

Detection of Video Sequences Using Compact Signatures

T. C HOAD and J. ZOBEL
RMIT University

Digital representations are widely used for audiovisual content, enabling the creation of large on-line repositories of video, allowing access such as video on demand. However, the ease of copying and distribution of digital video makes piracy a growing concern for content owners. We investigate methods for identifying coderivative video content—that is, video clips that are derived from the same original source. By using dynamic programming to identify regions of similarity in video signatures, it is possible to efficiently and accurately identify coderivatives, even when these regions constitute only a small section of the clip being searched. We propose four new methods for producing compact video signatures, based on the way in which the video changes over time. The intuition is that such properties are likely to be preserved even when the video is badly degraded. We demonstrate that these signatures are insensitive to dramatic changes in video bitrate and resolution, two parameters that are often altered when reencoding. In the presence of mild degradations, our methods can accurately identify copies of clips that are as short as 5 s within a dataset 140 min long. These methods are much faster than previously proposed techniques; using a more compact signature, this query can be completed in a few milliseconds.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Algorithms, Performance, Reliability

Additional Key Words and Phrases: Video similarity detection, dynamic programming, local alignment

1. INTRODUCTION

Digital formats have been adopted in all stages of the life cycle of video content, including preproduction, visual effects, editing, mastering, distribution, and display. These technologies have made it easier to produce copies of video data both legally and illegally, and to distribute those copies to a large audience. There are techniques that can be used to stem the unauthorised use of video

Some aspects of this work have previously been published by the authors in Hoad and Zobel [2003a, 2003c].

Authors' addresses: T. C. Hoad: 7853 149th Ave. NE, Redmond, WA 98052; email: tim@hoad.id.au; J. Zobel: School of Computer Science and Information Technology, RMIT University, GPO Vox 2476V, Melbourne, Victoria 3001, Australia; email: jz@cs.rmit.edu.au.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or permissions@acm.org.

© 2006 ACM 1046-8188/06/0100-0001 \$5.00

content, but in general these are either unreliable or restrictive upon the user. Physical copy prevention mechanisms can be bypassed, and can cause difficulty for law-abiding citizens when creating backups or using content as prescribed by copyright law.

Another way to address illegal copying is to detect copies of a piece of content and take action. As this does not involve a copy-prevention mechanism, the material can be copied within the bounds of the copyright laws, while still allowing content owners a mechanism to prevent widespread piracy. One vehicle for such an approach is video watermarking. Watermarking systems are limited by the fact that they either reduce video quality or are trivial to remove [Petitcolas et al. 1998; Langelaar et al. 1998; Hartung et al. 1999]. Also, a watermark must be present in the original content in order for copies to be traced. An alternative to watermarking is to detect copies based on the properties of the content itself. Content-based copy detection requires no watermark or other identifying feature embedded in the video and places no restrictions on the legal use of the content. Such techniques also have other applications, such as monitoring video streams for known content, and locating redundant or related clips within a collection.

Much of the research into video retrieval has concerned topic-based matching. While the general problem of locating video data that is relevant to an information need is important, we are not attempting to address this problem in this body of work. We are concerned with the more specific problem of identifying video content that is derived from the same source video as the user query, which we term *coderivative*. We define a pair of entities as being coderivative if they share the same original source. Thus, the 2002 remake of *The Time Machine* was not coderivative with the original from 1960, as there is no original film from which they were both created. Conversely, *Lord of the Rings—The Fellowship of the Ring* was coderivative with the DVD “Platinum Extended Edition” release, as the DVD extended version was created from the theatrical release. By definition, any piece of content is coderivative with itself.

Existing methods for searching video to identify coderivatives have substantial limitations: they are sensitive to degradation of the video; they are expensive to compute; and many are limited to comparison of whole clips, making them unsuitable for applications such as monitoring of continuous streams. Most of the previously proposed search methods require direct comparison of video features between the query clip and the data being searched, which is computationally expensive and sensitive to changes that can occur during “lossy” processes such as transcoding or analogue transmission.

We describe novel techniques for identification of coderivative video sequences that address the limitations of previous approaches. The methods proposed are all applicable to real-time monitoring of video streams, or to searching of large collections. The techniques we present are fast, accurate, and insensitive to video degradation, making them ideal for coderivative video search. Experiments on large quantities of real-world data demonstrate that our approach is robust and effective, even for short queries or queries that have been transcoded or degraded.

The methods we introduce use a two-stage process of signature generation and query evaluation. We have developed four novel methods for producing compact video signatures for coderivative detection that allow for fast, accurate search. The first method is based on the structure of the video. It uses the pattern of edits in the clip to produce a representation that is particularly compact, requiring about 5 kB/h of video. It is efficient to search, and is insensitive to changes in video quality. The second method uses the extent to which the color changes over time to determine similarity between the two clips being compared. Since the color information is not compared directly between video clips, this signature type is insensitive to changes in video quality. The third method compares the way in which the motion in the clip changes. This is computed using a novel algorithm that is substantially more efficient than traditional motion estimation: motion is estimated according to the position of centroids of luminance in the video frames. The final approach is a combination of the last two: by combining evidence based on color changes and centroid motion, we achieve substantial improvements in retrieval effectiveness.

We also describe a technique for searching video using these signatures. This method, based on approximate string matching, can identify regions of similarity between signatures, even when the video is substantially altered. Key to this search method is the function used to determine similarity between elements in the video signatures. We describe five functions that can be used for this task, and evaluate their comparative effectiveness experimentally.

We compared the effectiveness of these methods against a baseline by using them to retrieve known clips from a stream of free-to-air television content. Initial experiments employed a small collection of clips and explore the robustness of the proposed methods in the presence of various forms of degradation that occur in analogue and digital video. This was followed by larger-scale experiments in which clips of various lengths were randomly selected from a long sequence of video and then subjected to several forms of degradation before being used as queries to retrieve the clips from the original source. In these experiments, we demonstrated that our new methods are substantially more effective than previously proposed methods. In retrieval tests using a 140-min dataset, and randomly selected queries ranging from 5 to 120 s in length, we showed that one method is able to correctly identify the correct region as the highest-ranked match 85% of the time even when the queries are heavily degraded (by reducing video resolution), and 99% of the time when the degradation is minimal.

2. BACKGROUND

Obstacles to accurate matching include format variations and sources of degradation. A basic one is the broadcast standard, which varies from country to country. Standards in use include NTSC, SECAM, and PAL, and film; these vary in vertical resolution and frame rate. Digital formats are different again. Conversion between frame rates results in variation from the original signal, and in slight variations in speed of playback.

Another variable in video formats is the shape of the frames, or aspect ratio. The analogue broadcast standards use the aspect ratio of 4:3, while aspect ratios of 16:9 and 2.35:1 are common in cinema and digital broadcasting (including

HDTV). Corrections for aspect ratio involve forms of cropping or the addition of black regions, and can involve considerable manual intervention (for example, panning), thus potentially confusing standard video retrieval methods.

Color representation leads to further issues. Conversion between different color spaces is lossy. For example, RGB represents the three colors, while YCrCb represents color as luminance, red, and blue. In HSV, the dimensions are color, saturation, and brightness. Further problems are presented by the needs of compression; the number of bits per pixel may be low, and color subsampling may be used, so even within a format the color space is approximated.

Another source of degradation is the bitrate used to represent the content. Reducing video bitrate can introduce anomalies or *artifacts* into the picture. Using a lower frame rate or resolution is an effective way to eliminate these artifacts from the compressed video, but introduces other limitations as described above.

2.1 Video Structure

In addition to the inherent low-level formatting of video content, there is also a higher-level structure that is imposed in the process of creating a video. Films are structured similarly to printed books: where books have chapters and paragraphs, video content has chapters, *scenes*, and *shots*. A shot is a sequence of frames captured by a single continuous operation of a camera. The point at which one shot leads into another is called a *shot boundary*. There are many transition effects that can be applied at shot boundaries, but by far the most common is the *cut*, where one shot ends abruptly and the next begins at the next frame. We use shot boundaries in a signature method for finding coderivative video segments.

Research into automatic methods for determining the structure of video content has been pursued for some time. Algorithms for cut detection are now fairly robust, and detection of other transitions is possible with a reasonable level of accuracy [Boreczky and Rowe 1996; Naphade et al. 1998; Lienhart 2001].

Many algorithms have been proposed for detecting cuts in uncompressed video. These algorithms generally use features of the video frames to determine the level of discontinuity—that is, the amount of change between frames—and define a cut as a peak in the level of discontinuity. One feature that is widely used is the color histogram, which captures the distribution of colors in a video frame. Many variations of color histogram have been investigated for use in cut detection, including histograms using a variety of colorspace, histograms of different sizes, and the use of multiple histograms for different regions of the video frame [Ueda et al. 1991; Nagasaka and Tanaka 1992; Swanberg et al. 1993; Zhang et al. 1993; Hampapur et al. 1994; Patel and Sethi 1997]. Other features that have been successfully used to measure discontinuity for cut detection are edges [Zabih et al. 1995, 1999; Li and Lu 2000] and motion [Zhang et al. 1993; Boreczky and Rowe 1996; Liu et al. 2000; Naphade et al. 1998].

2.2 Previous Approaches to Video Search

It is easy to assume that methods for identifying semantically similar content would be applicable to coderivative detection, but this is not necessarily the

case. Some techniques that are used to locate similar content rely on metadata, which may be produced manually or automatically, but this data is likely to be different for each instance of a piece of content. Other techniques are limited to comparison of whole video clips, but much video content is distributed in a continuous stream that cannot be easily segmented for comparison. We explore a wide range of search techniques in this section, many of which are not suitable for copy detection for reasons such as those above. However, it is useful to understand how the related video search problems are addressed, and why these techniques are not appropriate for the specific task presented in this article.

There are many feature spaces that can be used for determining similarity between pairs of video, but all are derived from either the video frames or the soundtrack. Likewise, there are many ways in which these features may be abstracted and compared. Results are generally considered to be relevant to the query if they are visually similar. The extent of similarity that is required for a match to be considered correct is dependent on the application and the query posed, and relevance judgment is subjective.

Most early feature-based video retrieval systems were based on existing image retrieval engines [Yeo and Yeung 1997]. A key point that differentiates video comparison methods is the feature space used to compare the clips. A color histogram difference is often used to compare video [Yeung and Liu 1995; Tan et al. 1999; Yoon et al. 1999; Hauptmann et al. 2002a; Lee et al. 2003], although other feature representations have been used, including: intensity histograms, texture [Wu et al. 2000], facial-recognition features [Gupta and Jain 1997; Jaimes et al. 2002], and motion [Yeung and Liu 1995; Chang et al. 1997; Shan and Lee 1998; Ngo et al. 2001]. The range of specific color features is broad—color histograms, region histograms, color coherence vectors (CCVs), and ordinal signatures.

Liu et al. [1999] proposed that videos be segmented into shots, each of which is represented by a keyframe. A similar approach was described by Wu et al. [2000]. Fushikida et al. [1999] also used color features for retrieval of video clips, although they used region histograms, rather than a single histogram, for representative frames. There are major drawbacks to this approach. In particular, the techniques described are only able to compare whole clips: similarity is determined by the number of shots in common, and temporal relationships are generally ignored. The other significant drawback of this approach is that it uses color histograms, which are not robust to some of the problems that occur in the video domain. For content distributed in analogue form, noise and interference can cause color shifts, as well as alterations in texture.

Zhao et al. [2001] presented a method in which the clips are segmented into shots from which a small number of keyframes are selected for color analysis. Hauptmann and Papernick [2002] proposed that still images be partitioned into 16×16 pixel regions, with each region represented by the most common color (when images are reduced to 256 colors). Similarity is determined by the number of matching regions between query and image. Such systems are likely to be sensitive to shot length, as longer shots would be likely to contain more colors, causing these shots to be favored during querying.

Motion features are often used in conjunction with other features, but are less commonly used in isolation. One of the few examples of a motion-based retrieval system was presented by Ahanger et al. [1995], who discussed the use of object motion in formulating a query; object and camera motion is described graphically by the user and matched to data being searched. Yasugi et al. [2001] proposed a method for recognising identical events in video, based on the assumption that the same event will be tracked in a similar way with multiple cameras.

Some systems use speech transcripts or open captions as the dominant feature for retrieval [Hauptmann et al. 2002b]. Although the use of captions and transcripts is undoubtedly useful for general-purpose video search systems, it is unlikely that it would be effective for identifying coderivative content.

Perhaps the most common approach to video search based on low-level features is to use a combination of features. A limitation that can be observed in several hybrid systems, such as those described by Lienhart et al. [1998], Chang et al. [1998], and Yang et al. [2002], is that significant expertise and human intervention are required to assign weights to each of the features, or to provide relevance feedback to achieve acceptable results. A somewhat different approach is that of Ide et al. [2001]. In this system, information from open captions was used in conjunction with image attributes. Naphade et al. [2001] described a system that uses both audio and video features to determine similarity. This method is unsuitable for unsupervised use, as weights for the features must be assigned manually. A common attribute of these systems is that they are intended to achieve a high level of retrieval effectiveness, with little consideration of computational cost—many of these systems evaluate queries in approximately real time (that is, queries are evaluated in an amount of time comparable to the length of the data being searched).

2.3 Coderivative Identification

The feature-based querying systems described above can be used to identify coderivative material, but these systems are engineered to address the same needs as other semantic querying systems—to identify clips that are visually or semantically related. The limitations of these fields in detecting copied or coderivative material have led to an increasing interest in developing methods that effectively identify coderivatives in the video domain.

Lienhart et al. [1997] presented early work on video identification for recognizing and identifying television commercials. In order to identify these known advertisements, color coherence vectors (CCVs) were computed for each frame in the commercial blocks and for each frame in known advertisements, yielding a fingerprint. The minimal distance between query and data fingerprints was calculated by counting the number of insertions, deletions, and substitutions of CCVs required to convert the query fingerprint to the data signature. High accuracy was reported with the use of this method to identify commercials from a collection of 200; however, it is not clear that these results could be replicated if some degradation in video quality was present. A significant limitation of this method is that, once the fingerprints are computed, queries are processed in

approximately real time, making it unsuitable for retrieval from large collections. In our experiments, reported later, computational cost was a significant issue.

Mohan [1998] presented another approach to matching video sequences. A representation of the video is produced by computing an ordinal signature for a reduced-intensity version of each frame; these are then concatenated to form a vector, which is used to determine similarity. A sliding window is used to align the feature vectors, and distance is calculated by computing the average distance between frames, as determined by the distance function (in this case, an arithmetic difference). We used a similar technique as a baseline for our experiments in Section 5. This method is not limited to detecting coderivative content, but the experiments reported showed some success at identifying replays of a given piece of sporting content. The use of the ordinal measure is likely to make this approach less sensitive to video degradation than previous methods, but direct comparison of ordinal signatures is computationally expensive, making this method unsuitable for large collections.

Matching of video sequences based on color histograms has also been investigated. Adjeroh and Lee [1998] introduced a video representation where color histograms are computed for each frame in the sequence, which are then categorized into classes, based on the bin values. Adjeroh et al. [1998] described an algorithm for matching video sequences based on these representations. This involves a string-matching process similar to that used by Lienhart et al. [1997]. Based on our experiments reported in Section 5, which used comparison of color histograms between clips, it seems unlikely that this method would be effective for retrieval from large video collections, especially if the content is degraded.

Naphade et al. [2000] presented a system that uses YCC color histograms as a representation of a video sequence. Color histograms with 32 luminance bins and 16 bins for each chrominance channel are computed for each frame of the query clip and the target clip (the data being searched). A sliding window is used to compute a similarity measurement for every position in the target clip. Local maxima are extracted and sections of the target clip in which the local maximum similarity exceeds a fixed threshold are marked as a match. Results for effectiveness were not reported, but, because of the direct comparison of color information between clips, it is probable that this system would be sensitive to degradation of the video.

