

Video Similarity Detection for Digital Rights Management

Timothy C. Hoad

Justin Zobel *

School of Computer Science and Information Technology, RMIT University
GPO Box 2476V, Melbourne 3001, Australia
{hoad,jz}@cs.rmit.edu.au

* Contact author for all correspondence.

Abstract

Vast quantities of video data are distributed around the world every day. Video content owners would like to be able to automatically detect any use of their material, in any media or representation. We investigate techniques for identifying similar video content in large collections. Current methods are based on related technology, such as image retrieval, but the effectiveness of these techniques has not been demonstrated for the task of locating video clips that are derived from the same original. We propose a new method for locating video clips, shot-length detection, and compare it to methods based on image retrieval. We test the methods in a variety of contexts and show that they have different strengths and weaknesses. Our results show that the shot-based approach is promising, but is not yet sufficiently robust for practical application.

1 Introduction

The advent of DVD, digital broadcasting, and the use of broadband internet connections has led to a dramatic increase in the accessibility of digital multimedia content. As compression technologies continue to improve and internet bandwidth becomes more affordable, this use of digital media is set to continue to grow. The range of formats in which content is delivered is also increasing. For example, a given video might be initially produced and distributed in a high-resolution digital format, then re-distributed in a different aspect ratio for VHS and free-to-air broadcast. Low-resolution digital copies, possibly at a lower frame rate, might then be made of the VHS version.

The proliferation of digital media presents new opportunities to users in terms of distribution methods and interactivity, but also presents issues with regard to digital rights and ownership verification. There is a growing concern amongst content owners that video and audio piracy is increasing. The ad-hoc nature of user-created media repositories, such as KaZaA, presents its own problems. Frequently the same con-

Copyright ©2003, Australian Computer Society, Inc. This paper appeared at Twenty-Sixth Australasian Computer Science Conference (ACSC2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol. 16. Michael Oudshoorn, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

tent is present with different filenames, using different compression systems, and with different aspect ratios and frame rates; in addition there may be minor alterations, such as the removal of advertisements or a dubbed soundtrack. While the files are not bitwise identical, they are “co-derivative”—that is, they are derived from the same original and represent the same content. Another issue is that content owners provide material such as video clips and advertisements to broadcasters, and need to monitor whether and when the material is used.

There are a variety of solutions that have been proposed to prevent the distribution of material that is subject to copyright, but they all have limitations. Physical copy prevention techniques can be circumvented, and result in a reduction in the freedom of law-abiding users to keep backups of digital content and to transfer the content between different media for personal use. Watermarking either results in a noticeable degradation of quality, or can be easily removed. Thus it is also necessary to be able to detect copyright infringement.

In this paper we investigate the problem of locating instances of a query clip within a longer data clip. Applications include locating specific advertisements, identifying advertisement breaks, and monitoring of a video stream in order to identify copyright infringements. While the task of locating instances of a query is relatively straightforward when working with traditional media, it is problematic in the video domain due to the fact that two instances of the same clip may not be bitwise identical. They can differ in resolution, frame rate, bitrate, compression codec, and colour or signal quality.

There has been some investigation of the general problem of algorithms for video search engines, that is, of methods for finding fragments of video that are pertinent to a user query. However, these methods do not address the particular problem of finding video that is the same as the query—indeed, in much of the existing research, the query is expressed as text. To our knowledge, the problem of finding specific clips has not previously been addressed.

An obvious approach to finding a query clip is to extract images from it and find matching images in

text of image retrieval. If the matching images are found in the right order in a region in the data clip, then it seems likely that the region is a match. However, image comparison is not robust for data that has been transformed between formats. Changes in the colour map or the aspect ratio are likely to undermine such an approach.

Another approach that may be used in this field is video watermarking. This is a technique developed for use in digital rights management to verify ownership of a piece of content. The concept involves embedding a signature within the video data which will be preserved even after conversion between analogue and digital or changes in video encoding format. There are three drawbacks to this approach, however. Digital video encoders are designed specifically to compress video signals by discarding information that is imperceptible to humans. Therefore, for a watermark to be robust, it must be perceptible, thereby reducing video quality for the same encoded bitrate. Secondly, it is generally possible to render the watermarks unreadable with only minor effort. The third, and most significant problem with watermarking, is that it requires that the content that is being searched is watermarked. For this reason, it cannot be used to identify content that has already been released.

We propose a novel alternative: to seek matching patterns of *shots*. A characteristic of video is that, in general, it consists of continuous sequences of images from one scene, with occasional cuts from one scene to another. A single continuous operation of a camera is generally referred to as a shot. Cut-detection algorithms (developed in the context of general video retrieval) can be used to detect the point in time at which each cut event occurs, and hence the start and end points of the shots. Given the high frequency of cuts in most videos, the length of the intervals between cuts (the shot length) can provide a signature that can be used for matching, and cut-detection algorithms should produce consistent results even after transformation.

