F#	Sys	Rcl	Prc	F#
1	thu26	0.894	0.901	0.897486	thu25	0.93	0.941	0.935468	thu26	0.788	0.791	0.789497	thu13	0.88	0.821	0.849477

2	thu12	0.904	0.89	0.896945	thu13	0.949	0.921	0.93479	thu01	0.818	0.757	0.786319	thu25	0.876	0.818	0.846007
3	thu02	0.912	0.878	0.894677	thu26	0.93	0.939	0.934478	thu05	0.806	0.767	0.786017	thu12	0.859	0.824	0.841136
4	thu01	0.901	0.887	0.893945	thu02	0.941	0.928	0.934455	thu02	0.827	0.746	0.784414	thu26	0.86	0.821	0.840048
5	thu25	0.872	0.916	0.893459	thu12	0.949	0.919	0.933759	thu12	0.771	0.798	0.784268	thu05	0.847	0.831	0.838924
6	thu13	0.882	0.903	0.892376	thu01	0.929	0.936	0.932487	thu07	0.838	0.733	0.781991	thu02	0.845	0.828	0.836414
7	thu09	0.925	0.856	0.889163	thu09	0.949	0.914	0.931171	thu09	0.854	0.71	0.775371	thu07	0.846	0.825	0.835368
8	thu07	0.888	0.886	0.886999	thu07	0.905	0.948	0.926001	thu23	0.837	0.718	0.772947	thu01	0.841	0.829	0.834957
9	thu05	0.92	0.854	0.885772	thu23	0.957	0.892	0.923357	thu25	0.701	0.831	0.760484	thu23	0.831	0.83	0.8305
10	thu23	0.927	0.845	0.884103	thu05	0.959	0.883	0.919432	thu13	0.686	0.837	0.754014	thu09	0.834	0.825	0.8295

Table 2: Description and analysis of the submissions

sysid	w_x	description
thu01	w_{cut}=1, w_{gt1}=w_{gt3}=w_{gt5}=1	Default parameter
thu02	w_{cut}=2, w_{gt1}=w_{gt3}=w_{gt5}=1	Make more penalty for missing cuts, so as to increase the recall of cuts detection
thu05	w_{cut}=15, w_{gt1}=w_{gt3}=w_{gt5}=1	Make more penalty for missing cuts, so as to increase the recall of cuts detection
thu07	w_{cut}=0.2, w_{gt1}=w_{gt3}=w_{gt5}=1	Make more penalty for false alarms of cuts, so as to increase the precision of cut detection
thu09	w_{cut}=2, w_{gt1}=w_{gt3}=w_{gt5}=2	Make more penalty for missing cuts and gradual transitions, so as to increase the recall of boundary detection
thu12	w_{cut}=2, w_{gt1}=w_{gt3}=w_{gt5}=0.5	Increase the recall of cuts and increase the precision of gradual transitions
thu13	w_{cut}=2, w_{gt1}=w_{gt3}=w_{gt5}=0.25	Increase the recall of cuts and increase the precision of gradual transitions
thu23	w_{cut}=8, w_{gt1}=w_{gt3}=w_{gt5}=2	Increase the recall of both cuts and gradual transitions
thu25	w_{cut}=0.5, w_{gt1}=w_{gt3}=w_{gt5}=0.25	Increase the precision of both cuts and gradual transitions
thu26	w_{cut}=0.5, w_{gt1}=w_{gt3}=w_{gt5}=0.5	Increase the precision of both cuts and gradual transitions

2. Low Level Feature Extraction

2.1 System Overview

Camera motion estimation and classification is a fundamental problem in video analysis and retrieval. In the previous works, motion vectors are directly used to estimate the camera motion. In our system, a series of spatial features of motion vectors are proposed to select the motion vectors which describe the camera motion in deed. The four-parameter affine model is used to describe the camera motion, and the ILSE technique is used to calculate the parameters. Then camera motion will be classified into three classes: pan, tilt and zoom with an accurate classification method based on finite-state automata. The whole system can be described by Figure 2.

2.2 Motion Vector Extraction and Normalization

We extracted motion vectors directly from MPEG compressed video data. There are three types of frame and five types of macro-block. A macro-block can have zero, one or two motion vectors depending on its type and its frame type. Moreover, these motion vectors can be forward-predicted or backward-predicted with respect to the reference frame. In order to compare these vectors with each other, we need a uniform set of motion vectors, independent of the frame type, the macro-block type and the direction of prediction. Our normalizing method based on the method proposed by [Kim04].

2.3 Motion Vector Refinement

In the previous works [Bouthemy99, Sequeira93, Tan95, Rath99, Sorwar03], motion vectors are directly used to estimate the camera

motion. Since the motion vectors extracted from MPEG video describe the combination of the camera motion and the object motion, these methods can not distinguish the camera motion from the object motion accurately.

In generally, there are four differences between the motion vectors which describe camera motion and which describe object motion:

(1) The direction of the motion vectors which describe camera motion is coincident, while the direction of the motion vectors which describe object motion, especially a lot of small objects' motion, is disordered.

(2) The count of the motion vectors describing camera motion is larger than the count of the motion vectors describing object motion.