Hampapur and Bolle [2001] compared several different feature spaces for use in the detection of coderivative television commercials. In all cases, similarity was calculated using one frame from each second of video. In general, the shape-based feature spaces were found to be more discriminative when determining similarity between video clips; however, the data used was known to have variance in color fidelity. Based on our experience with methods involving direct comparison of visual features between clips, as reported in Section 5, it seems unlikely that these methods would be effective for the retrieval of coderivatives when the video is degraded. This work was followed up by Hampapur et al. [2001], with the investigation of a video search framework in which motion, ordinal, and colorbased signatures are compared. Short queries were evaluated in

approximately real time using these methods; retrieval of substantial queries from large collections would be impractical.

Another approach to similar video search was proposed by Ng et al. [2001]. Using shot boundary detection and shot clustering, a tree is built to represent the structure of the clips to be compared. Similarity is computed in a top-down manner, by recursively computing the similarity of the child nodes—the similarity of a node is defined as the normalized sum of the similarity of child nodes. At the leaf nodes, representing the shots, similarity is determined by a combination of color-histogram distances of key frames and the difference in shot length. This approach is limited to comparing whole clips—it is not suitable for finding regions of similarity within a longer stream. Another limitation is the direct comparison of color features, which is likely to be unreliable for video that has undergone changes in color.

Cheung and Zakhor [2000] described another approach to detecting coderivative video content, in which similarity between two clips is determined by the use of a video signature. The similarity between a pair of clips is determined by the similarity between the frames in the clips, and temporal information is ignored. To reduce the number of similarity computations, a subset of frames in the clips is used to produce the signature. A significant limitation of this work is that it is only useful for comparing whole clips: when evaluating similarity, sequences are treated as a “bag” of frames, which does not allow identification of similar subsequences. As with the approach of Naphade et al. [2000], this work also employs direct comparison of color information between clips, which would be likely to be susceptible to degradation, especially that involving color shifts.

DeMenthon [2003] described how *video strands* can be used to create spatio-temporal descriptions of video data. DeMenthon and Doermann [2003] reported experimental results using these descriptors to retrieve very short clips (15–100 frames) that exhibit distinctive patterns of motion, such as the graphic animations used by broadcasters to introduce programmes (referred to as *dynamic logos*).

All of the approaches described so far are computationally expensive. Hoi [2002] addressed this issue by proposing a two-stage search process. The first, *coarse search*, is conducted using a signature based on low-level features mapped to a low-dimensional feature space using the pyramid technique described in Berchtold et al. [1998]. This reduces the search space for the second, *fine search*, which uses a higher-dimensional feature space to improve the result rankings. Experimental results presented by Hoi et al. [2003] reported that the effectiveness of this method was limited: 85% of the correct matches were retrieved with precision of 90%—that is, 90% of the results listed were correct matches. It was assumed that the two-phase search process would result in a reduction in query evaluation times, but this was not tested experimentally.

Another proposal for addressing efficiency issues was described by Park et al. [2002], who used a trie-based index. To produce a representation that is compatible with a trie, low-level features are computed for each frame. Searching the trie involves producing an equivalent feature representation for the query clip, then traversing the trie to find the most similar frames in the data being

searched. Since only one matching region can be identified in each clip, the application of this technique to searching for coderivative sections in a large clip or continuous stream is limited. Park and Hyun [2004] reported efficiency tests conducted using 2 h of video from a movie, and 100 short queries. Our experience with methods that compare visual features directly between clips, presented in Section 5, suggests that this approach would not be sufficiently discriminatory for identification of copies of clips in large collections of video.

Another system that uses an index for similar video search was described by Hampapur and Bolle [2002]. It allows fast lookup of matching frames, but does not allow approximate matches: the local edge representations must be identical. It seems unlikely that this approach would be successful when the content is degraded, as differences will be present in the visual features. As lossy compression is used, even with nondegraded video, two copies of the same clip are not identical.

Pua et al. [2004] presented another indexing method for searching video. To evaluate a query, it is segmented into shots, and color moments for each frame are computed and quantized. A limitation of this approach is that similarity is computed on whole shots, making it unsuitable for finding sequences in which shots are truncated. It also relies on a large index structure being stored in memory. While the exact size of the structure was not disclosed, it seems likely that the index would require several megabytes per hour of data, in addition to the full (nonquantized) color moments, which would presumably be stored on disk. However, it is efficient.

3. REPRESENTATIONS FOR FAST IDENTIFICATION OF CODERIVATIVES

The search for coderivatives can use the same methods as similarity search. The frames of the query clip can be compared against the stored video one by one, and a sequence of matching or highly similar frames indicates that the same clip has been found. The main drawback of this method is that, for each frame, a color histogram or other descriptor must be stored, and the search process involves aligning (and thus comparing) sequences of descriptors—a costly process. Also, direct image comparison is sensitive to degradation: noise, artifacts due to encoding, or color shifts affect the accuracy of established image comparison techniques, resulting in a system that is unlikely to cope with these changes.

The methods presented in this article allow for both whole-clip comparisons in addition to the facility for long clips to be searched for sections of high similarity to a query clip. Once the collection has been preprocessed, it can be searched for instances of a query clip extremely fast.

We present a two-stage approach to coderivative identification. The first stage involves preprocessing of the video to be searched, which is followed by a query evaluation stage. The preprocessing, presented in this section, accesses the video clips sequentially to produce a compact representation. This representation can take many forms; we propose four methods for producing a representation, each of which uses different features of the video. Each method has strengths that are applicable to different kinds of content. The methods are described in detail below.

The first method—the *shot-length* method—is based on the structure of the video. The intuition is that copies of video can be identified by matching temporal patterns of events in the clips being compared. This requires the selection of events that can be efficiently and consistently identified, regardless of degradation or minor changes. Shot boundaries are an ideal candidate—they can be detected efficiently using robust, proven algorithms. The shot-length signature is described in detail in the following section.

Preliminary experiments demonstrated that the shot-length signature is suited to fast retrieval, but was not sufficiently discriminatory for use with very short queries, or with queries that contain less than four or five shots. This observation led to the development of two content-based signatures: the *color-shift* signature, and the *centroid-based* signature. These are less compact than the shot-length signature, but include substantially more information, making them more suited to short queries and to clips that contain relatively few shots. The color-shift method uses color distributions in the video frames to produce a signature that represents the change in color in the clip over time. The centroid-based signature represents the spatial movement of the lightest and darkest pixels in each frame over time. It is simple to determine a priori when the shot-length method is likely to fail, so the centroid-based or color-shift signatures can be used in these situations.

Preliminary experiments with these new signatures determined that they are each suited to particular types of queries. The color-shift signature, for example, is (unsurprisingly) less effective for black-and-white content, while the centroid-based signature is affected by degradations affecting pixel luminance. While it is generally possible to predict which method is most likely to be successful, a method that is effective in all situations is desirable. To address this, we developed the *combined* signature. This signature uses evidence from both the color-shift and centroid-based signatures to produce a composite signature that exhibits many of the strengths of each of the constituent signatures.

3.1 Shot-Length Signature

The shot-length method exploits the observation that almost all videos are prepared manually from distinct shots, resulting in any given clip having a unique pattern of edit operations. While there is room for improvement in detecting more complex transitions, such as dissolves and wipes, standard techniques for cut detection are robust. By segmenting the video into shots using a reliable cut detection algorithm, we can determine the length of each shot. Both the query clip and the data (whether this is one clip or many) are processed with the same cut-detection algorithm to produce a signature for the content.

Modern cut-detection algorithms are relatively reliable, but errors are still made. For example, a common weakness of cut-detection algorithms is that they are sensitive to lighting changes and fast object or camera motion. However, while these examples undoubtedly represent errors in cut detection, they do not necessarily have a detrimental effect on the cut-based signature presented here. Intuitively, if, for example, a sudden flash of light causes a cut to be marked at a particular point in a video sequence, it is likely that the same event in a copy of this sequence will result in the same erroneous cut to be identified.

Essentially, what is required for accurate matching of video sequences is a series of *events* that can be reliably identified. A cut is an ideal event as it can be identified using fast, reliable algorithms. Cut detection is insensitive to changes in color, intensity, bit rate, and resolution. Existing cut-detection algorithms can reliably determine the position of a cut to within one frame (around 30 to 40 ms) [Lienhart 2001]. Conceivably, this slight inaccuracy could cause difficulties when matching sequences that are recorded at different frame rates, as the intervals between cuts may be slightly different. Approximate matching techniques, such as those described in the following section, are designed to avoid the problems caused by such inaccuracies.

We used a color-histogram-based cut-detection technique in the uncompressed domain in our experiments, with a dynamic threshold to reduce inaccuracies caused by changes in the amount of camera motion. The procedure used for producing the shot-length signature is as follows:

- (1) Process the video to detect cuts by
 - (a) decoding the video frames sequentially; we used a modified version of the mpeg2dec video decoder for this task;
 - (b) computing color histograms for each frame; to avoid colorspace conversion, we used a YCrCb histogram; 24 bins were used for the luminance channel, and 12 for each of the chrominance channels, resulting in a vector of 48 values in the histogram;
 - (c) computing histogram differences for each pair of adjacent frames; the Manhattan distance measure is ideal, due to its computational simplicity;
 - (d) comparing each histogram difference with an adaptive threshold to determine the presence of a cut.
- (2) When a cut is identified, count the number of frames since the previous cut.
- (3) Convert the number of frames to an elapsed time in milliseconds.
- (4) Append this value to the signature.

Typical broadcast video contains cuts at the rate of around 1200/h, so the data can be reduced to a signature of around 5 kB (uncompressed) for each hour of video. The following is an example of a typical shot-length signature, which represents a 60-s video clip using only 27 scalar values:

1480, 3920, 10880, 3080, 2240, 80, 1360, 80, 1240, 1200, 720, 600,
360, 360, 360, 280, 200, 160, 160, 1880, 15640, 840, 4120, 1480, 4520,
1920, 560.

Since the shot-length signature uses a small amount of information from the clips being compared, it is expected that the effectiveness of retrieval using this signature will be lower than more computationally expensive approaches, such as frame-by-frame comparison of visual features. It is likely, however, that the shot-length signature will be fast to search due to its compactness. The robustness of cut-detection algorithms also indicates that this signature type will be relatively insensitive to degradation of the video.

3.2 Color-Shift Signature

Previous attempts at coderivative identification have used direct comparison of color features between clips to determine similarity. The color information that is present in a video clip is a valuable feature for video matching, but this approach is sensitive to degradation and color changes between the clips, and is expensive to compute. The color-shift signature proposed here uses the color information in the video, but avoids direct comparison by using the change in color, rather than the color itself, to represent the video.

Using small samples of video, we analyzed a wide range of properties to identify which of them were stable under degradation. An example of this kind of analysis is shown in Figure 1. We observed that, while the absolute color values were subject to various forms of degradation, the magnitude of the change in color between frames in the same clip was much more robust. In our method, the change in color between two frames is represented as a single scalar value. We computed a color histogram for each frame, and calculated the distance from a similar histogram for the previous frame to produce each symbol in the color-shift representation.

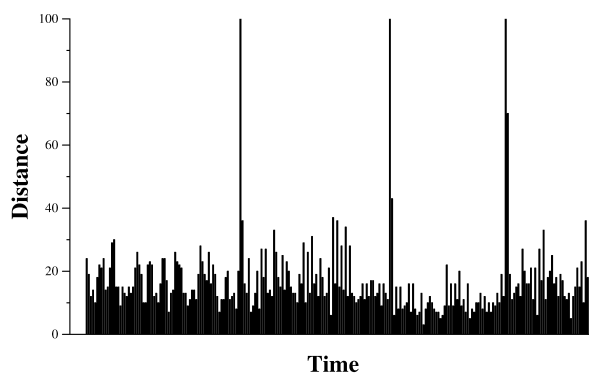
The first stage in producing the color-shift signature is to calculate a color histogram for each frame. We used 16 bins for each of the three color channels (luminance, red chrominance, and blue chrominance for YCrCb encoded video). While any number of bins can be used, in preliminary experiments we found that 8 or 16 bins gave the best results in practice. These histograms were then normalized according to the video resolution to allow accurate comparison of clips encoded at different resolutions.

To illustrate the process of computing the color-shift representation, we used a color histogram with four bins per channel. The resulting color histograms for an example sequence are shown in Table I.

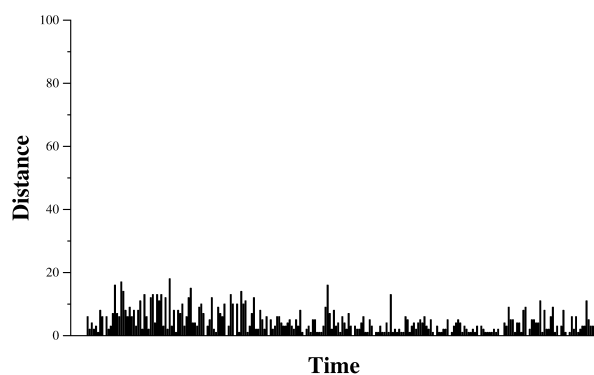
A distance measure was then used to calculate the change in color between adjacent frames. There are several potential measures. When comparing them, it is necessary to normalize the reported distances from each, to produce values in the same order of magnitude. This ensures that the scoring methods work equally well on all signatures. We normalized the values to fall in the range 0–200.

The intermediate stage of calculating the difference between the adjacent histograms for the example sequence is shown in Table I. For this example, we have used the Manhattan distance, producing the representation shown in the far-right column of Table I. To summarize, the method used to produce a color-shift signature is as follows:

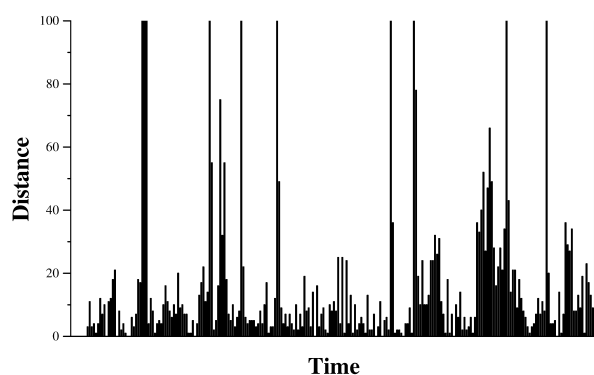
- (1) Decode the video frames sequentially as before, to produce color histograms in YCrCb colorspace.
- (2) Compute histogram differences for each pair of adjacent frames. There are many vector distance functions that can be used for this task. In Section 5 we compare effectiveness of Manhattan distance, Euclidean distance, histogram intersection, binwise histogram intersection, and chi-squar measures.
- (3) Append the histogram distance to the signature.



(a) Visualization of color-shift signature on sample clip



(b) Visualization of signature difference with degraded clip



(c) Visualization of signature difference with unrelated clip

Fig. 1. Color-shift signature differences. The color-shift signature for a 10-s video clip (a) shows several peaks. When the difference between symbol values is computed with a degraded version of the same clip, the high values are canceled out (b). In contrast, when compared with an unrelated clip, the peaks from the original signature, as well as new peaks from the second signature, can be observed (c).

Table I. Histogram Differences (YCrCb histograms are computed for the example video sequence and a vector distance function is used to compute histogram differences. The histogram differences for an example video sequence are shown here. The right-most column shows the final color-shift representation of the example video sequence, which consists of a single integer value for each frame in the clip.)

Frame no.	Difference histogram			Manhattan Distance
	Y	Cr	Cb	
0	[12, 4, 8, 18]	[3, 19, 16, 6]	[17, 14, 10, 1]	128
1	[2, 4, 2, 4]	[3, 1, 6, 2]	[5, 0, 2, 3]	43
2	[1, 1, 1, 3]	[1, 2, 1, 2]	[2, 1, 1, 0]	16
3	[2, 1, 2, 1]	[0, 2, 0, 2]	[1, 0, 2, 1]	14
4	[13, 3, 18, 2]	[6, 8, 1, 1]	[6, 2, 2, 6]	68
5	[0, 2, 3, 1]	[3, 2, 3, 2]	[1, 1, 1, 1]	20

This method is used to create representations for both the query and the data to be searched. The amount of storage required is around 180 kB/h of video data. The query is evaluated using methods described later. The following is an example of a color-shift signature for a typical 5-s query:

5, 2, 2, 3, 4, 0, 5, 0, 0, 118, 6, 5, 4, 7, 4, 2, 4, 2, 5, 9, 6, 7, 4, 3, 3, 2, 35,
3, 4, 2, 4, 4, 3, 5, 8, 7, 6, 5, 5, 4, 8, 8, 23, 12, 19, 27, 10, 26, 25, 26.

