Bounded Coordinate System Indexing for Real-time Video Clip Search

ZI HUANG, HENG TAO SHEN, JIE SHAO, and XIAOFANG ZHOU
The University of Queensland
and
BIN CUI
Peking University

Recently, video clips have become very popular online. Massive influx of video clips has created an urgent need for video search engines to facilitate retrieving relevant clips. Different from traditional long videos, a video clip is a short video often expressing a moment of significance. Due to high complexity of video data, efficient video clip search from large databases turns out to be very challenging. We propose a novel video clip representation model called Bounded Coordinate System (BCS), which is the first single representative capturing the dominating content and content changing trends of a video clip. It summarizes a video clip by a coordinate system, where each of its coordinate axes is identified by Principal Component Analysis (PCA) and bounded by the range of data projections along the axis. The similarity measure of BCS considers the operations of translation, rotation and scaling for coordinate system matching. Particularly, rotation and scaling reflect the difference of content tendencies. Compared with the quadratic time complexity of existing methods, the time complexity of measuring BCS similarity is linear. The compact video representation together with its linear similarity measure makes real-time search from video clip collections feasible. To further improve the retrieval efficiency for large video databases, a two-dimensional transformation method called Bi-Distance Transformation (BDT) is introduced to utilize a pair of optimal reference points with respect to bi-directional axes in BCS. Our extensive performance study on a large database of more than 30,000 video clips demonstrates that BCS achieves very high search accuracy according to human judgement. This indicates that content tendencies are important in determining the meanings of video clips and confirms that BCS can capture the inherent moment of video clip to some extent that better resembles human perception. In addition, BDT outperforms existing indexing methods greatly. Integration of BCS model and BDT indexing can achieve real-time search from large video clip databases.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*abstracting methods*; *indexing methods*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*; *search process*

General Terms: Design, Algorithms, Experimentation

Additional Key Words and Phrases: Video Search, summarization, indexing, query processing

An extended abstract appeared in ACM Multimedia 2007 as a demo paper [Shen et al. 2007].
Authors' addresses: Z. Huang, H. T. Shen, J. Shao and X. Zhou: School of Information Technology and Electrical Engineering, The University of Queensland, Australia; emails: {huang, shenht, jshao, zxf}@itee.uq.edu.au; B. Cui: Department of Computer Science, Peking University, China; email: bin.cui@pku.edu.cn.

1. INTRODUCTION

Nowadays lots of personal and organizational video data are accumulating with ever more usage of video devices and advances in video processing technologies. Meanwhile, online media and streaming videos are growing aggressively. As mass Internet audiences are emerging rapidly and technologies become more video-friendly, such as storage cost plummeting, CPU speed continuing to double, bandwidth quickly increasing and web continuing to propel scalable infrastructure for media communities, we are facing mass videos in our century.

In particular, with the rapid spread of broadband Internet access, video clips become very popular online. Different from traditional long videos such as TV programs and full movies, video clips are *short clips in video format* (mostly less than 10 minutes), and "predominantly found on the Internet where the massive influx of video clips has been dubbed as a new phenomenon which has a profound impact on both the Internet and other forms of media"[1]. Sources of video clips are various, typically including music videos, TV commercials, news and sporting events, movie trailers, video blogs, etc. Traditional long videos can also be segmented into clips, each of which may represent a scene or story. It is reported that as of mid 2006, tens of millions of video clips are available on the Internet. According to a July 16, 2006 announcement by YouTube, more than 100 million clips are viewed daily on YouTube, with an additional 65,000 new video clips uploaded per day.

Clearly, the increasing generation and dissemination of video clips have created an urgent need for video search engines to facilitate retrieving relevant clips. For example, since videos are easy to copy, reformat, modify and republish, duplicate or near-duplicate videos often spread in the absence of central management or monitoring, thus there are many almost-identical videos on the Internet. An important problem now faced by YouTube is how to perform real-time video clip search for a new video clip against its huge collection to avoid copyright violation, or perform database purge. Since the retrieval efficiency will be hampered if a large number of search results are essentially duplicates or near-duplicates of one another, to realize a better user experience for search and browsing, eliminating those redundant videos is desired before presenting them to users [Wu et al. 2007]. The emergence of video search engines on major web search portals and market forces such as IPTV and mobile video service deployments are enabling many new application domains for video clip search [Gibbon 2005]. Fuelled by strong market trends and motivated by great successes in conventional web search, video search becomes a key part of the future of digital media [Sarukkai 2005].

In this paper, we regard video as a sequence of frames, each of which is typically represented by a high-dimensional feature vector, such as color feature. The number of frames depends on the video length and frame rate. Given a large collection of video clips, effective management is needed to search for user interests. To do so, two essential issues have to be addressed: obtaining a compact video representation with an effective similarity measure, and organizing the compact representations with an indexing method for efficient search.

Due to tremendous volume of video data, effective summarization techniques

[1]Quoted from Wikipedia

that abstract videos into compact representations usually have to be applied before deploying any indexing structure. The main idea is to represent each video with a small number of representatives, and then estimate video similarity based on the integration of representative similarities [Cheung and Zakhor 2003; 2005; Shen et al. 2005]. Unfortunately, from the database research perspective, these methods suffer from the following drawbacks. First, before an overall video similarity can be aggregated from some representatives, each representative has to perform an individual search (range or kNN search) with an access method to retrieval its similar counterparts from database. Therefore, multiple representatives for a single query video require multiple accesses to the underlying indexing structure. Second, the time complexities of their similarity measures are quadratic in the number of representatives. Both drawbacks potentially lead to inefficiency of video search. This explains why existing content-based video search engines are usually tested with a relatively small number of videos only.

Interestingly, an important characteristic of video clips is that a clip often has a central theme that shows "moment of significance, humor, oddity, or prodigy performance". This leads to the intuition that a video clip is likely to have the dominating visual content and corresponding content changing trends to express the moment. Inspired by this, we propose a novel video clip representation model called Bounded Coordinate System (BCS), which is the first single video representative that captures the dominating content and content changing trends by exploiting frame content distribution. It summarizes a video clip by a coordinate system, where each of its coordinate axes is identified by Principal Component Analysis (PCA) and bounded by the range of data projections along the axis. That is, BCS is a coordinate system with bounded axes in feature vector space[2]. It describes a video clip by a system origin and some bounded coordinate axes. BCS is a single video representative with small size (only linear in the dimensionality of feature space). Consequently, the complexity of video data is reduced dramatically. With PCA, the most significant orientations of frame content distribution can be identified. As we can see later, in real-life video clip datasets, the energy (i.e., the information) carried by a large number of bottom ranked Principal Components are very close to zero. They can be discarded without losing much information, since the major information on those Principal Components still can be retained in the origin. A few top ranked Bounded Principal Components capturing the major orientations and ranges of video content together with the origin are sufficient for accurate search.

To measure the similarity between two BCSs, we consider the operations including translation, rotation and scaling for coordinate system matching. The similarity measure of BCS integrates two distances: the distance between two origins by translation, and the distance between each pair of bounded axes by rotation and scaling. The first distance indicates the global difference between two sets of frames representing the video clips, while the second distance indicates the difference of all the

[2]Note that the problem we addressed here is different from finding relevant subsequences from long videos. Video clip search deals with video databases of short clips. BCS is particularly devised for video clip. It does not lose generality since long videos can be easily segmented into clips as needed.

corresponding bounded axes which reflect the content changing trends and ranges. Compared with the quadratic (in the number of representatives) time complexity of existing methods for measuring video similarity, the time complexity of measuring BCS similarity is only linear in the dimensionality of feature space, which is small compared with the video length (i.e., the number of representatives). The significant reduction of time complexity makes real-time search from large video clip collections feasible. Our experiments with tens of thousands of video clips demonstrate fast response of BCS in milliseconds, in contrast to slow response of some previously proposed methods in seconds or minutes.

To further improve the retrieval efficiency for large video databases, effective indexing on BCSs is desired. We introduce a two-dimensional transformation method called Bi-Distance Transformation (BDT), which utilizes the power of a pair of optimal reference points with respect to bi-directional axes in BCS. BDT transforms each data point into two distance values with respect to two optimal reference points, and further extends the standard B^+-tree to index two distance values. Finally, how BDT is applied for BCS indexing is introduced. An extensive performance study on a large database of more than 30,000 video clips (more than 500 hours in total) demonstrates very high search accuracy of BCS model in both RGB and HSV feature spaces. It achieves prominent improvements on some previously proposed methods. This reveals that our BCS model exploiting the distribution of frame content features can better capture the inherent moment of video clip that matches human perception. Content tendencies are very crucial in determining the relevance of video clips. In addition, the proposed BDT indexing can outperform one-dimensional transformation greatly in terms of search speed.

The rest of this paper is organized as follows. In Section 2, we review the related research efforts. Bounded Coordinate System is presented in Section 3, followed by its similarity measure in Section 4. We introduce Bi-Distance Transformation in Section 5. An extensive performance study is reported in Section 6, and finally, we conclude in Section 7.

2. RELATED WORK

2.1 Video Summarization

There are a number of video summarization techniques to abstract videos into compact representations. Video similarity then can be estimated based on these summaries. Key-frame is a popular abstraction for videos, which is normally extracted following shot segmentation [Liu et al. 1999; Wu et al. 2000; Peng and Ngo 2006; Ho et al. 2006]. In [Chang et al. 1999], a set of key-frames can be identified by selecting the k feature vectors to minimize the semi-Hausdorff distance between them and the original feature vectors of entire segment. In [Ferman and Tekalp 2003], a two-stage framework is proposed to generate hierarchical key-frame summaries of video sequences by clustering. At the first stage, fuzzy clustering and data pruning methods are applied to obtain a non-redundant set of key-frames that comprise the finest level of hierarchical summary. The second stage computes a coarser summary by reducing the number of key-frames to match the desired number of key-frames for each shot. Key-frames can also be selected based on the property of video sequence curve such as sharp corners or transformations us-

ing multi-dimensional curve splitting algorithms [DeMenthon et al. 1998]. In [Zhu et al. 2004], a hierarchical video summarization strategy that exploits video content structure is proposed. A hierarchical structure with increasing granularity is constructed from the clustered scenes, video scenes, and video groups to key-frames. In [Dadason et al. 2007], frames are selected at a uniform sampling rate and each sampled frame is used to compute up to several hundreds of image local descriptors for matching. The video search results are aggregated first from frame level and then video level. Unfortunately, a limitation associated with key-frame based and sub-sampling based representations is that they usually ignore the information of temporal evolution embedded in contiguous video frames.

