Exploring Temporal Consistency for Video Analysis and Retrieval

Jun Yang
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
juny@cs.cmu.edu

Alexander G. Hauptmann
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alex+@cs.cmu.edu

ABSTRACT

Temporal consistency is ubiquitous in video data, where temporally adjacent video shots usually share similar visual and semantic content. This paper presents a thorough study of temporal consistency defined with respect to semantic concepts and query topics using quantitative measures, and discusses its implications to video analysis and retrieval tasks. We further show that, in interactive settings, using temporal consistency leads to considerable improvement on the performance of semantic concept detection and retrieval of video data. Specifically, an active learning method with temporal sampling strategy is proposed for building classifiers of semantic concepts, and a temporal reranking method is proposed for improving the efficiency of interactive video search. Both methods outperform existing methods by considerable margins on the TRECVID dataset.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing methods*

General Terms

Algorithm, Performance, Experimentations

Keywords

Video retrieval, Semantic concept detection, Temporal consistency, Active learning, Interactive search

1. INTRODUCTION

Video analysis and retrieval has become an active and challenging research area in recent years. Various approaches have been proposed for detecting semantic video concepts, i.e., finding video shots that match generic concepts such as *anchor*, *outdoor*, and *sports*, or video retrieval, i.e., finding shots that answer specific query topics such as *"Find President Bush speaking in front of a U.S. flag"*. A general approach to such tasks is to study the pattern in the textual and/or image features of sample data (i.e., labeled examples of concepts or query examples), typically using a machine learning method, which finds more data in the collection that match such pattern. Related research issues such as the fusion of multi-modal features [15, 11], inter-concept relationships[7], and query semantics [8, 15], have been extensively studied. However, an equally important issue, the temporal consistency of video data, has not been well studied in the context of semantic concept detection and retrieval despite its potential value to such tasks.

Video data exhibit strong consistency along the temporal domain, which ensures the footage is visually smooth and semantically coherent. In this paper, temporal consistency refers to the observation that temporally adjacent video shots have similar visual and semantic content. This implies that the relevant shots matching a specific semantic concept or a query topic tend to gather in temporal neighborhoods or even appear next to each other consecutively. As shown in Figure 1, if a shot in broadcast news video matches the concept "sports", chance is high that the previous and next few shots are also about sports.

Temporal consistency provides valuable contextual clues to video analysis and retrieval tasks. In most existing approaches, the relevance of a given shot with respect to a semantic concept or query topic is determined based on its own content and independently from its neighboring shots. With temporal consistency, one can make more informed prediction as to the relevance of the shot by considering the relevance of its neighboring shots, thus enhancing the overall performance of the predictions. This poses the question of how to quantitatively measure the strength of temporal consistency w.r.t different concepts or query topics, as well as how to design approaches that make use of such temporal consistency.

However, the potential value of temporal consistency is shadowed by the high variance of video data. It is common that the video shots relevant to the same concept (or query) are visually dissimilar, posing difficulties to machine learning methods due to the discrepancy of distribution between training examples and testing data. For example, in Figure 1 the first 4 sports shots are about basketball, which are visually similar with yellow as their dominant color, while the last 3 sports shots are about baseball and dominated by the green color of baseball court. A color-based "sports" classifier trained from the basketball shots are likely to misclassify the baseball shots as irrelevant since visually they

MIR'06, October 26–27, 2006, Santa Barbara, California, USA.
Copyright 2006 ACM 1-59593-495-2/06/0010 ...$5.00.

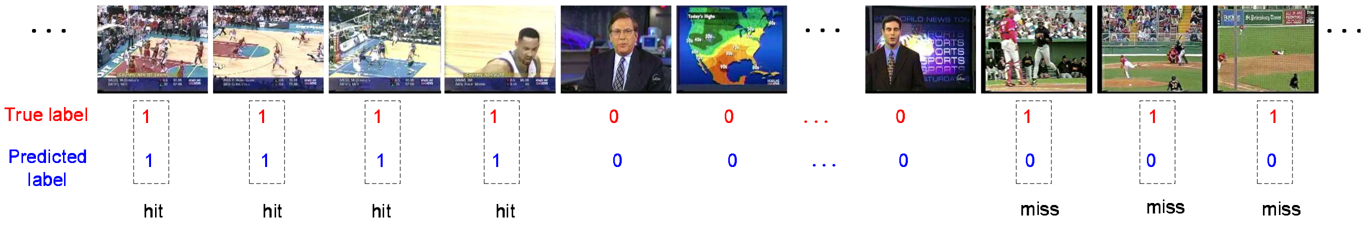


Figure 1: A sequence of example shots in news video footage, with their true and predicted labels with respect to a semantic concept "sports" .

look very different. While this data variance issue is a general problem of machine learning, it is particularly severe in video data because, due to the consistency of adjacent shots, the classifier tends to miss not a few individual outliers but a whole sequence of relevant shots. Furthermore, one can hardly rely on temporal consistency to correct the (mis)prediction on one shot from its adjacent shots, because the adjacent predictions also tend to be wrong.

To exploit temporal consistency for better video analysis and retrieval while circumventing the data variance problem is an important and challenging task. In this paper, we present a thorough study of temporal consistency in video data with ample statistics on a benchmark dataset, and discuss its relationship with the data variance problem and implications to video analysis and retrieval tasks. We then propose two simple but effective methods that utilize temporal consistency to improve the performance of semantic concept detection and retrieval in interactive settings. The major contributions of this paper includes:

1. We provide quantitative measures of temporal consistency defined in terms of semantic concepts or query topics in a benchmark video collection. We also discuss the relationship between temporal consistency and data variance, as well as their implications to video analysis and retrieval tasks.

2. We propose an active learning method with *temporal sampling strategy*, which exploits temporal consistency to build high-quality classifiers for semantic video concepts with minimum user efforts.

3. We propose a computationally efficient *temporal reranking method* for incremental improvement of the ranking list of relevant shots during interactive search.

Note that we do not solve or intend to solve the general problem caused by high data variance. We discuss high variance problem because it compromises the value of temporal consistency especially in a non-interactive setting. The methods to be presented in this paper negotiate a way of exploiting temporal consistency while going around the high variance problem in interactive settings.