(3) The distribution of the motion vectors which describe camera motion is wider in the frame than that of the motion vectors which describe object motion distribute.

(4) The motion vectors which describe object motion gather together around the objects, while the motion vectors which describe camera motion distribute separately all over the frame.

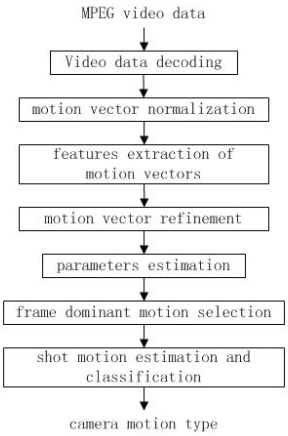

Figure 2. System overview

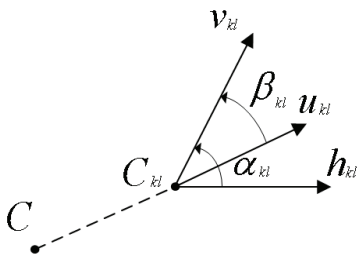

Figure 3. Angle α_{kl}, and β_{kl} of B_{kl}

2.3.1 Classification of Motion Vector

Firstly, we classify all motion vectors in a frame by calculating the histograms of α, β and intensity of blocks. The motion vectors which are classified in one class should have similar values of α, β or intensity. α, β are defined in Figure 3, where C is the center of the frame, C_{kl} is the center of block B_{kl}, u_{kl} is the motion vector of this block.

2.3.2 Spatial Features of Motion Vectors

Since the motion vectors classified into one class are similar, each class and the vectors in the class can represent certain motion.

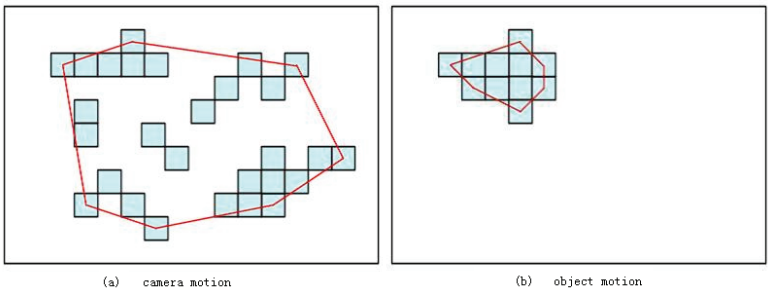

Figure 4. The convex polygon which can cover all the blocks, we can separate them by *Area*(*i*)

Let *Class*(*i*) be the *i*th class and *Count*(*i*) be the count of motion vectors in *Class*(*i*) to present whether the motion is the dominant motion in the frame. If a block is regarded as a point which locates at the center of the block, we can find the unique convex polygon *Poly*(*i*) which is the least polygon that can cover all the point in *Class*(*i*). Let *Area*(*i*) be the area of *Poly*(*i*) to represent the extent of the motion

vectors distribution. Let *Density*(*i*) be *Count*(*i*)/*Area*(*i*) to represent the density of the motion vectors in *Poly*(*i*).

Assume that there are *N* blocks in *Class*(*i*), let C_j be the center of *j*th block in the class, *Center*(*i*) be the center of this class. Let *Semidiameter*(*i*) = ($\sum_j$|*Center*(*i*) - C_j|) / *N* be the mean distance of *Class*(*i*) to represent the extent of the motion vectors distribution in the frame. If a block is regarded as a square, we can calculate the count of the borders when the neighbor block does not belong to the same class. Let *Perimeter*(*i*) be the sum of the count of the each block in *Class*(*i*). Let *Compactness*(*i*) be *Perimeter*(*i*)/*Count*(*i*) to represent the adjacency character of the blocks of *Class*(*i*).

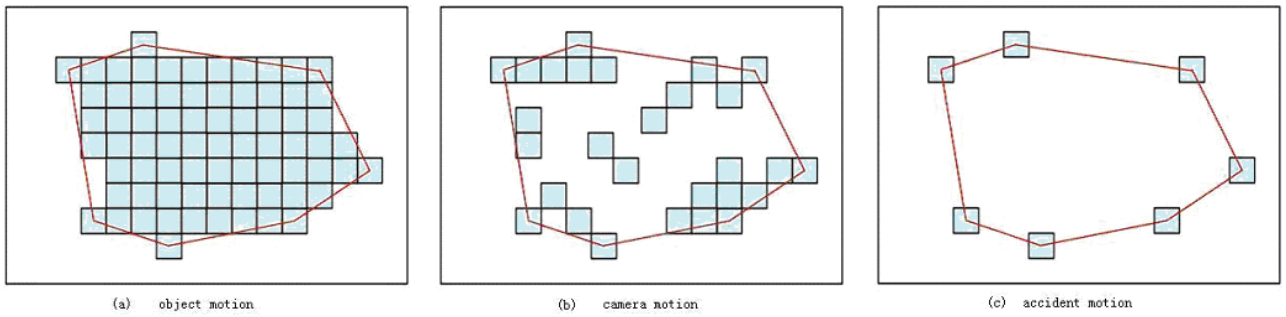

Figure 5. The classes belonging to different motion types have the same *Poly*(*i*), the same *Area*(*i*) and the different *Density*(*i*).