Videos can also be represented by a small number of representatives. In [Iyengar and Lippman 2000], video frames are clustered based on probability distributions by assuming that frames are distributed in a model like Gaussian, or mixture of Gaussian. Each frame is assigned to a cluster with a probability and each cluster is then represented by a higher level probability descriptor. A video is finally represented by a set of probability descriptors. To estimate the percentage of visually similar frames, in [Cheung and Zakhor 2003; 2005] a randomized algorithm is proposed to select a number of seed frames from a clustered training set that is representative of the ensemble of target video data under consideration, and assigns a small collection of closest frames named Video Signatures (ViSig) to the set of seed frames. The ViSigs used in [Cheung and Zakhor 2003] are several frame feature vectors whose total dimensionality is up to 712, which is too big to store and meet the need of fast search. A low dimensional feature extraction mapping [Cheung and Zakhor 2005] is needed for fast search. However, depending on the relative positions of the seed frames and ViSigs, this randomized algorithm may sample non-similar frames from two almost-identical videos. In [Shen et al. 2005], each video is summarized into a set of clusters, each of which is modelled as a hyper-sphere named Video Triplet (ViTri) described by its position, radius, and density. Each video is then represented by a much smaller number of hyper-spheres. Video similarity is approximated by the total volume of intersections between two hyper-spheres multiplying the smaller density of clusters. It also aims at deriving the overall video similarity by the total number of similar frames shared by two videos. However, for most above methods, there are many parameters which are hard to be tuned. A small change in a parameter value often ultimately affects performance greatly.

Recently, in [Hoad and Zobel 2006] some new techniques of producing video signatures including the shot length, color shift, centroid motion, and combined methods have been proposed for co-derivative video search. The shot length signature is based on the pattern of editings in video sequence. There are a number of transition effects that can be applied at shot boundary, among which the most common one is known as hard cut. While there are many efforts for detecting more complex transitions such as fades and dissolves, by far standard cut detection algorithm is fairly robust [Lienhart 1999]. The number of frames between two adjacent cuts is recorded as the length of each shot. In this way, each video is represented by a sequence of integers. It is assumed to be insensitive to changes in bitrate and resolution. The color shift signature captures the change of color in frames over time by a single integer. The centroid motion signature represents the

spatial movements of luminance centroids in frames, capturing the motion in video using a motion estimation algorithm. The combined signature uses evidence from both the color shift and centroid motion signatures to produce a combination that shares many of the strengths of constituent signatures. These pertinent properties are expected to be preserved even when the video is substantially altered. Correspondingly, some associated scoring functions are defined, in an ad hoc fashion, for video signature matching. The last three signatures introduced in [Hoad and Zobel 2006] consider the local context of neighboring frames, thus are probably more robust than conventional feature comparison methods [Hampapur et al. 2002; Mohan 1998; Kim and Vasudev 2005; Naphade et al. 2000; Kashino et al. 2003] that compare the frames directly between two videos (with an assumption that similar videos have the same length and temporal order). However, they contain one symbol for each video frame, thus might be less efficient, as those methods based on frame-by-frame comparison of visual features which are usually incapable of fast search in large video databases. The shot length signature summarizes a video clip globally rather than focusing on its sequential details, which is conceptually somewhat similar to our proposal. In the experiments, we compare this signature with BCS.

2.2 Video Similarity Measure

In content-based image retrieval, the similarity/distance of images can be computed by the well-known Minkowski-form distance L_p [Rubner et al. 2001], e.g., the sum of squared differences (Euclidean distance L_2) is justified when the underlying data have a normal distribution, while the sum of absolute differences (Manhattan distance L_1) is justified when the underlying data have an exponential distribution [Sebe et al. 2000], although there are indications that more complex similarity measures might be necessary [Santini and Jain 1999; Smeulders et al. 2000; Law-To et al. 2007]. Many proposals have been investigated to extend the distance function for measuring video sequences. When temporal information is considered, various time series similarity measures that handle temporal order, frame alignment, gap and noise can be applied. Mean distance [Lee et al. 2000] extends the traditional similarity search methods for time series to support multi-dimensional data sequences. However, it adheres to temporal order in a rigid manner and does not allow frame alignment or gap. Meanwhile, since it entirely relies on the Euclidean distance of frame pairwise similarity, it is sensitive to noise frames. Other distance functions, such as Dynamic Time Warping (DTW) [Keogh 2002], Longest Common Subsequence (LCSS) [Vlachos et al. 2002], and Edit distance (e.g., Edit Distance on Real sequence (EDR) [Chen et al. 2005]) can also be extended for measuring multi-dimensional time series. Just to mention a few, for measuring video similarity, DTW is adopted in [Chiu et al. 2006], LCSS is adopted in [Chen and Chua 2001], and Edit distance is adopted in [Adjeroh et al. 1999; Zhou and Zhang 2005; Bertini et al. 2006], etc.

Unfortunately, all these measures need to compare most, if not all, frames pairwise. The time complexities of similarity computations are quadratic in video length (i.e., the number of frames). Given that the video length is large (typically hundreds of frames or more), obviously, such expensive computations are strongly prohibitive for large video databases. Therefore, measuring video similarity based

on compact representations becomes a more practical solution. A widely used video similarity measure is the percentage of similar frames shared by two videos [Cheung and Zakhor 2003; 2005; Shen et al. 2005] which does not consider the temporal order of frames. The naive way is to find similar frames in one video for each frame in the other video. The time complexity is also quadratic in video length. The similarity of videos can be approximated by a smaller number of representatives such as ViSigs [Cheung and Zakhor 2003; 2005] and ViTris [Shen et al. 2005]. In this way, the time complexity can be reduced to be quadratic in the number of representatives.

2.3 High-dimensional Indexing

High-dimensional indexing is also related to our research. It has been extensively studied in the database community [Böhm et al. 2001]. Although tree-based index structures work well in low to medium dimensional spaces (up to 20-30 dimensions), a simple sequential scan usually performs better at higher dimensionality [Weber et al. 1998]. To tackle the notorious "curse of dimensionality", substantial progresses have been made in the past decade, which can be generally classified into five approaches: tree-like structure such as X-tree [Berchtold et al. 1996], data compression such as VA-file [Weber et al. 1998] and VQ-index [Tuncel et al. 2002], dimensionality reduction and hybrid of tree-like structure such as LDR [Chakrabarti and Mehrotra 2000] and MKL-tree [Franco et al. 2007], transformation to one-dimension such as Pyramid-technique [Berchtold et al. 1998] and iDistance [Jagadish et al. 2005] and approximate search such as LSH [Gionis et al. 1999] and SASH [Houle and Sakuma 2005]. Some indexing methods specially dedicated to video object indexing have also been investigated [Chen et al. 2004; Lee et al. 2005]. Our work is more closely related to the transformation-based category. iDistance [Jagadish et al. 2005] transforms each high-dimensional point to a one-dimensional distance value with respect to its corresponding reference point. The obtained one-dimensional distance values are then indexed by a standard B^+-tree. An optimal one-dimensional transformation method is introduced in [Shen et al. 2005] based on Principal Component Analysis (PCA) on the input dataset. More details will be reviewed in Section 5.

After each video is summarized into a small number of representatives, an indexing structure can be built on those representatives. Similar videos are then returned by searching the index. However, there is a problem with the state-of-the-art methods [Cheung and Zakhor 2003; 2005; Shen et al. 2005]. Given a query video, each of its representatives has to search the index to find the similar representatives before an overall video similarity can be aggregated. In other words, for a single query, the underlying indexing structure has to be accessed the same times as the number of representatives in the query video. Moreover, since the time complexity of similarity measure is quadratic in the number of representatives, multi-representative model less appealing for large video databases.

We propose a novel video clip representation model which captures the dominating content and content changing trends to summarize video clip. Interestingly, the time complexity of its similarity measure is only linear in the dimensionality of feature space.

3. BOUNDED COORDINATE SYSTEM

In this section, we introduce Bounded Coordinate System (BCS), a statistical model which exploits the distribution of frame content features for compact video representation.

3.1 BCS Representation

Given the understanding that a video clip often expresses a moment of significance, from the human perception point of view, different dominating content of video clips may show different significance. Meanwhile, different content changing trends may also suggest different meanings. For example, a major content change from light color to black in a movie trailer may express horror, mourning, depression, etc. On the other hand, a change from light color to red may suggest excitement, passion, etc. Intuitively, a video representation that can capture not only the dominating visual content but also corresponding content changing trends potentially improves the quality of content-based video search. While most existing work [Ferman and Tekalp 2003; Zhu et al. 2004; Cheung and Zakhor 2003; 2005; Shen et al. 2005] focus on summarizing video content, to the best of our knowledge, the information of content tendencies has not been utilized in video search. In this paper, we propose a novel model called Bounded Coordinate System (BCS) to capture the dominating content and content changing trends of a video clip by extending Principal Component Analysis (PCA). We begin with the illustration of PCA.

PCA [Jolliffe 2002] is a linear transformation that projects data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate axis (called the first Principal Component, or PC), the second greatest variance on the second coordinate axis, and so on. Each PC is associated with an eigenvalue, which is a measure of the variance in the PC. The first PC is the eigenvector corresponding to the largest eigenvalue of the dataset's covariance matrix $\mathbf{C}$, the second PC corresponds to the eigenvector with the second largest eigenvalue, and so on. All Principal Components are orthogonal to each other and can be ranked based on their eigenvalues. PCA amounts to a "rotation" of the coordinate axes to identify the Principal Components such that a more "natural" coordinate system of the input dataset can be found.

PCA has two important properties [Jolliffe 2002]: it maximizes the variances of projections on the Principal Components and minimizes the "least-squares" (Euclidean) reconstruction error between the original data and their projections on the Principal Components. A 2-dimensional example of PCA is shown in Figure 1, where the first PC indicates the direction that exhibits a larger variance, and the second PC orthogonal to the first one indicates the direction with a smaller variance.

Based on the first property, the direction along the first Principal Component exhibits the strongest changing trend of the dataset, the direction along the second Principal Component exhibits the second strongest changing trend, and so on. By analyzing Principal Components, the most significant orientations of the dataset can be understood. Meanwhile, from the second property, the projection on the first Principal Component carries the most information of the data, and so on. If the input dataset is skewed (which is very common for real-life feature data), the major information of the dataset can be captured by a few top ranked Principal

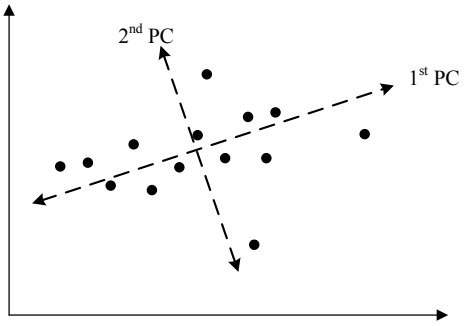


Fig. 1. Principal Components of a sample dataset in 2-dimensional space.

Components and discarding bottom ranked Principal Components leads to no much information loss.

It has to be noted that a Principal Component only provides directional information, i.e., a line (or direction) of the tendency. Next, we present a useful definition called Bounded Principal Component (BPC) for incorporating range information.