2. RELATED WORK

For video concept detection and retrieval, a general framework is to build a supervised classifier from sample shots, such as query examples or manually labeled shots, and use the classifier to find more relevant shots whose features match those of the samples. Some important issues have been explored within this framework, such as the fusion of multimodality information [15], the relationships between different semantic concepts [7], the modeling of query types and semantics [15, 8]. Given the relatively low performance of the automatic approaches, there has been also research on video analysis and retrieval in interactive settings. Active learning has been a popular method in this area, which is used in [3] and [11] for iteratively building better classifiers of semantic concepts by choosing the most informative shots for user feedbacks. The methods to be presented in this paper also consider the interactive setting of video analysis and retrieval, but with a focus on the temporal consistency issue which has not been much explored in existing work.

There have been many works on using the temporal information of video data. Visual content continuity (and discontinuity) has been the primary clue for both shot boundary detection methods [6] and story segmentation methods [16, 9]. Recently, hidden Markov model (HMM) and its variants have been used to discover meaningful patterns or events in soccer video [13] and news video [14, 4]. Moreover, motion trajectory which can be seen as microscopic temporal dynamics has been used for retrieving video objects [2]. The computer vision community has also used temporal constraint in video data to improve the tracking of people and physical objects, such as the work in [5]. More relevant to our work is that from Song et al. [11], which considers video temporal consistency when classifying semantic concepts such as indoor/outdoor. The idea is to cluster adjacent video shots with similar content into groups in order to avoid labeling (near-)duplicate shots, which improves the classifier with fewer manually labeled shots.

3. MEASURING TEMPORAL CONSISTENCY

In this paper, we examine the temporal consistency with respect to a given semantic concept or query topic, measured by the tendency that the relevant shots of this concept/query appear in temporal proximity. As different concepts and queries may exhibit varying levels of temporal consistency, providing a quantitative measurement is critical as to whether and how to use the temporal information for better video analysis and retrieval. In this section, we present two measurements of temporal consistency, namely transitional probability and pointwise mutual information, and apply them to a large number of concepts and queries on the TRECVID video collection [10]. This is to our knowledge the first work reporting quantitative measures of video temporal consistency on a benchmark dataset.

Transitional probability: Suppose $C_t \in \{0,1\}$ is a binary variable indicating whether shot t is relevant to semantic concept C. The transitional probability $P(C_t = 1|C_{t-1} = 1)$ is defined as the conditional probability of shot t being relevant to C given that its previous shot $t-1$ is

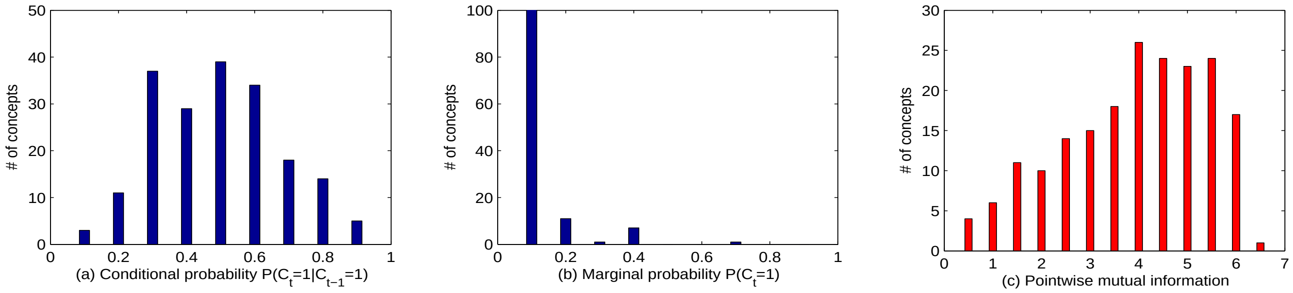


Figure 2: The distribution of transitional probability, marginal probability, and pointwise mutual information of 194 LSCOM concepts computed on the TRECVID 2005 development set. Note the scale difference.

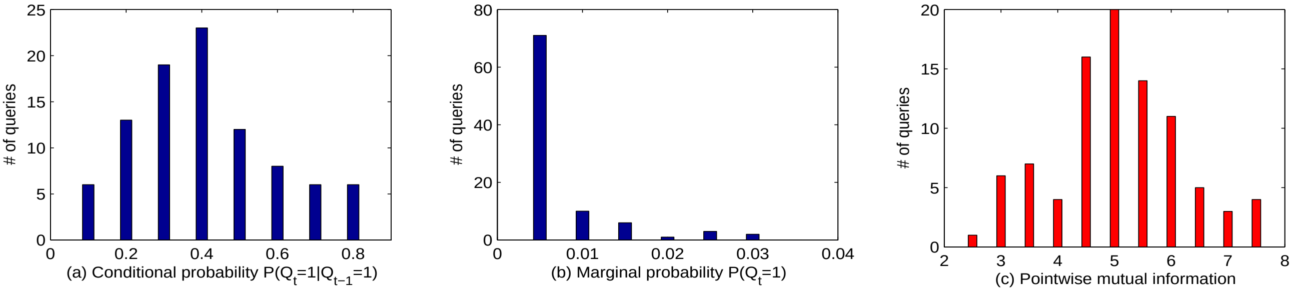


Figure 3: The distribution of transitional probability, marginal probability, and pointwise mutual information of 96 query topics from TRECVID 2002 to 2005 on the respective test set. Note the scale difference.

relevant to C, which is calculated as:

$$P(C_t = 1|C_{t-1} = 1) = \frac{\#(C_t = 1, C_{t-1} = 1)}{\#(C_{t-1} = 1)}. \qquad (1)$$

where $\#(C_{t-1} = 1)$ is the total number of relevant shots in the collection, and $\#(C_t = 1, C_{t-1} = 1)$ is the total number of consecutive shot pairs that are both relevant to C. Clearly, transitional probability provides a quantitative measure of the strength of the temporal consistency w.r.t a semantic concept.

We examine the transitional probability of the semantic concepts defined in LSCOM [1], or large scale concept ontology for multimedia, based on the TRECVID 2005 development set which contains over 74,000 shots. Among the 320 LSCOM concepts, we filter out the extremely rare concepts (with $< 0.1\%$ relevant shots), and compute the transitional probability of the remaining 194 concepts using the provided true labels. For comparison purpose, we also compute the (marginal) probability $P(C_t = 1)$ of each concept, which is equal to the ratio of its relevant shots in the whole collection.