2.3.3 Refinement

We calculate the following features for each class to separate camera motion from object motion:

(1) *Count*(*i*). The count is almost big when the class represents camera motion. On the contrary, the count is often little when the class represents object motion.

(2) *Area*(*i*). The area is almost large when the class represents camera motion. On the contrary, the area is often small when the class represents object motion.

(3) *Density*(*i*). The value of density is middle when the class represents camera motion. The value of density is high when the class represents object motion. The value is low when the class represents accidental and sporadic motion.

(4) *Semidiameter*(*i*). The value of semi-diameter is high when the class represents camera motion. On the contrary, the value is low when the class represents object motion.

(5) *Compactness*(*i*). The value of compactness is low when the class represents camera motion. On the contrary, the value is high when the class represents object motion.

We give each block a score by these features and experiential thresholds. We estimate each block by its score and select the blocks which represent camera motion.

2.4 Frame Dominant Motion Estimation

2.4.1 Motion Parameters Estimation

We use a four-parameter affine motion model [Rath99] to describe camera motion:

$$\begin{bmatrix} v_{klx} \\ v_{kly} \end{bmatrix} = \begin{bmatrix} z_x c_{klx} \\ z_y c_{kjy} \end{bmatrix} + \begin{bmatrix} p \\ t \end{bmatrix} \qquad (1)$$

where v_{kl} = (v_{klx}, v_{kly}) is the motion vector of B_{kl}, C_{kl} = (C_{klx}, C_{kly}) is the center of B_{kl},. z_x and z_y are scale parameters in order to represent zoom motion. *p* is the parameter to represent pan motion, and *t* represents tilt motion.

Then, the model parameters of *Frame*(*s*) are calculated using the Iterative Least Squares Estimation [Rath99] technique. The ILSE technique can provide *M* different results for *Frame*(*s*). Let *Para*(*s*, *j*) = [z_x, z_y, *p*, *t*] be the *j*th result of *Frame*(*s*), *j*=0, 1,..., *M*-1. If a motion vector matches with *Para*(*s*, *j*), it will belong to *Para*(*s*, *j*). Matching means that a motion vector lies within a threshold from

Para(*s*, *j*). *Sum*(*s*, *j*) is the number of the motion vector who matches with *Para*(*s*, *j*) in *Frame*(*s*).

2.4.2 Frame Dominant Motion Extraction

After the parameter estimation, we select the *Para*(*s*) which have most belongings to describe the dominant motion in *Frame*(*s*).

$$Para(s) = Para(s, t), \text{ if } Sum(s, t) = \text{Max}\{\text{Sum(s, 0), Sum(s, 1), ..., Sum(s, M-1)}\} > K*L/2 \qquad (2)$$

Then, *Para*(*s*) is regarded as the representation of the dominant motion in *Frame*(*s*).

2.5 Long Term Camera Motion Estimation and Classification

In generally, camera motion occurs continuously during a long term in a shot. A motion is considered to present in the shot if it occurs anytime within the shot. The camera motion will be classified into three classes: pan (left or right) or track, tilt (up or down) or boom, and zoom (in or out) or dolly.

Let P_k be the motion parameter of the motion type k, and $P_k(s)$ be the motion parameter of the motion type k in *Frame*(*s*), $k \in \{pan, tilt, zoom\}$. We detect and classify the camera motion in a shot using experiential rules and thresholds. A camera motion occurs in a shot if the following rules are satisfied:

(1) $\left|\sum_{s=i}^{j} P_k(s)\right| > T_k(sum)$. The $T_k(sum)$ is a threshold for the summation of the parameter P_k. The camera motion should occur continuously and noticeable, so the summation should exceed $T_k(sum)$.

(2) $j-i+1 > T_k(span)$. The camera motion should occur continuously, so the duration should exceed $T_k(span)$.

(3) $\frac{1}{j-i+1}\left|\sum_{s=i}^{j} P_k(s)\right| > T_k(avg)$. The camera motion should occur uninterrupted and perceptibly, so the average of the summation should exceed $T_k(avg)$.

(4) $\left|\sum_{s=i}^{j} P_k(s)\right| > T_{kl}(blend)\left|\sum_{s=i}^{j} P_l(s)\right|$. The camera motion should occur perceptibly if other type camera motion occurs at the same time, so the ratio of summation of this type and other types should exceed $T_{kl}(blend)$. In our system, the arrangement of $T_{kl}(blend)$ is [0.2, 0.4].

If the rules are satisfied, we declare that this type of camera motion occurs in the current frame.

2.6 Results and Evaluation

We have submitted 7 runs to TRECVID2005. The evaluation results of these runs are listed in Table 3~6.