A visualization of a color-shift signature suggests that this representation is likely to be effective when identifying coderivative video content in practice. Figure 1(a) shows the color-shift signature for a short clip. The vertical axis shows the value of the symbols in the signature. Most of the symbols are in the 10–30 range, with a few peaks exceeding 40. The second graph in this figure shows the result of subtracting symbol values for a degraded version of the same clip. All of the high peaks in this graph are eliminated, with most symbol differences being in the range 5–10. The third graph shows the symbol differences when the original signature is compared with an unrelated clip. Peaks from both signatures are present in this graph, showing that the clips are significantly different. This demonstrates that the color-shift signature contains sufficient information to determine coderivation.

Since the color-shift signature retains substantially more information about the content of the clips being compared than the shot-length signature, it is likely that retrieval using this representation will be more effective. The color-shift signature, however, is less compact than the shot-length representation, so longer search times can be expected. Searching video using the color-shift signature is still more efficient than frame-by-frame comparison of feature vectors, though the latter method is likely to be more effective, especially when the content has not been degraded.

3.3 Centroid-Based Signature

Our centroid-based method produces a video representation based on a simple motion detection algorithm that attempts to estimate the movement of areas of luminosity from frame to frame. The underlying intuition is that motion estimation should be a viable representation of the video, as it is generally not



Fig. 2. Centroid motion. The centroids of luminance vary substantially over this sequence of three frames. The bright region in the lower-left balances the bright parts of the person’s face in the first frame, resulting in a light centroid positioned to the left of the face. The bright region disappears in the next frame, resulting in the light centroid shifting to the right. Another bright object appears in the final frame, causing the centroid to shift back again.

perceptibly affected by changes in video quality. Our expectation is that this should make it a robust method for evaluating similarity between degraded clips.

Existing motion estimation algorithms, however, are computationally expensive. Block motion estimation, for example, requires that each frame be segmented into small regions (blocks) and motion estimated for each block individually. In addition to computational cost, many existing algorithms are intended to represent either camera motion or object motion in the video. The algorithm introduced in this section represents overall motion in the clip, so both camera motion and object motion have an influence the signature produced.

The algorithm that we propose for estimating the motion between frames is not suitable for the range of applications that block motion estimation is used for, such as video compression. However, it is substantially less computationally expensive and is insensitive to video degradation—an important property for the application considered in this article. As well as being expensive to compute, preliminary investigation suggests that motion direction is ineffective as a feature for coderivative identification. For this reason, we use the magnitude of the motion vector, rather than the direction, to compute the centroid-based signature.

To produce the centroid-based signature that we propose, two motion vectors are calculated for each frame: one for the darkest pixels and the other for the lightest pixels (as determined by the luminance value). The centroid of each of these collections of pixels is determined, and the position of the centroid is compared to the position of the corresponding centroid in the previous frame. Figure 2 illustrates the motion vectors of the light pixel centroids over a sequence of three frames. In the first frame, the centroid is positioned to the left of the person’s face. Changes in luminance values in the second frame cause the centroid to shift to the right. Finally, in the third frame, the centroid moved toward the left of the frame again.

The locations of the dark and light centroids are determined by identifying the lightest and darkest 5% of pixels in each frame, and calculating an average of the x and y coordinates to produce the location of the centroid, which is then

Table II. Centroid Representation (Once the centroids are located, the magnitude of the movement for each centroid is computed. In this example the magnitudes are combined to produce the example the centroid-based signature for the video sequence.)

Frame no.	Dark Movement	Light Movement	Final Representation
0	0	0	0
1	12	11	23
2	6	8	14
3	4	8	12
4	208	56	264
5	9	7	16

normalized according to the size of the frame. Using a fixed number of pixels is undesirable, as the position of the centroid could change substantially when the resolution is altered. For each frame, the Euclidean distance from the previous position is calculated for each of the two centroids.

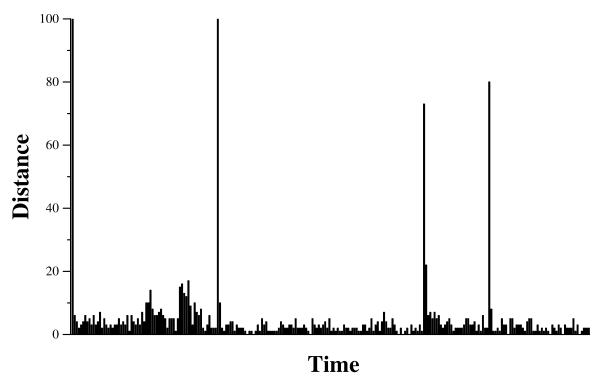
To summarize, the centroid-based video signature is produced as follows:

- (1) Decode the video frames sequentially. We used the same modified mpeg2dec video decoder as was used for the color-shift method.
- (2) For each frame, locate the bright centroid of luminance by
 - (a) identifying the pixels in the image that have the highest luminance levels; We limited the number of pixels used to 5% of the pixels in the image; this value was selected to ensure that the brightest pixels will still be included, even if the overall luminance of the image is shifted; a heap data structure alleviates the need for sorting the pixels for this task;
 - (b) locating the centroid of the most luminant pixels by computing the median horizontal and vertical position of these pixels.
- (3) Calculate the Euclidean distance from the bright centroid in the previous frame.
- (4) Normalize this distance according to the resolution of the image.
- (5) Locate the dark centroid by repeating this procedure using pixels with the lowest luminance levels.
- (6) Combine the distances using a simple function such as sum or product.
- (7) Append the combined value to the signature.

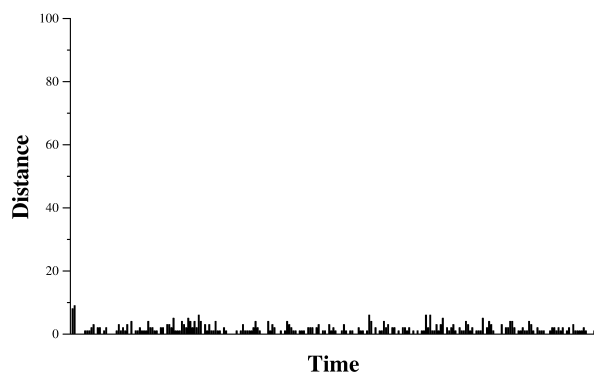
The final representation consists of a sequence of integers representing the distance these centroids have moved between adjacent frames, as shown in Table II. Using 2 B/frame, which is sufficient to represent the movement of the centroids, this representation requires about 180 kB/h of data. An example of a centroid-based signature is as follows.

7, 0, 9, 1, 6, 5, 4, 11, 1, 2, 73, 4, 3, 2, 4, 3, 2, 1, 1, 3, 8, 0, 2, 4, 2, 1, 2,
48, 2, 2, 3, 1, 4, 7, 4, 7, 2, 1, 3, 5, 11, 6, 3, 9, 14, 7, 14, 18, 23, 23, 31.

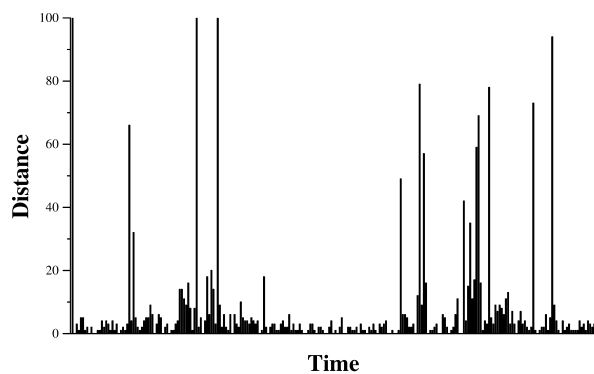
Figure 3 shows a similar visualization to that shown for the color-shift signature above. The first graph shows the centroid-based signature for a 10-s video



(a) Visualization of centroid motion on sample clip



(b) Visualization of signature difference with degraded clip



(c) Visualization of signature difference with unrelated clip

Fig. 3. Centroid signature differences. The centroid-based signature for a short clip (a), shows many low values, and a few high peaks in this visualisation. When the difference is computed between this signature and the signature for a degraded version of the same clip, the high peaks are eliminated (b). When compared with an unrelated clip, however, the symbol differences show many high values (c).

clip, which exhibits many low values and a few peaks. When compared with a degraded version of the same clip, these peaks are eliminated, leaving only small symbol difference values (b). When the original signature is compared with an unrelated clip, however, there are many high peaks, and a smaller number of very low values. This demonstrates that the centroid-based signature is suitable for coderivative detection, and is likely to be successful when automatic signature comparison methods are used.

Like the color-shift signature, the centroid-based signature contains one symbol for each frame in the video, so the efficiency of query evaluation using either method will be equivalent. The effectiveness of retrieval using the centroid-based signature is also likely to be equivalent, since both representations retain a similar amount of information about the clips being compared. Since neither of these representations rely on direct comparison of visual features, we expect that they will be less sensitive to degradation than previous methods, such as frame-by-frame comparison of color features, although the latter is likely to be substantially more effective for nondegraded content.

3.4 Combined Signature

The final signature type that we propose is a combined signature. Given that the color-shift and centroid methods use different properties of the video to produce a signature, it seems likely that these signatures will be effective in different circumstances. For example, given certain camera actions, such as dolly or rotation, the color distribution in the frame may remain relatively static, despite the camera movement; however, the centroids of luminance are likely to change. Similar situations exist that could have the opposite effect—footage of a moving object on a static background of similar luminance levels may cause minimal changes to the centroids of luminance, but the color balance could be affected.

By using evidence from both of these signature types, it is possible that a signature could be developed that has the strengths of both the color-shift and centroid signature types, without being affected substantially by their respective limitations. There are many ways that these signatures could be combined: products or sums of the symbols are two possibilities. However, we simply interleave the symbols produced by the two constituent signatures to produce the combined representation.

By combining the centroid-based and color-shift signatures in this way, a new signature is produced that uses two symbols to represent the change between each pair of frames. The combined signature does not distinguish between these two symbols—it would be possible, for example, for a symbol from the color-shift signature to be matched against a symbol from the centroid-based signature. Matches between symbols representing different features, however, are likely to be isolated. Color-derived symbols tend to be in different ranges to centroid-derived symbols for a given frame, so misalignment between these symbols is rare. The process of combining the signatures is illustrated in Figure 4.

This method of combining evidence has an overhead: the signature produced is double the length of the color-shift or centroid-based signatures. While this

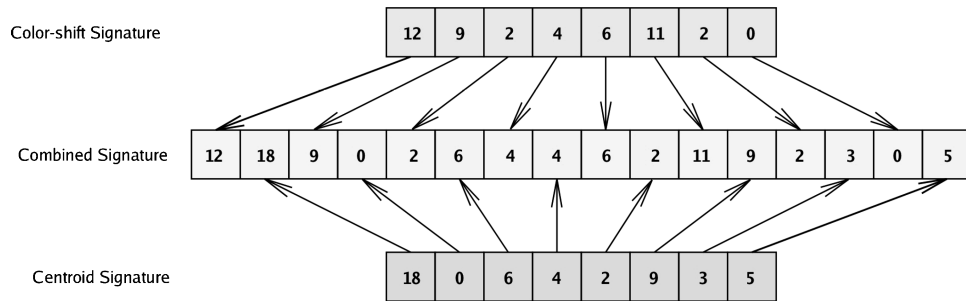


Fig. 4. Combined signature. The combined signature is created by interleaving symbol values from the color-shift and centroid-based signatures. This produces a signature that contains two integers for each frame in the sequence.

does not affect the complexity of the retrieval algorithm, twice as much data must be searched, resulting in greater query evaluation times. We expect, however, that the increase in efficacy will justify this additional cost.

3.5 Other Alternatives

As part of this research, other methods of this kind were explored, but were discarded after preliminary experiments. One of these was the centroid of pixels with luminance values closest to the average in each frame. In preliminary experiments we found this method to be expensive and we observed no improvement in effectiveness over the methods described above.

In all of the experiments using the centroid-based signature presented in this article, we used 5% of the pixels to compute the light and dark centroids. We also investigated variations of the centroid-based signature using 1%, 10%, and 20% of the pixels, but we found that these were less effective in preliminary tests, so they were disregarded. We propose the use of the Euclidean distance to determine the magnitude of the centroid movement between frames. We investigated other methods for calculating this distance, including the Manhattan distance, but found the Euclidean distance to be more promising, albeit by a small margin.

Another signature that we considered used the direction of the centroid movement rather than the magnitude, but we found this approach to be rather sensitive to changes in the video. For example, minor degradation of the video could cause the centroid to shift by a few pixels. If this centroid is close to the centroid in the previous frame, then the shift of a few pixels could substantially affect the direction of the motion vector, though the magnitude would change only slightly.

4. VIDEO SEARCH USING APPROXIMATE STRING MATCHING

Finding coderivatives using our video signatures requires a search method that is capable of efficiently comparing signatures to accurately locate similar sequences. In this section, we describe a novel technique for searching video signatures, based on an approximate string-matching method called *local*

alignment. Once signatures have been calculated for the data being searched and the query clip using one of the methods described in Section 3, this approximate string-matching technique can be used to align the query with segments of the data in the collection. Sections of video in the collection are ranked according to similarity to the query clip to allow the user to quickly identify the most similar parts of the collection, as well as giving an indication of the degree of similarity.

There are many difficulties in adapting local alignment to video search. *Dynamic programming*—the method we use for computing local alignment—has been applied to video search before, but previous adaptations of this technique have substantial limitations. Lienhart et al. [1997] described one approach for using dynamic programming to locate coderivative regions in video streams, but their proposal had two important weaknesses. First, the optimal alignment was found by computing a distance between every frame in the query and every frame in the data. This involved a computationally expensive vector distance calculation for each pair, making query evaluation costly: queries were reported to run in approximately real time. Second, similarity between clips was determined according to the number of exact matches between frame representations. Since color features are altered by processes such as transcoding and analogue transmission, exact matches are unlikely in clips that have been degraded, so the overall similarity of copies of a clip in different formats is likely to be low. Naphade et al. [2001] proposed a similar adaptation of dynamic programming, although their application was for semantic search rather than coderivative detection. This method also required exact matches between features for similarity to be identified, so it would likely be sensitive to degradation.

Another method was proposed by Adjeroh et al. [1998]. This addressed the efficiency problems in the above techniques by quantizing feature vectors into a discrete set of symbols to compute an approximate alignment. Thus, the number of symbol comparisons was limited to the number of features used in the representation. A more accurate alignment was then produced for candidate regions using a similar method to that described by Naphade et al. [2001]. While this method was likely to substantially reduce the cost of query evaluation, it still required exact matches between symbols for regions of similarity to be identified, so it was unlikely to be effective for identifying clips that have been degraded or transcoded.

To evaluate queries efficiently, we use the compact signatures described in the previous section to compute the similarity between clips. Since these signatures use a small number of symbols to represent a video sequence, query evaluation requires only a fraction of the number of operations of previous methods. In contrast to previous applications of dynamic programming, the methods presented in this section do not rely on exact matches between the symbols being compared. In early experiments we found that simplistic alignment using only exact matches was not successful. When video clips are subjected to slight degradation, such as by a minor change to bitrate, the visual features are altered.

4.1 Applying Approximate String Matching to Video Search

With each of the representations described in Section 3, approximate string-matching techniques can be used to locate parts of the collection that are similar to the query clip. The query clip is preprocessed in the same way as the collection, producing an equivalent representation of the data to be identified. Each of the representations described in the previous section produce a series of integer numbers representing the content of the video sequence. These sequences can be aligned using a variation of local alignment.