We have implemented image-based and shot-based algorithms for clip detection, and have evaluated them on free-to-air broadcast material. Each method was tested with several query clips, and robustness was investigated by applying a range of transformations and signal degradations to the queries. Our experiments showed that the two methods have different strengths, but that the shot-based method is superior if sufficient numbers of shots are present.

2 Measuring similarity

There has been extensive research into methods for assessing the similarity of pairs of videos (Cheung & Zakhor 2002, Dimitrova & Abdel-Mottaleb 1997,

1999, Zhao, W.Qi, S.Z.Li, S.Q.Yang & Zhang 2001). Much of this work proceeds by analogy with information retrieval: the task is to find clips that concern similar themes or material, rather than clips that are drawn from the same source. One of the difficulties in determining similarity between any two pieces of information is in the definition of similarity. Unfortunately, the definition of similarity, in general, is dependent on the application. For example, consider two video clips, one depicting a pair of dancers and another showing a pair of wrestlers. In one context, they may be considered similar, as they both show a pair of people involved in a sport that involves physical contact. In another context, dancing and wrestling may be considered as totally unrelated.

While our research is indeed addressing video similarity, our definition of similarity is concrete. For the purposes of this work, we consider a pair of clips to be similar if they are both derived from the same original. For example, the director's cut of the movie *Blade Runner* would be considered co-derivative with the original theatrical release. Similarly, two instances of the same piece of content would also be considered co-derivative. A remake of *Beau Geste* is not co-derivative with the original version, however.

There are many aspects of a video clip that can be changed without making significant changes to the content. A robust system for co-derivation detection should not be affected by these changes. Some examples of attributes that can change are:

- TV standards (PAL, NTSC, SECAM)
- Aspect ratio (3:4, 16:9, 2.35:1)
- Digital video codec (MPEG-1, DivX, MJPEG)
- Frame rate (25 fps, 29.97 fps, 30 fps)
- Resolution (160×120 , 320×240 , 640×480)
- Bitrate (64 kbps, 320 kbps, 1.5 Mbps)
- Edits (theatrical release, director's cut, TV release with ads)
- Soundtracks (dubbed videos)
- Media (VHS, SDTV, DVD, HDTV)

The two major fields that are related to this problem are content-based image retrieval and content-based video retrieval. We investigate the techniques used in these disciplines for measuring similarity, as possible bases for methods for accurately detecting co-derivation.

A range of image matching techniques have been proposed.

Pixel-based comparison The similarity of a pair of images is computed by adding the differences of pixel colour values for each pixel in the frame. This approach is very sensitive to noise. A minor shift in

logue noise, can add up to a substantial difference over a whole frame. This means that changes in resolution, aspect ratio, or bitrate can have dramatic effects. For video, changes in frame rate would be problematic.

This approach also suffers from serious performance issues. In an average video clip with a resolution of 320x288 pixels, each frame comparison involves a comparison of over 100,000 pixels. Given that the colour of each pixel is represented by three values (red, green, and blue), more than 300,000 integer comparisons would be required to compare a pair of frames.

Colour-histogram-based comparison In histogram-based techniques, each pixel of the images is categorised into a bucket in a colour histogram. The comparison then involves calculating the distance between the histograms representing the two images, usually by calculating a Manhattan or Euclidean distance.

The representation of the pixels in each frame is by a set of integer values, which, combined, describe the colour of the pixel. Each of the values describes an aspect of the colour. The aspect described by these values depends on the *colourspace* that is used for that image (or set of images). The most widely used colourspace in digital media is RGB, in which the three values describe the proportion of red, green, and blue in the pixel. The YUV (or YCrCb) colourspace is also widely used in digital media. This colourspace also used three values per pixel, which represent luminance, red chrominance and blue chrominance. This colourspace is popular due to the fact that it more closely models the way in which the human eye perceives colour, and conversion between YUV and RGB is relatively inexpensive. HSV is another colourspace which is fairly widely used. Its three values represent colour, saturation and value (brightness). It is another attempt to model human perception of colour, although this model is based more on the way that people describe colour, rather than the way the eye senses it. $L^*a^*b^*$ (also called CIELAB) is another colourspace which is slightly different again. The philosophy with this system is that a pair of colours differing in value by a single unit should be perceived by a human as having a similar difference as any other pair of colours that differ in value by a single unit. For example, the colours represented by the values (0, 0, 50) and (0, 0, 51) should be perceived as being different to each other to the same degree as the colours represented by the values (100, 0, 0) and (101, 0, 0). Generally, colour histograms are calculated using the colourspace used to encode the image, however it is also possible to convert between colourspaces in order to more accurately model human perception.

to noise as the pixel-based method, and would not be as dramatically affected by changes in resolution or bitrate. Changes in frame rate would present problems—frames present in the query would be absent in the data, or vice versa—as would alterations in colour balance. Performing a frame-by-frame comparison of image histograms would still be computationally expensive, though not to the same degree as the pixel-based method.