Table 3. The evaluation results of our runs for pan

sysId	TP	TN	FP	FN	Prec.	Recall	F1
THU_06	518	1138	21	69	0.961	0.882	0.919
THU_03	484	1154	5	103	0.99	0.825	0.900
THU_07	485	1148	11	102	0.978	0.826	0.895
THU_02	461	1156	3	126	0.994	0.785	0.877
THU_05	492	1141	18	95	0.965	0.838	0.897
THU_01	459	1156	3	128	0.994	0.782	0.875
THU_04	437	1159	0	150	1.000	0.744	0.853

Table 4. The evaluation results of our runs for tilt

sysId	TP	TN	FP	FN	Prec.	Recall	F1
THU_06	167	1158	1	43	0.994	0.795	0.883
THU_03	163	1157	2	47	0.988	0.776	0.869
THU_07	151	1158	1	59	0.993	0.719	0.834
THU_02	143	1159	0	67	1.000	0.681	0.810
THU_05	158	1158	1	52	0.994	0.752	0.856
THU_01	134	1159	0	76	1.000	0.638	0.778
THU_04	130	1159	0	80	1.000	0.619	0.764

Table 5. The evaluation results of our runs for zoom

sysId	TP	TN	FP	FN	Prec.	Recall	F1
THU_06	400	1150	9	111	0.978	0.783	0.869
THU_03	393	1149	10	118	0.975	0.769	0.859
THU_07	370	1155	4	141	0.989	0.724	0.836
THU_02	364	1157	2	147	0.995	0.712	0.830
THU_05	312	1152	7	199	0.978	0.611	0.752
THU_01	350	1157	2	161	0.994	0.685	0.811
THU_04	311	1159	0	200	1.000	0.609	0.756

Table 6. The evaluation results of our runs for all motion types

sysId	Mean_precision	Mean_recall	Mean_F1
THU_06	0.978	0.820	0.892
THU_03	0.984	0.790	0.876
THU_07	0.987	0.756	0.856
THU_02	0.996	0.726	0.839
THU_05	0.979	0.734	0.838
THU_01	0.996	0.702	0.823
THU_04	1.000	0.657	0.792

3. High-Level Feature Detection

3.1 The TSINGHUA TRECVID 2005 Concept Detection System

The systems we built this year rely heavily on the visual information. Visual features include Color Auto-Correlograms, Color Coherence Vector, Color Histogram, Color Moment, Edge Histogram and Wavelet Texture. Two different systems using regional and global image features are compared to explore the effectiveness of regional features. In the regional system, keyframes are segmented and regional feature of all the six types mentioned above are extracted. Then support vector machine (SVM) classifier with Earth Mover Distance (EMD) kernel is built [Jing04]. In the global system, the six types of global features are extracted for each keyframe directly. Then the classifier ensembles for detecting each concept are formed by using the Relay Boost algorithm explained below. This is followed by a concept context module. We mainly tried two approaches, one based on stacked SVM and the other based on weighted sum of the confidence scores of the related concepts. We then apply time clustered post-filtering to remove false positive shots. Based on the two systems we have our 7 runs.

Often the concept detection task suffers both from few positive instances and from large imbalanced data sets in which negative instances heavily outnumber the positive instances. These two problems are intrinsic for the concept detection task. We propose a Relay Boost (RL.Boost) approach to tackle them. Using SVM as the base classifier, RL.Boost actively selects representative instances from the imbalanced data set for each base classifier and combines these classifiers in a boosting-like ensemble. RL.Boost also shares the accumulated training error pattern to explore the multiple features for better performance. In this way, the traditional Low-level Feature based model and fusing process for these models are integrated into one module, our Relay Boost Ensemble module.

The TSINGHUA TRECVID 2005 Concept Detection systems are shown in the Figure 6.

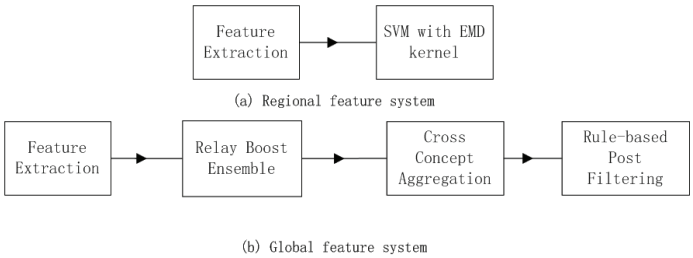

Figure 6. Concept detection system

The table below lists the name of the TSINGHUA run and its description.

Table 7. The description of each submission

Run	Description
A_ua_1	Regional feature system: regional features + EMD + SVM
A_ua_2	Global feature system: Relay.boost+ filtering
A_ua_3	Global feature system: Semi-supervised learning with test data
A_ua_4	Global feature system: a mixed run[1]
A_ua_5	Global feature system: 34 concept Relay.boost + stacked svm
A_ua_6	Global feature system: related concept weighting
A_ua_7	Global feature system: Relay.boost + stacked svm +filtering

3.2 Concept Detection Results

Figure 7 compares TSINGHUA performance with the best and the median performance across all runs for the 10 concepts. Figure 8 compares TSINGHUA runs' Average Precision at a depth of 2000.

3.3 Concept Detection Lessons

The following lessons were learnt from across all seven runs submitted:

1. Multi-feature fusion improves over any single modality significantly.
2. The regional features are less expressive than the global one in the current implementation.
3. Sometimes the concept context model is useful, but we do not know exactly when will be so.
4. Filtering by time clustered analysis can not substantially improve concept detection.
5. We should pay more attention to the characteristics of concept detection itself and devise more suitable learning algorithms beyond

[1] 39 and 44 (fire and mountain) come from regional Relay boost results. Others are slightly modified semi-supervised results.

