Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Vision and Image Understanding 103 (2006) 80–88
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Finding and identifying unknown commercials using repeated video sequence detection

John M. Gauch [a,*], Abhishek Shivadas [b]

[a] EECS Department, University of Kansas, USA
[b] Veatros LLC, USA

Received 7 March 2005; accepted 18 March 2006

Abstract

Automated commercial detection can be performed by matching features extracted from commercials or by detecting embedded codes that are hidden within the commercial. In both cases, it is necessary to create a database of known commercials that contain the information necessary for detection. In this paper, we present an automated technique for locating previously unknown commercials by continuously monitoring broadcast television signals. Our system has two components: repeated video sequence detection, and feature-based classification of video sequences as commercials or non-commercials. Our system utilizes customized temporal video segmentation techniques to automatically partition the digital video signal into semantically sensible shots and scenes. As each frame of the video source is processed, we extract auxiliary information to facilitate repeated sequence detection. When the video transition marking the end of the shot/scene is detected, we are able to rapidly locate all previous occurrences of the video clip. In order to classify video sequences as commercials or non-commercials, we extract a number of features from each video sequence that characterize the temporal and chromatic variations within the clip. We have evaluated three classification approaches using this information and have consistently achieved over 93% accuracy identifying new commercials and non-commercials as they are broadcast.

Keywords: Digital video; Temporal segmentation; Repeated sequence detection; Video classification; Commercial detection

1. Introduction

The ever-growing television broadcast market has spawned a second industry, one that tracks advertising activity in across a variety of media. This information is valuable for competitive marketing analysis, advertising planning, and as a barometer for the advertising (and thus broadcasting) industry health. In order to carry out a successful marketing campaign it is important to monitor the commercials of competitors and plan television/cable advertising accordingly. This can be done by hiring employees to watch broadcasts 24 by 7 and manually record new commercials as they appear. Given the large number broadcasters and channels, the cost of this approach is error prone and prohibitively expensive.

The goal of our research is to develop automated techniques for detecting previously unknown commercials as they are broadcast. This task is significantly harder than detecting the occurrence of known commercials because we do not know exactly what audio/video signals the new commercials contain.

2. System architecture

To identify new commercials we have adapted and combined three video processing techniques: temporal segmentation, repeated video sequence detection, and video sequence classification. The new commercials we identify are then used by our commercial detection system to generate marketing reports for the broadcasting industry. The flow of information is illustrated in Fig. 1.

Video signals are digitized using commodity hardware and features are extracted by our preprocessing software.

* Corresponding author.
E-mail address: jgauch@ku.edu (J.M. Gauch).

1077-3142/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.03.002

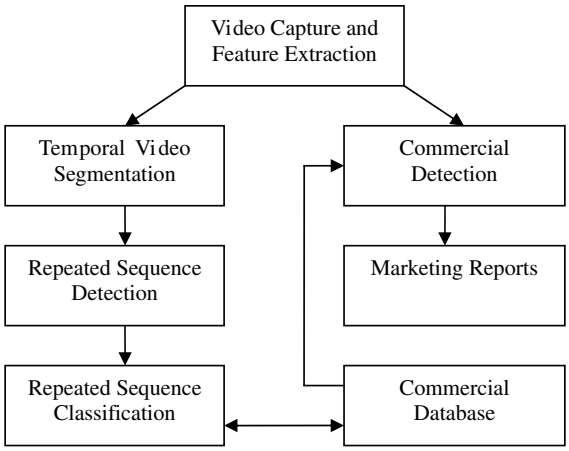

Fig. 1. System architecture.

We then perform temporal segmentation to partition the video signal we are monitoring into individual shots, which are meaningful sized units of video content for subsequent analysis. Our system is designed to run continuously in real time, so we use temporal properties of color moments to detect cuts, fades and dissolves. We have intentionally chosen our segmentation parameters so the video is over-segmented. The resulting shots are typically 1–10 s long.

The repetition patterns of shots provide a valuable commercial identification cue. To detect repeated shots in real-time, we use shot level hashing based on color moment features of video frames. Our repeated sequences consist of sets of shots that occurred in the same order two or more times during the 24 h time period being monitored. These videos are typically 10–60 s long for commercials and promotional materials and may be several minutes long for non-commercials.