We plot the distribution of the transitional and marginal probability of the 194 concepts in Figure 2 (a) and (b). We see that while the marginal probability of most concepts are below 0.1 with an average of 0.038, their transitional probability is distributed in much higher range with an average of 0.452. This means that once the label of the previous shot is known, one can improve the prediction on the label of the current shot from hopeless guess (0.038) to a coin-flip chance (0.452). This sharp contrast reveals the strong temporal consistency of most semantic concepts.

The transitional probability $P(Q_t = 1|Q_{t-1} = 1)$ of a query topic Q and its marginal probability $P(Q_t = 1)$ can be computed in exactly the same way. We repeat the above experiments using the 96 benchmark query topics collected from the TRECVID 2002 through 2005 data, for which the true labels are available on the respective test set. The distribution of their transitional and marginal probability are plotted in Figure 3(a) and (b). We observe even stronger temporal consistency among most of the query topics, with the average transitional probability (0.353) much larger than the average marginal probability (0.0043).

Pointwise mutual information: The transitional probability can be biased, because for a frequent concept (i.e., a concept with many relevant shots) the probability that we see two relevant shots being consecutive due to sheer chance is higher than a rare concept. A better metric of temporal consistency is the ratio of transitional probability against marginal probability. The logarithm of this ratio is the point-wise mutual information (PMI) metric, defined as:

$$PMI = log\frac{P(C_t = 1|C_{t-1} = 1)}{P(C_t = 1)} = log\frac{P(C_t = 1, C_{t-1} = 1)}{P(C_t = 1)P(C_{t-1} = 1)} \qquad (2)$$

The distribution of PMIs of the 193 concepts and 96 query topics is shown in Figure 2(c) and 3(c), respectively. We see that 163 out of 194 concepts, and all the 96 query topics, have PMI larger than 2, implying that their transitional probability is at least 7 times (e^2) of its marginal probability. In average, the transitional probability is 90.5 times larger than its corresponding marginal probability for a semantic concept, and 271.2 times larger for a query. These statistics show that, for most semantic concepts and query topics, knowing the relevance label of a shot is tremendously useful for predicting the label of the next shot.

The statistics of transitional probability and PMI shows strong temporal consistency in video data. Given that the transitional probability is way larger than the marginal prob-

Table 1: **Prediction results of 40 LSCOM-freq concepts on the TRECVID 2005 development set**

Prediction type	Average # of shots
Hit	785.4
Miss	**1836**
False positive	413.2
Correct reject	33653

Table 2: **Transitional probability between hit and miss of 40 LSCOM-freq concepts on the TRECVID 2005 development set**

Transitional probability	mean (±standard deviation)
$P(Hit_t \mid Hit_{t-1})$	0.654 (±0.112)
$P(Miss_t \mid Hit_{t-1})$	0.346 (±0.112)
$P(Miss_t \mid Miss_{t-1})$	**0.869** (±0.068)
$P(Hit_t \mid Miss_{t-1})$	0.131 (±0.068)

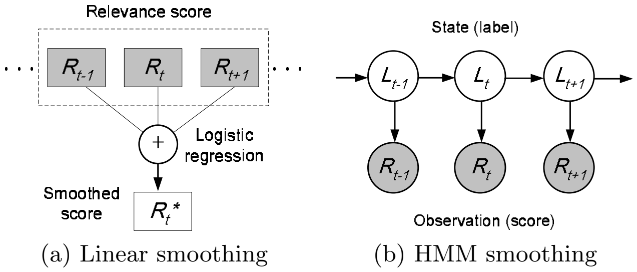
ability, we can significantly improve the prediction on the (relevance) label of a shot by consulting the prediction on the previous shot, and therefore boost the overall performance of concept detection or retrieval. As will be shown in the next section, however, this is not necessarily the case due to the high variance of video data.

4. TEMPORAL CONSISTENCY AND DATA VARIANCE

An intuitive idea of using temporal consistency is to "smooth" the prediction of a shot w.r.t to a concept or query using the predictions of the neighboring shots. However, the effectiveness of this approach is compromised by another common property of video data – high variance. Here, high variance is reflected in the fact that video shots relevant to the same semantic concept or query topic have dissimilar content. This poses a difficulty to the machine learning methods used for classifying this concept/topic, since the relevant shots in the testing set can be very different from those in training set, as exemplified in Figure 1.

High data variance is a general issue in machine learning and exists in various types of data. However, we argue that it is more severe in video data. Because of temporal consistency, there exist not a few individual outliers but sequences of dissimilar relevant shots along the temporal domain in the testing data (e.g., the sequence of baseball shots in Figure 1). As the classifier makes continuous mistakes on the whole sequence of outliers, the performance suffers. Moreover, we can hardly correct such a misclassification by smoothing with the predictions on neighboring shots, since most of the neighboring predictions tend to be wrong. The temporal consistency is of little help if majority of the predictions in the temporal neighborhood are wrong. In the following, we provide evidence on this argument by examining the distribution of concept classification errors, and demonstrate that simple "smoothing" methods can hardly achieve consistent improvement on video concept detection.

Distribution of concept classifier errors. We build SVM classifiers based on 256-d HSV color histogram feature for 40 frequently-used LSCOM concepts (called LSCOM-freq concepts), whose labels are available on the TRECVID 2005 development set. These 40 concepts vary from those related to genre (e.g., sports, entertainment) and scene (e.g. studio, outdoor, waterscape), to those related to object (e.g., airplane, car) and people (e.g., political leader, prisoner). We believe they form a faithful representation of the video concept space.



(a) Linear smoothing (b) HMM smoothing

Figure 4: Smooth the relevance score of a shot by (a) linear combination of neighbors' scores (b) hidden Markov model

To evaluate the classifiers of the 40 concepts, we partition the TRECVID 2005 development data into 3 sets: 40% as a training set, 20% as a held-out set, and the rest 40% as a test set. Each classifier is built from the training set and applied to predict the (binary) label of each shot in the test set. Comparing the predicted label with the true label results in one of the four cases, namely hit (relevant shot labeled as relevant), miss (relevant shot labeled as irrelevant), false positive (irrelevant shot labeled as relevant), and correct reject (irrelevant shots labeled as irrelevant). Among them, miss and false positive are errors. Table 1 summarizes the average frequency of these 4 cases in the classification results of the 40 LSCOM-freq concepts. We see that the number of misses is about four times of the number of the false positives, showing that miss is the primary type of error a concept classifier makes. This is a partial evidence on the high variance of video data.