Relay Boost.

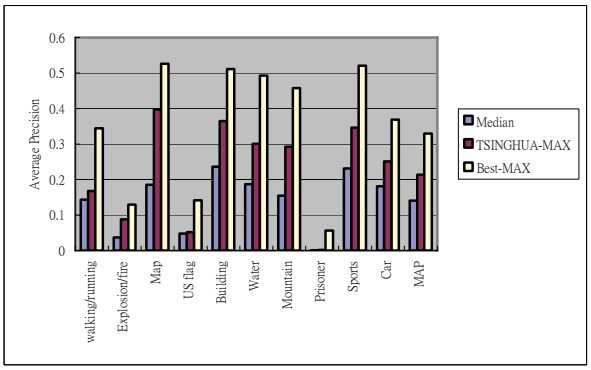

Figure 7. Performance of our system and other systems

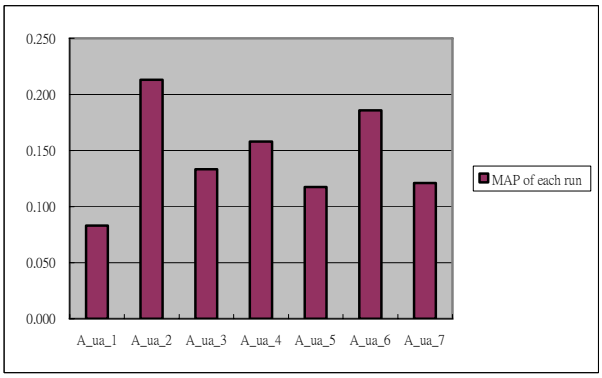

Figure 8. AP of our systems at a depth of 2000

4. Automatic search

4.1 System Overview

Our video search systems have three basic retrieval models: a text model, which searches video shots through ASR (automatic speech recognition); an image model, which searches video shots via region-based image matching; a concept model, in which queries and shots are automatically parsed into vectors in a concept space and relevant shots are then retrieved by measuring vector similarity. Based on these models, we further develop some combination systems. In the score fusing system, search results are ranked by fusing the scores generated from the basic retrieval models. While in the query type based fusing system, the queries are first classified into two classes, and then retrieval is performed over different models. An overview of the system is shown in Figure 9.

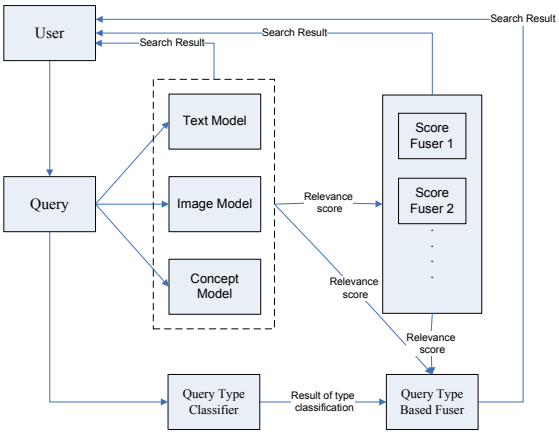

Figure 9. Overview of automatic search system

Among our 7 submissions, the results show that when searching for general topics, which are always less related to person, combining text and concept models does better than only using text model.

4.2 Basic Retrieval Model

4.2.1 Text Model

Two text-based retrieval systems have been implemented, a baseline system and the other system with an additional query expansion module.

The baseline system is a fully automatic system entirely based on the transcripts from the English ASR output provided by NIST. For those non English video materials (i.e., video data from Chinese and Arabic TV stations), we use the corresponding machine-translated

transcripts. An OKAPI-TF formula [Robertson95][Zhai01] is adopted. Simple preprocessing such as removal of stop words and the phrase "find shots of" is performed automatically on the NIST supplied topic text before submitting queries to the system. As a variation of the traditional TF-IDF model, the retrieval model used here is relatively and empirically robust. Heuristic parameters, e.g., k1 and b in the document TF function, are trained to maximize mean average precision (MAP) score on TRECVID2004 data. Besides, pseudo feedback based on a simple version of Rocchio algorithm [Zhai01] is performed by using the top twenty results as relevant to execute one round relevance feedback. Our experiments on TRECVID2004 data showed that with pseudo feedback, the performance of the text retrieval system was improved obviously.

In many cases, user query keywords are very limited. And words people use to describe the same object or topic may have tremendous diversity which makes lexical matching methods incomplete and imprecise. Thus query expansion is necessary to acquire a better retrieval. There are various ways to do the task, automatically or manually. Qi Tian et al [Chua04] utilized Google and WordNet to expand query. Ellen M. Voorhees [Voorhees94] used lexical-semantic relations and examined it in the large, diverse TREC collection. Here we use a latent semantic analysis method [Landauer98][Dumais88] to build semantically correlated word sets for each query term and then use these correlated words to reform and expand query. The main idea of LSA is to map each document (any query can be considered as a document) vector to a lower dimensional representation in a so-called latent semantic space. Experiments show that for the TREC video search task, recall can be obviously enhanced with the LSA query expansion. However, because the expansion task was executed automatically without perceptual or thesaurus knowledge, some words which are indeed not relevant will be also included in the expanded query, which may have harmful influence on precision.