The final phase of our automated system is video sequence classification. In order to evaluate the effectiveness of our classification algorithms, we have established ground truth for a large collection of repeated video sequences, labeling each as either a commercial or non-commercial.

As we describe the details of our system, we review related work in this field that has influenced our research.

3. Temporal video segmentation

The process of dividing a video signal into semantically sensible video clips has been an important component in video indexing and retrieval systems for some time. Early work in this area [1–3] described techniques for detecting editing effects such as cuts, fade, dissolves, and spatial effects. These temporal video segmentation methods focused primarily on inter-frame pixel differences. To improve the quality of temporal segmentation, a wide range of low level and high level features have been proposed. For example, color histogram similarity [4,5], change detection based on edge detection features [6–8], video self-similarity analysis [9], difference detection of foveation points [10], and explicit modeling of fade and dissolve transitions [11,12]. To address speed requirements, a number of temporal segmentation approaches have been developed for the compressed video domain, operating directly on MPEG streams without full video decompression [13–16]. Many of these approaches for shot boundary detection have been quantitatively compared at annual TRECVID. The most successful systems typically combine a wide variety of features described above with careful selection of thresholds and other parameters to obtain recall and precision rates up to 94% for cut detection and 83% for gradual transitions such as fades and dissolves [17].

For our current application, computational speed and high recall are more important than high precision or labeling of transition types. For this reason, we make use of temporal variations in color moments to locate the video transitions that partition the video into shots. We calculate our *video change metric* C by integrating the sum absolute differences of the first three color moments for a range of temporal displacements. Specifically, we calculate:

$$C(t,n) = \sum_{dt=1..n} |V(t) - V(t-dt)|, \tag{1}$$

where our color moment vector is given by

$$V(t) = (M(t,r), M(t,g), M(t,b), S(t,r), S(t,g), S(t,b), K(t,r), K(t,g), K(t,b)) \tag{2}$$

and first, second, and third order color moments are calculated for each color channel using:

$$M(t,c) = \frac{1}{N} \sum_{xy} I(x,y,t,c), \tag{3}$$

$$S(t,c) = \sqrt{\frac{1}{N}\left(\sum_{xy} [I(x,y,t,c) - M(t,c)]^2\right)}, \tag{4}$$

$$K(t,c) = \sqrt[3]{\frac{1}{N}\left(\sum_{xy} [I(x,y,t,c) - M(t,c)]^3\right)}. \tag{5}$$

In a previous project, we calculated $C(t,n)$ for $n = 1, 5, 10$ and used multiple thresholds to detect and classify shot transitions [18]. For this project, we conducted a series of experiments and calculated $C(t,n)$ values for 24 h of video with $n = 1, 5, 10, 15, 20$. We found that these functions were qualitatively very similar, with local maxima occurring at almost identical times. This can be explained because the summation for $C(t,n) = C(t,n-1) + |V(t) - V(t-n)|$. The major difference between these functions is the width of the peaks, which is proportional to the window size. To improve the speed of our temporal segmentation algorithm, we now use a single $C(t,n)$ function to identify video transitions. Specifically, we find local maxima of $C(t,10)$ that are greater than our derived segmentation threshold T_{10}, and mark these points as video transitions. We do not attempt to

label transitions as either cuts, fades or dissolves because this classification information is not needed by our subsequent algorithms.

As shown in Fig. 2, our video change metric has a different dynamic range at different times. Since it varies over time and/or channel, we base our segmentation threshold T_{10} on the statistical properties of $C(t,10)$. Specifically, we estimate the mean M_c and the standard deviation σ_c of $C(t,10)$ over a specified time period, and then calculate our threshold using $T_{10} = M_c + 2\sigma_c$. In Fig. 2, we show the statistical properties of $C(t,10)$ for 24 h of video data taken from a typical television channel. The mean and standard deviation of $C(t,10)$ vary from 1.5 to 4.2 and from 3.0 to 7.2, respectively, during the day as different programs are broadcast and different commercials are played. As a result, T_{10} based on hourly statistics varies between 7.5 and 18.5. This is a larger variation in values than we expected given the number of video frames in one hour of video. Since our ultimate goal is to identify commercials using repeated video sequences, it is important to perform temporal segmentation in a consistent way during the whole day. For this reason, we base our threshold on the daily statistics. In this case, the daily mean, standard deviation for $C(t,10)$ were 2.7 and 5.2, respectively, so $T_{10} = 13.1$.