Texture-based comparison Texture-based methods are similar to the colour-histogram method. Instead of using a feature vector based on colour, however, similarity is computed based on a feature vector that represents the contrast, grain, and direction properties of the image (Huang & Rui 1997). As with any frame-by-frame comparison, it has the drawbacks of being sensitive to changes in frame rates. This method also has performance problems, as texture histograms are generally more expensive to produce than are colour histograms. This method would also be sensitive to encoding artefacts and changes in encoding bitrate, as texture information is often lost at low bitrates.

Other feature spaces There are other feature spaces, such as shape (Aslandogan & Yu 1999), which could be used in a similar manner to colour or texture. Unfortunately, they all have inherent problems. Most of them are not robust with respect to changes in colour, compression, or resolution, and some of them are expensive to compute.

3 Video matching techniques

Image-based methods

Many techniques have been proposed for evaluating the similarity between pairs of video clips.

This simplest method for evaluating the similarity of a pair of video clips is to compare the metadata that is associated with them, such as title, date of production, and other ancillary information. While there are text annotation standards such as MPEG-7 available for video data (Abdel-Mottaleb 1999, Day & Martinez 2001), these are not widely used due to the cost of annotation and the inherent inaccuracy and subjectivity of human-entered text. For this reason, metadata comparison is not considered to be a serious solution. Nor is it applicable to the problem we are considering.

Content-based methods for comparing video data can be broadly categorised into two groups, both of which are query-by-example methods, intended for use in general-purpose video databases for identifying content that meets an information need. The

niques to perform a frame-by-frame comparison, as outlined above. We refer to this class of techniques as the *frame-by-frame* methods. Some of these approaches have variations that are designed to cope with certain problems, such as the mis-alignment of frames due to differing frame rates (Tan, Kulkarni & Ramadge 1999), and normalising the data to allow for differing resolutions (Tan et al. 1999). The second group perform comparisons based on sections of the clip that are larger than one frame. The unit that is generally used is a shot although sometimes the segmentation is simply a fixed-length string of frames. We refer to this group as the *segment-based* methods.

The segment-based methods aim to address the efficiency problems of the frame-by-frame methods by reducing the size of the features that are used to perform the comparison. The frame-by-frame methods compare every frame, but the segment-based methods generally select one or two representative frames from the segment (Fushikida, Hiwatari & Waki 1999), and perform feature-space comparisons based only on these frames, or calculate a single feature vector based on several frames from the segment (Wu, Zhuang & Pan 2000). As in the frame-by-frame methods, the segment-based systems employ various feature spaces to determine similarity.

The other feature that differentiates video comparison methods is the feature space that is used to compare the clips. These methods generally use a colour histogram difference to compare frames (Tan et al. 1999, Yeung & Liu 1995), although other feature spaces, such as intensity (Yeung & Liu 1995), texture (Wu et al. 2000), motion (Shan & Lee 1998) and shape are possible. Colour histograms are often favoured due to the simplicity and efficiency, however motion is also a commonly used discriminator in these query-by-example approaches, as it is considered to be a good indication of the style or mood of a piece of video.

Lienhart et al. (1998) proposed measurement of video similarity based on a conglomeration of a large number of features. They consider colour, motion intensity, face recognition, framing, and camera motion. This list covers most of the feature spaces that have been proposed. Their work focuses on the integration of these measures and the way in which a query should be expressed.

While they demonstrated promising experimental results, there two major problems with their approach that makes it less valuable for finding co-derivatives. The first drawback is that of efficiency. Because their methods consider so many different feature spaces, the cost of evaluating queries is very high. The selection of feature spaces is strongly focused on effectiveness, with little consideration of the cost. For example, the colour features were repre-

a colour histogram. While this approach promises a more discriminative representation (Pass, Zabih & Miller 1996), there is an additional cost in evaluating it. In addition, the colourspace used is $L^*a^*b^*$. Since most digital video is represented using YUV, this represents a costly colourspace conversion. It is not likely that similarity can be evaluated in real time using the techniques they describe.