4.2.2 Image Model

We use a region-based image retrieval model. Several image and video examples are given by TRECVID for every query. Start time and stop time are included for each of the video examples, from which one keyframe is extracted. Image examples and keyframes of video examples constitutes the whole set of image examples.

First, region-based features are extracted for every image in either the example set or TRECVID 2005 test collection. In the experiments, JSEG algorithm is adopted for image segmentation. Each region is represented by its feature and region importance, which can be defined as (r_f, r_i). Feature r_f may include color moment, color histogram, and color-autocorrelogram. Region importance r_i is initialized as the area ratio of the region. Therefore, the feature of an image is described as a set of region features, which can be denoted as {(r_{f1}, r_{i1}), (r_{f2}, r_{i2}), ..., (r_{fn}, r_{in})}. Distance between two images is calculated by EMD distance.

Second, for a given query, distances between every image in the query example set and TRECVID 2005 test collection are calculated. For a specified keyframe in the test collection, the minimum of distances from every image in the query example set to the keyframe is viewed as the similarity of the keyframe to the given query. The minus minimum is used as the confidence value that a keyframe in the test collection is relevant to a given query.

Finally, the confidence on keyframes for a give query is converted to that on shots, which is ranked according to the confidence value. The top 1000 shot list from the converted confidence file is submitted for evaluation of TRECVID 2005 search task.

4.2.3 Concept Model

Concept based video retrieval methods have high expressive power, and somewhat solve the semantic gap. Much work has been done on concept based video retrieval. Snoeck et al[Snoek04] developed a semantic video search engine for the interactive search task, which included a concept retrieval module, and the relations of search topics to concepts were judged by expert users.

In our opinion, concept based video retrieval is the research direction of video retrieval. We propose a multimodal understanding method

to map queries and video shots to concept space, and then evaluate query-shot similarity with vector computing in concept space. We use the HLFD (High Level Feature Detection) method of our group to parse video shots into concept vectors. For queries, besides HLFD, we also use a text understanding method to get a more precise mapping.

Using textual information, the basic and intuitive idea is to explore the latent relationships between text feature and concepts, and then obtain semantic relations between concepts and queries resultantly. For text always explicitly contains most semantic information, our approach has inherent semantic meanings. As we have applied LSA to mine semantic correlations among text terms, naturally if visual concepts could be represented by some kind of text pattern, the LSA method would be applicable to analyze relationships between text feature and visual concepts. For each concept detected in the TRECVID high-level feature extraction task, we consider it as a virtual term in corresponding shots. For example, use word "CONCEPT_BOAT" as representation if the "boat" concept is detected in some shots. By adding all these conceptual, virtual terms to the original corpus of ASR transcripts, we could apply the same LSA method to dig up the correlations between query terms and conceptual terms. Consider a "concept space" with each concept as a dimension, every term in a query could then be mapped to a quantified conceptual vector. Using term weights measured by TF-IDF weighting technique in the whole collection, conceptual vector for the query is just the linear combination of vectors of query terms.

Therefore, similar to vector model of text IR, a video shot s and a user query q are represented as t-dimensional vectors in concept space. Then we can evaluate the degree of similarity of the video shot s with regard to the query q as the correlation between the vectors s and q. This correlation can be quantified by the dot product of these two vectors or by the cosine of the angle between these two vectors, and so on. Then we can retrieve video shots and get a ranked result in the concept space.

Compared to traditional methods, our concept model has two great advantages. First, our concept model can automatically assign the queries and video shots with concepts, which can save the human labor and doesn't require the user familiar with the concepts. So the model is practicable and can be easily adapted to a higher dimension concept space. Secondly, our concept model assigns non-binary weights to concepts for queries and video shots, which can make the result more precise.

4.3 Fusion

Fusion of multiple modalities, e.g., text features, low level visual features and high level features, is crucial for video retrieval systems, especially for automatic types without any user interaction. In the concept model, we have used the multimodal information of video. However, as the concept model is still too shallow, it may induce information loss and still can not get a satisfying result. So, to fully exploit the multimodal information of video, we combine our three basic retrieval model to get some hybrid systems. In the score fusing system, the results are ranked by fusing the scores generated from the basic retrieval models. While in the result fusing system, the queries are classified into two classes, and then retrieved using different retrieval models.

4.3.1 Score Fusing

Many related works have been done on designing strategies of multi-modality combination. Most common method is to fuse the similarity score of different retrieval results.

For a video shot s and a user query q, our three basic retrieval models will give three similarity score S_T, S_I, S_C, which have been normalized into rank-based probability scores (i.e., within [0,1]).