Further, we look at the temporal distribution of the misses by computing the transitional probability between hits and misses. As shown in Table 2, the average transitional probability of miss is as high as 0.87, indicating that if the classifier just misses a relevant shot, it is very likely to miss the next relevant shot. On the other hand, if the classifier (correctly) finds a relevant shot, it has a fairly good chance (0.65) to find the next one. This shows a pattern of "find all, or miss all", which confirms our conjecture about the continuous misclassifications caused by temporal consistency and data variance.

Temporal Smoothing for concept detection. The idea behind temporal smoothing is to "smooth" the predicted relevance score of a shot w.r.t to a concept using the scores of its adjacent shots, in the hope that it corrects misclassifications using the correct predictions in the adjacency. We implement two smoothing methods as illustrated in Figure 4, both of which try to improve the relevance score R_t of shot t generated by the SVM classifier trained earlier (denoted as baseline classifier). Linear smoothing computes the updated score R_t^* as a weighted combination of the old scores of itself and its neighboring shots in a window (R_{t-1}, R_t, and R_{t+1}), with the weights learned by logistic regres-

Concept	Baseline	Smoothing		
		Oracle	Linear	HMM
Basketball	0.358	0.741	0.360	0.317
Building	0.314	0.455	0.311	0.311
Car	0.307	0.514	0.306	0.305
Commercials	0.830	0.991	0.836	0.836
Computers	0.387	0.458	0.427	0.402
Explosion	0.149	0.318	0.131	0.142
......	...	...	...	...
......	...	...	...	...
Sports	0.603	0.872	0.608	0.614
Still image	0.119	0.341	0.107	0.097
Studio	0.721	0.767	0.752	0.698
Text only	0.662	0.678	0.668	0.636
Urban	0.178	0.434	0.179	0.178
Vegetation	0.310	0.491	0.314	0.302
Waterscape	0.388	0.554	0.374	0.396
Weapons	0.378	0.676	0.380	0.384
Weather	0.196	0.774	0.201	0.213
average (MAP)	**0.399**	**0.567**	**0.403**	**0.397**

Table 3: Average precision (AP) of 40 LSCOM-freq concepts by smoothing methods

sion on the held-out set. Hidden Markov model (HMM) models the concept label of a shot $L_t \in \{0, 1\}$ as hidden state generating its score R_t as the observation, and uses the posterior probability of positive label as the new relevance score, i.e., $R_t^* = P(L_t = 1|\vec{R})$. The parameters of HMM is also trained on the held-out set, and the Viterbi algorithm is used to compute $P(L_t = 1|\vec{R})$.

Table 3 summarizes the average precision (AP) of the 40 concepts on the test set achieved by the two smoothing methods as well as the baseline classifiers. For comparison purpose, we also include an (unrealistic) oracle method as a variant of linear smoothing which takes true labels of the neighboring shots as input. That is, it combines L_{t-1}, R_t, and L_{t+1} to compute R_t^*. This oracle represents the performance upper bound one can possibly reach by using the temporal knowledge. As we can see, the oracle method outperforms the baseline on almost every concept, resulting in a 17% improvement on mean average precision (MAP). This shows that the labels of the neighboring shots are tremendously useful for predicting the label of the current shot. When using the real scores instead of true labels, however, the linear smoothing method does not significantly outperform the baseline, with less than 1% improvement on MAP. Comparing it with the baseline classifiers, we find that linear smoothing helps on some concepts (e.g., Computers, Studio) but hurts on some others (e.g., Explosion). The same situation is observed on results of the HMM smoothing method.

Since the only difference between Oracle and Linear Smoothing is the use of true labels or predicted scores, it is odd to see the large difference of their performance, especially on concepts with high baseline AP (e.g. "Commercials") where the predictions are almost as good as the true labels. This can be explained by the distribution of classification errors. Temporal smoothing improves performance by correcting the misclassification on a shot using the predictions on its neighbors. Its success is therefore based on the assumption that the majority of the predictions in that neighborhood are correct. However, as we observed in Table 2, misclassifications (e.g., misses) tend to appear in a row, which makes temporal smoothing method unable to correct any of the mistakes.

The high variance of video data poses a serious challenge for using temporal consistency for tasks like concept detection in an automatic setting. In the following, we propose two methods that can effectively exploit temporal consistency for better concept detection and retrieval in interactive settings while going around the problem caused by high data variance. To be specific, the idea is to rely on users to spot misclassified relevant shots, and then use heuristics based on temporal consistency to find more relevant shots in the neighborhoods around these "seed" shots.

5. ACTIVE LEARNING WITH TEMPORAL SAMPLING STRATEGY FOR CONCEPT DETECTION

Active learning technique [12] has been used to build high-quality classifiers for semantic video concepts [3, 11]. The idea is to improve the current classifier by asking users to label informative shots and adding the labeled shots into the training set of the classifier. The major difference between conventional relevance feedback and active learning is that the former only selects top-ranked examples for user labeling, while the latter adopts more intelligent *sampling strategies* to choose informative examples from which the classifier can learn the most. A general assumption on the informativeness of examples is that an example is more useful if the classifier's prediction on it is more uncertain. Based on this assumption, active learning methods typically sample examples close to the classification hyperplane. Another general belief is that a relevant example is more useful than an irrelevant one especially when the number of relevant examples is small compared with that of the irrelevant ones.

Based on our discussion in Section 4, the shots most useful for improving a video concept classifier are those from the sequences of relevant shots misclassified by the current classifier. However, we have no idea on what these shots are (otherwise we would be able to build a better classifier from the very beginning) and thus cannot sample from them. Nevertheless, we find that these misclassified relevant shots are more likely to appear close to the classification hyperplane than anywhere else in the feature space. To see this, we examine the distribution of the missed relevant shots at different distances to the classification hyperplane, and find that an average of 31.3% of the missed shots are distributed among the 5% of shots closest to the classification hyperplane of the SVM classifiers used. Therefore, an active learning method that samples data close to the classification hyperplane has a good chance of finding the missed shots and building a better classifier.