The computational speed of our algorithm was evaluated using pre-recorded color moment values. We obtained results that were over 1300 times faster than real-time. Segmentation of 24 h of video was completed in 65.9 s on a 2 GHz Xeon processor running Linux. The accuracy of our temporal segmentation algorithm was evaluated by calculating the recall and precision over one hour of a typical news broadcast obtained from CNN. For this hour we found 961 video transitions, of which 618 were cuts, 84 were fades, 189 dissolves, and 70 were other transitions. Using our statistically derived threshold for T10 we obtained a recall of 92% and precision of 79% for this data set. Informal experiments with other broadcast news videos, e.g., Fox News, produce similar results.

4. Repeated sequence detection

The problem of repeated sequence detection appears in several guises. In database applications, an image or video clip could be used as a query to search for similar images or video clips in a large archive. To avoid the computational expense of brute force matching, image features can be pre-calculated and used to define metric spaces and multi-dimensional trees can be created to search for images or video clips of interest [19–22]. As with any database application, these data structures must be updated as images or videos are added or removed from the database.

Several of methods have been devised to detect repeated video sequences within news or sports broadcasts for the purposes of identifying interesting content. [23] uses a hidden Markov model to identify slow motion replay segments in sports videos. Their system uses the zero crossings of intensity differences and color histogram features to classify video as either slow motion, still, editing effect, normal play or normal replay. Satoh [24] decomposes input video into fragments of highly similar frames and uses brute force color histogram intersection to detect identical shots. This approach is too slow to scale to large video archives. Ide et al. [25] performs temporal segmentation to partition news videos into short clips and then uses closed captions to track the repetition of topics in news videos. Because their approach is text-based, it may miss video repetitions where the audio/text have changed.

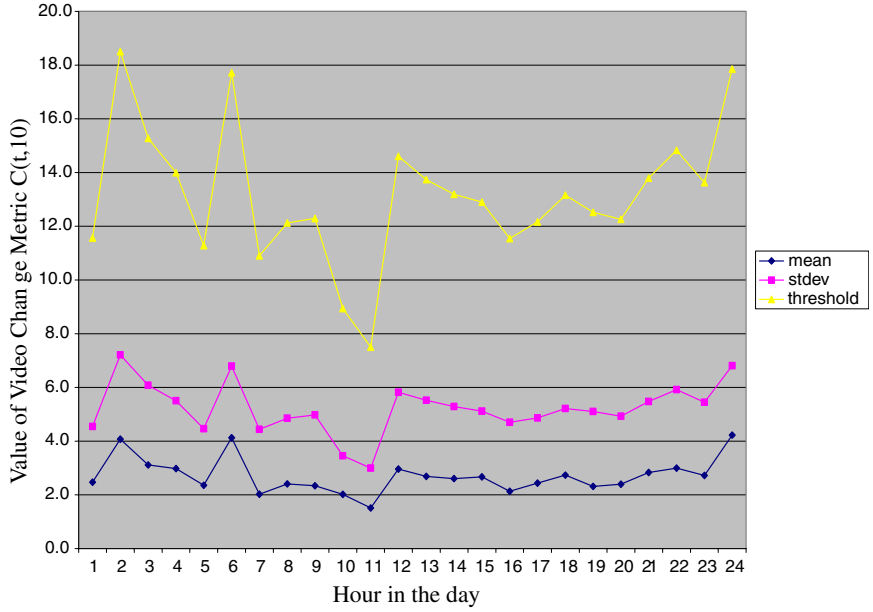

Fig. 2. Statistical properties of $C(t,10)$ over a 24 h period for a typical television channel.

Finally, a number of techniques have been developed for hashing images and/or video clips to improve the performance of image/video retrieval. These approaches differ in which features are used to characterize the image and/or video, and also in how the hash table is represented. Color image features are used by [26,27] to create compact binary signatures for hashing. Their approaches outperform constant and variable bin allocation approaches for color-based image retrieval tasks. Temporal variations in luminance are encoded by [28] to create M-bit hash words for video hashing. Their approach is robust to bit-errors introduced by minor video distortion, but it ignores colors properties. Perfect and near-perfect hashing functions have been proposed to minimize the number of collisions that occur while retrieving images based on feature-based hash indices [29,30]. These approaches naturally provide rapid search, but the devising perfect or near-perfect hash functions is complex and costly.