As the three models have evaluated the similarity in different aspect, we use multiply operation instead of weighted sum. And the score of text or image retrieval model is normalized to [1, e]. That is because we assume that the concept model score is meaningful for all the shots, but the text or image model score is meaningful for the top portion, while for the remainders the score is not reasonable. So we

normalize the score to [1, *e*] to reduce the distance in the sense of multiple. The fusing result S_F can be got through equation 3.

$$S_F = S_C \times e^{S_T} \quad \text{or} \quad S_F = S_C \times e^{S_I} \qquad (3)$$

4.3.2 Query Type Based Fusing

There have been much attention on query type classification. It is hypothesized that we can classify the query into proper type and there is a best retrieval model for queries of the same type. A recent work from Rong Yan et al [Yan04] proposed using query-class dependent weights within a hierarchical mixture-of-expert framework to combine multiple retrieval results. Compared with previous fixed weighting methods, this approach showed a considerable improvement. However, using EM algorithm to learn weights, it was somewhat like data fitting and had less semantic meanings. Differently, we use the query understanding result to get a simple classifier.

As mentioned before, we propose a multimodal understanding method to map queries to concept space. Assume the relation of a query and "person" concept is C-Q, we can find that a large C-Q always means query of a specific person. So we can classify the queries into person / non-person categories with a threshold for C-Q values. In our system, we just set a heuristic threshold of 0.076. And we use text model for queries of person category, while concept model for queries of non-person category.

4.4 Experiment and Result

We submitted 7 video automatic search runs. We submitted one baseline run for each basic retrieval model, and an additional query expansion run for text model. Then we fused the concept model with text model and image model separately and obtained two runs. Finally we did the query type classification and use text model for queries of person category, while concept model for queries of non-person category. It was the last run. Detail description of these 7 runs is as follows:

Table 8. The description of each submission

Run	Description
THU01	uses Text (ASR) and Concept Model, and the topics are classified into two classes: person, non-person, by the result of Person Model detection.
THU02	uses only Text (ASR) Model.
THU03	uses only Concept Model.
THU04	uses only Image Model.
THU05	uses Text (ASR) and Concept Model.
THU06	uses Image Model and Concept Model.
THU07	uses Text (ASR), with query expand and studio removing.

And the evaluation result is illustrated in Table 9. We can see that, although the result of concept model (run3:THU03) seems too bad, when combined with text model, the result (run5:THU05) is comparative with text model (run2:THU02). Especially, the combined result does well in the generic topics, for instance, '0166: palm trees','0168: a goal being made in a soccer match', and so on. So by using query type classification, we can utilize the different characteristics of the text model and the combined model. And the result (run7:THU07) shows its advantage.

Table 9. Automatic search results for all the queries

queryid	THU01	THU02	THU03	THU04	THU05	THU06	THU07
149	0.1097	0.1097	0.0004	0	0.0244	0.0001	0.0482
150	0.0139	0.0139	0	0	0.0001	0	0.0117
151	0.1101	0.1101	0.0172	0.0748	0.0998	0.0501	0.2514

152	0.2741	0.2741	0.0031	0.0051	0.1619	0.0052	0.2536
153	0.2651	0.2651	0.0004	0.0001	0.2043	0.0001	0.2386
154	0.1799	0.1799	0.0006	0	0.1731	0.001	0.1945
155	0.0002	0.0004	0.0002	0.0853	0.0002	0.0046	0.0001
156	0.047	0.1019	0.002	0.0006	0.047	0.0021	0.0828
157	0.0063	0.0063	0.0021	0.0008	0.0041	0.0032	0.0036
158	0.0866	0.1356	0.0041	0.0021	0.0866	0.0052	0.0376
159	0	0.0004	0	0	0	0	0.0001
160	0.0034	0.0016	0.0042	0.0022	0.0034	0.0074	0.0003
161	0.0455	0.0358	0.0128	0.0025	0.0455	0.0108	0.0426
162	0.0081	0.0021	0.0073	0.0064	0.0081	0.0115	0.0024
163	0.0045	0.0045	0.001	0.0039	0.0032	0.0015	0.0046
164	0.1216	0.0906	0.1143	0.0009	0.1216	0.1032	0.0983
165	0.0791	0.019	0.0659	0.0178	0.0791	0.1096	0.0084
166	0.0226	0.0176	0.0088	0.0018	0.0226	0.0105	0.0134
167	0.0002	0.0005	0.0001	0.0005	0.0002	0.0004	0
168	0.0768	0.0414	0.0758	0.0146	0.0768	0.0716	0.0312
169	0.0336	0.0272	0.0131	0.0073	0.0336	0.0158	0.0147
170	0.0299	0.0008	0.0344	0.0121	0.0299	0.0379	0.002
171	0.4133	0.1592	0.1918	0.0029	0.4133	0.2081	0.3051
172	0.0065	0.0065	0.0024	0.005	0.0024	0.0018	0.0018
173	0.0808	0.0668	0.0234	0.0103	0.0684	0.0276	0.0686