The second drawback to their approach is that considerable human intervention is required when formulating the query. It is necessary for the user to set the parameters for use in evaluating the query. This involves selecting the feature spaces that should be used in query evaluation, and assigning weights to each of these feature spaces. While this system may be useful as a generalised query engine, the inefficiency and intervention required make it unsuitable for unsupervised co-derivative detection, nor is it clear that users will in general be sufficiently expert to make the necessary choices.

Liu et al. (1999) propose a more efficient approach to evaluating video similarity. The videos are segmented into shots, each of which are represented by a keyframe. They use a colour histogram in HSV colourspace, along with a texture vector consisting of values to represent coarseness, contrast and direction to determine frame similarity. Whole clips are compared by finding the shots that are common to both clips (pairs are judged as “similar” or “not similar” by comparing the similarity values of their representative keyframes to an adaptive threshold). The sequence of these common shots is then compared to evaluate the final similarity score.

There are some major drawbacks to this approach as well. The most significant problem is that the technique is only able to compare whole clips. This may be useful in databases where the clips are carefully segmented prior to insertion, but in our application this is not the case. As video is frequently distributed in a continuous stream, it is common for programs to be interleaved with ads, or for more than one program to be in the same clip. It also makes it difficult to locate a short clip that is a subset of a large one, as the overall similarity would be quite low, even though the local similarity may be very strong.

The other significant drawback of this approach is that it uses a colour histogram as the dominant discriminator. Colour histograms are widely recognised as being useful in image comparison (Aslandogan & Yu 1999), however they are not robust to some of the problems that occur in the video domain. Most video content is currently distributed in analogue form. One of the common analogue encoding formats, NTSC, is notorious for its problems in colour fidelity.¹ VHS is a medium that is also well known for

¹Hence the nickname “Never Twice the Same Colour”.

logue noise and interference can cause colour shifts, as well as alterations in texture.

Zhao et al. (2001) present a different approach, referred to as the Nearest Feature Line method. This is a segment-based method in which the clips are segmented into shots from which a small number of keyframes are selected. The colour histograms are then used to build a vector—the feature line—that captures the “trajectory” of the segment. The distance between feature lines determines the similarity of a pair of segments. While this approach still employs colour histograms, the histograms are not directly compared between clips. This means that the system is less subject to colour variations or different encoding methods.

Adjeroh, Lee & King (1998) use colour-histogram-based comparison in a segment-based method. Most techniques that combine these approaches are intended for comparing whole clips, but this work uses dynamic programming to locate areas of local similarity. Dynamic programming is discussed further below.

Shot-length comparison

We propose a new method for estimating the similarity of video clips. Most of the existing methods calculate similarity based on features of the frames in the video. These methods are appealing due to the successes that have been seen in the image retrieval domain. However, we do not believe that they are reliable or accurate enough for identifying co-derivative video clips. The frame-by-frame techniques are susceptible to alignment problems when comparing clips of different bitrates, and are expensive to compute. Most of the segment-based methods are sensitive to small changes in the video, especially changes to the bitrate or colour.

Our work 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. Video cut-detection, has been investigated in detail, and is reliable and accurate (Boreczky & Rowe 1996). 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 an index to the data. Typical broadcast video contains cuts at the rate of around 1200 per hour, so the data can be reduced to an index of around 5 kilobytes for each hour of video for querying purposes.

There are two classes of cut-detection techniques. The first is the *compressed-domain* techniques, which make use of the motion information contained in compressed video data. The second class is the

data must be decoded before cut-detection can be performed. The compressed-domain methods are generally faster, due to the fact that video decoding can be bypassed, but the uncompressed-domain techniques are generally considered to be more accurate (Lienhart 2001). We used a colour-histogram based 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 query is then evaluated using local alignment. Dynamic programming techniques, including local alignment, are generally applied to problems that involve matching strings that are made up of symbols from a finite alphabet (Sankoff & Kruskal 1999). The similarity between two strings is defined as the minimum number of insertion and deletion operations that must be performed on one string to transform it into the other.

Dynamic programming has been applied to a wide variety of string-matching problems, including genomic retrieval, music matching, and text comparison. It is relatively efficient, having asymptotic complexity $O(mn)$, where m is the length of the query and n is the length of the data being searched (Gusfield 1997).

In this application, the strings consist of a sequence of integers representing the length of the shots, which can be any value greater than zero. We tested several alignment scoring techniques.

Binary scoring. In a traditional dynamic programming approach, a pair of shots the same length would be awarded a positive score, and a pair with different lengths would be awarded a negative score. Each interval is treated as a token that can either match or mismatch.

This approach would present problems when comparing videos of different frame rates, however, as mis-alignment by one or two frames would be common but should not prevent a match from being flagged. For example, in comparing video at 12 frames per second to video at 30 frames per second, some shots will occur at slightly different times.