Adding only a few missed relevant shots to the training data is not sufficient to largely improve a classifier. Since the relevant shots of a concept are usually rare compared with the irrelevant ones, we hope that the sampled data contain as many relevant shots as possible. Given that relevant shots are likely to appear consecutively due to temporal consistency, a simple way of finding more relevant shots is to choose the shots close the those already labeled as relevant in the previous iterations of active feedbacks. Therefore, our sampling strategy needs to balance between two factors on

Algorithm 1 Temporal sampling strategy

Input: labeled set L, unlabeled set U, set of relevant shots R, the number of samples N;
Output: sample set S;
Functions: $f()$ is the distance to SVM classification hayperplane, $dist()$ is temporal distance between shots, $D()$ is the density factor

```
$T \longleftarrow \varnothing$
For each $s_i \in U$,
    $F(s_i) = \alpha|f(s_i)| + \beta \min_{s_j \in \{R\}} dist(s_i, s_j) + (1-\alpha-\beta)D(s_i)$
While $(|S| < N)$,
    $s* \longleftarrow argmin_{s_i \in U} F(s_i)$
    $S \longleftarrow S \cup \{s*\}$
    $U \longleftarrow U - \{s*\}$
```

each unlabeled shot, namely 1) its distance to the classification hyperplane in the feature space, and 2) its temporal distance to the closest relevant shot. The informativeness score $F(s_i)$ of a shot s_i is computed as a linear combination of these two distance factors:

$$F(s_i) = \alpha|f(s_i)| + (1-\alpha) \min_{s_j \in \{R\}} dist(s_i, s_j) \qquad (3)$$

where $|f(s_i)|$ is the distance function of shot s_i to the classification hyperplane in the feature space, R is the set of relevant shots labeled so far, and $dist(s_i, s_j)$ is the distance between two shots measured by the number of shots in between. Here α is a constant that balances the contribution of the two factors, and its value needs to be determined experimentally. In our active learning method, the shots with the smallest F scores are chosen as samples to be labeled by users and added into the current training data to update the classifier. Unlike the temporal smoothing methods described in Section 4, this method is not as vulnerable to the high data variance problem since its close-to-hyperplane sampling strategy helps discover missed relevant shots, while using temporal consistency heuristics expedites the improvement of the classifier by finding as many relevant shots as possible.

According to Eq.3, once a relevant shot is labeled, this temporal sampling strategy quickly “grows” into the neighborhood around this shot and labels more relevant shots in it. While this strategy is efficient in terms of quickly sampling a large number of relevant shots, it has a potential weakness that its samples concentrate in only a few temporal neighborhoods. When the total number of samples is fixed, this strategy may result in excessive samples from the same neighborhoods, which carry little additional information as they are all similar to each other, and meanwhile leave many other temporal neighborhoods unexplored. To remedy this problem, we introduce a density factor $D(s_i)$ to penalize samples from highly sampled temporal neighborhoods, where $D(s_i)$ is defined as the ratio of labeled shots among all the shots in a window around s_i. The informativeness score of an unlabeled shot s_i is now computed as:

$$F(s_i) = \alpha|f(s_i)| + \beta \min_{s_j \in \{R\}} dist(s_i, s_j) + (1-\alpha-\beta)D(s_i) \quad (4)$$

where α and β are trade-off factors to be determined. The density factor $D(s_i)$ favors shots from less sampled neighborhoods and therefore achieves a wider temporal spread of the samples. The procedure of this sampling strategy is described in Algorithm 1.

6. TEMPORAL RERANKING FOR INTERACTIVE SEARCH

We can improve the performance of interactive video search by exploiting the temporal consistency using a similar idea. The common scenario of interactive video search is as follows: given a query topic, a user sends manually formulated and reformulated queries to a retrieval system, the system returns a list of shots ranked in descending order of their (predicted) relevance score, and the user browses through this list to label relevant shots. The effectiveness of interactive search is measured by the number of relevant shots labeled by the user within a fixed time interval. The methods used for TRECVID interactive search vary in terms of the query type(text query, image query), the number of queries used, and the visualization and interaction techniques. Nevertheless, the basic scenario shared by these methods is that a user goes down a ranking list and labels relevant shots from it. Therefore, the success of interactive search is largely influenced by, among many factors, the quality of the ranking list, or more specifically, the number of relevant shots within a certain depth of the ranking list.

The *initial* ranking list of a query is determined by the retrieval system and its quality is not the focus of this paper. As user goes down the ranking list and starts to label relevant shots, valuable knowledge can be learned from the labels and used to change *on the fly* the ranking of the remaining shots in the list. A well-studied solution is to perform incremental relevance feedback, where one expands the current query with shots that have just been labeled as relevant and rerank the remaining shots according to the expanded query. In the following, we describe a computationally efficient method for improving the ranking list on the fly by pushing the neighbors of the labeled relevant shots to high ranks, which is orthogonal to existing relevance feedback methods. This idea is related to the local browsing/expansion technique used in interactive video search, which allows users to conveniently examine the temporally neighboring shots of any given shot. The major difference is that in our method such expansion is done automatically and based on user's previous labels.

In our method, when a shot is labeled as relevant by user during the labeling process, we push its temporally neighboring shots into high ranks up to the position of the current shot, provided that they have not seen by the user. The assumption behind is that the neighbors of a relevant shot are likely to be relevant too. A straightforward implementation of this idea is to push all the unseen neighboring shots in a temporal window of fixed size around the current shot (which is just labeled relevant) into the next a few slots following the current shot in the ranking list. This reranking method is illustrated in Figure 5, which shows the list before and after the reranking triggered by labeling shot t as relevant. We call this method *fixed-window temporal reranking.* The choice of window size reflects how aggressive the method is in terms of exploiting the temporal consistency of the query topic. Using a large window brings more relevant shots into the front but meanwhile can also brings in more irrelevant ones. Hence, one needs to experiment with multiple window sizes to find the best tradeoff. As an advantage essential for interactive search, this reranking method requires basically no extra computation.