To perform repeated video sequence detection, we need a compact description of video frames that can be used to rapidly locate similar video frames. In this section, we describe the image features we used to characterize video frames as they are processed, and the hash table we have developed to rapidly locate repeated frames based on these image features.

A video sequence can be characterized by the distribution of (r,g,b) intensities within each frame $I(x,y,t,c)$. There are numerous ways to describe this distribution. It can be represented using a single three-dimensional color histogram with 256^3 entries, or three one-dimensional histograms with 256 entries, but these representations are too large for 24 h video analysis applications. For our video processing applications, we have found that color moments $V(t)$ described above provide a compact and effective representation of these distributions.

Our goal is to identify all repeated video sequences that occur within a 24 h period. At 30 frames per second, there are $T = 2{,}592{,}000$ video frames in one day. Storing the nine color moments above as floating point values requires 36 bytes per frame or 88 MB per day. To reduce our storage requirements, we quantize each color moment to 256 levels so they can be stored in nine bytes per frame or 22 MB per day.

To rapidly identify repeated video frames based on color moment properties, we store all of the color moment vectors $V(t)$ for a 24 h period in a hash table. When data are inserted, we calculate the hash index $H(t)$ based on the moment values $V(t)$ and store the video frame index t in the appropriate location in the hash table. Later, when we search the hash table for similar video frames, we again calculate $H(t)$ and return the set of all frame indices that have been previously stored in this hash table location.

As shown in Fig. 3, the distribution of standard deviation and skew moments resembled normal distributions with means of 54.7 and 46.5 and standard deviations of 16.6 and 16.9, respectively. The color means had a much broader distribution with mean 97.1 and a standard deviation of 45.4.

To minimize the computational cost of inserting data into the hash table, we must define a hash function $H(t)$ based on the values in $V(t)$ that minimizes the number of hash table collisions. When $V(t)$ values are uniformly distributed the quantized $V(t)$ values can be concatenated together to form a single large value $H(t)$ that is also uniformly distributed. Unfortunately, the color moments we obtain from typical news broadcasts are not uniformly distributed (see Fig. 3). For this reason, we have developed and compared a number of functions that attempt to obtain uniformly distributed hash values from color moments. The three functions that we found to be most effective are:

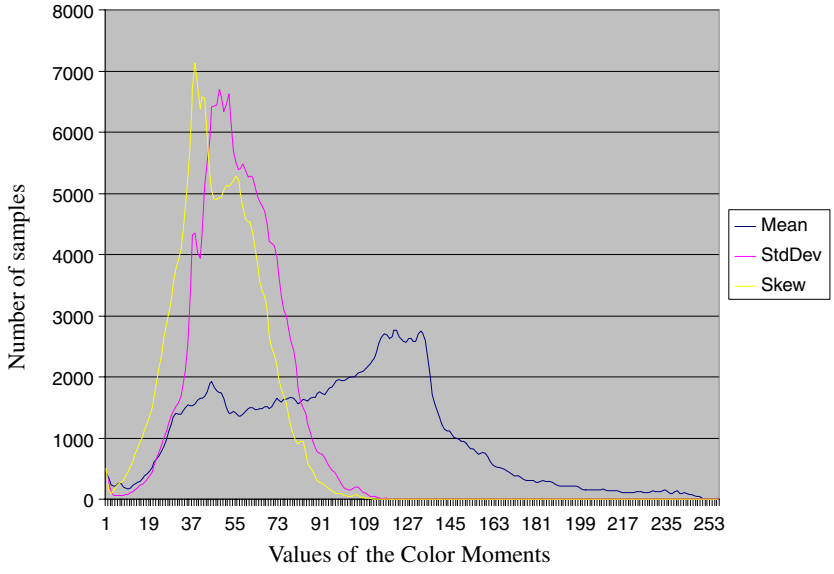

Fig. 3. Distributions of mean $M(t,c)$, standard deviation $S(t,c)$, and skew moments $K(t,c)$ derived from 24 h of broadcast television.