Definition 1 Bounded Principal Component. *For a Principal Component, denoted as Φ_i which identifies a line (or direction), its Bounded Principal Component, denoted as $\ddot{\Phi}_i$, identifies a segment of the line bounded by two furthermost projections on Φ_i for all data points. The length of Bounded Principal Component, denoted as $||\ddot{\Phi}_i||$, is the length of the segment.*

In the example shown in Figure 2, on the line identified by the first Principal Component Φ_1, two furthermost projections (signified with circles on the line) determine a Bounded Principal Component by bounding the first PC. Similarly, a Bounded Principal Component for the second PC Φ_2 is also identified. Clearly, a Bounded Principal Component is determined by its Principal Component and two furthermost projections. The distance between two furthermost projections is the length of Bounded Principal Component. Therefore, given a coordinate system identified by PCA, Bounded Principal Components carry the information of both direction and length, i.e., they can indicate the major orientations and ranges of data distribution.

Given a dataset, it is not uncommon that there exist some abnormal points (or noises). Note that the length of a Bounded Principal Component determined by two furthermost projections along the Principal Component is very sensitive to noises. Figure 2 shows such a scenario, where $||\ddot{\Phi}_2||$ is overlarge due to a noise point at the bottom. To capture the data information more accurately and avoid the negative effect of noises, we re-define the length of Bounded Principal Component by the standard deviation (i.e., σ) of data projections.

Definition 2 Bounded Principal Component by σ. *For a Principal Component, denoted as Φ_i which identifies a line (or direction), its Bounded Principal Component, denoted as $\ddot{\Phi}_i$, identifies a segment of the line bounded by σ_i, where σ_i is the standard deviation of projections on Φ_i away from the mean for all data points. The length of Bounded Principal Component $||\ddot{\Phi}_i||$ is $2\sigma_i$.*

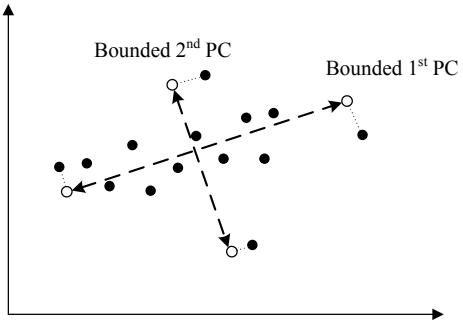


Fig. 2. Bounded Principal Components.

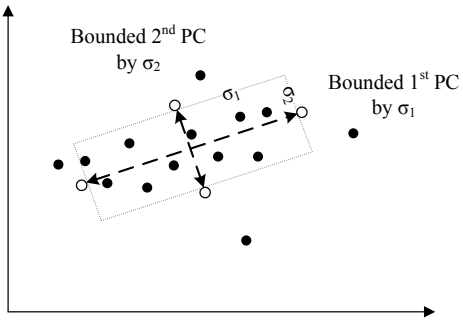


Fig. 3. Bounded Principal Components by σ.

Standard deviation is the most common measure of statistical dispersion for one-dimensional data. Simply put, standard deviation measures how data points are spread out in a dataset. More precisely, it is a measure of the average distance of data points from their mean. If the data points are all close to the mean, the standard deviation is small (closer to zero). If many data points are very distant from the mean, the standard deviation is large (further from zero). Here each σ_i indicates the average distance from the mean (i.e., the origin of coordinate system) to a projection on Φ_i. Following Figure 2, Figure 3 shows the corresponding Bounded Principal Components by σ_i. As we can see, Bounded Principal Components are now bounded by tighter ranges which include most of the data points (in the dashed rectangle).

Assume that the projections along a Principal Component follows the normal distribution. Statistically, about 68% of the data points are at within 1 standard deviation of the mean, about 95% of the data points are within 2 standard deviations, and about 99.7% lie within 3 standard deviations. This is known as the "68-95-99.7 rule". Naturally, this provides a reference to flexibly determine the scale of $||\ddot{\Phi}_i||$ by σ_i. That is, we can further relax the length of Bounded Principal Component to be $||\ddot{\Phi}_i|| = 2c\sigma_i$, where c is a user-defined constant.

There is yet another advantage of using σ_i for Bounded Principal Component. In Definition 2, Bounded Principal Component is represented by the Principal Compo-

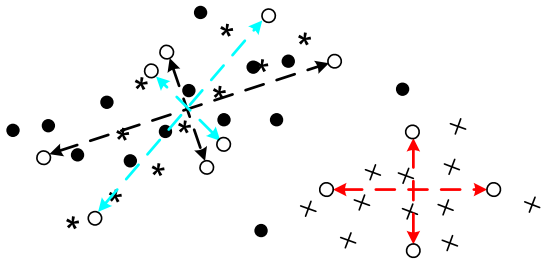


Fig. 4. Three sample video clips and their BCSs.

nent and two projections. Since Bounded Principal Component by σ_i is symmetric with respect to the origin, i.e., the data range away from the origin in either direction is the same, we can represent Bounded Principal Component by the Principal Component and its σ_i. Moreover, if the length of Principle Component is represented by σ_i, the Principal Component itself is sufficient to represent its Bounded Principal Component (i.e., $\Phi_i = \ddot{\Phi}_i$) since the mean (the origin of coordinate system) is known. In the following, we refer Bounded Principal Component as the one bounded by σ_i.

Now, we are ready to define Bounded Coordinate System for compact video clip representation.

Definition 3 Bounded Coordinate System. *Given a video clip in the form of $X = \{x_1, x_2, \ldots, x_n\}$, where x_i is a d-dimensional point (or feature vector) and assume $n \geq d$, Bounded Coordinate System of X is described by the mean of all x_i denoted as O (i.e., the origin of coordinate system) and d Bounded Principal Components (i.e., the bounded axes of coordinate system). In short, $BCS(X) = (O, \ddot{\Phi}_1, \ldots, \ddot{\Phi}_d)$.*

Assume Figure 3 depicts a set of frames representing a video clip, then it shows a Bounded Coordinate System (BCS) with two Bounded Principal Components. As we can see, BCS represents a video clip by an origin which reflects the dominating content and some Bounded Principal Components which capture the content changing trends and ranges by directions and lengths. It is a global and compact description. Independent of the number of points, BCS uses an origin and d Bounded Principal Components to summarize a video clip. Since both the origin and Bounded Principal Components are d-dimensional points/vectors, a BCS consists of $(d+1)$ d-dimensional points. Graphically, BCS can be viewed as a d-dimensional coordinate system centered at the mean of all points, where each of its axes is bounded. Figure 4 shows two more sample video clips and their corresponding BCSs, where the $\star$ video clip (signified with $\star$) has very similar origin to that of the $\bullet$ video clip (signified with $\bullet$) but different in orientations, and the $\times$ video clip (signified with $\times$) has different origin and orientations from those of the $\star$ and $\bullet$ video clips.

3.2 Further Discussion

In this subsection, we have a more detailed discussion on BCS.

3.2.1 *Robustness.* Theoretically, BCS can be regarded as a general model for characterizing a set of data points. It becomes more distinctive when the dataset exhibits higher degree of correlation (i.e., tendency) in feature space. Given the characteristic of a video clip that it often shows a moment of significance, it is highly expected that a video clip has strongly correlated content in order to express the moment. Therefore, BCS tends to be particularly suitable for video clip representation.

Since BCS characterizes the content distribution of a video clip, it is robust to many video editions which have none or little effect on overall content distributions, such as changing video format, changing frame rate, scaling, resizing, shifting, flipping, inserting, deleting, swapping, and so on. Variations caused by different view points, lighting condition and coloring may change the origin of BCS. However, the content orientation information could still be highly preserved in BCS.

It is understood that BCS model ignores the temporal and length information of video data. However, this may not necessarily degrade the performance. Actually, it makes BCS robust to temporal re-ordering and length variation. As indicated in previous work [Cheung and Zakhor 2003; 2005], video clips of different lengths or temporal orders may also be very relevant and effective approaches should be capable of measuring videos of different lengths and robust to temporal re-ordering. For example, a movie trailer may have two different versions. One is a short version for advertising in peak period while the other is a long version for off-peak period.

Existing distances that consider temporal order and video length, such as Edit distance based methods, use frame pairwise comparison and do not allow moderate tolerance on temporal order and length. Thus, they are potentially unable to retrieve relevant videos of different temporal orders or lengths, although they may provide better ranking for videos with exact temporal order and length. While relevant video clips with exact temporal order and length can be retrieved, BCS is also able to retrieve other relevant video clips with variations on temporal order and video length. This is further supported by our experiments on real-life video clips which show the superior search accuracy over the methods which consider temporal order. More details are reported in Section 6.

3.2.2 *Selection of Bounded Principal Components.* In BCS, Bounded Principal Components with smaller eigenvalues also have smaller lengths since the length of each Bounded Principal Component is computed based on the standard deviation of projections. Given a BCS, many Bounded Principal Components may have very small lengths (close to zero). A question naturally arises: is it necessary to retain all Bounded Principal Components in a BCS?

Obviously, Bounded Principal Components ranked higher indicate the most significant content tendencies. On the contrary, along those Bounded Principal Components with very small lengths (i.e., their variances are close to zero), all values are very close to each other and the content tend to be stable. Consequently, we can represent the stable information by their average in the system origin and discard those Bounded Principal Components with small lengths since their tendencies are negligible. Clearly, discarding insignificant ones can further reduce the size of BCS representation since a smaller number of Bounded Principal Components are used. Note that along a Bounded Principal Component $\ddot{\Phi}_i$, the average of all points is

O_i (i.e., its i^{th} dimensional value). We only discard those bottom ranked Bounded Principal Components representing negligible tendency information, while the stable information still can be retained in the origin O. For example, if $\ddot{\Phi}_i$ is discarded, the major information along $\ddot{\Phi}_i$ is still retained in O_i.

There are some guidelines which can be used in discarding negligible Bounded Principal Components. One way is to pick Bounded Principal Components with the largest lengthes. A fixed number of Bounded Principal Components can be retained and a consistent representation for all video clips can be achieved. This preserves a varying amount of tendency information of video content. However, it is possible that many Bounded Principal Components have zero lengths. Therefore, retaining the fixed number of Bounded Principal Components is less appealing. Alternatively, we favor the same amount of tendency information measured by variances and choose a varying number of Bounded Principal Components. This will give (approximately) consistent amount of information to be retained at the expense of varying-sized representations regarding the number of Bounded Principal Components in BCSs. Therefore, a video clip can be summarized by the system origin O (a d-dimensional point) and d' Bounded Principal Components, where d' is expected to be much smaller than d.

As demonstrated by our experiments reported in Section 6, for the large collection of video clips each of which contains thousands of high-dimensional feature vectors, a small number of Bounded Principal Components is usually sufficient to represent each video clip for very satisfactory search accuracy. The complexity of video data is then reduced dramatically. This has a significant impact as real-time search can be achieved practically by indexing small-sized video representations - BCSs. Note that BCS is a single representative for each video clip, in contrast to existing multi-representative methods.