Using the same window size for all the query topics is

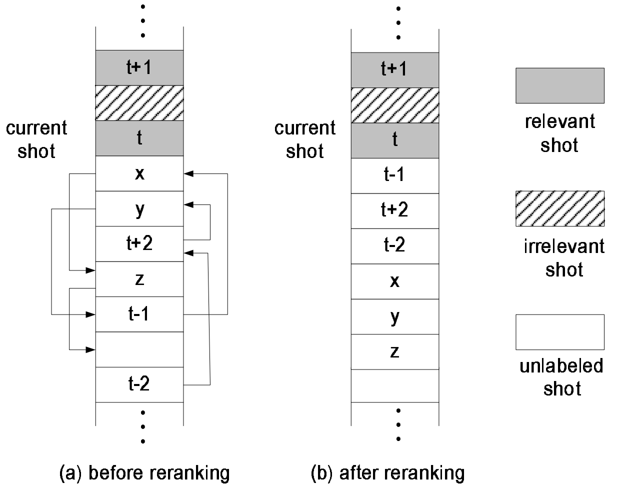


Figure 5: The shot ranking list before and after temporal reranking with window size of 4, triggered when the current shot t is labeled as relevant. The unlabeled temporal neighbors of t, including $t-1$, $t+2$, and $t-2$, are moved forward into the next a few slots under t, ranked by their distance to t.

sub-optimal since it ignores the difference on the strength of temporal consistency of different topics. It is intuitive to use larger windows for topics whose relevant shots exhibit stronger temporal consistency, and vice versa. However, one has no clue as to the strength of temporal consistency of each query topic before the search process, and it is illegitimate in the search paradigm to acquire such knowledge from, say, labeled training data. Nevertheless, we suggest to estimate this information *on the fly* as the user browses the ranking list and labels relevant and irrelevant shots. Following this idea, we propose an *adaptive-window temporal reranking* scheme, in which the window size is dynamically determined by the transitional probability $P(Q_t = 1|Q_{t-1} = 1)$ estimated from the pool of shots that have been labeled so far. We estimate $P(Q_t = 1|Q_{t-1} = 1)$ in the same way suggested in Section 3, except that here we only use the shots that are labeled. The window size is given as

$$window_Q = max\_window \times P(Q_t = 1|Q_{t-1} = 1) \qquad (5)$$

As our estimation of $P(Q_t = 1|Q_{t-1} = 1)$ improves as the user labels more data, the window size used for reranking gradually reflects the strength of temporal consistency of this topic, so it is expected to perform better than the fixed-window method. In terms of efficiency, this method involves slightly more computation for updating the estimate of transitional probability, but is still much faster than necessary for interactive search.

7. EXPERIMENTS

7.1 Active learning with temporal sampling for concept detection

The proposed active learning method is evaluated by experiments on 40 LSCOM-freq concepts and the TRECVID 2005 development set, which contains diversified news video footage from 13 programs in 3 different languages. The whole data set is split into 3 sets in temporal order, including an initial set I of 3050 shots, a test set T of 31686 shots, and a held-out set H of the remaining 39787 shots. These three sets have no overlap on temporal domain. For each of the 40 concept, we manually label the relevant shots in set I and train a SVM classifier with RBF kernel for this concept based on 256-d HSV color histogram. The classifier is then applied to predict the relevance scores of the shots in set T and H, and we perform active learning to improve this classifier by choosing unlabeled shots from T for user feedback (The feedback is simulated using computer programs.) To be close to the real case, only 100 shots are chosen for feedback in each iteration, and only 8 feedback iterations are performed. We test three sampling strategies of active learning, namely the conventional strategy **AL** [3] which samples unlabeled shots closest to classification hyperplane, and the proposed temporal sampling strategy **TempAL1** (Eq.3) and its variant **TempAL2** (Eq.4) with the additional density factor. We also experiment with different tradeoff factor α and β, and choose the ones giving the best performance.

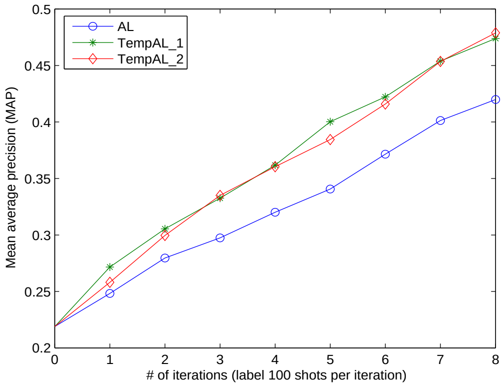


Figure 6: MAP of 40 LSCOM-freq concepts on test set T in 8 feedback iterations. The methods compared are conventional active learning (AL), and active learning with two temporal sampling strategies (TempAL1 and TempAL2).

We first examine the mean average precision (MAP) of the 40 concepts on set T during the 8 feedback iterations. It is still an open question whether the shots that have been labeled in feedbacks should be included for evaluation. Here we decide to include them (and thus use the whole T set for evaluation) for two reasons. First, excluding the labeled shots from T result in *different* test set for different methods, because they may choose different shots for feedback. Second, even after 8 iterations only 800 shots are labeled, which is a very small portion of the 31686 shots in T. The performance on T is a meaningful metric as it shows how much we can improve an existing concept classifier on a large set with limited user intervention.

As shown in Figure 6, active learning methods based on the two proposed temporal sampling strategies outperforms that based on conventional method by large margins. The improvements are consistent across the iterations, which means that we can benefit from the temporal sampling strategy no matter how many iterations are performed. It is a

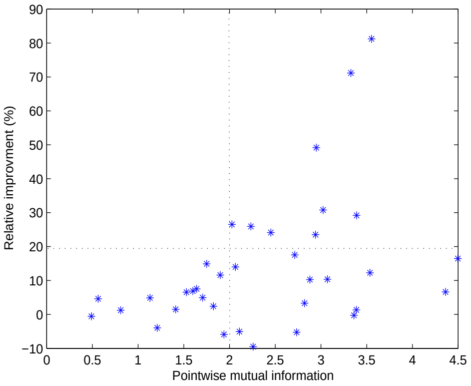


Figure 7: The relative improvement of TempAL2 over AL on the 40 LSCOM-freq concepts after 8 iterations against their pointwise mutual information.

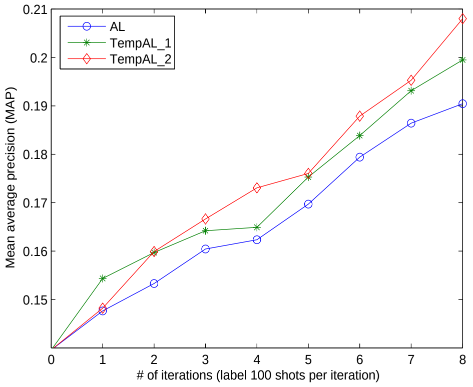


Figure 8: The MAP of the three methods on 40 LSCOM-freq concepts evaluated on held-out set H in 8 feedback iterations.

bit surprising to see that **TempAL2** does not perform significantly better than **TempAL1**. A possible explanation is that while **TempAL2** achieves a wider temporal spread of labeled shots, it may not label as many relevant shots as **TempAL1** does.