Local alignment is widely applied to string-matching problems, where the strings are comprised of symbols from a finite alphabet: similar sequences between the two strings are identified by comparing pairs of symbols to compute an edit distance between the strings. In contrast, the video representations that we describe consist of integer values that are uncapped. When applied to a finite alphabet, the edit distance is usually scored using a simple metric, where a match is awarded a positive score and a mismatch is awarded a negative score. In our application, a more flexible way of computing the edit distance is required, as nonidentical symbols may not represent a lack of similarity. For example, in string-matching, the symbol “A” is considered to bear no similarity to the symbol “B,” so a negative score would be awarded. On the other hand, when comparing centroid movements, a distance of 99, while not an exact match, is very similar to a distance of 100, so a positive score may be appropriate.

Alternative scoring systems can correct this limitation. We investigated several scoring systems intended to address the problems caused by nonidentical symbols. All of the systems awarded scores based on the absolute difference between the symbol in the query and the symbol in the data. With each scoring system, a positive score was given when the difference between the symbols being compared was small or zero, and a negative score was given when the difference was large.

4.2 Scoring Functions for Video Signature Matching

Key to the success of the alignment algorithm is the method used to determine an appropriate score, given a pair of symbols. An important property of a scoring function is the point at which a score of zero is awarded. When the difference between the symbols exceeds a given value, a negative score is awarded, and when the difference is less than this value, a positive score is given. We refer to this point as the *indecision point*. In order to manipulate the indecision point, the scoring functions include a scaling factor, k , that operates on the difference. We refer to this *scaled symbol difference* as δ :

$$\delta = k \cdot |s_q - s_d|. \quad (1)$$

The scaling factor, k , allows us to modify the indecision point without affecting the magnitude of the scores awarded by the scoring function, and we demonstrate the effects of doing this in Section 5.

4.2.1 Binary Scoring. The simplest scoring system presented here is binary scoring. In a traditional dynamic programming problem, a match between

symbols would be awarded a positive score, and a mismatch would be awarded a negative score. We awarded a score of 20 for a pair of matching entries in the video signature, and a score of -5 for a mismatch. These scores were selected to prevent individual mismatches from influencing the outcome of the scoring process.

This scoring system is intended as a baseline. Because no allowance is made for a near match, this scoring function is unlikely to be effective on material where either the query or the data being searched has undergone degradation. Degradation of the video will cause slight changes in the color-shift and centroid-based video signatures, which will prevent the matches from being detected. It would also be likely to fail when used to compare shot-length signatures of videos recorded at different frame rates, as misalignment by one or two frames would be common but should not prevent a match from being flagged. For example, in comparing video at 25 frames/s to video at 30 frames/s, some shots will occur at slightly different times.

An extension to the binary scoring system would be to award a fixed positive score for differences less than a given threshold, and a negative score for differences greater than a second threshold. This would result in more effective retrieval for nonidentical clips, however, the scoring functions described below are more likely to be effective, so we have not tested this variation.

4.2.2 Linear Scoring. The binary scoring system is based closely on the scoring methods used for many string-matching problems, in which symbols are considered to either be identical, or have no relation at all. In the signature data produced by the methods described in Section 3, however, this is not the case. The linear scoring system, and all of the other scoring systems described below, address this limitation by awarding a score that is proportional to the magnitude of the difference in the symbols in the signatures. This allows for some tolerance when the video has undergone changes and the signature varies slightly.

The linear scoring system uses a simple linear function to award a score based on the difference of the two symbols being compared:

$$S = 20 - \frac{3 \cdot \delta}{2}. \quad (2)$$

From the maximum score of 20, we subtract the scaled symbol difference, δ . The scaled symbol difference is multiplied by a factor of $\frac{3}{2}$, which alters the gradient of the function in order to set the indecision point at 12. The choice of 12 as the indecision point was somewhat arbitrary; it was made based on the results of preliminary experimentation on small data sets, which demonstrated this to be reasonably effective. It was expected that this scoring method would be more effective than binary scoring, especially when searching for nonidentical clips.

4.2.3 Categorical Scoring. The categorical scoring system is tailored for use with the shot-length signature. It awards high scores where the difference in shot length is less than one frame, lower scores for a difference of two or three frames, and a negative score where the difference exceeds five frames. A difference of three to four frames is considered indeterminate, so a score of

zero is used. This method is not likely to be effective with the color-shift or centroid-based signature data.

4.2.4 Cubic Scoring. The indecision point of a scoring function represents the point at which the similarity between two symbols is indeterminate—that is, it can be considered neither a strong match nor a strong mismatch. At this point, a score of zero is given. Conceptually, a symbol difference that is close to the indecision point is not a strong indication of overall similarity between clips. It is logical, therefore, that these differences should be awarded very low scores. Extending this intuition, the scores awarded should increase exponentially in both directions as the symbol difference moves away from the indecision point. These attributes can be realized by using a cubic function to determine the similarity score:

$$S = \frac{20 - (\delta - 10)^3}{50}. \quad (3)$$

This function gives a maximum possible score of 20 for an exact match, and a negative score for strong mismatches, while enabling the indeterminate range to be quite large, which is useful for this application. As with the logarithmic functions, the constants in this equation are included to ensure a maximum score of 20, and an indecision point of 12.

4.2.5 Mean-Weighted Scoring. The previous scoring techniques produce a score based on the absolute value of the symbols in the representation. For example, when comparing the symbols 50 and 100, all the previous scoring methods will consistently produce the same score, regardless of context. In practice, it is more useful to determine the score according to the nature of the content being examined. If the difference between a given pair of symbols is 50, then this would be significant when the majority of the differences are less than 10, but insignificant when the differences frequently exceed 500.

Where m_q is the mean value of all symbols in the query, the mean-weighted scoring system defines the score according to the function below:

$$S = \frac{2(2 - \delta)}{m_q}. \quad (4)$$

The outcome of this is that this scoring system is applicable to many different types of content. It is also likely to achieve good results with many different video signatures, as different representations use symbols of varying magnitudes.

5. MEASURING RETRIEVAL EFFECTIVENESS

In this section we explore the effectiveness of the proposed video matching methods with a series of experiments. The first experiment explored the new search methods introduced in the previous section using each of the signature types described in Section 3. We tested each of the scoring methods and investigated several other parameters of the search algorithm to determine which

variations are the most effective in practice, and to compare these results with a baseline that represents the current state of the art. These tests were conducted using a real-world dataset with a small selection of queries. We then present a second experiment in which we selected a large number of queries at random from a dataset and explored the effectiveness of the new methods under varying levels and kinds of signal degradation.

5.1 A Baseline for Comparison

In order to evaluate the effectiveness of the new methods, a point of reference is required. Section 2 describes the previous research in the area of video retrieval, and identification of coderivative video. In selecting an appropriate baseline, we first eliminated any methods that did not have the capabilities required for this problem. Specifically, some of the methods described previously are unable to identify a region of similarity in a long clip—that is, comparisons are performed on a whole-clip basis. These are unsuitable for the problem addressed in this article, so were disregarded.

Along with applicability to the task that we address, the other key consideration is efficacy. Thus it is appropriate to compare our new methods to one that demonstrates high retrieval effectiveness, possibly at the cost of efficiency. To reflect this, we used a baseline method that uses color-histogram comparison on a frame-by-frame basis, which we judged to be the most robust of the previous methods. This method is based on a combination of the ideas proposed by Lienhart et al. [1997], Naphade et al. [2000], and Hampapur and Bolle [2001]. We believe that is a plausible choice as it involves examining the video in detail—our signature methods discard a great deal of information—while being relatively insensitive to noise.

The technique used is as follows:

- (1) Preprocess the data clip by
 - (a) decoding each frame sequentially; we used a customised MPEG-2 decoder¹ for this task;
 - (b) computing a color histogram for the frame; we used a color histogram in YCrCb colorspace, with 16 bins for each of the three channels; the histogram was normalized according to the resolution of the image;
 - (c) writing the full histogram to a signature file; this allows the query to be repeated, or other queries to be run on the same data without requiring a full decode of the data clip.
- (2) Preprocess the query clip in the same manner, storing the signature in memory.
- (3) Read the data's signature file sequentially.
- (4) Create a sliding window over the data signature, with the size of the window slightly larger than the query. We used a window 130% of the size of the query, to allow matches to be identified even when the frame rate is altered.

¹The decoder used for this project was a modified version of mpeg2dec, which is part of the libmpeg2 project—an open source MPEG-2 video stream decoder library.

- (5) Use a distance function to compare histograms in the query signature to those in the sliding window.
- (6) Count the number of pairs where the distance is less than a fixed threshold. The threshold value was selected according to effectiveness in preliminary tests. This count was used as the similarity score for the current window.
- (7) Shift the sliding window forward by one frame and repeat the last two steps until the end of the signature file is reached.
- (8) Select locally maximal scores. This reduces redundancy in the results and emphasizes high-scoring regions.
- (9) Sort all window positions according to the number of similar histograms and output results. The output then consists of a ranked list of regions, with the strongest matches listed first.

This algorithm is computationally expensive. Step 5 of the search requires $|Q| \times |D| \times |H|$ integer comparisons, where $|Q|$ is the number of frames in the query, $|D|$ is the number of frames in the data, and $|H|$ is the number of bins in the image histograms. To alleviate this cost slightly, we considered only two frames from each second of the data clip. This reduced the computational cost by a factor of 10–15 (depending on the frame rate of the clip). Because we still used every frame from the query clip, the effectiveness was not noticeably affected.

5.2 Preliminary Experiments

In Section 3, we introduced four methods for building signature data for video matching: the color-shift method, the centroid method, the combined method, and the shot-length method. Several variants of the color-shift method and the centroid method were proposed, each of which produces a different signature, although based on the same properties of the video. In Section 4, we discussed several methods of calculating similarity scores that can be applied to the alignment algorithm proposed for matching regions of similarity.

The first group of experiments that we present used a small number of queries to explore the impact that these variables have on retrieval effectiveness. These experiments compared the scoring methods, signature variations, and indecision point. The results were then compared with the baseline method described above.

In order to quantify the effectiveness of the techniques and variables that are explored in this section, we use the measures *precision* and *recall*. These measures are used in the information retrieval domain to evaluate the efficacy of retrieval techniques [Baeza-Yates and Ribeiro-Neto 1999].

The results reported in this section showed average precision and recall scores over all of the queries. The precision was recorded after n results, where n was the number of correct matches for the given query that existed in the data set. The number of correct matches was determined by exhaustive human relevance assessment, which was possible due to the small size of the datasets used for this experiment. Recall was measured after a maximum of 20 results, or fewer in cases where fewer than 20 results were listed.

5.2.1 *Data Sets.* We used two sets of test data (that is, two data clips) in the preliminary experiments. We used a different set of queries for each data set. Video was recorded at a resolution of 352×288 pixels at the PAL frame rate of 25 frames/s (fps) and compressed using MPEG-1 at a bitrate of 1.2 Mb/s, unless otherwise noted.

—*Single query, multiple variants.* The first data set was a 170-min clip recorded from broadcast commercial television.² The content was a movie, *Star Wars: Episode I The Phantom Menace*, which was interleaved by 13 blocks of advertisements, with a total of around 120 commercials. The data clip was 1.6 GB when compressed using MPEG-1. The color-shift and centroid signatures consisted of 256,467 symbols, while the shot-length signature consisted of 1430 symbols.

We chose one advertisement to use as a query in this dataset, which was selected due to its being repeated five times over the duration of the clip. This data set was used to explore the robustness of the methods as the data was exposed to various methods of degradation. To represent this, we modified the query as follows:

- Query 1 was the original ad with no modification.
- Query 2 had increased brightness.
- Query 3 had increased contrast.
- Query 4 was recorded at a lower bitrate, 125 kb/s.
- Query 5 was converted to a the NTSC frame rate (29.97 fps).
- Query 6 had analogue noise added.
- Query 7 had increased color saturation.
- Query 8 had reduced brightness.
- Query 9 had reduced contrast.
- Query 10 had reduced color saturation.
- Query 11 was recorded at a very low resolution, 96×80 . pixels.

—*Multiple queries, single variant.* The second data set was a 180-min clip consisting of a stream of video recorded during prime time from commercial television. The content was comprised of a current affairs program and two dramas. This clip required 1.8 GB when compressed with MPEG-1, and the signature data was a little under 270,000 symbols using the color-shift and centroid methods, and 1571 symbols using the shot-length method.

We chose seven commercials from the clip to use as queries, four of which were 30 s long; the others were 15 s. Each query was chosen to represent a different style of content, and preference was given to advertisements that occurred more than once in the data set. There was a total of 15 occurrences of the seven queries in the data.

- Query 12 was a 15-s advertisement for a furniture retailer that represented a typical “sale” advertisement.
- Query 13 was for a food product and was delivered in a “documentary” style.

²Under Australian copyright law, free-to-air broadcast material can be recorded and stored for research purposes.

Table III. Scoring Methods (The categorical scoring system was the most effective in these tests, as indicated by the precision and recall scores shown here. This was probably due to the higher indecision point with this scoring function. The cubic and mean-weighted scoring functions are likely to be more effective in practice.)

Scoring Method	Color Shift		Centroid		Shot Length	
	$p(n)$	$r(20)$	$p(n)$	$r(20)$	$p(n)$	$r(20)$
Binary	0.19	0.30	0.29	0.47	0.60	0.75
Categorical	0.76	0.79	0.61	0.71	0.66	0.77
Linear	0.19	0.30	0.29	0.47	0.60	0.75
Cubic	0.59	0.71	0.52	0.65	0.60	0.75
Mean-weighted	0.66	0.71	0.65	0.74	0.59	0.67

- Query 14 was for another food product, and attempted to appeal to a consumer’s emotional needs.
- Query 15 was a short advertisement (15 s) for a fast food chain.
- Query 16 was for a car and was chosen for the fact that it contained only two edits.
- Query 17 was also for a car, and represented the “cinematic” style of advertising.
- Query 18 was a 15-s commercial for a department store, and was chosen for its visual similarity to other advertisements in the clip.

Experiments on the two data sets were independent—that is, queries from one dataset were not run on the other data set; however, the experimental results shown below are an aggregate of queries 1–11 on data set 1 and queries 12–18 on data set 2.

5.2.2 Scoring Methods. The first stage of experimentation compared the effectiveness of the scoring methods described in Section 4.2. For each query, we used an insertion and deletion penalty of 5, which effectively limits the penalty that is applied by a mismatch between symbols. Table III shows the results of these tests. The first column shows the average precision score for the 18 queries using each of the five variations of the color-shift signature. The second column shows the recall score for the same set of tests. The next four columns show precision and recall for the same queries, evaluated using the centroid and shot-length methods. The combined signature was not tested, as these experiments were intended to determine the optimal parameters for retrieval. We show results of experiments using the combined signature in Section 5.2.6 below.

The first line of Table III shows the results for the binary scoring system. As would be expected, the effectiveness of this scoring method was lacking, with an average of only a little more than 50% of the correct regions reported in the top 20 results, and just 30% in the top 20 when the color-shift signature was used. The categorical scoring system however, shows an unexpected result: this appears to be the most effective scoring system, with an average of 76% of the correct matches reported in the top 20. The other scoring systems, with the exception of binary scoring, have an indecision point of 12, while the categorical

scoring system has an indecision point of 120. This was due to the fact that this scoring system was designed to work with the shot-length signature to allow a tolerance of a few frames difference between shot lengths. Interestingly, the categorical scoring system was not substantially more effective with the shot-length signature than with binary scoring.

To understand what properties are desirable in a scoring function, it is useful to compare the effectiveness of the methods we tested. The linear scoring system was more tolerant to slight mismatches than the binary system, so it could be expected to be more effective; however these results indicate no improvement. As expected, the cubic scoring system was substantially more effective. This indicates that awarding high scores to identical matches, and dropping off sharply as the difference increases, results in substantially improved effectiveness. It also suggests that a wide range of near-zero scores near the indecision point is beneficial, and that the magnitude of the penalty should continue to rise as the symbol difference increases beyond the indecision point.

The mean-weighted scoring system performed very well, with an average of 71% of correct results identified in the top 20. The performance was particularly good with the centroid signature, where it was the most effective of all the scoring systems. The effectiveness was slightly lower with the color-shift signature, though the difference was marginal. The success of this method indicates that it is useful to consider the nature of the signatures being compared—if the symbols are generally small, then a small difference is significant, while if the symbols values are high, a small difference is inconsequential.