3.2.3 *Comparison between BCS and PCA.* It is also worthwhile to distinguish the design philosophy of BCS from that of dimensionality reduction based on PCA. The idea of conventional PCA is to seek a subspace of lower dimensionality than the original space, by rotating the coordinate system such that the first coordinate axis (or Principal Component) exhibits the largest variance, the second coordinate axis exhibits the second largest variance, and so on. Most existing studies such as [Chakrabarti and Mehrotra 2000; Franco et al. 2007; Cui et al. 2006] apply PCA to project the data to the first few Principal Components so that the dimensionality can be reduced without losing much information. Here, PCA is employed as an intermediate to identify Bounded Coordinate System for statistical summarization of a video clip. However, compared with PCA, BCS has the following distinct features:

(1) BCS is a completely different representation model. Given a video clip in the form of $X = \{x_1, x_2, \ldots, x_n\}$, where x_i is a d-dimensional point (or frame feature vector), BCS is a single representative summarized from the whole set of data points for each video clip, to capture the content changing trends and ranges. That is, $BCS(X) = (O, \ddot{\Phi}_1, \ldots, \ddot{\Phi}_d)$, where O is the origin of coordinate system and Φ_i is the i^{th} Bounded Principal Component.
PCA, however, can be viewed as a "lossy data compression" technique of the same cardinality. It projects a data set into a lower dimensional subspace. The

representation provided by performing PCA is still a set of n points, but each with a lower dimensionality. That is, denoting the representation of a video clip projected by PCA as X', $X' = \{x'_1, x'_2, \ldots, x'_n\}$, where x'_i is a d'-dimensional point projected from the d-dimensional x_i and $d' < d$.
Clearly, two different representations also lead to different similarity measures. X' derived from PCA retains the same number of points in a video clip. The similarity measure based on the percentage of similar points (or frames) shared by two video clips [Cheung and Zakhor 2003; 2005; Shen et al. 2005] requires a quadratic time complexity to n since pairwise point comparisons are involved. However, as we will see in the next Section shortly, our BCS similarity has a linear time complexity to d, where d is typically much smaller than n. Such an efficient similarity measure paves the way for real-time search.

(2) BCS involves a bounded scheme to incorporate range information, by considering two furthermost projections, or preferably, the standard deviation of projections. PCA does not care about the range of data distribution long a principal component, but only identifies its directional information.

(3) BCS still holds if d'=d, in which case full fidelity is used to retain 100% energy (i.e., the number of Bounded Principal Components is not reduced for even more compact representation). It has to be noted that, the process of BCS generation itself discussed previously is not intended as a dimensionality reduction. In other words, d' is not always necessarily to be smaller than d.

Like PCA, BCS is an unsupervised technique and as such does not require any label information of the data. On the other hand, Linear Discriminant Analysis (LDA) relies on the label information to find informative projections that best discriminate among classes (rather than those that best describe the data) [Fukunaga 1990]. Given a training set containing frames from multiple classes, Fisher's linear discriminant is a specific choice of direction for projection of the data down to one dimension. We report some experiments in Section 6 to contrast BCS from a PCA implementation.

4. BCS SIMILARITY MEASURE

In the above section, we have shown how a video clip can be summarized into a compact representation - BCS. This section presents how the similarity between two BCSs is computed as an measure of their video similarity.

Besides the system origin and axes identified by Principal Components, BCS also bounds each axis as shown in Figures 3 and 4. The system origin measures the average position of all points, while Bounded Principal Components indicate the orientations of large tendencies together with their ranges.

Two coordinate systems can be matched by performing two operations: translation and rotation. A *translation* allows one to move its origin to wherever wanted. Using translation, we can move one system's origin to the position of the other. A *rotation* defines an angle which specifies the amount to rotate for an axis. Using rotation, we can rotate an axis in one system to match its correspondent in the other. Clearly, the translation distance from one origin to the other and the angles of rotations for all axes can be used to distinguish two coordinate systems. However, how to integrate the translation distance and rotation angles is unclear.

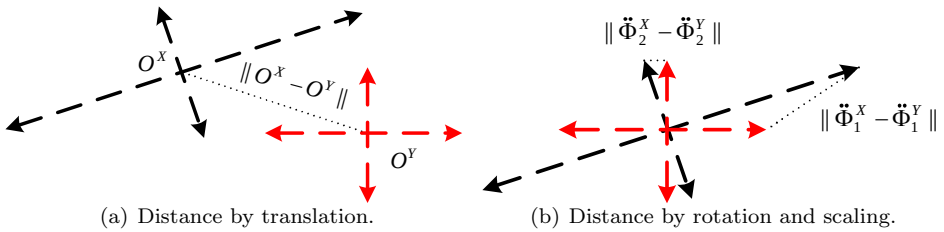
Uniquely, BCS also specifies the length of each axis. A *scaling* operation can be used to stretch or shrink one axis to be of equal length to the other. Matching two Bounded Principal Components requires both rotation and scaling. Very interestingly, in vector space, the difference of two vectors (in our case, two Bounded Principal Components) is given as the length of their subtraction, which nicely takes both rotation and scaling operations into consideration. Using this property, we are now ready to present the similarity measure of BCS. Note that each BCS may retain different number of Bounded Principal Components after discarding the insignificant ones.

Given two video clips X and Y and their corresponding BCS representations $BCS(X) = (O^X, \ddot{\Phi}_1^X, \ldots, \ddot{\Phi}_{d^X}^X)$ and $BCS(Y) = (O^Y, \ddot{\Phi}_1^Y, \ldots, \ddot{\Phi}_{d^Y}^Y)$, where d^X and d^Y are the numbers of Bounded Principal Components of $BCS(X)$ and $BCS(Y)$ respectively and assume $d^X \geq d^Y$, their distance is computed by:

$$D(BCS(X), BCS(Y)) = \underbrace{||O^X - O^Y||}_{by\ translation} + \underbrace{(\sum_{i=1}^{d^Y} ||\ddot{\Phi}_i^X - \ddot{\Phi}_i^Y|| + \sum_{i=d^Y+1}^{d^X} ||\ddot{\Phi}_i^X||)/2}_{by\ rotation\ and\ scaling}$$

The above distance function considers two factors: the distance between two origins by translation, and the distance between each pair of bounded axes by rotation and scaling. The first distance indicates the global difference between two sets of points, while the second distance indicates the average difference (i.e., $||\ddot{\Phi}_i^X - \ddot{\Phi}_i^Y||/2$) of all the corresponding Bounded Principal Components which reflect the content changing trends and ranges. If one BCS has more Bounded Principal Components than the other, half of their lengths (i.e, $||\ddot{\Phi}_i^X||/2$) will be added directly as the differences on those extra Bounded Principal Components. Recall that the origin indicates the dominating content of a video clip, and Bounded Principal Components indicate the content tendencies. Given two video clips, if their major visual contents are very similar, their origins are expected to be close to each other. Using Bounded Principal Components, two video clips having similar dominating content can be distinguished as well since Bounded Principal Components capture possible content tendencies if any. For example, given two different TV commercials of shampoo, their dominating contents can be similar, e.g., the major colors are both white with different minor colors. Therefore, their origins are expected to be close. However, the different correlations of the similar major and different minor colors carried by Bounded Principal Components can be used to better distinguish these two video clips. Clearly, our BCS similarity measure addresses more on content changing trends and ranges since the distance by rotation and scaling involves multiple bounded principle components.

The geometric interpretation of BCS similarity measure can be illustrated in Figure 5 with two BCSs. In Figure 5(a), the distance between two origins is the length of their subtraction, i.e., $||O^X - O^Y||$, which is also the Euclidean distance between O^X and O^Y. It represents the average distance between two sets of points. In Figure 5(b), the maximum distance on Bounded Principal Component is computed as $||\ddot{\Phi}_i^X - \ddot{\Phi}_i^Y||$, which considers the factors of rotation and scaling. Based on the triangle property, the average distance on each Bounded Principal Component is simply $||\ddot{\Phi}_i^X - \ddot{\Phi}_i^Y||/2$. In short, the proposed BCS similarity measure integrates the



(a) Distance by translation. (b) Distance by rotation and scaling.

Fig. 5. BCS similarity measure.

operations of translation, rotation and scaling to effectively compute the similarity between two BCSs. Particularly, rotation and scaling take the content tendency information into consideration for measuring the relevance of video clips.

Let $BCS(X) = (O^X, \ddot{\Phi}_1^X, \ldots, \ddot{\Phi}_{d^X}^X)$, $BCS(Y) = (O^Y, \ddot{\Phi}_1^Y, \ldots, \ddot{\Phi}_{d^Y}^Y)$, and $BCS(Z) = (O^Z, \ddot{\Phi}_1^Z, \ldots, \ddot{\Phi}_{d^Z}^Z)$ be three BCS representations. BCS similarity measure satisfies the following properties of being metric:

$$D(BCS(X), BCS(Y)) \geq 0 \ (positivity)$$

$$D(BCS(X), BCS(Y)) = 0 \Longleftrightarrow BCS(X) = BCS(Y) \ (symmetry)$$

$$D(BCS(X), BCS(Y)) \leq D(BCS(X), BCS(Z)) + D(BCS(Z), BCS(Y)) \ (triangle\ inequality)$$

The time complexity of BCS similarity measure is linear in the number of Bounded Principal Components. Note that the time complexities of existing methods for measuring video similarity are normally quadratic. Extended Graph Edit Distance (EGED) [Lee et al. 2005]) has an NP-complete time complexity and heuristic methods are strongly desired. The time complexity of Edit Distance on Real sequence (EDR) [Chen et al. 2005] is quadratic in video length. Given two videos $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$ in the d-dimensional feature space, EDR requires $m \times n$ inter-frame distance computations in the d-dimensional space. Take two video clips of 1,000 frames as an example. It will compute 1000^2 inter-frame Euclidean distances before an overall similarity can be aggregated. ViSig [Cheung and Zakhor 2003; 2005] and ViTri [Shen et al. 2005] improve the complexity to be quadratic in the number of representatives, which is much smaller than the number of frames. Assume two videos X and Y are summarized into n' and m' representatives, then $m' \times n'$ inter-representative distance computations are required. However, given $BCS(X) = (O^X, \ddot{\Phi}_1^X, \ldots, \ddot{\Phi}_{d^X}^X)$ and $BCS(Y) = (O^Y, \ddot{\Phi}_1^Y, \ldots, \ddot{\Phi}_{d^Y}^Y)$ for videos X and Y respectively and assume $d^X \geq d^Y$, the BCS similarity measure only requires $(d^X + 1)$ distance computations, and d^X is expected to be even smaller than d. For example, for two BCSs with 10 Bounded Principal Components respectively, the similarity measure only involves (10+1) Euclidean distance computations for system origins and corresponding Bounded Principal Components. It is evident that BCS improves the computation complexity of video similarity measure greatly. The improvement becomes more significant as the video length increases. As we can see later in the experiments, for our real-life video feature data in up to 64-dimensional RGB color space, normally less than 10 Bounded

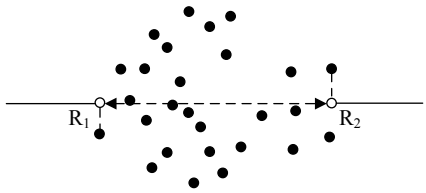


Fig. 6. Both sides out of the first Bounded Principal Component for optimal reference points.