It is also interesting to explore the relationship between the strength of temporal consistency of each concept and the improvement achieved by temporal sampling. For each concept, we compute the relative improvement (in percentage) of **TempAL2** over **AL** on the final AP achieved after 8 feedback iterations, and plot it in Figure 7 against the pointwise mutual information of this concept. From their distributions, we see that all the concepts on which **TempAL2** produces an 20+% improvement satisfy $PMI > 2$, and for the concepts with $PMI < 2$ using temporal strategy is not very helpful. This is intuitive because temporal sampling strategy is expected to be more effective for concepts with stronger temporal consistency. However, the dots in the bottom-right corner of Figure 7 shows that it is unable to significantly improve the performance of every concept with high PMI. Viable explanations of this include the complexity of the concept and the limitation of color histogram feature.

A more challenging evaluation metric is to examine the performance on a unpolluted held-out set H, which shows how much the improvement on T can be generalized to the future data. Figure 8 has the performance comparison between the three methods on H. We see that while the two temporal sampling methods still outperform the conventional method, the gap is not as impressive. Here **TempAL2** is clearly better than **TempAL1**, which can be contributed to the wide temporal spread of its samples due to the introduction of the density factor.

7.2 Temporal reranking for interactive search

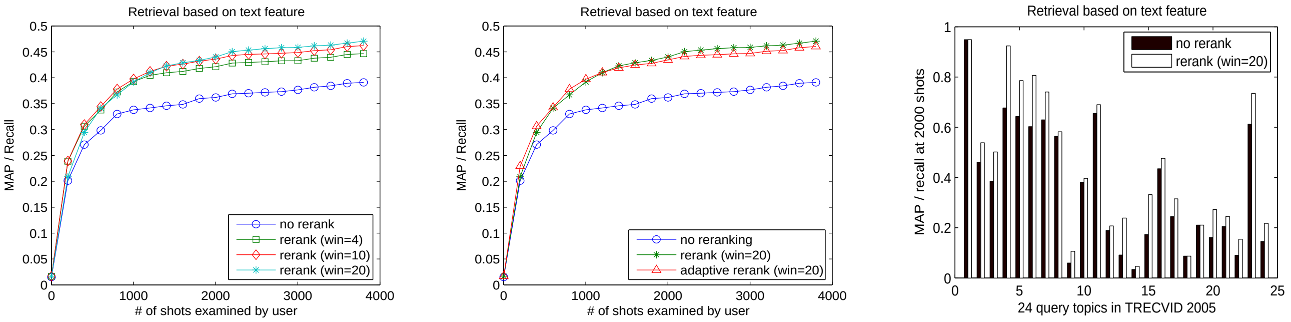
The temporal reranking method is evaluated by how much it improves a ranking list of relevant shots for a query topic. We follow the paradigm of TRECVID interactive search where a user is given a fixed time interval (15 minute) for each topic to find as many relevant shots as possible. For simplicity, we make three assumptions in our experiment. First, we assume that a user issues only one query for each topic, and spends all the time examining, in a top-down manner, the ranking list of shots retrieved by a retrieval system and labeling the relevant shots in it. Even if in reality a user may try multiple queries for each topic, he/she merely repeats the above process for many times, so the improvement within one iteration carries over to the whole process. Second, all users are assumed to be equally efficient, meaning that everyone examines the same number of shots in the fixed time interval. Finally, we assume users are error-free, i.e., a user makes no mistake as to the judgment on the relevance of a shot.

With these assumptions, for each topic the number of relevant shots a user labels is equal to the number of relevant shots among top N shots in the ranking list, where N is the number of shots the user is able to examine within the time interval. We evaluate the quality of the ranking list by the recall of the top N shots, defined as $Rec_N = N_{rel}/M$, where N_{rel} is the number of relevant shots within the top N shots, and M is the total number of relevant shots in the collection[1]. Obviously, whether we rerank the list or not, Rec_N improves as N gets larger, but we expect that with temporal reranking method it increases at a faster rate than it does without reranking.

Our experiment is conducted on the TRECVID 2005 collection using the 24 benchmark query topics. We generate the ranking list of candidate shots using two strategies, one using only text retrieval (text-only run), and the other based on the combination of multiple features including keywords and image color histogram (multi-feature run) using the method described in [15]. For each query topic, we trace the recall (or AP) up to the top 4000 shots in the list, which is the about the maximum number of shots a user can possibly see in the given 15 minute.

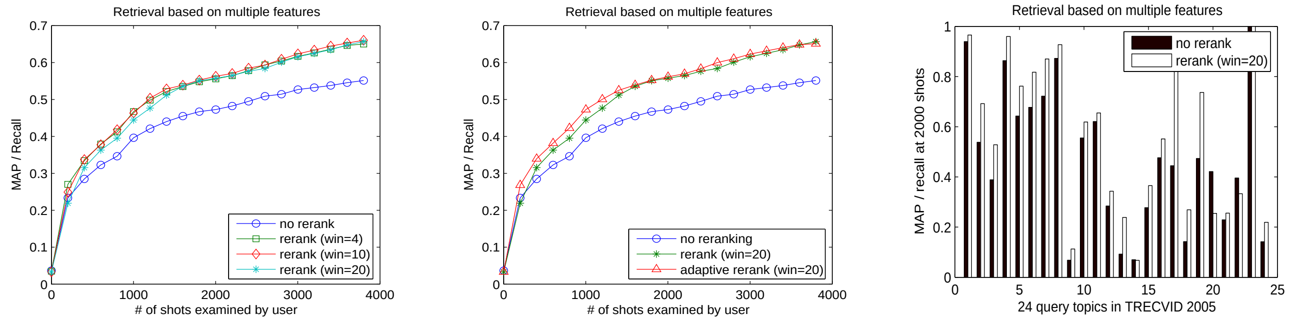
Figure 9(a) shows the performance comparison in the text-only run using fixed-window temporal reranking with window size of 4, 10, and 20, as well as the baseline method without reranking. We see that using the right window size, the reranking method improves the baseline by about 7% at

[1] Although TRECVID uses average precision (AP) of the user-sorted ranking list as evaluation metric, in our case the recall approximates AP assuming that the user puts all the relevant shots found at the top of the final list.