Table 1
Comparison of the collision rates for three color moment hash functions

Function	0 collisions (%)	<5 collisions (%)	<10 collisions (%)	Average collisions
$H1$	86.58	99.41	99.74	1.386
$H2$	86.77	99.85	100.00	0.181
$H3$	86.94	99.88	100.00	0.177

$$H1(t) = \prod_{i=1..n} V_i(t) \bmod T, \tag{6}$$

$$H2(t) = \sum_{i=1..n} i! V_i(t) \bmod T, \tag{7}$$

$$H3(t) = \sum_{i=1..n} Q^i V_i(t) \bmod T. \tag{8}$$

Hash function $H1(t)$ calculates the product of moment values $V(t)$, producing values in the range $[0..256^9 - 1]$, which are mapped to size of the hash table $[0..T-1]$ using the modulo function. Hash function $H2(t)$ calculates a weighted sum of moment values $V(t)$. Weights are calculated based on the moment index in order to increase the distribution $H2(t)$ values. Without these $i!$ weights, the range would be [0..255*9] which is not acceptable for a hash function. Hash function $H3(t)$ calculates a weighted sum of moment values where the Q^i weights combine the $V(t)$ values in base Q. When Q is equal to 2^k, this is equivalent to shifting and adding the k most significant bits of the moment values to obtain $H3(t)$.

To select a hash function for this application, we compared the number of collisions that occurred while inserting 24 h of moment data (2,592,000 moment vectors) into a hash table with size $T = 10{,}000{,}001$. All three hash functions behaved similarly (see Table 1). Over 86% of the time moments are inserted into the hash table with zero collisions, and 99% of the data was inserted with 10 or fewer collisions. Hash function $H1$ had a moderate number of insertions that resulted more than 10 collisions. For this reason, the average number of collisions for $H1$ was significantly worse than $H2$ and $H3$. Since $H3$ with $Q = 32$ provided a modest performance improvement over $H2$, we use $H3$ for our repeated sequence detection research.

Our repeated sequence detection process consists of three components, video frame hashing, similar video sequence filtering, and repeated sequence validation. When each new shot is added to the hash table, we use hash table lookup to identify video sequences in the archive that are potentially similar to the input video sequence using a voting scheme. We then use the video sequence filtering component to determine if these potentially similar video sequences are truly similar to the input video sequence. This filtering is based on the number of frames that have the same hash values, the relative lengths of the shots, and the mean color moment vectors for each shot.

To perform repeated sequence validation, we align and compare each new shot A to all potentially similar shots B that remain after video sequence filtering. If we let $V_a(t)$ represent the color moment vectors for shot A of length L_a, and let $V_b(t)$ represent the color moment vectors for shot B of length L_b, the mean color moment vectors for both shots can be calculated using

$$M_a = \frac{1}{L_a} \sum_{t=1..L_a} V_s(t) \text{ and } M_b = \frac{1}{L_b} \sum_{t=1..L_b} V_b(t). \tag{9}$$

The difference between shot A and shot B at displacement dt can be calculated using

$$D_{a,b}(dt) = \frac{1}{L} \sum_{t=1..L} |(V_a(t) - M_a) - (V_b(t+dt) - M_b)|, \tag{10}$$

where the length of overlap is given by L. To determine if clip A matches clip B, we perform a brute force search of all possible dt values with non-zero shot overlap and we compare the minimum value of $D_{a,b}(dt)$ to our predefined matching threshold T_D. When a video clips starts with a sharp cut or easily detected transition, the optimal alignment displacement dt is zero. Hence, we avoid unnecessary calculations by terminating the moment comparison loop as soon as we find $D_{a,b}(dt) < T_D$. Finally, we merge adjacent shots that are repeated in the same order elsewhere in our archive. By combining multiple 2–3 s shots, we are able to obtain repeated video sequences that are typically 10–60 s long.

In our experiments with 24 h of digital video, the repeated sequence detection phase required 245 s on a 2 GHz Xeon processor running Linux. Because hash table insertion is a constant time operation, our repeated sequence detection algorithm would scale linearly if longer time periods are considered and the hash table size is increased accordingly. Hence, we would expect to be able to process 72 h worth of digital video in approximately 750 s of cpu time. To evaluate the accuracy of our repeated sequence detection algorithm we selected 50 news story shots and manually identified all repetitions within a 24 h period. For this collection we achieved 86% recall and 91% precision detecting repeated video sequences.