5. Conclusions

The paper presents our approaches in four tasks in TRECVID 2005. Our novel shot boundary detection system consists of three modules to detect FOI, CUT and long GT in turn. CUT and long GT detectors are support vector machines taking score vector generated by graph partition model. The system achieves highest F1 value for all transition among all the runs submitted to TRECVID. In our camera motion detector, some well defined features and rules are used to distinguish camera motion from object motion. After estimating affine model parameters of camera motion and finding the dominant one in a frame, camera motion type of a shot is determined with a finite-state automata. The evaluation shows that the system is very accurate. We try regional feature based approach and global feature based approach in concept detection. In the regional system, keyframes are segmented and features are extracted for each region. Then a support vector machine classifier with Earth Mover Distance (EMD) kernel is built. In the global system, features are extracted for each keyframe directly. Then the classifier ensembles for detecting each concept are formed by using the Relay Boost algorithm. This is followed by a concept context module and a time clustered post-filtering module to remove false positive shots. From the results, we find that multi-feature fusion improves over any single modality significantly. Our automatic video search systems have three basic retrieval models: a text model based on script generated by ASR, an image model based on region-based image matching and a concept model which automatically parses the queries and video shots into concept vectors, and then searches video shots based on query-shot similarity computing in concept space. Besides these models, we also develop some combination systems, i.e. score fusing system, and query type based fusing system. Among our 7 submissions, the results show that when searching for general topics, which are always less related to person, combining text and concept models performs better than only using text model.

Acknowledgements

This work is partially supported by National Key Basic Research Project of China (2004CB318108) and National Natural Science Foundation of China (60135010).

References

[thu_notebook04] J.Yuan, W.Zheng, L.Chen, D.Ding, et al. Tsinghua University at TRECVID 2004: Shot Boundary Detection and High-level Feature Extraction. In: Proceedings of TRECVID 2005 Workshop.

[vcip05] W.Zheng, J.Yuan, H.Wang, F.Lin, B.Zhang. A Novel Shot Boundary Detection Framework. In: Proc. of VCIP 2005. Beijing.

[pakdd05] J.Yuan, B.Zhang, F.Lin. Graph Partition Model for Robust Temporal Data Segmentation. In: Proc. of PAKDD 2005, 758-764, May 18-20 2005.

[acmmm05] J. Yuan, J.Li, F. Lin, B. Zhang. A Unified Shot Boundary Detection Framework Based on Graph Partition Model. In: Proc. of ACM Multimedia 2005，November 6-11,2005

[libsvm] C.-W. Hsu, C.-C. Chang, C.-J. Lin. A practical guide to support vector classification. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[Kim04] N. W. Kim, T. Y. Kim, and J. S. Choi, “A Probability-Based Flow Analysis Using MV Information in Compressed Domain,” in Mexican International Conference on Artificial Intelligence, pp. 592-601, 2004.

[Bouthemy99] P. Bouthemy, M. Gelgon, and F. Ganansia, “A unified approach to shot change detection and camera motion characterization,” IEEE Transaction on Circuits and Systems for Video Technology, vol. 9, pp. 1030-1044, 1999.

[Sequeira93] M. M. de Sequeira, and F. Pereira, “Global motion compensation and motion vector smoothing in an extended H.261 recommendation,” in Video Communications and PACS for Medical Applications, Proc. SPIE, pp. 226-237, 1993.

[Tan95] Y. P. Tan, S. R. Kulkarni, and P. J. Ramadge, “A new method for camera parameter estimation,” in Processing of International Conference Image Processing, vol. 1, pp. 405-408, 1995.

[Rath99] G. B. Rath and A. Makur, “Iterative least squares and compression based estimations for a four-parameter linear global motion model and global motion compensation,” IEEE Transactions on Circuits & Systems for Video Technology, vol. 9, pp. 1075-1099, 1999.

[Sorwar03] G. Sorwar, M. Murshed, and L Dooley, “Fast global motion estimation using iterative least-square estimation technique,” Fourth International Conference on Information, Communications & Signal Processing and Fourth IEEE Pacific-Rim Conference On Multimedia 15-18 December 2003, Singapore.

[Jing04] Feng Jing, Mingjing Li, Hong-Jiang Zhang, Bo Zhang: An efficient and effective region-based image retrieval framework, IEEE Transactions on Image Processing, Vol. 13(5) pp 699-709, May 2004

[Robertson95] S.E. Robertson, S. Walker, and M. Sparck Jones, et al., "Okapi at TREC-3," Proc. Second Text Retrieval Conf. (TREC-3), 1995.

[Zhai01] Zhai, C. Notes on the Lemur TFIDF model. Unpublished report. 2001

[Landauer98] Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. In Discourse Processes, 25, 259-284. 1998

[Chua04] T.-S. Chua, S.-Y. Neo and et al, TRECVID 2004 Search and Feature Extraction Task by NUS PRIS, TREC Video Retrieval Evaluation Online Proceedings, 2004

[Voorhees94] Ellen M. Voorhees , Query expansion using lexical-semantic relations, In Proceedings of ACMSIGIR. Dublin, Ireland, pages 61--69. ACM/Springer, 1994

[Dumais88] Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Using Latent Semantic Analysis to Improve Access to Textual Information, Bell Communications Research, 1988

[Yan04] Rong Yan, JunYang, Alexander G. Hauptman, Learning Query-Class Dependent Weights in Automatic Video Retrieval, Proceedings of the 12th annual ACM international conference on Multimedia, 2004

[Snoek04] C.G.M. Snoek, M. Worring, J.M. Geusebroek, D.C. Koelma, F.J. Seinstra, The MediaMill TRECVID 2004 Semantic Viedo Search Engine, TREC Video Retrieval Evaluation Online Proceedings, 2004