Principal Components are sufficient for accurate search, and the improvement from the quadratic complexity of existing methods to linear complexity can benefit video search in large databases greatly.

5. BCS INDEXING

So far, we have discussed how video clips can be modelled as BCSs and how BCS similarity can be measured. The compact BCS representation reduces the complexity of video data greatly and the linear BCS similarity measure further improves the efficiency of retrieval. However, as the collection of video clips becomes very large, exhaustive scan on all BCSs to search similar clips is still undesirable. To further improve the retrieval efficiency for real-time search, effective indexing on BCSs is investigated.

The one-dimensional transformation accommodated with B^+-tree has shown the superiority due to its simplicity and efficiency [Jagadish et al. 2005]. Given a dataset, the one-dimensional transformation for a point P can be simply achieved by a mapping function $D(P, R)$ which computes the distance between the point and the selected reference point R, where D is the distance function used. The derived one-dimensional distance values for all the points are then sorted and indexed by a B^+-tree. Given a query Q and a search radius r, an efficient range search $[D(Q, R) - r, D(Q, R) + r]$ in B^+-tree is performed. All the points whose distances in the range are eligible for actual distance computations.

The performance of such a one-dimensional transformation is greatly affected by the selection of reference point which eventually influences the amount of information lost during the process of transformation. It is shown in [Shen et al. 2005] how to select an optimal reference point that can maximally preserve the original distance of any two points after transformation. It proves that the optimal reference points for such a one-dimensional transformation lie on the first Principal Component of the input dataset, but out of the first Bounded Principal Component. Since a Principal Component is bi-directional, optimal reference points fall into either side out of the first Bounded Principal Component. Figure 6 shows the possible positions of optimal reference points on either side (solid line segment) out of the first Bounded Principal Component (dashed line segment). Optimal reference points on the same side are expected to have the same effect on the transformation. Naturally, either furthermost projection which bounds the first Bounded Principal Component (Definition 1) can be selected as an optimal reference point (circles in Figure 6).

As we can see, there are two choices for reference point selection with respect to both directions of the Principal Component. One major weakness of the one-

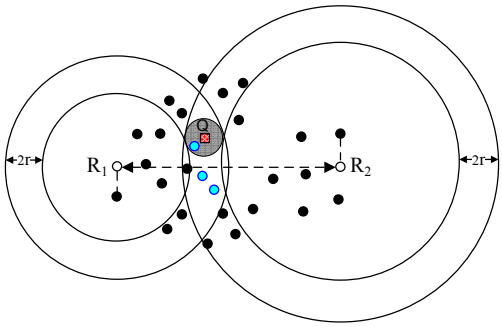


Fig. 7. Search space for using two optimal reference points.

dimensional transformation is that only a single reference point is used, since one reference point corresponds to one distance value for each point to be indexed by B$^+$-tree. To utilize the power of optimal reference points, in the following, we first introduce a two-dimensional transformation method called Bi-Distance Transformation (BDT) which further extends the standard B$^+$-tree to index two distance values for each point. Then, we look at how BDT can be applied to index BCSs.

The intuition of BDT is that two furthermost projections (i.e., two optimal reference points R_1 and R_2 as shown in Figure 6) separated by the first Bounded Principal Component are far away from each other. Points that are close to one optimal reference point will be far from the other. Given a query point Q and a search radius r, the probability for P of satisfying both $D(Q,R_1)-r \leq D(P,R_1) \leq D(Q,R_1)+r$ and $D(Q,R_2)-r \leq D(P,R_2) \leq D(Q,R_2)+r$ simultaneously is much less than that of satisfying either $D(Q,R_1)-r \leq D(P,R_1) \leq D(Q,R_1)+r$ or $D(Q,R_2)-r \leq D(P,R_2) \leq D(Q,R_2)+r$. Figure 7 shows the different search spaces of using a single optimal reference point (R_1 or R_2) and both together. The points in the annulus centered at R_1 (or R_2) need to be accessed when R_1 (or R_2) is used, while only the points in the intersection of both annuluses will be accessed when both R_1 and R_2 are used simultaneously.

Here, Bi-Distance Transformation is introduced to generate two distance values (i.e., two indexing keys) by using two furthermost projections on the first Principal Component as reference points. Formally, given a point P, its indexing keys are a pair of distance values computed as follows:

$$\begin{cases} K1(P) = D(P,R_1) \\ K2(P) = D(P,R_2) \end{cases}$$

Therefore, each point is associated with two indexing keys. Given a query Q and a search radius r, a point P can be safely pruned if it does not satisfy both $D(Q,R_1)-r \leq D(P,R_1) \leq D(Q,R_1)+r$ and $D(Q,R_2)-r \leq D(P,R_2) \leq D(Q,R_2)+r$. It has to be noted that there is no need to add any more optimal reference points since optimal reference points on the same side out of the first Bounded Principal Component are expected to have the same effect on the transformation.

B$^+$-tree is a one-dimensional indexing structure. Since each point is now repre-

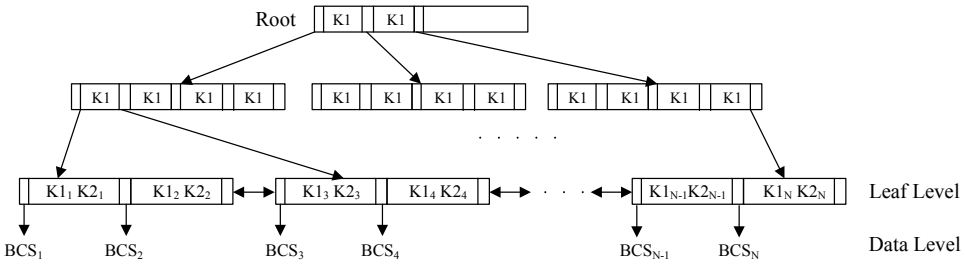


Fig. 8. An extended B$^+$-tree with two indexing keys.

sented by two indexing keys, B$^+$-tree has to be extended to utilize both keys[3]. A simple way is to sort and index the keys $K1$s (or $K2$s) with respect to R_1 (or R_2) by a B$^+$-tree while the keys corresponding to R_2 (or R_1) are embedded into the leaf node of a B$^+$-tree. A sample extended B$^+$-tree is given in Figure 8, where $K1$s are indexed and $K2$ is embedded into leaf nodes.

Given a query Q, its two indexing keys are first computed. Assume $K1$s are indexed. A range search on $K1$s is then performed in the extended B$^+$-tree. At leaf node level, the points whose $K1$s are in the range of $[D(Q, R_1) - r, D(Q, R_1) + r]$ are then checked whether their $K2$s are in the range of $[D(Q, R_2) - r, D(Q, R_2) + r]$. Only the points falling into both ranges are accessed for actual distance computations.

Compared with the standard B$^+$-tree, the extended version nearly doubles the number of leaf nodes. By expending a little on one-dimensional key checking, we may reduce a large number of candidates to save expensive disk accesses and actual distance computations. For dynamic insertion and deletion, the similar techniques introduced in [Shen et al. 2005] can be applied.

So far, we have discussed Bi-Distance Transformation which produces two keys for each point and uses the extended B$^+$-tree to index two keys. Our BCS model is a coordinate system described by an origin and some Bounded Principal Components. It not only indicates the dominating content of a video clip, but also content changing trends and ranges. To index BCSs using BDT, two optimal reference points have to be found first. Since BCS is a composite structure, we use the origins of BCSs to determine the optimal reference points. To do so, we first identify the first Bounded Principal Components for all BCSs' origins. The two BCSs corresponding to the two furthermost projections on the identified first Bounded Principal Component are selected as the two optimal reference points. Two indexing keys for each BCS of database video clips are then computed based on the proposed BCS similarity measure. As shown in Figure 8, BCSs are contained in the data level of extended B$^+$-tree.

Note that Bounded Principal Component is used twice. One is to generate a BCS for each video clip. The other is to index BCSs, where we identify the first Bounded Principal Component of all BCSs' origins to determine two optimal reference points for BDT indexing.

[3]A two-dimensional R-tree is not applicable since the pruning condition here is applied on each dimension, not on the aggregated distance from all dimensions as R-tree does.

6. EXPERIMENTS

In this section, we test the effectiveness and efficiency of our proposal on a large database of video clips by extensive performance evaluations.

6.1 Setup

Our database consists of about 30,000 non-identical video clips varying from seconds to minutes (more than 500 hours in total), including TV commercials, movies, news, documentaries, etc. They are recorded by VirtualDub[4] at PAL frame rate of 25fps in AVI format at the resolution of 384 × 288 pixels. For the sake of keeping the computational cost low and achieving high performance without resorting to complicated frame content representations, we extract the most perceptive visual feature - color histograms, as a generally applicable method of exploiting frame content distribution. Each video frame is processed and two popular color spaces are used: RGB color space and HSV color space. Four feature datasets in 8-, 16-, 32- and 64-dimensionality for each color space were extracted for test purpose. In our experiments the BCS representations of video clips are therefore based on content delivered from contiguous video frames in the high-dimensional RGB and HSV feature spaces. Both feature extraction and BCS generation are offline processes.

Since our proposal does not involve any domain-specific premise, it is applicable to all genres of videos. Except where specified, we select TV commercial as the default query genre since it has been extensively used in other studies on video search. Commercial videos often contain highly paced video editings and visual effects. Moreover, similar commercials sometimes have different lengths, e.g., several versions of a commercial are edited to be broadcasted in different TV stations or different time slots. All the results reported are the average based on 20 queries of commercial video clips randomly selected from the database, except for one experiment to study the effect of different program genres in which a variety of video types are used for comparison. The lengths of query video clips vary from seconds to minutes with an average of about 60 seconds, and the average number of shots of 20 commercial queries is about 18.25. The average number of relevant video clips is about 50. All the experiments were performed on Window XP platform with Intel Core 2 CPU (2.4 GHz) and 2.0 GB RAM.