5. Video sequence classification

The identification of commercials in television signals for filtering or monitoring purposes is an important video classification task. The most common features used to identify commercials are the black/silent frames that mark the start of commercials and the relatively short lengths of shots within commercials [31,32]. Other useful features include the amount of object/camera motion and the ration of hard cuts to dissolves/fades [33], color histograms of keyframes [34,35], the presence of channel logos [36,37], and letterbox detection [38].

Classification of video clips as commercials or non-commercials based on these incommensurate features presents a number of challenges. One approach is to select separate thresholds for each feature. Another approach is to select relative weights for each feature and a single threshold to identify commercials. Genetic algorithms can be used to

choose these weights and thresholds [38]. More advanced classification algorithms make use of hidden Markov models and a Viterbi decoder to detect and label commercials [36,37], or k-means clustering where k is chosen using the Dunn index [39].

Many commercial classification algorithms operate at the shot level, so a number of short shots must be heuristically combined to identify whole commercials [31]. This is a difficult task which is complicated by potential classification errors at the shot level. To address this problem, temporal coherence (similarity of shots within a commercial) has been used to develop an adaptive classification algorithm using a *time-constraint boost* [40].

The repetition of video content can be used for a number of purposes. For example, in many news broadcasts video clips of important events are replayed as the story develops. This can be used to track the evolution of news stories [12,41]. Repetition of commercials provides valuable information about which shots belong together in a single commercial. This information was used by [42] who used repeated keyframes together with shot features to identify commercials and remove them from broadcasts.

The primary goal of our video sequence classification process is to classify a repeated sequence either as commercial or non-commercial. As a first step, we calculate five features for each repeated video sequence: The number of sharp cuts per second C in the video sequence. The number of dark monochrome video frames two seconds before D_b and two seconds after D_a the shot that have mean intensity and intensity variance values below given thresholds. The mean μ and standard deviation σ of the absolute color moment differences between adjacent frames in the sequence. These values are calculated using

$$\mu = \frac{1}{N-1}\sum_{n=1}^{N-1}\frac{1}{M}\sum_{m=1}^{M}|V(n,m)-V(n+1,m)|, \tag{11}$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N-1}\left(\left(\frac{1}{M}\sum_{m=1}^{M}V(n,m)-V(n+1,m)\right)-\mu\right)^2}. \tag{12}$$

We implemented and evaluated three video classification algorithms based on the feature vectors above: nearest centroid classification, weighted binary classifier voting, and k-nearest neighbor (kNN) classification.

Our first classification algorithm is the fastest and easiest to implement. The centroids are calculated for all commercials and for all non-commercials feature vectors in the training set. The normalized Euclidian distances between the input sequence features and these centroids are calculated, and the sequence is classified based on which set it is closest to.

Our second approach assumes that commercials and non-commercials have distinct ranges of values for each of our five video sequence features. Hence, we apply thresholds to obtain five binary classifiers. These results can be combined using a simple majority rules approach, or relative weights can be assigned to each feature based on classification accuracy. To optimize the performance of this approach, we performed an exhaustive search for the set of thresholds and weights that has the fewest classification errors.

Finally, to implement kNN classification, we calculated the normalized Euclidean distance between the feature vector for the input sequence and all pre-classified sequences, and selected the closest k results. We use the standard deviations of the six features in the training set to determine the normalizing factors. The input sequence is classified according to a simple majority rule. If $k/2$ of the closest k shots are commercials, the input sequence is a commercial. Otherwise it is identified as a non-commercial.

6. Experimental results

The temporal video segmentation and repeated sequence detection algorithms described in previous sections were used to process 72 h of television programming. This created a data set of 575 repeated video clips that were viewed and manually classified. Clips (390) were identified as commercials and 185 were identified as non-commercials. For testing purposes, we chose half of these clips for training, and used the other half for validation. We used even/odd shot identifiers to partition the data set to ensure that clips were uniformly distributed over time and to minimize experimental bias.