Categorical scoring. The time difference between shot lengths is categorised and the result scored appropriately. Exact matches were given a score of 15 and matches that differed by less than 10 milliseconds were awarded a score of 13. A difference of between 42 ms and 84 ms (a difference of about one frame) was awarded a score of 8. Differences of more than 200 milliseconds were given a score of -3 , while mismatches of more than a second were given a score of -7 .

plied to the difference (δ) to give a score for the match.

$$score = 3D \frac{20}{\log_e(\delta+3)} - 4$$

In addition to expected benefits to effectiveness, matching based on shots has considerable potential for improving the efficiency of detecting co-derivatives. Due to the fact that the index to the data can be so compact, for a short query local alignment requires less than a second per hour of content being searched. Because the actual query is so efficient, it is conceivable that hundreds or even thousands of queries may be executed simultaneously on a video stream in real time. Cut-detection is about as expensive as computing a colour histogram, but is a once-off cost.

4 Experiments

In order to evaluate the effectiveness of the shot-length comparison, we compared it to a baseline method. The baseline that we used was a colour-histogram based technique based on the frame-by-frame methods, with some modifications to improve efficiency and robustness. That is, we took what we judged to be the best of the video retrieval methods, and adapted it to this application.

The baseline method calculates colour histograms for each frame in the query clip and stored them in memory. It then processes the data clip sequentially, computing colour histograms for one frame per second. The histograms from the query are compared to the histograms in a sliding window of the data clip. The size of the window was set 30% larger than the length of the query. For each pair of these histograms where the Manhattan distance exceeded a threshold, the similarity score for that section of the clip is incremented by one. Local maxima are then identified, and ranked according to the similarity score.

The most common metrics for determining the effectiveness of an information retrieval task are precision, the proportion of results returned that are relevant, and recall, the proportion of relevant results that are returned (Salton & McGill 1983). However, when the task involves detecting co-derivative documents, these metrics are not always sufficient (Hoad & Zobel To appear), as we have explored in our previous work on detection of plagiarism in assignments.

In our experiments, the precision reported is Precision(n), where n is the number of occurrences of the query clip in the data clip. The recall reported is Recall(10), that is, the number of correct matches in the top 10 matches. In addition to these statistics, the highest false match (HFM), separation, and separation-to-HFM ratio are reported. The HFM is the percentage similarity that is reported for the highest-ranked irrelevant result. The separation is

reported for the lowest relevant result, and the HFM. Where the lowest relevant result is ranked lower than the HFM, the separation will be reported as a negative value. In cases where the precision and recall are very high, the HFM, separation and separation-to-HFM ratio give a good indication of how well the relevant and irrelevant matches are separated.

We used two sets of test data (that is, two data clips) in our experiments. We used a different set of queries for each data set. All video was recorded at a resolution of 352x288 pixels at the VHS frame rate of 25 fps and compressed using MPEG-1 at a bitrate of 1.2 Mbps, unless otherwise noted.

Single query, multiple variants

The first data set was a 170 minute clip recorded from broadcast commercial television.² The content was a movie, *Star Wars: Episode I The Phantom Menace*, which was interleaved by thirteen blocks of advertisements, with a total of around 120 commercials. We chose one advertisement to use as a query in this dataset, which was selected due to the fact that it was 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 is the original ad with no modification
- Query 2 has increased brightness
- Query 3 has increased contrast
- Query 4 was recorded at a lower bitrate, 125 kbps
- Query 5 was converted to a different frame rate, 29.97 fps, which is the rate used for NTSC video (movies in theatres are shown at 24 fps, and TV in Australia is broadcast at 25 fps)
- Query 6 had analogue noise added
- Query 7 had increased colour saturation
- Query 8 had reduced brightness
- Query 9 had reduced contrast
- Query 10 had reduced colour saturation
- Query 11 was recorded at a very low resolution, 96×80 pixels

Table 1 shows the results of executing each of these queries on the first clip using the baseline method. This shows an obvious trend. Query 1 is easily identified by this method, with all of the correct matches separated by nearly 40% from the incorrect ones. Queries 4, 5, and 11 are also identified, but with much lower separation from the incorrect matches. This

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

olution have some impact on the effectiveness, but not to a devastating degree. When watching these clips, the degradation in quality is evident, but the colour is well preserved.

The rest of the queries presented substantial difficulties to the baseline method. This method was unable to find any instances of the ad when using queries 2, 3, 7, 8, 9, and 10. To a human, all of these queries are quite watchable, but show some changes to the colour of the original clip. Four of the occurrences were identified using query 6, but they were not well separated from the incorrect matches.