(a) Fixed-window reranking (b) Adaptive-window reranking (c) Comparison on each query

Figure 9: Temporal reranking on 24 TRECVID 2005 query topics in text-only run



(a) Fixed-window reranking (b) Adaptive-window reranking (c) Comparison on each query

Figure 10: Temporal reranking on 24 TRECVID 2005 query topics in multi-feature run

top 2000 shots and 8% at top 4000 shots. The optimal window size depends on the number of shots the user is able to examine. The **win20** outperforms **win10** after 2000 shots, but it is slightly worse than **win10** within the first 1000 shots. Therefore, a quick user who can see over 2000 shots in 15 mins would prefer **win20**, and vice versa. In Figure 9(b), we compare the performance of fixed-window reranking and adaptive-window reranking, both using (maximal) window size of 20. The adaptive method is slightly better within the top 1500 shots, after which its performance starts to go below that of the fixed-window method. To show that the improvement of temporal reranking is consistent across different query topics, Figure 9(c) shows on a per query basis the recall of the baseline and of the reranking method. We find that reranking method either outperforms the baseline (20+% relative improvement in 14 out of 24 topics) or performs as well as the baseline, but never hurts the performance of any single query topic.

Figure 10 shows the comparison on multi-feature run, where the baseline is higher than the baseline in text-only run due to the integration of multimodal features. Despite a higher baseline, the temporal reranking method generates in a even larger improvement of about 12% at top 4000 shots when using window of size 10. This means that our method is orthogonal to the multimodal features used for generating the ranking list, and the improvement is not sensitive to the baseline. Similarly, the adaptive reranking is slightly better than its fixed-window counterpart, especially within the top 2000 shots in the list. All the results show that temporal reranking is an efficient approach for significantly improving the performance of interactive video search.

8. CONCLUSION AND DISCUSSION

This paper has presented a thorough study on the temporal consistency of video data and discussed its impacts to semantic video concept detection and retrieval tasks. It has proposed a temporal sampling strategy for active learning method used to iteratively build classifiers of video semantic concepts, and a temporal reranking method for interactive video search. Extensive experiments on the TRECVID collection have shown considerable improvements of the proposed methods over the existing approaches.

This work is a rather preliminary exploration on the video temporal consistency issue in terms of both data variety and approach. The presented observations are mainly based on broadcast news video, especially the TRECVID dataset, due to its availability and popularity. Studying the temporal consistency in non-news video, and comparing the observations with those of news video will be interesting future research. While there have been works on using the temporal information in home video [11] and surveillance video [5], a thorough study of the issue has not been seen on these video genres. Besides the data issue, a similar question can be raised on whether the 40 LSCOM-freq concepts and the 96 TRECVID queries used in our analysis, despite the fact that they are part of a benchmark dataset, constitute a faithful representation of the real video concept/topic space. Analysis based on different concept and/or query sets is desirable to show whether our finding is generalizable.

On the other hand, our analysis focuses on the consistency of *adjacent* shots rather than the shots in a *neighborhood*. This is a limitation if the consistency of video data is beyond the adjacent shots. For example, a typical news footage on an interview consists of shots alternating between two subjects (an interviewer and an interviewee), in which case our method finds little temporal consistency as the shots containing the same person are not consecutive. Moreover, our method does not consider the length of consecutive shots that are relevant to the same concept/query. Another type of temporal information not addressed by our method is the exclusive relationship between the relevant shots of a concept or query. As an example, anchor shots rarely appear consecutively; instead, a shot being an anchor shot usually indicates that the previous and next few shots are NOT anchor shots. Both the long-term consistency and the exclusive relationship deserve future research and may contribute to further improvement on video analysis and retrieval tasks.

9. REFERENCES

[1] LSCOM lexicon definitions and annotations version 1.0. In *DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia, Columbia University ADVENT Technical Report 217-2006-3*, 2006.

[2] S. Chang, W. Chen, H. Horace, H. Sundaram, and D. Zhong. A fully automated content based video search engine supporting spatio-temporal queries. *IEEE Trans. on Circuit System and Video Technology*, 8(5):602–615, 1998.

[3] M. Chen, M. Christel, A. Hauptmann, and H. Wactlar. Putting active learning into multimedia applications: dynamic definition and refinement of concept classifiers. In *Proc. of the 13th ACM Int'l Conf. on Multimedia*, pages 902–911, New York, NY, USA, 2005. ACM Press.

[4] S. Ebadollahi, L. Xie, S.-F. Chang, and J. Smith. Visual event detection using multi-dimensional concept dynamics. In *Proc. IEEE Int'l Conf. on Multimedia and Expo (ICME 2006)*, 2006.

[5] R. Khalaf and S. S. Intille. Improving multiple people tracking using temporal consistency. In *MIT Dept. of Architecture House N Project Technical Report*, 2001.

[6] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *SPIE Conf. on Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 290–301, 1999.

[7] M. R. Naphade, T. Kristjansson, B. Frey, and T. Huang. Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems. In *Proc. of ICIP*, 1998.

[8] A. P. Natsev, M. R. Naphade, and J. Tesic. Learning the semantics of multimedia queries and concepts from a small number of examples. In *Proc. of the 13th ACM Int'l Conf. on Multimedia*, pages 598–607, New York, NY, USA, 2005. ACM Press.

[9] Y. Rui, T. S. Huang, and S. Mehrotra. Constructing table-of-content for videos. *Multimedia Syst.*, 7(5):359–368, 1999.

[10] A. Smeaton and P. Over. Trecvid: Benchmarking the effectiveness of infomration retrieval tasks on digital video. In *Proc. of the Intl. Conf. on Image and Video Retrieval*, 2003.

[11] X. Song, C.-Y. Lin, and M.-T. Sun. Autonomous visual model building based on image crawling through internet search engines. In *Int'l Workshop on Multimedia Information Retrieval*, pages 315–322. ACM Press, 2004.

[12] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proc. of the 9th ACM Int'l Conf. on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

[13] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Structure analysis of soccer video with hidden markov models. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, Orlando, FL, May 2002.

[14] L. Xie, L. Kennedy, S.-F. Chang, A. Divakaran, H. Sun, and C.-Y. Lin. Layered dynamic mixture model for pattern discovery in asynchronous multi-modal streams. In *Int'l Conf. on Acoustic, Speech and Signal Processing*, Philadelphia, PA, March 2005.

[15] R. Yan, J. Yang, and A. G. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *Proc. of the 12th ACM Int'l Conf. on Multimedia*, pages 548–555. ACM Press, 2004.

[16] H. Zhang, S. Y. Tan, S. W. Smoliar, and G. Yihong. Automatic parsing and indexing of news video. *Multimedia Syst.*, 2(6):256–266, 1995.