To evaluate our classification algorithms, we used two approaches. We calculated classification *accuracy* A based on the number of correct and incorrect commercial and non-commercial classifications using the following: $A = (\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})$, where TP = number correct commercials (true positives), FP = number incorrect commercials (false positives), FN = number incorrect non-commercials (false negatives), TN = number correct non-commercials (true negatives). We computed the *precision* and *recall* for both commercials (C_r, C_p) and non-commercials (N_p, N_r) using: $C_\mathrm{p} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, $C_\mathrm{r} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, $N_\mathrm{p} = \mathrm{TN}/(\mathrm{TN}+\mathrm{FN})$, $N_\mathrm{r} = \mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$. One effective way to combine recall and precision values is to calculate α-weighted harmonic mean, known as the F-measure.

$$F = \left(\frac{\alpha}{R}+\frac{(1-\alpha)}{P}\right)^{-1}. \tag{13}$$

When $\alpha = 0.5$ this yields the following F-measures for commercials F_c and non-commercials F_n.

$$F_\mathrm{c} = \frac{2C_\mathrm{p}C_\mathrm{r}}{(C_\mathrm{p}+C_\mathrm{r})}, \tag{14}$$

$$F_\mathrm{n} = \frac{2N_\mathrm{p}N_\mathrm{r}}{(N_\mathrm{p}+N_\mathrm{r})}. \tag{15}$$

These values were combined into a single *quality* metric Q by weighting commercials and non-commercials equally and averaging their corresponding F measures.

$$Q = \frac{C_p C_r}{(C_p + C_r)} + \frac{N_p N_r}{(N_p + N_r)}. \quad (16)$$

In the special case where the number of correctly identified commercials TP equals the number of correct non-commercials TN, Q becomes identical to A. When TP $\gg$ TN or vice versa, the Q metric is more sensitive to classification errors than the A measure, and will yield lower values.

In our first experiment, we calculated the centroids of all of the commercials, and the centroid of all non-commercials in our training set. The commercials in the testing set were classified based on the Euclidean distances to these centroids. We found that F_c, F_n, and Q were all equal to 0.86, and below our shot classification target.

In our second experiment, we performed classification using our five features separately and then using a weighted binary classification. To identify the thresholds for classification, we conducted a brute force search of potential threshold values from zero to the maximum value of each feature at 0.01 increments. The thresholds that maximized our Q and A metrics are shown in Tables 2 and 3 together with corresponding F-measures for commercials and non-commercials. Our best $Q = 0.87$ and our best $A = 0.88$, which is only a modest improvement over our nearest centroid-based classification.

We considered three methods for combining binary classifications. In the first, we averaged the output of our five classifiers. If the number of commercial classifications was greater than the number of non-commercials, the shot was classified as commercial. Otherwise the shot was classified as a non-commercial. We also considered weighted averages where the weights were given by the best Q value or best A value obtained by the individual binary classifiers. This gives more weight to the binary classifiers that perform best. Our results are shown in Table 4. Our three weighted binary classification methods produced identical results, with $Q = 0.88$ and $A = 0.89$, which is a modest improvement over previous methods.

Table 2
Experimental results for binary classification optimizing Q metric

Feature	Best Q threshold	Best F_c	Best F_n	Best Q
D_a	1	0.84	0.76	0.80
D_b	1	0.85	0.78	0.82
C	0.34	0.91	0.81	0.86
μ	0.67	0.91	0.83	0.87
σ	1.50	0.91	0.82	0.87

Table 3
Experimental results for binary classification optimizing A metric

Feature	Best A threshold	Best F_c	Best F_n	Best A
D_a	1	0.84	0.76	0.81
D_b	1	0.85	0.78	0.83
C	0.26	0.91	0.80	0.88
μ	0.67	0.91	0.83	0.88
σ	1.50	0.91	0.82	0.88

Table 4
Experimental results for weighted binary classification

Combination	F_c	F_n	Q	F_c	F_n	A
Equal weights	0.92	0.84	0.88	0.92	0.84	0.89
Q weighted	0.92	0.84	0.88	0.92	0.84	0.89
A weighted	0.92	0.84	0.88	0.92	0.84	0.89

Table 5
Equally weighted kNN classification results for $k = 1..5$

k	F_c	F_n	Q	A
1	0.94	0.88	0.91	0.92
2	0.94	0.87	0.91	0.92
3	0.93	0.86	0.90	0.91
4	0.93	0.85	0.89	0.91
5	0.92	0.84	0.88	0.89

Table 6
Rank weighted kNN classification results for $k = 1..5$

k	F_c	F_n	Q	A
1	0.94	0.88	0.91	0.92
2	0.94	0.88	0.91	0.92
3	0.95	0.89	0.92	0.93
4	0.94	0.88	0.91	0.92
5	0.94	0.88	0.91	0.92

In our final experiment, we evaluated two methods for kNN classification based on our feature vectors. Our first approach weighted the k classifications equally when calculating the result. We also experimented with a weighted classification scheme where the relative contribution of each neighbor's classification decreases linearly with rank position k. Specifically we used $W(k) = (K - k + 1)/K$. We varied the number of neighbors k from 1 to 5 and calculated the F measures for commercials and non-commercials, and the corresponding quality Q, and accuracy A metrics. Both approaches performed similarly and produced excellent classification results (see Tables 5 and 6). Our best classification results were obtained for weighted kNN classification with $k = 3$, where $Q = 0.92$ and $A = 0.93$.