From the results of this experiment, it is difficult to draw a conclusion as to which scoring method is likely to be the best in practice. The categorical method appears to be the most effective, but it is likely that this was due to the higher indecision point. The effectiveness on this data set, however, cannot reasonably be dismissed. Binary scoring performed poorly, as was expected, and this trend is likely to continue with larger sets of data. Substantially more effective was the cubic scoring system. The mean-weighted scoring system was quite different from the rest, and this data demonstrates that it warrants further investigation. The remainder of the experiments with this data set were conducted using the three most promising scoring methods: categorical, cubic, and mean weighted.

5.2.3 Variations of the Color-Shift Method. To evaluate the comparative effectiveness of each of the variations of the color-shift signature, we executed all of the queries on their respective data sets, using categorical, cubic, and mean-weighted scoring methods. For all of these tests, we used an indecision point of 12, which was selected according to a cursory inspection of the distribution of symbols in typical data. We used an insertion and deletion penalty of 5, to prevent mismatches from dominating the scoring.

Table IV shows the effectiveness of the five variations. The Manhattan and Euclidean distance both yielded very strong results, with an average of 90% of the correct results listed in the top 20. The Euclidean measure showed slightly higher precision, meaning that the correct results were more likely to be listed in the top few. Histogram intersection and binwise histogram intersection showed comparable performance, which is not surprising, given that

Table IV. Variations on Color-Shift Method (Precision and recall scores for queries 1–18 using each variation of the color-shift signature. Manhattan distance and Euclidean distance are both extremely effective, while chi square performs poorly.)

Color-Shift Signature Variation	$p(n)$	$r(20)$
Manhattan distance	0.84	0.90
Euclidean distance	0.88	0.90
Histogram intersection	0.75	0.85
Binwise intersection	0.71	0.80
Chi square	0.17	0.22

Table V. Centroid Method Variations (Precision and recall scores for queries 1–18 using each variation of the centroid signature show that the light centroid is the most effective for these queries. It seems unlikely, however, that the light centroid would be any more effective than the dark centroid in wider tests.)

Centroid Signature Variation	$p(n)$	$r(20)$
Light Centroid	0.82	0.86
Dark Centroid	0.44	0.63
Centroid Sum	0.65	0.75
Centroid Difference	0.58	0.76
Centroid Product	0.50	0.46

these two measures are largely similar. These two measures, however, were unable to discriminate as strongly as Manhattan or Euclidean distance. Chi square showed very poor results, with an average of only 22% of the results being listed in the top 20, though the reasons for this are unclear.

Although Euclidean distance exhibited slightly higher precision than Manhattan distance, the difference was marginal, so we selected Manhattan distance for the remainder of our experiments due to the lower computation costs.

5.2.4 Variations of Centroid Method. We used the same method to compare the variations of the centroid signature as we used for the color-shift signature—that is, we ran all 18 queries using categorical, cubic, and mean-weighted scoring systems. We used insertion and deletion penalty of 5 and an indecision point of 12. The results of these tests are shown in Table V.

This shows that the magnitudes of both light and dark centroid vectors were effective discriminators, as were the sum and difference of the magnitudes. The product of the magnitudes performed very poorly, which was likely due to the fact that errors are amplified by the use of multiplication, while addition is more tolerant to small deviations. The centroid difference had roughly equivalent recall to the centroid sum, but the precision using this method was substantially lower.

Table V shows a marked difference in performance between dark and light centroid vectors, but this was probably due to the small sample and no conclusion should be drawn. Note that, while the sum of the two vectors sometimes

Table VI. Indecision Point for Cubic Scoring (Cubic scoring is extremely sensitive to indecision point. With an indecision point of 12, we observed that 82% of the correct results can be identified in the top 20 with the color-shift signature. Changing the indecision point to 9 results in only 2% of the correct results being identified. This is not substantially superior to selecting a point in the clip at random, which has approximately 0.2% chance of being correct.)

	Recall(20) at Indecision Point (cubic scoring)									
	3	6	9	12	15	18	25	50	75	100
Color shift	0.44	0.13	0.02	0.82	0.05	0.00	0.00	0.00	0.00	0.00
Centroid	0.51	0.05	0.02	0.61	0.00	0.00	0.00	0.00	0.00	0.00
Shot length	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.02	0.02

Table VII. Indecision Point for Mean-Weighted Scoring (Mean-weighted scoring is highly robust to changing indecision point; however, the optimal recall is achieved in the indecision point range 9–15 for all three signature types.)

	Recall(20) at Indecision Point (mean-weighted scoring)									
	3	6	9	12	15	18	25	50	75	90
Color shift	0.32	0.94	0.96	0.96	0.96	0.94	0.93	0.91	0.83	0.80
Centroid	0.34	0.69	0.84	0.84	0.83	0.83	0.80	0.77	0.65	0.64
Shot length	0.63	0.67	0.67	0.67	0.67	0.67	0.63	0.62	0.68	0.62

yielded a less discriminatory result than one of the vectors individually, it almost always gave a result at least as good as the weaker of the two individual vectors. Despite the sum producing a slightly lower average than the light vector on its own, this observation suggested that it was likely to be more robust, so we selected this variation for the remainder of our experiments.

5.2.5 Indecision Point. The final variable that we explored in this series of experiments was the indecision point. We used an indel penalty of 5 for all 18 queries. Extensive preliminary experimentation demonstrated that this value gave the most accurate results. We ran each query with all three signature types using cubic and mean-weighted scoring, which we present separately.

Table VI shows the recall results for queries using the cubic scoring system. This scoring system is extremely sensitive to varying the indecision point. Using an indecision point of 12, we were able to achieve an average of 82% of the queries in the top 20. Changing this value by 3 in either direction had a catastrophic effect on retrieval accuracy—with the indecision point set at 15 for the color-shift signature, only 5% of the results were correctly identified on average, while, with the indecision point set at 9, only 2% were identified.

In contrast, the mean-weighted scoring system is insensitive to changes in indecision point. The recall results for mean-weighted scoring are shown in Table VII. The best accuracy was achieved using an indecision point between 9 and 12 for all three signature types; however, acceptable results were still achieved with the indecision point as low as 6 and as high as 50. Similar patterns were observed about query precision: the highest precision was achieved in the same range, 9–15. The insensitivity to changing indecision point may not appear to be a useful attribute for a scoring system, as the indecision point

Table VIII. Comparison of Effectiveness
 (When comparing precision and recall of all five methods, we observe that all of the new methods performed better than the baseline, with the combined method displaying the strongest result.)

Retrieval Method	$p(n)$	$r(20)$
Baseline	0.55	0.64
Color shift	0.94	0.96
Centroid	0.72	0.84
Shot length	0.59	0.67
Combined	0.99	1.00

can be set arbitrarily when executing a query. While this is true, the fact that mean-weighted scoring is not substantially affected by using a suboptimal indecision point is an indication that it is likely to also be robust to different types of data and different kinds of degradations and edits.

By comparing the results of the cubic and mean-weighted scoring systems, we can observe that, given optimal selection of variables, the mean-weighted scoring system achieved substantially higher accuracy than cubic scoring. For this reason, as well as its robustness to differing indecision points, we used mean-weighted scoring for the remainder of the experiments in this section.

5.2.6 Evaluation of Effectiveness. The previous sections explored variations of the signature types and alignment techniques that are proposed in this article, but the question of how these methods compare with the current state of the art has not yet been addressed. In Section 5.1 we described the method that we used as a baseline for comparing these new techniques with current approaches.

In order to draw a comparison with this baseline, we first selected the appropriate variations of signature types and scoring systems that we found to be most effective in previous sections. We used these parameters to execute each of the queries on the two datasets. We compared these results with the results of executing the same queries with the baseline method. We also compared these results with those achieved using the combined signature type, using the same parameters as were found to be successful with the color-shift and centroid methods. A summary of these results is shown in Table VIII. The baseline method identified approximately two-thirds of the correct answers in the top 20 on average. The shot-length method had comparable results. The centroid method performed much better than the baseline and shot-length methods, and the color-shift method was even more effective, with an average of 96% of correct results listed in the top 20, and 94% of the top- n results being correct. The combined signature method was substantially more effective than either the color-shift or centroid methods. It correctly identified every instance of all 18 queries in the top 20, with all but one of these being listed in the top n .

In order to gain a more in-depth understanding of the comparative strength of these methods, we used three additional measures for comparing retrieval effectiveness: *highest false match* (HFM), *separation*, and *separation/HFM ratio* [Hoad and Zobel 2003b]. These measures are intended to compare retrieval

methods where the precision and recall are close to 1.0. Ideally, a retrieval system should not only find the correct matches, it should also be able to automatically differentiate between correct and incorrect matches. These new measures attempted to encapsulate the likelihood that a retrieval method could succeed in this task.

The HFM measure reports the highest score, expressed as a percentage, given to an incorrect match in the result ranking. The separation is the difference between the HFM and the lowest score given to a correct match in the result ranking (the *lowest correct match* or LCM). Note that when precision and recall are both 1.0 the separation will be positive, and, when precision is less than 1.0, the separation will be negative. If the recall(m), where m is the total number of answers reported by the retrieval system, is less than 1.0, then separation cannot be accurately calculated, as the LCM is unknown. In cases such as this, it is assumed that the LCM is some arbitrarily low score, such as 5%. The third measure, the separation/HFM ratio (or simply ratio), is the quotient of the separation and HFM. When this score exceeds 1.0, it is likely that a simple filter could be used to discard all incorrect matches, while still identifying all correct matches.

Further analysis of the results using these measures yields interesting results. The centroid and color-shift representations had a strong average HFM-to-separation ratio, which was consistent with the high-precision and recall scores discussed earlier. Although the average performance of these two methods was comparable, the color-shift representation was much more consistent, with only query 10 causing significant difficulties. This is not surprising, given that the color saturation in query 10 was reduced markedly, which would not affect the luminance part of the color histogram but would have a dramatic effect on the chrominance channels. The color-shift method would fail to identify, for example, a black-and-white version of a color movie. The most effective method, however, was the combined signature. This method had the highest average separation/HFM ratio and in many cases it was more effective than either the color-shift or centroid methods.

Query 1 was easily identified by four of the five methods. The shot-length method identified four of the five occurrences, but the fifth was missed entirely. Further investigation revealed that this was due to a cut-detection error in the data clip. Occasional cut-detection errors are likely to have a noticeable effect on retrieval accuracy with the shot-length method when the query is short, as a single mismatch contributes substantially to the alignment score when the signature is only a few symbols long, as was the case with this query.

Queries 4 and 11 demonstrated the effect of dramatic reductions in bitrate or resolution. These degradations had an effect on cut-detection accuracy, and hence the effectiveness of the shot-length method was reduced. This was likely to be less problematic with longer queries. The other methods were able to cope with these degradations. Interestingly, the color-shift and combined methods achieved higher ratios with the low-resolution version (query 11) than the nondegraded version (query 1), although the reason for this is unclear.

The effect of changes in frame rate was shown by query 5. While the ratio was reduced substantially for the baseline and shot length methods, they were still

able to list all the correct matches first. Early experiments with the shot-length methods were unsuccessful at identifying the correct results for query 5, due to slight mismatches in shot length caused by changing the frame rate; however, this result indicates that this shortcoming has been resolved.

The presence of analogue noise, as in query 6, did not affect any of the new methods; however, the effectiveness of the baseline method was substantially reduced. Queries 2, 3, and 7–10, in which the color saturation, brightness, and contrast had been altered, did not present significant problems for the new methods, with the exception of the centroid method in the presence of changes to brightness. This is intuitive, as changes to the brightness will have an effect on the centroids of luminance. The increase in contrast in query 3 appeared to have a positive impact on the combined signature method: for this query, the lowest correct match was more than seven times higher than the highest false match. The baseline method was unable to cope with any of these degradations.

Query 12 presented difficulties for all of the methods, with the color-shift and combined methods being the only ones that identified all the correct matches first, albeit with very low separation/HFM ratios. The reason for this is unclear, as the sequence appears quite unique to a human viewer. Query 15 caused some problems for the shot-length and baseline methods, which was likely due to the length of the query. While most of the queries were 30 s long, query 15 was only 15 s. This was related to the problems caused by query 16. This query caused difficulties to the cut-based method due to the small number of cuts, but also presented problems to the color-shift and centroid methods. In addition to the small number of cuts, this query displayed a very static video sequence, with no camera movement, and very little object movement. It was only with the baseline method, which compares the color properties directly, that this advertisement could be easily identified. The color-shift and combined methods identified this query correctly, but with very low ratios. Similarly, query 18 presented problems when using the shot-length method, as it also contains only a few cuts. The baseline method had difficulty with this query as well, but this was due to the fact that there was another advertisement in the data which was visually similar to this query (though advertising a different company). In contrast, query 17 displayed several cuts and a wide range of both camera and object motion. All five methods were able to identify this query.

From these preliminary experiments, it is evident that all of the methods proposed are useful for identifying coderivative video data. It is also clear that each of the methods has some shortcomings. The shot-length method, for example, is unreliable when the query clip contains insufficient cuts; experimental results suggest that five or six cuts are required to achieve good accuracy. The color-shift and centroid methods, on the other hand, are inherently insensitive to the number of edits in the query. These methods are significantly more expensive, in terms of storage requirements and query evaluation time, than the cut-based method, but provide superior differentiation between correct and incorrect results. The color-shift method is very robust, being sensitive only to dramatic changes in color information in the query. The centroid method is also robust, but because this method relies on identifying the centroids according

to the luminance value of the pixels, it is sensitive to changes to these values. The coupling of these two signatures results in the most robust and effective method. In all but one case, the combined signature performed at least as well as the lesser of the two constituent signatures, and in many cases the effectiveness was substantially better than either of them. The baseline method, too, has strengths. Each of the other methods relies, in some way, on the change in the video over time. As a result, they tend to yield inferior effectiveness when searching for clips which contain very little motion.

These preliminary experiments, however, were limited, both in the breadth of queries and the size of the dataset. Section 5.3 investigates how these methods performed on a large range of queries, subject to real-world degradations.

5.3 Robustness Experiments

The preliminary experiments enabled us to select scoring systems and other parameters that are likely to result in effective retrieval. In the following section, we present the results of a series of experiments intended to assess performance of these methods in an environment that is consistent with the realities of digital video. To achieve this, we experimented with queries of varying lengths, and with degradations that are likely to be observed in practice. These degradations include transcoding errors and changes to bitrate, frame rate, and resolution—all of which are common in digital video. We compare all four of the proposed signature types in this experiment: shot-length, color-shift, centroid, and combined.

For the dataset in this experiment, we used a 141-min clip, recorded from a commercial television station during prime time. The content of this dataset included two sit-coms, which constituted approximately 1 h of the clip, and a talk show, which made up the rest. The talk show included, among other things, two live music performances and several interviews. This content was interleaved with numerous advertisements and two news bulletins. The dataset was encoded in MPEG-1 at CIF resolution (352×288 pixels). Since the data source was in PAL format, we retained the PAL frame rate of 25 fps for the dataset. We encoded the data at 1200 kb/s—the data rate used for VCD encoding.

We selected 450 clips at random from the data to use as queries. This included equal numbers of clips of 5, 10, 20, 30, 40, 50, 60, 90, and 120 s in length. By sourcing the query clips from the dataset, we could be certain that every query had at least one correct match in the data, and the position in the data from which the clip was taken could be used as an “automatic” relevance judgment. It is possible that some of the queries that we selected were from sequences repeated in the dataset: that is, the query occurred more than once. Because human relevance assessment in this experiment was prohibitively expensive, we were unable to identify these situations. We continued on the assumption that each query occurred exactly once in the dataset, so it was possible that a match that would be judged correct by a human observer would be judged incorrect by this system. It was unlikely, however, that more than a few of the selected queries actually occurred more than once, so the overall results should not be substantially affected.