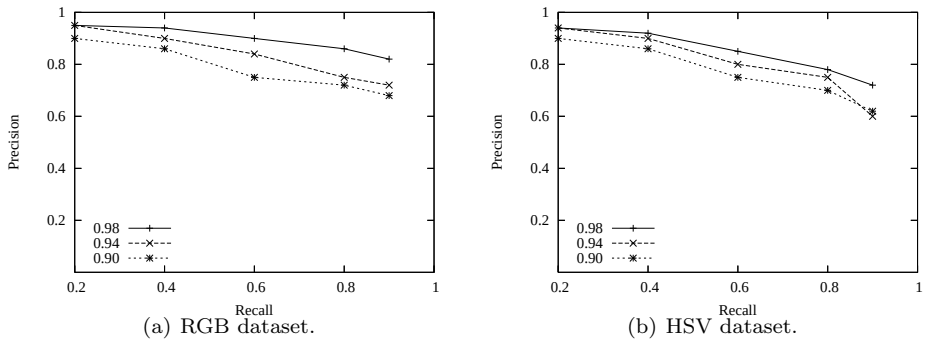
6.2 Effectiveness

In this subsection, we test the effectiveness of our BCS model by conducting a user study. For each query, a group of five users manually browse and classify the clips in the database to find its relevant clips based on their own perceptions, by taking various factors in their minds, such as objects, events, visual information, temporal information, for a whole judgement. It is doubtless that different people may have different perceptions on the judgement of relevance. For each query, the results determined by all users are intersected as the ground-truth for that query. Each query has at least 10 relevant video clips. The standard *precision-recall* curve is used to evaluate the effectiveness, where precision is defined as *the fraction of the number of relevant results returned to the total number of results returned*, while

[4]http://www.virtualdub.org

Percentage of Energy	90%	94%	98%
Average $\lvert\ddot{\Phi}\rvert$ of RGB Dataset	5.27	6.12	7.63
Average $\lvert\ddot{\Phi}\rvert$ of HSV Dataset	9.27	12.26	15.87

Table I. Percentage of energy vs. average $|\ddot{\Phi}|$.

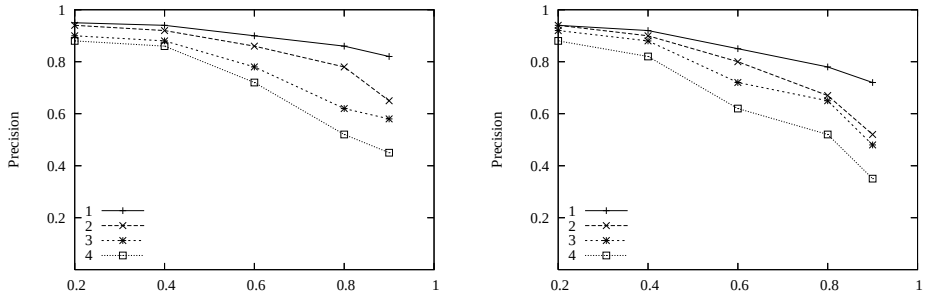


(a) RGB dataset. (b) HSV dataset.

Fig. 9. Effect of $|\ddot{\Phi}|$ on effectiveness.

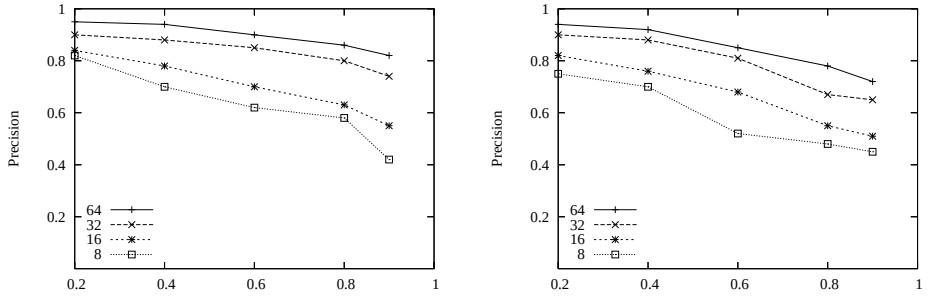
recall is defined as *the fraction of the number of relevant results returned to the total number of relevant results in the database*. We first investigate how the parameters affect the effectiveness of BCS, followed by a comparison study with some other methods.

6.2.1 *Effect of* $|\ddot{\Phi}|$. In the first experiment, we test the effect of the number of Bounded Principal Components, i.e., $|\ddot{\Phi}|$, on precision-recall using the 64-dimensional RGB and HSV datasets, and set c=1, where c is a constant in determining the length of Bounded Principal Component. As mentioned, each BCS may have different number of Bounded Principal Components by retaining the same amount of energy measured by their eigenvalues (i.e., the relative variances). Table I shows the average number of Bounded Principal Components for all video clips in relation to different percentages of energy retained. When 98% of energy is retained, the average numbers of Bounded Principal Components in 64-dimensional RGB and HSV datasets are 7.63 and 15.87 respectively. Undoubtedly, a larger percentage of energy corresponds to a larger average number of Bounded Principal Components. Given a video clip with the number of frames about 1,500 (for 60 seconds), 8-16 Bounded Principal Components are sufficient to retain most of its information. This suggests that video clips do present certain degree of tendencies on a small number of intrinsic directions. Uniform distribution of content rarely occurs for real-life video clips. Correspondingly, Figure 9 shows the precision and recall with respect to different percentages of energy retained (or $|\ddot{\Phi}|$) for each dataset. As we can see, as the amount of energy retained (or $|\ddot{\Phi}|$) increases, the precision and recall gain, since more information is embedded in BCS. When about 98% energy is retained, in both cases, the precisions are above 80% when the recalls are near 80%, which is quite satisfactory for video search based on human perception. For the following experiments, we retain 98% energy for video clips.



(a) RGB dataset. (b) HSV dataset.

Fig. 10. Effect of c on effectiveness.

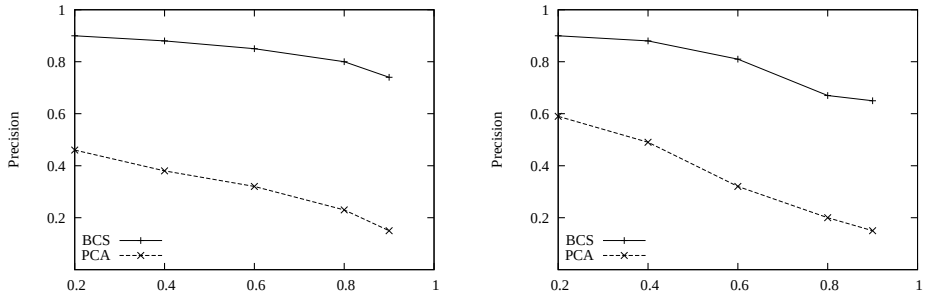


(a) RGB dataset. (b) HSV dataset.

Fig. 11. Effect of dimensionality on effectiveness.

6.2.2 *Effect of c.* In this experiment, we test the effect of c in scaling the length of Bounded Principal Component $||\ddot{\Phi}||$, by $c\sigma$. The datasets used for this experiment are also 64-dimensional RGB and HSV color features. Figure 10 shows that the precision-recall of our BCS becomes worse as c increases. When c reaches 4, the precision and recall drop considerably for both datasets. Although a larger c includes more data points in BCS representation, very few points will be further included for an increment of c when c is greater than 1. Thus, the accuracy of representation is expected to drop. Observing from Figure 10, we set c=1 for the following experiments due to its high search accuracy.

6.2.3 *Effect of Dimensionality.* In this experiment, we test the effect of dimensionalities on precision-recall using all the datasets in 8-, 16-, 32- and 64-dimensional RGB and HSV color spaces respectively. Figure 11 shows that generally the effectiveness becomes better as the feature dimensionality increases for both of them. This is reasonable since in general higher dimensional feature spaces carry more information about the content of video frames, which in turn benefits search accuracy.



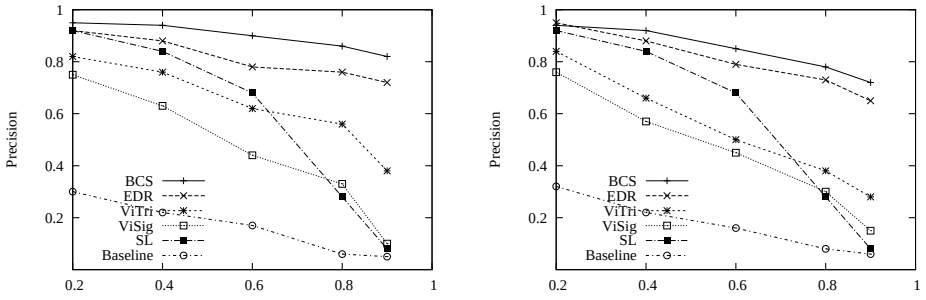
(a) RGB dataset. (b) HSV dataset.

Fig. 12. BCS vs. PCA on effectiveness.

6.2.4 *BCS vs. PCA.* In this experiment, we compare BCS with a PCA implementation. Recall that the representation projected by PCA is a set of n points in a lower dimensional subspace. The percentage of similar points shared by two video clips is used for its similarity measure [Cheung and Zakhor 2005]. For fair comparisons, the number of Principle Components retained in BCS is always the same as the subspace dimensionality in PCA.

Figure 12 shows the results of the effectiveness comparison between BCS and PCA in RGB and HSV datasets. We set to reduce the 64-dimensional RGB features to a 8-dimensional subspace and the 64-dimensional HSV features to a 16-dimensional subspace respectively. It is evident that BCS which treats each point set as a whole entity shows better results. This is due to the fact that BCS considers some important correlations among the individual elements for search. After PCA projections, the resulting lower dimensional representations still need aggregate point-to-point comparison results for measuring video similarity. However, the experiment results suggest it may have some limitations and could be unreliable, besides its quadratic time complexity.

6.2.5 *More Extensive Comparison Study.* In the following, we compare our proposal with five other methods to contrast BCS with them in terms of effectiveness. The global color histogram, which is the mean of all frame feature vectors, is selected as the baseline for comparison. ViSig [Cheung and Zakhor 2003] and ViTri [Shen et al. 2005] are typical multi-representative methods which represent a video clip by a small number of representatives. To estimate video similarity by the percentage of similar frames shared by two videos, the similarity of any pair of representatives is measured by the volume of intersection between two representatives and then the overall video similarity can be aggregated. In ViSig, 18 frames (i.e., the average number of shots in query clips) are randomly selected as seed vectors for Voronoi cell construction. In ViTri, the Euclidean distance for frame clustering affects the number and compactness of frame clusters, and here this parameter value is set to be 0.3 following the original paper. Edit Distance on Real sequence (EDR) [Chen et al. 2005] takes the temporal information into consideration for similarity computation. The distance threshold ϵ is set to be 0.3 as the best performing parameter value. We also compare BCS with the compact shot length signature (denoted as



(a) RGB dataset. (b) HSV dataset.

Fig. 13. BCS vs. EDR vs. ViTri vs. ViSig vs. SL vs. baseline.