Table 2 shows the results of applying the shot-based method to the same queries. The binary scoring variation of this technique was able to identify all five of the correct instances of the advertisement using nine of the queries. This method was unable to allow for changes in frame rate, as evidenced by the very poor result for query 5. This is to be expected, as the change of frame rate will cause slight changes (usually in the order of a few milliseconds) to the length of the shot, thereby causing the binary method to penalise the mismatch.

The separation shown for these results is unimpressive, however it is interesting to note that, in all cases except query 5, four of the five instances of the ad were identified as having a similarity of at least 75%, while the highest false match was always less than 20%. In each case, the same instance of the advertisement was identified with a similarity score only slightly higher than that of the false matches. Due to limitations in the cut-detection techniques used in this research, some errors are unavoidable. Examining the output from the cut-detection software, it was evident that there were several errors in the cut-detection for the fifth instance of the advertisement, despite the fact that all five instances are indistinguishable to a human observer. This accounts for the poor performance in identifying this instance of the query.

Both the logarithmic and categorical scoring methods had similar results to the binary method, although in all cases, the fifth instance of the query was not ranked in the top 5. In all cases, except for query 5, the fifth instance was listed in the top 10 results. While these methods perform slightly less strongly for most of the queries, the result for query 5 is much better. Both of these methods are able to clearly identify four of the five instances of the advertisement when using this query.

Multiple queries, single variant

The second data set was a 180-minute 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. We chose

Each query was chosen to represent a different style of content.

- Query 12 is an advertisement for a furniture retailer that represents a typical “sale” advertisement.
- Query 13 is for a food product and is delivered in a “documentary” style.
- Query 14 is for another food product, and attempts to appeal to a consumers emotional needs.
- Query 15 is a short advertisement (15 seconds) for a fast food chain.
- Query 16 is for a car and was chosen for the fact that it contains only two edits.
- Query 17 is also for a car, and represents the “cinematic” style of advertising.
- Query 18 is for a department store, and was chosen for its visual similarity to other advertisements in the clip.

Table 3 shows the results of querying the second data set with each of the queries using the baseline method. The recall for each of the queries is very good, and the precision is high for most of them. These results alone, however, do not give an accurate picture of the effectiveness. For query 12, one instance was identified very confidently, with a score of 97.37%. The other two were ranked below 40%, resulting in poor separation. All instances were identified and well separated for query 13, and the effectiveness for queries 14 and 16 was acceptable. The results for query 15 and 17 were weak, with false matches ranking at nearly 80% and more than 70% respectively. The three instances of query 18 were ranked above 80%, but six other advertisements were ranked the same, resulting in a separation of zero.

Table 4 shows the results of querying the second data set with each of the queries using the shot-length comparison method. The results here are more polarised than for the baseline method. Queries 13, 14, 15 and 17 all yielded very good results, with perfect precision and high separation. The results for query 12 were mediocre, with all three correct matches scoring in the top ten, but only one of them in the top three. Query 16 performed poorly, with none of the instances clearly identified. Query 18 was also disappointing, with only one of the three instances identified.

The reason for the poor performance for query 16 is easily explained, as the whole advertisement consists of only three shots, with two of these being very short and occurring at the end of the clip. Because of the small number of shots, it is much more likely that false matches will score highly. It also makes it more sensitive to errors in the cut-detection process.

Table 1: *Results of baseline queries on dataset 1.*

	Precision(s)	Recall(10)	HFM	Sep.	Sep./HFM
Query 1	1.00	1.00	34.21%	39.47%	1.15
Query 2	0.00	0.00	0.00%	0.00%	0.00
Query 3	0.00	0.00	18.42%	-18.42%	-1.00
Query 4	1.00	1.00	34.21%	26.32%	0.77
Query 5	1.00	1.00	50.00%	15.79%	0.32
Query 6	0.80	1.00	68.42%	-13.16%	-0.19
Query 7	0.00	0.00	0.00%	0.00%	0.00
Query 8	0.00	0.00	0.00%	0.00%	0.00
Query 9	0.00	0.00	0.00%	0.00%	0.00
Query 10	0.00	0.00	0.00%	0.00%	0.00
Query 11	1.00	1.00	31.58%	7.89%	0.25