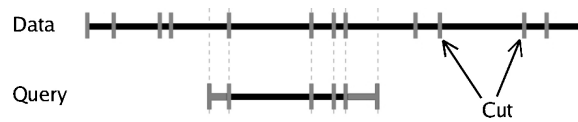


Fig. 5. Shot misalignment. Randomly selected start and end points for queries extracted from a data clip will result in a mismatch of shot lengths for the first and last shots in the query. It is likely that this will have a detrimental impact on retrieval effectiveness.

For each result reported by the search functions, a start-time and score were listed by the software. We considered a result to be a correct match if it had any overlap with the region from which the query was extracted, with a small tolerance, t . For these experiments, we used a tolerance of 1 s. Thus, given a query of length l and the start time s of a matching sequence in the dataset, we considered a match reported to begin at time m to be correct if $s - l - t < m < s + l + t$. While this is a fairly broad tolerance, we found that when a correct match was reported, it was usually much more precise. This is discussed further in Section 5.3.6 below.

It is possible that random selection of queries will present problems for the shot-length method, as it relies on accurate measurement of the intervals between edits in a clip. The randomly selected start and end points of the query are unlikely to occur at shot boundaries, which will result in the query beginning and ending with a fragment of a shot, as illustrated in Figure 5. To assess the impact of this, for half of the queries, we aligned the beginning and end of the clip to the nearest shot-boundary before extracting the query from the data file. In cases where this altered the length of the query by more than 5 s, the query was discarded and another selected at random. After alignment, the queries ranged in length from 3.84 to 124.40 s.

To extract queries from the source, the selected clip was decoded, then reencoded using `mpeg2enc`—an open-source MPEG-2 encoder. The extracted queries were reencoded using the same parameters as the dataset—1200 kb/s at 25 fps and a resolution of 352×288 . Degradation of query files involved reencoding the extracted query using different parameters, again using `mpeg2enc`. When degrading the query, all other parameters were unchanged. For example, when reencoding video at 176×144 , the bitrate remained at 1200 kb/s.

5.3.1 Sensitivity to Change in Bitrate. Some of the most common forms of degradation in digital video are caused by a reduction in the bitrate used to encode the clip. The use of different bitrates to encode a single video clip is frequently seen in cinema content. Content is often captured at 80 Mb/s or more, then reencoded at around 40 Mb/s for digital cinema projection, 4–6 Mb/s for DVD, and 2–3 Mb/s for digital television. For unauthorized distribution, it is often captured and transcoded at about 1.5 Mb/s for VCD, then even lower for distribution on the internet—400–600 kb/s is not uncommon.

It is important, therefore, that any tool designed to identify copies of a piece of content is still effective when the bitrate has been reduced. To assess the effectiveness of our new methods when the bitrate was reduced, we transcoded the extracted queries using five different bitrates and used them to search the

Table IX. Color-Shift Signature (The results of the experiments using the color-shift signature showed that the best results are achieved when the data is not heavily degraded. The table shows the results for each query length under each level of degradation. The first number in each triplet is the proportion of queries for which the correct result was identified as the first match in the ranked list. The second and third numbers show the proportion of queries for which the correct result was listed in the top 3 and top 10, respectively. For example, the first triplet in the first row shows that the correct result was listed first for 78% of the queries, while the correct region was listed in the top 3 for 78% of the queries, and in the top 10 for 84%.)

Query Length	1200 kb/s	600 kb/s	300 kb/s	100 kb/s
	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10
5	0.78/0.78/0.84	0.48/0.52/0.52	0.28/0.40/0.46	0.28/0.40/0.44
30	0.98/1.00/1.00	0.86/0.92/0.94	0.68/0.78/0.84	0.62/0.74/0.84
120	0.98/1.00/1.00	0.92/0.94/0.98	0.90/0.92/0.94	0.86/0.92/0.94
Total	0.96/0.97/0.98	0.82/0.85/0.88	0.65/0.73/0.80	0.62/0.70/0.77

nondegraded data clip. After extracting the query clip from the data, we encoded it at 1200 kb/s. We then transcoded this clip at 600, 300, and 100 kb/s. Using the MPEG-1 codec, video encoded at less than 600 kb/s is approaching unwatchable. The use of very low bitrates in this experiment allowed us to explore the limits of the methods that we tested.

Table IX shows the results of all the queries on this dataset using the color-shift signature. Each triplet shows the proportion of queries for which the correct region was identified in the top 1, top 3, and top 10 ranked results. The first column shows the results with query clips encoded at 1200 kb/s—the same bitrate as was used in the data clip. Even for very short queries, the color-shift method identified the relevant region of the data clip as the highest-ranked result more than three-quarters of the time, and in the top ten 84% of the time. As should be expected, the accuracy of this method increased proportionally to the query length. For queries longer than 30 s, the correct region was listed in the top 3 results for every query, and almost always as the highest-ranked match. Of all 450 queries, the color-shift method listed the correct match first 96% of the time. In 11 of the 450 queries, the correct match was not listed in the top 10. Only two of these cases, however, failed to list the correct result in the top 100—both of these queries were 5 s long, and the correct results were listed at positions 161 and 731.

Retrieval effectiveness suffers as the bitrate is reduced. Even at an extremely low bitrate, however, the correct result was identified first 62% of the time, and in the top 10 more than three-quarters of the time. Regardless of the bitrate, the likelihood of finding a correct match increased with a longer query. It seems probable that the correct region could be identified almost all of the time for queries several minutes in length, even at extremely low bitrates.

Similar patterns can be seen when using the centroid-based signature on the same queries. Table X shows the results of queries on this dataset using the centroid-based signature. For very short queries that were not substantially degraded, the centroid-based signature method identified 94% of the correct regions as the highest-ranked result. Effectiveness improved for longer queries, and all queries longer than 30 s were correctly identified as the highest-ranked

Table X. Centroid-Based Signature (The results of queries using the centroid-based signature exhibit similar trends to those observed using the color-shift signature. This table shows the proportion of correct regions listed in the top 1, top 3, and top 10 ranked results. The correct region was listed as the first result for 98% of queries when the degradation was minimal. As the degradation become stronger, the retrieval effectiveness dropped—at 100 kb/s, 71% of queries had the correct match listed first.)

Query	1200 kb/s	600 kb/s	300 kb/s	100 kb/s
Length	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10
5	0.94/0.96/0.96	0.40/0.54/0.66	0.34/0.48/0.58	0.32/0.44/0.56
30	1.00/1.00/1.00	1.00/1.00/1.00	0.90/0.96/0.96	0.86/0.92/0.96
120	1.00/1.00/1.00	1.00/1.00/1.00	0.94/0.94/0.96	0.82/0.86/0.96
Total	0.98/0.99/0.99	0.86/0.90/0.93	0.79/0.83/0.87	0.71/0.77/0.82

Table XI. Shot-Length Signature (Using the shot-length signature on the small dataset was less effective than the color-shift or centroid-based methods. The proportions of queries for which the correct region was listed in the top 1, top 3, and top 10 ranked results using the shot-length signature are shown in this table. On queries that had not been degraded substantially, the shot-length signature was able to correctly identify only 14% of the 5-s queries as the strongest match, but for longer queries, 82% were correctly identified. The shot-length method, however, appeared to be insensitive to degradation: 64% of the queries were listed in the top 10 with only minor degradation and 63% were still listed in the top 10 with the lowest bitrate.)

Query	1200 kb/s	600 kb/s	300 kb/s	100 kb/s
Length	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10
5	0.14/0.20/0.24	0.12/0.20/0.24	0.12/0.20/0.24	0.12/0.20/0.24
30	0.60/0.68/0.76	0.56/0.66/0.74	0.52/0.64/0.72	0.52/0.64/0.72
120	0.82/0.82/0.84	0.80/0.80/0.82	0.84/0.84/0.84	0.82/0.82/0.84
Total	0.55/0.60/0.64	0.54/0.60/0.64	0.53/0.58/0.62	0.53/0.58/0.63

result. Over queries of all lengths, the centroid-based method identified the correct region as the first result 98% of the time. This method failed to identify the correct result in the top 10 in only five of the 450 queries. In four of these cases, the correct match was listed in the top 30, and the fifth was listed at 129.

Table XI shows the results using the shot-length signature on the same 450 queries. At first glance, these results seem much less satisfactory than the results achieved using the color-shift or centroid-based signatures. Only 24% of the 5-s queries identified the correct region in the top 10, and only 14% as the best match. These scores improved for longer queries—for 2-min queries, 84% were identified in the top 10, and in almost all of these cases the best match was listed first. These results are still well short of those achieved using the other two signature types, but this is to be expected. A typical shot-length signature for a 2-min query contains around 20 symbols, while the equivalent color-shift or centroid-based signature contains about 3000 symbols. Given a query containing a similar number of symbols (in typical broadcast video, this would be about 2.5 h long), it is likely that the shot-length method would achieve a level of effectiveness at least as high as either of the other methods.

Table XII. Combined Signature (Using the combined signature type resulted in similar results as the other methods. Effectiveness improved for longer queries, but dropped as the query bitrate was reduced.)

Query Length	1200 kb/s	600 kb/s	300 kb/s	100 kb/s
	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10
5	1.00/1.00/1.00	0.54/0.66/0.68	0.44/0.56/0.60	0.40/0.50/0.58
30	1.00/1.00/1.00	0.98/0.98/0.98	0.82/0.92/0.96	0.82/0.86/0.92
120	1.00/1.00/1.00	0.94/0.98/0.98	0.92/0.98/0.98	0.92/0.98/0.98
Total	1.00/1.00/1.00	0.89/0.92/0.93	0.82/0.87/0.90	0.77/0.81/0.86

Another interesting observation is that the shot-length method did not appear to be substantially affected by changes to the encoded bitrate. For 2-min queries, the results were identical for the 1200 kb/s and 100 kb/s queries. Overall, 55% of the queries resulted in the correct region being listed first when minimal degradation had occurred. When reduced to 100 kb/s, the shot-length signature method was still able to identify 53% of the correct results first.

The final signature type that we used for these queries was the combined signature. The results of these queries are shown in Table XII. As with the other three signature types, the retrieval effectiveness with this signature type was higher for longer queries, and lower for more heavily degraded queries. Almost all of the queries were identified as the strongest match when the degradation was minimal. Again, retrieval effectiveness suffered as the bitrate was reduced; however, even at the lowest bitrate, 98% of 2-min queries were still listed in the top 3 results.

5.3.2 Sensitivity to Change in Resolution. Another property that is frequently changed when the content is reencoded is the resolution. Digital cinema, for example, is often encoded using the $2k$ resolution (2048×1556 pixels); HDTV can be encoded at 1920×1080 or 1280×720 pixels; VCD uses 352×288 pixels³; and Internet streaming often uses even lower resolutions. A reduction in resolution is usually performed to conserve bandwidth, so is generally associated with a reduced bitrate. To isolate resolution as a form of degradation, however, we used the same bitrate (1.2 Mb/s) for all resolutions.

The original video was encoded at CIF resolution (352×288). To evaluate the effect of lowering the resolution, we reencoded the queries using QCIF (176×144), SQCIF (128×96), and a very low resolution (64×48 —a little larger than a desktop icon). The lowest of these resolutions is rarely used in practice; however, we included it to test the limits of accurate retrieval.

Table XIII shows the results of all 450 queries on the small dataset with the queries encoded at each of these resolutions. The first column shows the results with queries at the original resolution. The second column shows results at QCIF resolution. All of the methods showed some reduction in effectiveness at this resolution, but the difference was quite small. The third column, using SQCIF, shows that three of the methods were affected by this further reduction in resolution (which contained less than one-eighth of the number of pixels as

³The quoted resolution is for PAL VCD. NTSC uses a slightly lower resolution— 352×240 .

Table XIII. Sensitivity to Resolution Change (As would be expected, the retrieval effectiveness for each of the methods is lower when the query is more severely degraded. The shot-length signature method is less susceptible to changes in query resolution than are the other methods.)

Signature Type	352×288	176×144	128×96	64×48
	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10
Color shift	0.96/0.97/0.98	0.92/0.94/0.96	0.89/0.91/0.93	0.68/0.73/0.79
Centroid	0.98/0.99/0.99	0.77/0.82/0.86	0.57/0.62/0.70	0.37/0.42/0.47
Shot length	0.55/0.60/0.64	0.54/0.58/0.63	0.52/0.57/0.62	0.48/0.52/0.56
Combined	1.00/1.00/1.00	0.94/0.96/0.97	0.88/0.90/0.93	0.58/0.67/0.74

Table XIV. Sensitivity to Frame Rate Change (Changes in frame rate have a noticeable impact on retrieval effectiveness. The color-shift, centroid, and combined methods appear to be affected fairly equally regardless of the frame rate chosen. While the shot-length method is not sensitive to changes in bitrate or resolution, changes to the frame rate have a substantial impact.)

Signature Type	25 fps	24 fps	29.97 fps	30 fps
	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10	Top 1 / 3 / 10
Color shift	0.96/0.97/0.98	0.90/0.91/0.94	0.88/0.90/0.93	0.88/0.90/0.92
Centroid	0.98/0.99/0.99	0.94/0.95/0.96	0.92/0.92/0.94	0.91/0.94/0.95
Shot length	0.55/0.60/0.64	0.38/0.46/0.53	0.30/0.36/0.46	0.30/0.36/0.46
Combined	1.00/1.00/1.00	0.95/0.96/0.96	0.94/0.95/0.96	0.94/0.95/0.96

the original). The shot-length method does not appear to have been substantially affected by this change. A further reduction in resolution, shown in the final column, had a dramatic effect on the effectiveness of the color-shift, centroid, and combined signature methods, but, again, the shot-length method was affected only slightly.

5.3.3 Sensitivity to Change in Frame Rate. Frame rate is another property that is often changed when video is reencoded. The effect of changing the frame rate is not as noticeable to a human viewer as changes to bitrate or resolution; however, it has the potential to affect signature alignments substantially, by introducing repeated frames into the video sequence, which will cause insertions in the signature alignment. Changes to the frame rate will also affect the shot-length method: altering the frame rate is likely to change the length of a shot. In most cases, the change will be a few tens of milliseconds; however, this is enough to result in an inexact match rather than an exact match.

The results of these tests are shown in Table XIV. Changes to the frame rate have a noticeable effect on retrieval accuracy for all four signature types. With the color-shift method, for example, the proportion of correct results in the top 10 dropped from 98% to 94% when the frame rate was changed from 25 fps to 24 fps. Changing the frame rate to 29.97 fps had a more noticeable impact, but the result was almost identical to that achieved using 30 fps. The shot-length method was found to be very resilient to changes in bitrate and resolution, but changes to frame rate had a much stronger effect. The hit rate dropped from 64% in the top 10 to 53% when the query was reencoded at 24 fps, and to 46% using 29.97 or 30 fps. It is unlikely that these changes affected the

Table XV. Effects of Query Alignment on Shot-Length Method (Aligning queries to the nearest shot boundary had a positive effect on all of the signature types, but the effect was most noticeable with the shot-length signature. Each pair of values in this table shows the proportion of correct results listed as the highest-ranked match for nonaligned and aligned queries at different bitrates.)

Query Length	1200 kb/s	600 kb/s	300 kb/s	100 kb/s
	Random/Aligned	Random/Aligned	Random/Aligned	Random/Aligned
5	0.04 / 0.24	0.04 / 0.20	0.04 / 0.20	0.04 / 0.20
30	0.44 / 0.76	0.36 / 0.76	0.32 / 0.72	0.32 / 0.72
120	0.68 / 0.96	0.68 / 0.92	0.68 / 1.00	0.68 / 0.96
Total	0.33 / 0.77	0.33 / 0.75	0.32 / 0.75	0.31 / 0.74

accuracy of the cut-detection algorithm, but the length of the detected shots would have been slightly different after changing the frame rate. This would result in lower scores being awarded to matching shots, resulting in less discrimination between correct and incorrect matches.

5.3.4 Impact of Query Alignment. The results presented in the above section suggest that the shot-length method is substantially inferior in retrieval accuracy to the other three signature types. There are two likely explanations for this observation. First, the shot-length signature uses much less information to determine coderivation, which results in poor effectiveness for very short queries. Second, random selection of queries results in truncated shots at the beginning and end of each query, causing mismatching symbols in the signature.