SL in the following) introduced in [Hoad and Zobel 2006]. In our implementation, video clips are segmented with a reliable cut detection utility[5] based on color histograms in the uncompressed domain with a dynamic threshold to determine the presence of a cut, which is exactly the same process of shot detection for generating the compact signatures in the original paper. In the approximate string matching, the categorical scoring technique [Hoad and Zobel 2006] which is particularly tailored for SL signature is used as the similarity measure.

Figure 13 shows the results of six different methods using 64-dimensional RGB and HSV datasets. Note that for SL method, since the shot boundaries of video clips are obtained with a publicly available software, its effectiveness is independent of the color space of test dataset. However, we plot it in the corresponding figures for clearer comparison. From Figure 13(a), we have the following observations. First, based on manually judged ground-truth, our BCS model normally outperforms EDR by 5-10%, which in turn outperforms ViTri and ViSig by at least 10%. SL does not perform very well in our test database and baseline method does not work. When the recall is 60%, BCS achieves more than 90% precision, while EDR, ViTri, ViSig, SL and baseline are able to achieve around 80%, 60%, 45%, 70% and 20% precision respectively. Figure 13(b) shows similar results for HSV feature space. This reveals that the moment of significance embedded in video clip can be better described with the help of content tendencies. Baseline method basically represents a video clip by the mean of all feature vectors (i.e., the origin of BCS). Clearly, although the origin of BCS captures the dominating content, it is not very informative. BCS improves the baseline significantly by further emphasizing on the content changing trends and ranges to measure the similarity along the tendencies for effective relevance discovery. On the contrary, ViTri and ViSig which estimate video similarity based on the percentage of similar frames sometimes fail to resemble the relevance of video clips according to human perception. ViSig performs worse than ViTri since it may generate quite different signatures for similar videos. This suggests that the simple integration of representative similarities may not sufficiently reflect the significance of video clips. Some embedded characteristics

[5]http://www.informatik.uni-mannheim.de/pi4/projects/MoCA/downloads.html

that can better express the meanings of video clips need to be exploited.

While EDR considers the temporal order of frames for video sequence matching, it is also based on directly comparing individual frame features, which might be unreliable for videos that have undergone transformations. Moreover, EDR potentially cannot find the edited videos which are different in temporal orders or lengths. This may happen for example in the case of commercials, where some editions of the same commercial are made to produce shorter versions from a longer one. We also observe the search accuracy of SL method is not very good due to some reasons. First, the shot boundaries of relevant videos could be inconsistent. Second, although this method enjoys its representation compactness, a limitation is that it is not sufficiently discriminatory for short queries, or queries with relatively few shots. Third, SL does not carry content information. In practice, similar versions, in part or as a whole, can be edited from the original video content in many ways. Generally speaking, SL signature without content information is hard to find these similarities, thus its precisions become quite low particulary for large recalls. Last but not least, while BCS, EDR, ViTri, ViSig and baseline are not only suitable for clip level but also applicable to sub-shot or shot level, SL can only search video clips that consist of multiple shots. In short, EDR, ViTri, ViSig, SL and baseline all fail to exploit the potential *correlations* of frame content existing in feature space.

Figure 14 shows a sample query and its top 5 results returned from the first three search methods, where each video clip (with its ID) is displayed with 5 key-frames. As we can see, BCS retrieves the best ranked results, while EDR ranks the same results differently, and ViTri provides the worst top 5 results. Note that the query and the top 3 results of BCS are actually different versions of TV commercials for a same product with editing effects (e.g., different prices and contact numbers are displayed when the commercials are broadcasted in different cities/states).

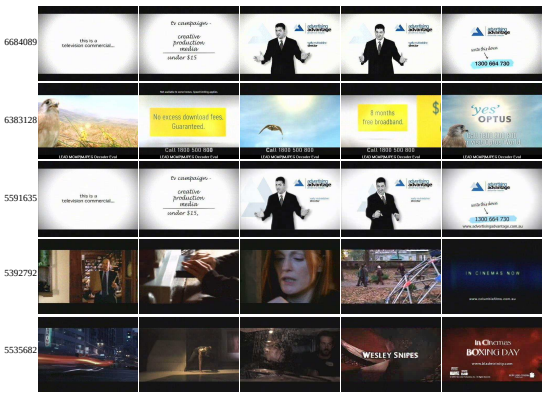
The comparison experiment shows that BCS captures the dominating content and content tendencies which probably better reflect the meanings of video clips, and its similarity model can accurately measure the relevance of videos from the human perception point of view. To further consolidate the claim that the content tendencies among frames better reflect the significance (the meaning that is expressed) of a video clip, more extensive user studies on even larger collection of more diverse video clips are planed in future.

6.2.6 *Effect of Program Genre.* In this experiment, we further test the effect of video genres on BCS using the 64-dimensional RGB and HSV datasets. The four different program genres and the average number of shots for the query video clips are summarized in Table II. Figure 15 shows the results of commercial, movie, news and documentary queries separately. It can be observed from Figure 15 that, compared with commercial queries, searching relevant news videos appears to be most difficult. The influences of program genre on effectiveness are mainly from the nature of video content as well as the feature used for frame representation. Specifically, while color histograms tent to be more distinct for different TV commercials, they can be close to each other for irrelevant news stories. Another important reason is a same event judged by users can be captured from the different cameras with substantial variations in viewpoint, angle, etc. In this case, accurate search for video representations using global features can be very challenging. Despite

(a) A sample query video.

(b) BCS search results.

(c) EDR search results.

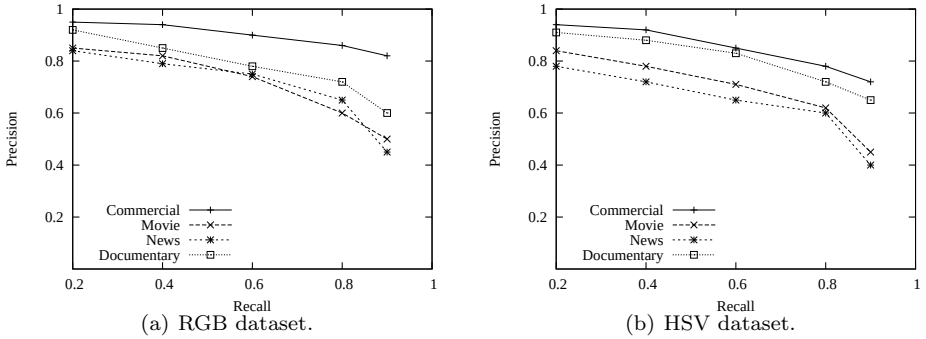


(d) ViTri search results.

Fig. 14. An example of top 5 results searched by BCS, EDR and ViTri.

Program Genre	Commercial	Movie	News	Documentary
Number of Queries	20	20	10	10
Duration in Minutes (Mean±Std-dev)	1.00±0.64	1.23±0.83	0.95±0.23	1.55±0.98
Average Number of Shots	18.25	15.26	13.20	13.54

Table II. Summary of query statistics.



(a) RGB dataset. (b) HSV dataset.

Fig. 15. Effect of program genre on effectiveness.

Method	BCS	EDR	ViTri	ViSig	SL
Time (s)	0.24	1600	38	44	12

Table III. Comparison of computational time.

these difficulties, our proposal still provides relatively satisfactory results. Since we have similar results for both RGB and HSV feature spaces, for the following experiments, we report the results on RGB feature space only.

6.3 Efficiency

In this subsection, we test the efficiency of BCS search by investigating our indexing method BDT. By default, we set k=20 in kNN search.

Our BCS model reduces the complexity of video data dramatically. From the previous subsection, we know that the number of Bounded Principal Components is very small. Meanwhile, BCS is a single video representative for each clip. Given a query video, only a single access to the underlying indexing structure is required. Moreover, the time complexity of BCS similarity measure is linear. We compare its computational time with EDR, ViTri, ViSig, and SL. In our experiment, for the dataset of 30,000 video clips in 32-dimensional RGB color space, the average CPU computational time without deploying any indexing structure for each method is shown in Table III, where key-frames (one key-frame per second) are used for EDR computation. As we can see, the efficiency of other methods is not comparable to that of BCS, given that their time complexities of similarity measures are all quadratic. Particularly, EDR involves too many key-frame pairwise comparisons. ViTri, ViSig and SL improve EDR by a smaller number of representative pairwise comparisons. Clearly, expensive computations caused by quadratic time complexity are undesired for real-time search.

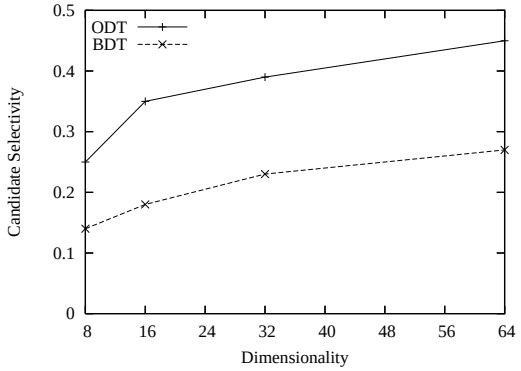


Fig. 16. Effect of dimensionality on efficiency.

Thus, in the following we focus on studying the performance of our indexing method BDT, compared with the optimal One-Dimensional Transformation, denoted as ODT, introduced earlier in [Shen et al. 2005]. ODT randomly selects one as the reference point from two optimal reference points derived by BDT. The measure we used here is *candidate selectivity*, which is defined as *the percentage of BCSs accessed*, i.e., the number of BCSs accessed in the extended B^+-tree divided by the total number of BCSs.

6.3.1 *Effect of Dimensionality.* In this experiment, we test the effect of dimensionality on efficiency using 30,000 BCSs of video clips generated from 8-, 16-, 32- and 64-dimensional RGB color datasets respectively. Figure 16 is the performance comparison between BDT and ODT. As we can see, the candidate selectivity increases as dimensionality goes up, due to "dimensionality curse". However, BDT improves ODT by further pruning nearly half of ODT's candidates for various dimensionalities. This demonstrates that the pruning power of BDT utilizing a pair of optimal reference points can save nearly half of disk accesses and expensive distance computations, with small overhead in leaf node level of B^+-tree.

6.3.2 *Effect of Data Size.* In this experiment, we test the effect of data size on efficiency using 32-dimensional RGB dataset. The results are shown in Figure 17. Clearly, as the number of database video clips increases, more BCSs are expected to be included in the candidate set. Nevertheless, BDT outperforms ODT by large margins for various data sizes.

6.3.3 *Effect of k.* In the last experiment, we test the effect of k in kNN search on efficiency using 32-dimensional RGB dataset. Figure 18 shows the changing trend of candidate selectivity as k increases. Both BDT and ODT go up as k increases, since more candidates are expected for more results. Again, BDT outperforms ODT by large margins for various searched results.