Table 2: *Results of shot-length queries on dataset 1.*

	Precision(s)	Recall(10)	HFM	Sep.	Sep./HFM
<i>Binary Scoring</i>					
Query 1	1.00	1.00	14.58%	2.09%	0.14
Query 2	1.00	1.00	18.18%	2.27%	0.12
Query 3	1.00	1.00	18.18%	2.27%	0.12
Query 4	1.00	1.00	14.58%	2.09%	0.14
Query 5	0.00	0.00	14.58%	n/a	n/a
Query 6	0.80	1.00	20.00%	-2.50%	-0.13
Query 7	1.00	1.00	18.18%	2.27%	0.12
Query 8	1.00	1.00	18.18%	2.27%	0.12
Query 9	1.00	1.00	18.18%	2.27%	0.12
Query 10	1.00	1.00	18.18%	2.27%	0.12
Query 11	1.00	1.00	14.29%	3.57%	0.25
<i>Categorical Scoring</i>					
Query 1	0.80	1.00	27.22%	-11.11%	-0.41
Query 2	0.80	1.00	30.30%	-8.48%	-0.28
Query 3	0.80	1.00	30.30%	-8.48%	-0.28
Query 4	0.80	1.00	27.22%	-11.11%	-0.41
Query 5	0.80	0.80	18.89%	-10.00%	-0.53
Query 6	0.80	1.00	29.33%	-6.66%	-0.23
Query 7	0.80	1.00	30.30%	-8.48%	-0.28
Query 8	0.80	1.00	30.30%	-8.48%	-0.28
Query 9	0.80	1.00	30.30%	-8.48%	-0.28
Query 10	0.80	1.00	30.30%	-8.48%	-0.28
Query 11	0.80	1.00	23.33%	-10.00%	-0.43
<i>Logarithmic Scoring</i>					
Query 1	0.80	1.00	28.82%	-11.76%	-0.41
Query 2	0.80	1.00	37.82%	-17.95%	-0.47
Query 3	0.80	1.00	37.82%	-17.95%	-0.47
Query 4	0.80	1.00	28.82%	-11.76%	-0.41
Query 5	0.80	0.80	8.82%	-2.94%	-0.33
Query 6	0.80	1.00	41.55%	-23.24%	-0.56
Query 7	0.80	1.00	37.82%	-17.95%	-0.47
Query 8	0.80	1.00	37.82%	-17.95%	-0.47
Query 9	0.80	1.00	37.82%	-17.95%	-0.47
Query 10	0.80	1.00	37.82%	-17.95%	-0.47
Query 11	0.80	1.00	19.70%	-0.09%	-0.05

Table 3: *Results of baseline queries on dataset 2.*

	Precision(s)	Recall(10)	HFM	Sep.	Sep./HFM
Query 12	0.67	1.00	34.21%	-2.63%	-0.08
Query 13	1.00	1.00	23.68%	55.27%	2.33
Query 14	1.00	1.00	51.35%	29.73%	0.58
Query 15	0.50	1.00	78.95%	0.00%	0.00
Query 16	1.00	1.00	44.74%	34.21%	0.76
Query 17	1.00	1.00	71.05%	15.79%	0.22
Query 18	1.00	1.00	83.33%	0.00%	0.00

Table 4: *Results of shot-length queries on dataset 2.*

	Precision(s)	Recall(10)	HFM	Sep.	Sep./HFM
<i>Binary Scoring</i>					
Query 12	0.33	1.00	19.83%	-9.85%	-0.50
Query 13	1.00	1.00	35.71%	50.00%	1.40
Query 14	1.00	1.00	16.67%	80.00%	4.80
Query 15	1.00	1.00	30.56%	47.22%	1.55
Query 16	0.00	0.00	<10.00%	n/a	n/a
Query 17	1.00	1.00	55.56%	44.44%	0.80
Query 18	0.33	0.33	50.00%	< -40.00%	n/a
<i>Categorical Scoring</i>					
Query 12	0.33	1.00	42.07%	-22.99%	-0.55
Query 13	1.00	1.00	32.38%	60.95%	1.88
Query 14	1.00	1.00	27.11%	72.00%	2.66
Query 15	1.00	1.00	48.89%	44.44%	0.91
Query 16	0.00	0.00	<10.00%	n/a	n/a
Query 17	1.00	1.00	60.00%	40.00%	0.67
Query 18	0.33	0.33	81.33%	< -71.33%	n/a
<i>Logarithmic Scoring</i>					
Query 12	0.33	1.00	27.25%	-9.98%	-0.37
Query 13	1.00	1.00	43.43%	41.42%	0.95
Query 14	1.00	1.00	22.54%	71.36%	3.17
Query 15	1.00	1.00	29.13%	51.97%	1.78
Query 16	0.00	0.00	<10.00%	n/a	n/a
Query 17	1.00	1.00	49.61%	50.33%	1.01
Query 18	0.33	0.33	50.70%	< -40.70%	n/a

succession. Due to the nature of the advertisement, many of the cuts are difficult to detect, so errors are common. Query 18 is a short advertisement (half the length of the rest) and consists of only three shots, which again makes it sensitive to errors.