To determine whether this is a significant limiting factor, we aligned half of the queries to the nearest shot boundary as described in Section 5.3, so that the query would contain only whole shots. Table XV compares the results of each of these two sets of queries (nonaligned and aligned). Each pair of results in the table shows the proportion of queries for which the correct match was ranked highest for nonaligned and aligned queries. The table shows the results for each bitrate used in our experiments to determine what effect query alignment has at different levels of degradation. For example, with 5-s queries encoded at 1200 kb/s, 4% of the correct matches were listed first for the nonaligned queries, compared with 24% of the aligned queries. This is a vast difference in accuracy, but it is easily explained. An average shot in broadcast video is about 2 s, so a 5-s query is likely to contain only two to three shots. If the first and last shots are truncated, the average query would be left with a signature containing two erroneous symbols and, at most, one correct symbol. The two erroneous symbols are likely to contribute more to the score than the one correct one, making a positive identification unlikely.

Similar results were seen for all query lengths, and all levels of degradation—the aligned queries were much easier to find than the nonaligned ones. In short queries, the difference was frequently a factor of 5 or more between the two groups. The difference was lower for longer queries, as the erroneous shot lengths contributed proportionally less to the overall score.

In real applications, it is unlikely that the users would have the choice to align their queries to shot boundaries, so an alternative solution is required.

Table XVI. Shot-Length Signature Truncation (By removing the first and last shots from the shot-length signature, the impact of truncated shots is reduced.

This table shows the proportion of queries for which the correct match was listed first when the full shot-length signature was used, and when the first and last shots were truncated. For example, for the 5-s aligned queries at 1200 kb/s, 24% of correct results were identified using the full signature, and 20% when the signature was truncated.)

Query Length	1200 kb/s	600 kb/s	300 kb/s	100 kb/s
	Full/Trunc.	Full/Trunc.	Full/Trunc.	Full/Trunc.
Nonaligned queries				
5	0.04 / 0.00	0.04 / 0.00	0.04 / 0.00	0.04 / 0.00
30	0.44 / 0.52	0.36 / 0.48	0.32 / 0.40	0.32 / 0.40
120	0.68 / 0.68	0.68 / 0.68	0.68 / 0.68	0.68 / 0.68
Total	0.33 / 0.40	0.33 / 0.40	0.32 / 0.37	0.31 / 0.37
Aligned queries				
5	0.24 / 0.20	0.20 / 0.16	0.20 / 0.16	0.20 / 0.16
30	0.76 / 0.80	0.76 / 0.80	0.72 / 0.72	0.72 / 0.72
120	0.96 / 1.00	0.92 / 0.92	1.00 / 0.92	0.96 / 0.92
Total	0.77 / 0.77	0.75 / 0.75	0.75 / 0.72	0.74 / 0.72

One possible approach is simply to assume that the first and last shots in a query are truncated, and therefore a hindrance to finding correct matches. These shots are then dropped from the signature, and query evaluation uses only the remaining shots. This is a compromise that is likely to have a negative impact on retrieval accuracy for aligned queries, but a strong positive impact on nonaligned queries.

Table XVI shows the effectiveness of this method. The top section of the table shows that this has a positive effect on most nonaligned queries, but, for very short queries, the effectiveness is reduced. It is likely that many of the 5-s queries contained only two shots, so dropping the first and last of them leaves a zero-length signature. In all other cases, however, query effectiveness improved. The improvement was less prominent in longer queries, as an alignment error has comparatively less impact on the overall score for the match. The second part of the table shows the impact of truncating the signature on queries that are already aligned to shot boundaries. The difference is noticeable on 5-s queries, but, contrary to our expectations, did not have a significant impact on longer queries. In a few cases, truncation of aligned query signatures actually improved retrieval effectiveness, although the reason for this is not apparent.

It is also interesting to note the effect that alignment of the queries has on other signature types. Table XVII compares the retrieval effectiveness for nonaligned and aligned queries using all four signature types. Although the most pronounced difference is observed when using the shot-length signature, the other methods also benefited when the queries were aligned to shot boundaries. One possible explanation for this is that, during shot transitions, the symbols produced by the centroid and color-shift methods will be high (as there is substantial change in the frames). It is possible that degradations of the queries results in these symbols being substantially different. This suggests that all these methods could be improved with further investigation of scoring methods.

Table XVII. Effects of Query Alignment (Aligning queries to the nearest shot boundary has a positive effect on all of the signature types, but the effect is most noticeable with the shot-length signature. This table shows the proportion of correct results listed as the highest-ranked match for nonaligned (random) and aligned queries at different bitrates.)

Signature Type	1200 kb/s	600 kb/s	300 kb/s	100 kb/s
	Rand./Algn.	Rand./Algn.	Rand./Algn.	Rand./Algn.
Color shift	0.95 / 0.97	0.73 / 0.90	0.52 / 0.78	0.48 / 0.76
Centroid	0.97 / 0.99	0.81 / 0.91	0.76 / 0.82	0.67 / 0.76
Shot length	0.33 / 0.77	0.33 / 0.75	0.32 / 0.75	0.31 / 0.74
Combined	1.00 / 1.00	0.85 / 0.93	0.75 / 0.88	0.70 / 0.84

Table XVIII. Comparison of New Methods with Baseline (All of the new methods were more effective than the baseline method on the small dataset. This table shows the proportion of queries where the correct result was listed first, followed by the proportion where the correct match was listed in the top 10. While not as effective as the new methods, the baseline method was not substantially affected by most of the degradations.)

	Color Shift	Centroid	Combined	Shot Length	Truncated Shot-len.	Baseline
	Top 1/10	Top 1/10	Top 1/10	Top 1/10	Top 1/10	Top 1/10
Original query (nondegraded)						
	1.00/1.00	1.00/1.00	1.00/1.00	0.60/0.73	0.67/0.76	0.22/0.47
Reduced bitrate						
600 kb/s	0.82/0.96	0.91/0.98	0.93/0.98	0.51/0.71	0.64/0.73	0.27/0.47
300 kb/s	0.58/0.82	0.76/0.89	0.82/0.96	0.53/0.69	0.58/0.69	0.27/0.44
100 kb/s	0.53/0.78	0.67/0.84	0.78/0.89	0.53/0.69	0.58/0.69	0.27/0.44
Reduced resolution						
176 × 144	0.98/1.00	0.73/0.82	0.98/1.00	0.56/0.71	0.67/0.73	0.22/0.44
128 × 96	0.93/1.00	0.51/0.67	0.96/1.00	0.56/0.71	0.62/0.73	0.29/0.49
64 × 48	0.69/0.89	0.40/0.49	0.60/0.73	0.53/0.69	0.58/0.69	0.02/0.02
Altered frame rate						
24 fps	0.91/1.00	1.00/1.00	0.98/1.00	0.38/0.58	0.31/0.53	0.22/0.49
29.97 fps	0.89/1.00	0.98/1.00	0.98/1.00	0.27/0.49	0.31/0.53	0.22/0.49
30 fps	0.91/1.00	0.96/1.00	1.00/1.00	0.27/0.51	0.27/0.56	0.22/0.49

5.3.5 *Comparative Effectiveness of Retrieval Techniques.* To determine the effectiveness of the new retrieval methods, we compared them to the baseline method described in Section 5.1. Due to the computation cost of the baseline method, we limited the number of queries to 45 for this comparison (five of each length). Using all of the degradations, this resulted in 495 individual queries, each of which took an average of 71 min to complete.

Table XVIII shows the results of these tests. The color-shift method, shown in the first column, was found to be very effective in most situations. In the presence of very pronounced degradation, the effectiveness of this method suffered, though most of the queries were still listed in the top 10, even when the query was reduced to 100 kb/s (78% listed in the top 10) or 64 × 48 pixels (89% listed in the top 10).

The centroid method shows similar results, though the strengths of this method are a little different. This method was less affected by changes to the encoded bitrate, with 76% of correct results listed first for 300-kb/s queries,

compared with 58% for the color-shift method. In contrast, the color-shift method was more resilient to changes in resolution, with 93% of correct results scoring highest for queries encoded at 128×96 pixels, compared with 51% for the centroid method. The effectiveness of these two methods was comparable when the frame rate was changed.

By combining these two signature types, the strengths of both are retained. The combined signature works better than either of the other two when the bitrate is reduced, and the effectiveness on reduced-resolution queries is similar to the better of the two (the color-shift method). The combined signature is the most effective of any of the methods tested.

The shot-length method is not as effective as the centroid, color-shift, or combined signature methods for these short queries. Truncating the first and last shots from the signature improves effectiveness somewhat, but it is still well short of the accuracy achieved using the other methods. It is likely that this discrepancy would be less apparent for very long queries (of several minutes or hours), as query signatures for short queries do not contain sufficient information for accurate retrieval.

All of the new methods tested, however, are more effective than the baseline, which was able to identify the correct matches in the top 10 less than half the time, even when the query was not degraded. Since the baseline uses detailed information about every frame in the query and compares these frames using established image comparison methods, it was expected that this method would be substantially more effective than the new signature-based techniques, though much slower. A possible explanation for this discrepancy was the nature of the data clip being searched—the clip consisted largely of a talk show, which contained many visually similar segments, such as relatively static shots of the host. Given the similarity of these images, it is unsurprising that image comparison techniques, such as the color histogram comparison used in the baseline method, are unable to discriminate between them. It is probable that other image comparison methods, such as region histograms, edge comparison, or color coherence vector differences would be equally ineffective. It is also likely that this shortcoming would be noticeable in numerous other types of content, including sports, game shows, and news broadcasts, as these genres also include large numbers of visually similar sections.

Interestingly, the baseline method was not dramatically affected by most of the degradations that we tested—the only exception being a dramatic reduction in resolution. This suggests that the overall color distribution is not significantly altered by these digital degradations, though the preliminary experiments in Section 5.2 showed that analogue degradation does have a substantial impact on the color. Transcoding the content using a different codec would also affect the baseline method, as color information is often altered in this process.

The results presented in this section indicate that the new signature-based retrieval methods are effective at identification of coderivative clips in real-world data. Using the combined signature on queries with low to moderate degradation, the correct regions were almost always listed as the highest-ranking result. It is useful, however, to consider how well these methods separate the correct and incorrect results. The separation/HFM ratio is a good

measure of the strength of this separation. If the ratio is high, it would be trivial to automatically eliminate the false matches, leaving only the correct regions. Such a method could then be used to accurately identify all the coderivatives in a collection or stream, and appropriate action could be taken without the need for human intervention.

Using the combined signature, an average ratio of 2.5 or more was achieved regardless of degradation. A ratio of 1 is sufficient to easily separate correct and incorrect matches by automatic means, so, with the combined signature, this would be trivial. The ratios using the centroid and color-shift signatures were also high enough for automatic elimination of false positives in most cases. The shot-length method was generally not sufficiently discriminatory for unsupervised matching with the short queries tested in this experiment, although it is likely that a satisfactory ratio could be achieved with longer queries.

In addition to being more effective, we found that the methods we propose are substantially faster than the baseline. We used a larger dataset (23 h) to evaluate the query execution time using the new methods with queries ranging from 5 to 60 s in length. Using the baseline method, the evaluation time averaged over 4 h. With the combined signature, queries took an average of 9 min, 18 s—25 times faster than the baseline. Queries on the color-shift signature data took an average of 2 min, 43 s and the centroid signature took 2 min, 24 s. Using the shot-length signature, query evaluation took an average of 30 ms—nearly 3 million times faster than real time.

5.3.6 Query Exactness. As discussed near the start of Section 5.3, we considered a result to be a correct match if the region indicated has any overlap with the known match in the dataset, with a small additional tolerance. For a 2-min query, this can result in a correct match being reported more than 120 s before or after the actual occurrence. For many applications, this is sufficiently exact to be considered correct, but for some tasks, especially where a high level of automation is desired, this is insufficient. In practice, we found our methods to be substantially more exact than this tolerance requires. We refer to the difference between the start time of an occurrence of a clip and the start time reported by the retrieval algorithm as the *exactness* of a query.

The average exactness of the queries on this dataset is shown in Table XIX. This table shows the average number of seconds difference between the actual occurrence of a clip and the match reported by the query evaluation. The results in this table are averaged over all of the queries that were correctly identified using each technique. For the baseline method with nondegraded queries, for example, this result is the average exactness for the 24 queries that were correctly identified. For the combined method, the result is the average exactness for all 450 queries, as they were all identified correctly.

For nondegraded queries, the color-shift method had an average error of 0.12 s—three frames at the PAL frame rate used in these experiments. Even better results were achieved using the centroid method, which identified nondegraded queries within less than two frames, on average, and the combined method, where the average error was just over one frame (0.05 s). Degradation of the query reduced the exactness of these methods, with exactness tending to

Table XIX. Query Exactness (While the tolerances used to determine whether a match is correct were forgiving, we observed that the actual results obtained were significantly more exact. This table shows the average error, in seconds, of the highest-ranked correct match (if any). In all cases, the exactness of the match was affected by degradation to the query clip.)

	Color Shift	Centroid	Combined	Shot Length	Truncated Shot-len.	Baseline
Original query (nondegraded)	0.12	0.07	0.05	4.01	6.08	5.15
Reduced bitrate						
600 kb/s	0.74	0.29	0.25	4.21	6.06	5.36
300 kb/s	1.49	0.46	0.33	4.83	6.35	6.18
100 kb/s	1.81	0.92	0.59	4.86	6.49	6.52
Reduced resolution						
176 × 144	0.20	0.51	0.10	4.11	5.92	5.58
128 × 96	0.38	1.62	0.16	4.32	5.77	6.64
64 × 48	0.55	4.76	1.03	5.55	6.62	0.55
Altered frame rate						
24 fps	0.43	0.14	0.13	5.80	7.10	6.02
29.97 fps	0.63	0.17	0.18	6.63	8.09	6.30
30 fps	0.57	0.17	0.16	6.60	7.87	6.30
Average	0.68	0.85	0.29	5.00	6.59	5.82

follow the trends observed for retrieval effectiveness described in Section 5.3.5. For instance, exactness for the color-shift method suffered most when the bitrate was reduced, while reduction in resolution had the most pronounced effect on the centroid method.

The shot-length method was less exact than the other signature types, with an average error of just over 4 s for nondegraded queries. In broadcast video, such as the data used in this experiment, the average shot length is around 2 s. The search algorithm actually reports the end-point of matching regions in the data, from which the start-point is calculated. If queries were aligned with the closest shot boundary to the actual end point, the average error would be around 2 s. Like effectiveness, the exactness using the shot-length signature was not affected to the same degree as with the other methods. The worst result—6.6 s—was just over 50% higher than the nondegraded queries, whereas the difference between the worst result with the centroid method (4.76 s) and the nondegraded case (0.07 s) was a factor of 68. As should be expected, the exactness of the shot-length method when the first and last shots were truncated was lower. It is probable that this could be improved simply by subtracting the length of the first (truncated) shot from the start time reported.

The baseline method was comparable in exactness to the shot-length signature method. This outcome was expected, as pairs of frames that are considered to be matches by the baseline method are not necessarily the exact same frame—it is more likely that they are a different frames from the same shot, or even another nearby shot. This would result in misalignment of the reported results.

Using any of the methods described, a postprocessing stage could be used to perform a finer-grained alignment of a small section of the data being searched in order to achieve a more exact result, but this has not been tested.

5.3.7 Robustness Experiment Summary. The data used in these experiments contains material that is, conceptually, difficult to search for coderivative content. The majority of the data clip consists of a talk show that contains a large number of similar shots. This repetition is typical of a broad range of broadcast television content, including sport, news and current affairs, documentaries, and situation comedies. It seems probable that repetition of visually similar content would cause difficulties for retrieval methods that rely on comparison of visual features between clips such as the baseline method used in this experiment. This intuition is confirmed by the poor effectiveness of the baseline method in these tests.

The newly proposed methods easily outperformed the baseline method in terms of query effectiveness, and the color-shift, centroid, and combined methods achieved substantially more exact results. It is clear that retrieval techniques based on the patterns of change in the video content are reliable for searching small collections such as this. In most cases, the new methods are able to distinguish between correct matches and false matches with sufficient separation for unsupervised matching.