7. CONCLUSIONS

We introduce a novel video clip representation model called Bounded Coordinate System (BCS), which is a single video representative capturing the dominating con-

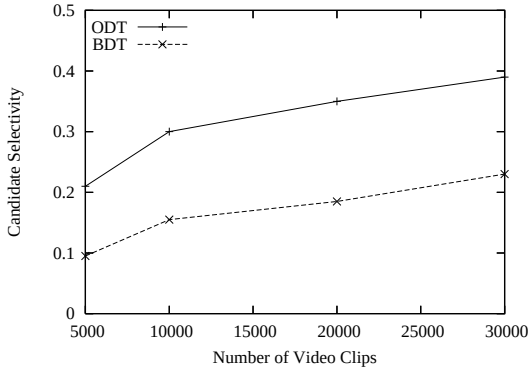


Fig. 17. Effect of data size on efficiency.

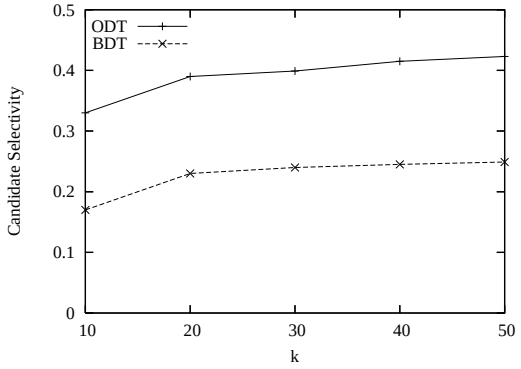


Fig. 18. Effect of k on efficiency.

tent and content changing trends of a video clip and reduces the complexity of video data dramatically. It summarizes a video clip by a coordinate system, where each of its coordinate axes is identified by PCA and bounded by the range of data projections along the axis. The similarity measure of BCS integrates the distance between the origins of two BCSs by translation, and the distance of all the corresponding Bounded Principal Components by rotation and scaling. This model improves the traditional quadratic time complexity of video similarity measure to linear. To further improve the retrieval efficiency for large video databases, we introduce a two-dimensional transformation method called Bi-Distance Transformation (BDT) which utilizes a pair of optimal reference points with respect to bi-directional axes in BCS. Our experiments on the large collection of real-life video clips prove the effectiveness and efficiency of our proposal. Although only conceptually simple and computationally inexpensive color feature is used in our experiments, the proposed method inherently supports other features of frame content representation.

REFERENCES

Adjeroh, D. A., Lee, M.-C., and King, I. 1999. A distance measure for video sequences. *Computer Vision and Image Understanding 75,* 1-2, 25–45.

Berchtold, S., Böhm, C., and Kriegel, H.-P. 1998. The pyramid-technique: Towards breaking the curse of dimensionality. In *SIGMOD Conference*. 142–153.

Berchtold, S., Keim, D. A., and Kriegel, H.-P. 1996. The x-tree : An index structure for high-dimensional data. In *VLDB*. 28–39.

Bertini, M., Bimbo, A. D., and Nunziati, W. 2006. Video clip matching using mpeg-7 descriptors and edit distance. In *CIVR*. 133–142.

Böhm, C., Berchtold, S., and Keim, D. A. 2001. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys 33,* 3, 322–373.

Chakrabarti, K. and Mehrotra, S. 2000. Local dimensionality reduction: A new approach to indexing high dimensional spaces. In *VLDB*. 89–100.

Chang, H. S., Sull, S., and Lee, S. U. 1999. Efficient video indexing scheme for content-based retrieval. *IEEE Trans. Circuits Syst. Video Techn. 9,* 8, 1269–1279.

Chen, L. and Chua, T.-S. 2001. A match and tiling approach to content-based video retrieval. In *ICME*. 417–420.

Chen, L., Özsu, M. T., and Oria, V. 2004. Mindex: An efficient index structure for salient-object-based queries in video databases. *Multimedia Syst. 10,* 1, 56–71.

Chen, L., Özsu, M. T., and Oria, V. 2005. Robust and fast similarity search for moving object trajectories. In *SIGMOD Conference*. 491–502.

Cheung, S.-C. S. and Zakhor, A. 2003. Efficient video similarity measurement with video signature. *IEEE Trans. Circuits Syst. Video Techn. 13,* 1, 59–74.

Cheung, S.-C. S. and Zakhor, A. 2005. Fast similarity search and clustering of video sequences on the world-wide-web. *IEEE Transactions on Multimedia 7,* 3, 524–537.

Chiu, C.-Y., Li, C.-H., Wang, H.-A., Chen, C.-S., and Chien, L.-F. 2006. A time warping based approach for video copy detection. In *ICPR (3)*. 228–231.

Cui, B., Shen, J., Cong, G., Shen, H. T., and Yu, C. 2006. Exploring composite acoustic features for efficient music similarity query. In *ACM Multimedia*. 412–420.

Dadason, K., Lejsek, H., Ásmundsson, F. H., Jónsson, B. T., and Amsaleg, L. 2007. Videntifier: identifying pirated videos in real-time. In *ACM Multimedia*. 471–472.

DeMenthon, D., Kobla, V., and Doermann, D. S. 1998. Video summarization by curve simplification. In *ACM Multimedia*. 211–218.

Ferman, A. M. and Tekalp, A. M. 2003. Two-stage hierarchical video summary extraction to match low-level user browsing preferences. *IEEE Transactions on Multimedia 5,* 2, 244–256.

Franco, A., Lumini, A., and Maio, D. 2007. Mkl-tree: an index structure for high-dimensional vector spaces. *Multimedia Syst. 12,* 6, 533–550.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition.*, second ed. Academic Press.

Gibbon, D. C. 2005. Introduction to video search engines. In *WWW, Tutorial*.

Gionis, A., Indyk, P., and Motwani, R. 1999. Similarity search in high dimensions via hashing. In *VLDB*. 518–529.

Hampapur, A., Hyun, K.-H., and Bolle, R. M. 2002. Comparison of sequence matching techniques for video copy detection. In *Storage and Retrieval for Image and Video Databases (SPIE)*. 194–201.

Ho, Y.-H., Lin, C.-W., Chen, J.-F., and Liao, H.-Y. M. 2006. Fast coarse-to-fine video retrieval using shot-level spatio-temporal statistics. *IEEE Trans. Circuits Syst. Video Techn. 16,* 5, 642–648.

Hoad, T. C. and Zobel, J. 2006. Detection of video sequences using compact signatures. *ACM Trans. Inf. Syst. 24,* 1, 1–50.

Houle, M. E. and Sakuma, J. 2005. Fast approximate similarity search in extremely high-dimensional data sets. In *ICDE*. 619–630.

Iyengar, G. and Lippman, A. 2000. Distributional clustering for efficient content-based retrieval of images and video. In *ICIP*. 81–84.

Jagadish, H. V., Ooi, B. C., Tan, K.-L., Yu, C., and Zhang, R. 2005. idistance: An adaptive b^+-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst. 30,* 2, 364–397.

Jolliffe, I. T. 2002. *Principal Component Analysis.*, second ed. Springer-Verlag.

Kashino, K., Kurozumi, T., and Murase, H. 2003. A quick search method for audio and video signals based on histogram pruning. *IEEE Transactions on Multimedia 5,* 3, 348–357.

Keogh, E. J. 2002. Exact indexing of dynamic time warping. In *VLDB*. 406–417.

Kim, C. and Vasudev, B. 2005. Spatiotemporal sequence matching for efficient video copy detection. *IEEE Trans. Circuits Syst. Video Techn. 15,* 1, 127–132.

Law-To, J., Chen, L., Joly, A., Laptev, I., Buisson, O., Gouet-Brunet, V., Boujemaa, N., and Stentiford, F. 2007. Video copy detection: a comparative study. In *CIVR*. 371–378.

Lee, J., Oh, J.-H., and Hwang, S. 2005. Strg-index: Spatio-temporal region graph indexing for large video databases. In *SIGMOD Conference*. 718–729.

Lee, S.-L., Chun, S.-J., Kim, D.-H., Lee, J.-H., and Chung, C.-W. 2000. Similarity search for multidimensional data sequences. In *ICDE*. 599–608.

Lienhart, R. 1999. Comparison of automatic shot boundary detection algorithms. In *Storage and Retrieval for Image and Video Databases (SPIE)*. 209–301.

Liu, X., Zhuang, Y., and Pan, Y. 1999. A new approach to retrieve video by example video clip. In *ACM Multimedia (2)*. 41–44.

Mohan, R. 1998. Video sequence matching. In *ICASSP*. 3697–3700.

Naphade, M. R., Yeung, M. M., and Yeo, B.-L. 2000. A novel scheme for fast and efficient video sequence matching using compact signatures. In *Storage and Retrieval for Image and Video Databases (SPIE)*. 564–572.

Peng, Y. and Ngo, C.-W. 2006. Clip-based similarity measure for query-dependent clip retrieval and video summarization. *IEEE Trans. Circuits Syst. Video Techn. 16,* 5, 612–627.

Rubner, Y., Puzicha, J., Tomasi, C., and Buhmann, J. M. 2001. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding 84,* 1, 25–43.

Santini, S. and Jain, R. 1999. Similarity measures. *IEEE Trans. Pattern Anal. Mach. Intell. 21,* 9, 871–883.

Sarukkai, R. 2005. Video search: opportunities & challenges. In *Multimedia Information Retrieval, Keynote Speech.*

Sebe, N., Lew, M. S., and Huijsmans, D. P. 2000. Toward improved ranking metrics. *IEEE Trans. Pattern Anal. Mach. Intell. 22,* 10, 1132–1143.

Shen, H. T., Ooi, B. C., Zhou, X., and Huang, Z. 2005. Towards effective indexing for very large video sequence database. In *SIGMOD Conference*. 730–741.

Shen, H. T., Zhou, X., Huang, Z., and Shao, J. 2007. Statistical summarization of content features for fast near-duplicate video detection. In *ACM Multimedia*. 164–165.

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. 2000. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell. 22,* 12, 1349–1380.

Tuncel, E., Ferhatosmanoglu, H., and Rose, K. 2002. Vq-index: an index structure for similarity searching in multimedia databases. In *ACM Multimedia*. 543–552.

Vlachos, M., Gunopulos, D., and Kollios, G. 2002. Discovering similar multidimensional trajectories. In *ICDE*. 673–684.

Weber, R., Schek, H.-J., and Blott, S. 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*. 194–205.

Wu, X., Hauptmann, A. G., and Ngo, C.-W. 2007. Practical elimination of near-duplicates from web video search. In *ACM Multimedia*. 218–227.

Wu, Y., Zhuang, Y., and Pan, Y. 2000. Content-based video similarity model. In *ACM Multimedia*. 465–467.

Zhou, J. and Zhang, X.-P. 2005. Automatic identification of digital video based on shot-level sequence matching. In *ACM Multimedia*. 515–518.

Zhu, X., Wu, X., Fan, J., Elmagarmid, A. K., and Aref, W. G. 2004. Exploring video content structure for hierarchical summarization. *Multimedia Syst. 10,* 2, 98–115.