5 Conclusions

We have investigated several candidate methods for identifying co-derivative video clips. Because the existing methods are generally designed with a different application in mind, they have distinct disadvantages when used for detecting co-derivation. We have presented a new approach to this problem and found that it performs favourably in comparison to a baseline based on existing methods.

The shot-length comparison method was found to be extremely robust to changes in the video, including alterations to the colours as well as changes in bitrate, frame rate, resolution and introduction of analogue interference. Whilst the shot-length comparison method performs well in many situations, we found some limitations when it is applied to certain content. Queries that contain only a small number of edits could not be reliably identified. Similarly, errors in cut-detection led, in some cases, to considerable reduction in query effectiveness.

Both of these problems may be alleviated. By improving the cut-detection methods, errors due to cut-detection errors may be dramatically reduced. While the colour-based methods performed poorly, they may be a useful tool to confirm that a result identified by the shot-length comparison technique is accurate. Where insufficient shots are detected, colour or other characteristics may be used as a substitute; that is, it is evident prior to running the query that due to lack of information it is unlikely to succeed, so other fallback methods should be used.

References

Abdel-Mottaleb, M. (1999), Mpeg-7: A content description standard beyond compression, in 'Proc. of IEEE Midwest Symposium on Circuits and Systems', Las Cruces, New Mexico.

Adjeroh, D. A., Lee, M. C. & King, I. (1998), A distance measure for video sequence similarity matching, in 'Proc. of Intl. Workshop on Multimedia Database Management Systems', Dayton, OH, pp. 72–79.

Aslandogan, Y. A. & Yu, C. T. (1999), 'Techniques and systems for image and video retrieval', *Knowledge and Data Engineering* **11**(1), 56–63.

Boreczky, J. S. & Rowe, L. A. (1996), Comparison of video shot boundary detection techniques, in 'Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases', pp. 170–179.

surement with video signature, in 'Proc. of International Conference on Image Processing', Rochester, New York.

Day, N. & Martinez, J. M. (2001), 'Introduction to MPEG-7'.

Dimitrova, N. & Abdel-Mottaleb, M. (1997), 'Content based video retrieval by example video clip', *Storage and Retrieval for Still Image and Video Databases V, Vol. SPIE-3022* pp. 59–70.

Fushikida, K., Hiwatari, Y. & Waki, H. (1999), A content-based video query agent using feature-based image search engine, in 'Proc. of Third Intl. Conference on Computational Intelligence and Multimedia Applications', New Delhi, India.

Gusfield, D. (1997), *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press.

Hoad, T. & Zobel, J. (To appear), 'Methods for identifying versioned and plagiarised documents', *Journal of the American Society for Information Science and Technology*.

Huang, T. & Rui, Y. (1997), Image retrieval: Past, present, and future, in 'Intl. Symposium on Multimedia Information Processing'.

Lienhart, R. (2001), 'Reliable transition detection in videos: A survey and practitioner's guide.', *International Journal of Image and Graphics (IJIG)* **1**(3), 469–486.

Lienhart, R., Effelsberg, W. & Jain, R. (1998), VisualGREP: A systematic method to compare and retrieve video sequences, in 'Storage and Retrieval for Image and Video Databases (SPIE)', pp. 271–283.

Liu, X., Zhuang, Y. & Pan, Y. (1999), A new approach to retrieve video by example video clip, in 'ACM Multimedia (2)', pp. 41–44.

Pass, G., Zabih, R. & Miller, J. (1996), Comparing images using color coherence vectors, in 'ACM Multimedia', pp. 65–73.

Salton, G. & McGill, M. J. (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill, New York.

Sankoff, D. & Kruskal, J. B. (1999), *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, CSLI Publications.

Shan, M.-K. & Lee, S.-Y. (1998), Content-based video retrieval based on similarity of frame sequence, in 'Proc. of Intl. Workshop on Multimedia Database Management Systems', Dayton, OH, pp. 72–79.

Tan, Y. P., Kulkarni, S. R. & Ramadge, P. J. (1999), A framework for measuring video similarity and its application to video query by example, in 'Proc. of ICIP', Kobe, Japan.

Wu, Y., Zhuang, Y. & Pan, Y. (2000), Content-based video similarity model, in 'ACM Multimedia'.

Yeung, M. & Liu, B. (1995), Efficient matching and clustering of video shots, in 'Proc. of the IEEE Intl. Conference on Image Processing', Vol. 1, Washington, D.C., pp. 338–341.

Zhao, L., W.Qi, S.Z.Li, S.Q.Yang & Zhang, H. (2001), Content-based retrieval of video shot using the improved nearest feature line method, in 'Proc. of ICASSP', Salt Lake City.