6. CONCLUSION

We have proposed new methods for the coderivative search of video, introducing four new techniques for producing video signature data: the shot-length, color-shift, centroid-based, and combined methods. Each of these use different properties of the video to produce compact signatures. The most compact signature—the shot-length signature—is based on the pattern of edits in the video. It is fast to search and insensitive to changes in bitrate and resolution. The color-shift signature captures the way in which the color in the frames changes over time, making it more robust than existing feature-comparison methods. The centroid-based signature represents the movements of the centroids of luminance in the frames, capturing the motion in the clip using a novel and efficient motion-estimation algorithm. Finally, the combined signature uses evidence from both the centroid and color-shift signatures to produce a composite that has many of the advantages of each.

We also presented methods for searching these signatures, based on local alignment. This efficient algorithm is capable of accurately identifying coderivative content even when it comprises only a small part of a long clip. The local alignment algorithm makes use of a scoring function to determine similarity between sequences of symbols; we introduced several scoring systems that are applicable to coderivative search and compared their effectiveness. Experimental results using real-world data confirmed that the selection of scoring method has a substantial impact on query effectiveness.

The methods we proposed were intended to address the limitations in efficiency and sensitivity to degradation that are present in existing solutions; however, we also found that our methods were substantially more effective than previous approaches, even when minimal degradation was present. In experiments on a 2.5-h dataset, the most effective of the new methods correctly identified 100% of the queries as the highest-ranked match. For the same test,

the baseline method identified just 22%. Extending the results to include correct matches in the top 10 ranked matches, the baseline was still only able to identify 47% of the queries.

While, in general, our methods were substantially more effective than the baseline, each of them has specific strengths and limitations. The color-shift signature method, for example, was effective when the resolution of the query clip differed from the data being searched, but effectiveness suffered when the bitrate was reduced. In contrast, the centroid-based signature was relatively insensitive to changes in bitrate, but was less effective when the resolution was altered. The combined signature method was less sensitive to changes in bitrate than the color-shift method, but more sensitive than the centroid method. Similarly, it was affected less by changes to resolution than the centroid method, but more than the color-shift method. For queries that were not substantially degraded, the combined signature method was more effective than either of the other two. It is straightforward to determine, a priori, which of these methods is likely to be most effective: if the bitrate of the query is low, the centroid method should be used; if the resolution is low, the color-shift method is likely to be better; if neither (or both) of these problems exist, then the combined signature is likely to yield better results.

The level of effectiveness achieved using the shot-length method was substantially lower than these three methods. It seems likely, however, that for queries longer than 10 or 20 min, the effectiveness would be comparable. In these situations, the shot-length method would be a preferable solution, due to the lower computation costs. The shot-length method is also insensitive to changes in both bitrate and resolution. Changes to the query frame rate had a significant impact on efficacy using this method—further investigation is required to develop techniques to address this limitation. It is likely that alternate scoring functions could be developed for use when the frame rates of query and data do not match.

The methods presented in this article are dramatically more efficient, robust, and effective than previous approaches. However, further improvements may be available. We tested a range of scoring methods for local alignment, but it is likely that even more effective scoring functions could be developed with a statistical approach using larger quantities of data. The audio tracks that are associated with video content contain a substantial amount of information that is likely to be useful as an additional indicator of similarity in many circumstances. Another area for further investigation is to use a two-stage approach, with an initial search using the shot-length method to reduce the search space, followed by a fine-grained search with a more discriminatory signature to determine similarity ranking.

However, while the efficacy of the proposed methods could undoubtedly be improved by further research, the experimental results presented in this article indicate that effective retrieval can be achieved with these new techniques as they stand. Efficient and accurate retrieval of coderivative video content using these novel methods is possible for collections of thousands of hours of video even when the content is significantly degraded or altered.

REFERENCES

- ADJEROH, D. A. AND LEE, M. C. 1998. Video sequence similarity matching. In *Proceedings of the Conference on Multimedia Information Analysis and Retrieval* (Hong Kong).
- ADJEROH, D. A., LEE, M. C., AND KING, I. 1998. A distance measure for video sequence similarity matching. In *Proceedings of International Workshop on Multimedia Database Management Systems* (Dayton, OH). 72–79.
- AHANGER, G., BENSON, D., AND LITTLE, T. D. C. 1995. Video query formulation. In *Proceedings of the Conference on Storage and Retrieval for Image and Video Databases* (SPIE). 280–291.
- BAEZA-YATES, R. A. AND RIBEIRO-NETO, B. 1999. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Reading, MA.
- BERCHTOLD, S., BOHM, C., AND KRIEGEL, H. P. 1998. The pyramid-technique: Towards breaking the curse of dimensionality. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Seattle, WA). 142–153.
- BORECZKY, J. S. AND ROWE, L. A. 1996. Comparison of video shot boundary detection techniques. In *Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases*. 170–179.
- CHANG, S., CHEN, W., MENG, H., SUNDARAM, H., AND ZHONG, D. 1997. VideoQ: An automated content based video search system using visual cues. In *Proceedings of the ACM Multimedia Conference*. 313–324.
- CHANG, S., CHEN, W., MENG, H., SUNDARAM, H., AND ZHONG, D. 1998. A fully automated content based video search engine supporting spatio-temporal queries. *IEEE Trans. Circ. Syst. Video Tech.* 8, 5, 602–615.
- CHEUNG, S. AND ZAKHOR, A. 2000. Estimation of web video multiplicity. In *Proceedings of the SPIE Conference on Internet Imaging* (San Jose, CA).
- DEMENTHON, D. 2003. Video retrieval of near-duplicates using k -nearest neighbor retrieval of spatio-temporal descriptors. In *Proceedings of the Third International Workshop on Content-Based Multimedia Indexing* (CBMI 2003, Rennes, France).
- DEMENTHON, D. AND DOERMANN, D. 2003. Video retrieval using spatio-temporal descriptors. In *Proceedings of ACM Multimedia 2003* (Berkeley, CA).
- FUSHIKIDA, K., HIWATARI, Y., AND WAKI, H. 1999. A content-based video query agent using feature-based image search engine. In *Proceedings of the 3rd International Conference on Computational Intelligence and Multimedia Applications* (New Delhi, India).
- GUPTA, A. AND JAIN, R. 1997. Visual information retrieval. *Commun. ACM* 40, 5, 70–79.
- HAMPAPUR, A. AND BOLLE, R. 2001. Comparison of distance measures for video copy detection. In *Proceedings of the International Conference on Multimedia and Expo*.
- HAMPAPUR, A. AND BOLLE, R. M. 2002. VideoGREP: Video copy detection using inverted file indices. Tech. rep. IBM Exploratory Computer Vision Group, Yorktown Heights, NY.
- HAMPAPUR, A., BOLLE, R. M., AND HYUN, K.-H. 2001. Comparison of sequence matching techniques for video copy detection. In *Proceedings of SPIE: Storage and Retrieval for Media Databases*.
- HAMPAPUR, A., WEYMOUTH, T. E., AND JAIN, R. 1994. Digital video segmentation. In *Proceedings of the ACM Multimedia Conference*. 357–364.
- HARTUNG, F., SU, J. K., AND GIROD, B. 1999. Spread spectrum watermarking: Malicious attacks and counterattacks. In *Proceedings of the SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging* (San Jose, CA). 147–158.
- HAUPTMANN, A. G., CRISTEL, M. G., AND PAPERINICK, N. D. 2002a. Video retrieval with multiple image search strategies. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*. 376–376.
- HAUPTMANN, A. G., JIN, R., AND NG, T. D. 2002b. Multi-modal information retrieval from broadcast video using ocr and speech recognition. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*. 160–161.
- HAUPTMANN, A. G. AND PAPERINICK, N. D. 2002. Video-Cuebik: Adapting image search to video shots. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*. 156–157.
- HOAD, T. C. AND ZOBEL, J. 2003a. Fast video matching with signature alignment. In *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval* (Berkeley, CA). 262–269.

- HOAD, T. C. AND ZOBEL, J. 2003b. Methods for identifying versioned and plagiarised documents. *J. Amer. Soc. Inform. Sci. Tech.* 54, 3 (Jan.), 203–215.
- HOAD, T. C. AND ZOBEL, J. 2003c. Video similarity detection for digital rights management. In *Proceedings of the Australasian Computer Science Conference* (Adelaide, Australia). 237–245.
- HOI, C.-H. 2002. Similarity measurement and detection of video sequences. Tech. rep. Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.
- HOI, C.-H., WANG, W., AND LYU, M. R. 2003. A novel scheme for video similarity detection. In *Proceedings of the Conference on Image and Video Retrieval*. 373–382.
- IDE, I., HAMADA, R., SAKAI, S., AND TANAKA, H. 2001. An attribute based news video indexing. In *Proceedings of the 2001 ACM Workshops on Multimedia*. 70–73.
- JAIMES, A., CHANG, S.-F., AND LOUI, A. C. 2002. Duplicate detection in consumer photography and news video. In *Proceedings of the ACM Multimedia Conference* (Juan Les Pines, France).
- LANGELAAR, G. C., LAGENDIJK, R. L., AND BIEMOND, J. 1998. Removing spatial spread spectrum watermarks by nonlinear filtering. In *Proceedings of the European Signal Processing Conference*. vol. 4. 2281–2284.
- LEE, S.-L., CHUN, S.-J., AND LEE, J.-H. 2003. Effective similarity search methods for large video data streams. In *Proceedings of the International Conference on Computational Science*. 1030–1039.
- LI, D. AND LU, H. 2000. Avoiding false alarms due to illumination variation in shot detection. In *Proceedings of the IEEE Workshop on Signal Processing Systems*.
- LIENHART, R. 2001. Reliable transition detection in videos: A survey and practitioner’s guide. *Int. J. Image Graph.* 1, 3, 469–486.
- LIENHART, R., EFFELSBERG, W., AND JAIN, R. 1998. VisualGREP: A systematic method to compare and retrieve video sequences. In *Proceedings of the Conference on Storage and Retrieval for Image and Video Databases (SPIE)*. 271–283.
- LIENHART, R., KUHMUNCH, C., AND EFFELSBERG, W. 1997. On the detection and recognition of television commercials. In *Proceedings of the International Conference on Multimedia Computing and Systems*. 509–516.
- LIU, T., ZHANG, X., WANG, D., FENG, J., AND LO, K.-T. 2000. Inertia-based cut detection technique: A step to the integration of video coding and content-based retrieval. In *Proceedings of the International Conference on Signal Processing* (Beijing, China). Vol. 2. 1018–1025.
- LIU, X., ZHUANG, Y., AND PAN, Y. 1999. A new approach to retrieve video by example video clip. In *Proceedings of the ACM Multimedia Conference*. Vol. 2. 41–44.
- MOHAN, R. 1998. Video sequence matching. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seattle, WA).
- NAGASAKA, A. AND TANAKA, T. 1992. Automatic video indexing and full-video search for object appearances. In *IFIP Proceedings of Visual Database Systems*. 113–127.
- NAPHADE, M. R., MEHROTRA, R., FERMAN, A. M., WARNICK, J., HUANG, T. S., AND TEKALP, A. M. 1998. A high performance shot boundary detection algorithm using multiple cues. In *Proceedings of the IEEE International Conference on Image Processing* (Chicago, IL).
- NAPHADE, M. R., WANG, R., AND HUANG, T. S. 2001. Supporting audiovisual query using dynamic programming. In *Proceedings of the Ninth ACM International Conference on Multimedia* (Ottawa, Ont., Canada). 411–420.
- 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 *Proceedings of SPIE, Storage and Retrieval for Media Databases* (San Jose, CA). Vol. 3972. 564–572.
- NG, C. W., KING, I., AND LYU, M. R. 2001. Video comparison using tree matching algorithms. In *Proceedings of the International Conference on Imaging Science, Systems and Technology* (Las Vegas, NV). Vol. 1. 184–190.
- NGO, C.-W., PONG, T.-C., AND ZHANG, H.-J. 2001. On clustering and retrieval of video shots. In *Proceedings of the ACM Multimedia Conference* (Ottawa, Ont., Canada).
- PARK, S., CHO, J.-S., AND HYUN, K.-H. 2002. Indexing technique for similarity matching in large video databases. In *Proceedings of the SPIE Conference on Storage and Retrieval for Media Databases* (San Jose, CA). 214–222.

- PARK, S. AND HYUN, K.-H. 2004. Trie for similarity matching in large video databases. *Inform Syst.* 29, 8 (Dec.), 641–652.
- PATEL, N. V. AND SETHI, I. K. 1997. Video shot detection and characterization for video databases. *Patt. Recog.* 30, 4 (Apr.), 583–592.
- PETITCOLAS, F. A., ANDERSON, R. J., AND KUHN, M. G. 1998. Attacks on copyright marking systems. In *Proceedings of the Second International Workshop on Information Hiding*. 218–238.
- PUA, K. M., GAUCH, J. M., GAUCH, S. E., AND MIADOWICZ, J. Z. 2004. Real time repeated video sequence identification. *Comput. Vis. Image Understand.* 93, 3, 310–327.
- SHAN, M.-K. AND LEE, S.-Y. 1998. Content-based video retrieval based on similarity of frame sequence. In *Proceedings of the International Workshop on Multimedia Database Management Systems* (Dayton, OH). 72–79.
- SWANBERG, D., SHU, C.-F., AND JAIN, R. 1993. Knowledge guided parsing in video databases. In *Proceedings of the Conference on Electronic Imaging: Science and Technology* (San Jose, CA).
- TAN, Y. P., KULKARNI, S. R., AND RAMADGE, P. J. 1999. A framework for measuring video similarity and its application to video query by example. In *Proceedings of the IEEE International Conference on Image Processing* (Kobe, Japan).
- UEDA, H., MIYATAKE, T., AND YOSHIZAWA, S. 1991. IMPACT: An interactive natural-motion-picture dedicated multimedia authoring system. In *Proceedings of the ACM International Conference on Computer-Human Interaction* (SIGCHI, New Orleans, LA).
- WU, Y., ZHUANG, Y., AND PAN, Y. 2000. Content-based video similarity model. In *Proceedings of the ACM Multimedia Conference*.
- YANG, J., LI, Q., AND ZHUANG, Y. 2002. Octopus: Aggressive search of multi-modality data using multifaceted knowledge base. In *Proceedings of the Eleventh International Conference on the World Wide Web*. 54–64.
- YASUGI, Y., BABAGUCHI, N., AND KITAHASHI, T. 2001. Detection of identical events from broadcasted sports video by comparing camera works. In *Proceedings of the 2001 ACM Workshops on Multimedia*. 66–69.
- YEO, B.-L. AND YEUNG, M. M. 1997. Retrieving and visualizing video. *Commun. ACM* 40, 12 (Dec.), 43–52.
- YEUNG, M. AND LIU, B. 1995. Efficient matching and clustering of video shots. In *Proceedings of the IEEE International Conference on Image Processing* (Washington, DC). Vol. 1. 338–341.
- YOON, M. H., YOON, Y. I., AND CHUNG KIM, K. 1999. Hybrid video information system supporting content-based retrieval. In *Proceedings of the 3rd International Conference on Computational Intelligence and Multimedia Applications* (New Delhi, India).
- ZABIH, R., MILLER, J., AND MAI, K. 1995. A feature-based algorithm for detecting and classifying scene breaks. In *Proceedings of the ACM Multimedia Conference* (San Francisco, CA). 189–200.
- ZABIH, R., MILLER, J., AND MAI, K. 1999. A feature-based algorithm for detecting and classifying production effects. *Multimed. Syst.* 7, 2, 119–128.
- ZHANG, H., KANKANHALLI, A., AND SMOLIAR, S. W. 1993. Automatic partitioning of full-motion video. *Multimed. Syst.* 1, 1, 10–28.
- ZHAO, L., QI, W., LI, S. Z., YANG, S. Q., AND ZHANG, H. J. 2001. Content-based retrieval of video shot using the improved nearest feature line method. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing* (Salt Lake City, UT).

Received January 2005; revised June 2005, October 2005; accepted October 2005