When we performed a statistical analysis of the classification results of our three approaches, we found that the performance of the kNN classifier was significantly better than both the nearest centroid classification ($t = 0.018$) and weighted binary classifier voting ($t = 0.033$) using a traditional two tail t test on our 575 classification decisions. For this reason, we have deployed this classifier in our new commercial detection system.

7. Conclusions

This paper describes an effective automated technique for locating previously unknown commercials by continuously monitoring broadcast television signals. Our temporal video segmentation uses changes in color moments to

identify cut, fade, and dissolve transitions in real time as the video is being captured. As the resulting shots are archived, we store frame descriptors in a hash table, which are later used to perform repeated video sequence detection. Adjacent repeated shots are combined to obtain video sequences corresponding to entire commercials or repeated programming.

To classify video sequences as commercials or non-commercials, we extract five features from each video sequence that characterize the temporal and chromatic variations within the clip (cuts per second, number of monochrome frames before and after the shot, and the mean and standard deviation of color moment differences within the clip). After manually classifying the 575 repeated sequences detected in 72 h of programming, we evaluated weighted binary classification, kNN classification, and nearest centroid classification, and found that the kNN approach yielded the best results, with an F measure of 0.95 for commercial detection, 0.89 for non-commercial detection, and overall classification quality of 0.92 and accuracy of 0.93.

Our system for identifying new commercials is now in continuous operation monitoring multiple channels and building a library of known commercials. This library is used by our known commercial detection system to gather marketing information in real-time. As our commercial database grows in size, we will explore additional video sequence features and classification algorithms to continue to improve overall performance.

References

[1] A. Nagasaka, T. Tanaka, Automatic video indexing and full-video search for object appearances, Visual Database Systems II, Elsevier Science Publishers B.V., 1992, pp. 119–133.

[2] A. Hampapur, T. Weymouth, A.R. Jain, Digital video segmentation, in: Proc. of the Second ACM Int. Conf. on Multimedia, San Francisco, 1994, pp. 357–364.

[3] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, Automatic partitioning of full-motion video, Multimedia Syst. 1 (1993) 10–28.

[4] M.J. Swain, D.H. Ballard, Color Indexing, Int. J. Comput. Vis. 7 (1) (1991) 11–32.

[5] U. Gargi, S. Oswald, D. Kosiba, S. Devadiga, R. Kasturi, Evaluation of video sequence indexing and hierarchical video indexing, in: Proc. of SPIE Conf. on Storage and Retrieval in Image and Video Databases, San Jose, 1995, pp. 1522–1530.

[6] R. Zabih, J. Miller, K. Mai, A feature-based algorithm for detecting and classifying scene breaks, in: Proc. of the ACM Int. Conf. on Multimedia, San Francisco, 1995, pp. 189–200.

[7] R. Zabih, J. Miller, K. Mai, A feature-based algorithm for detecting and classifying production effects, Multimedia Syst. 7 (2) (1999) 119–128.

[8] G. Gormley, Scene break detection and classification of digital video sequences using the method of edge detection, Technical Report, School of Computer Applications, Dublin City University, 1999, 119–128.

[9] M. Cooper, J. Foote, Scene boundary detection via video self-similarity analysis, in: Int. Conf. on Image Processing, Thessalonki, Greece, 2001, pp. 378–381.

[10] G. Boccignone, A. Chianese, V. Moscato, A. Picariello, Foviated shot detection for video segmentation, IEEE Trans. Circuits Syst. (in press).

[11] B.T. Truong, C. Dorai, S. Venkatesh, New enhancements to cut, fade, and dissolve detection processes in video segmentation, in: Proc. of the Eighth ACM Int. Conf. on Multimedia, Marina del Rey, 2000, pp. 219–227.

[12] J. Miadowicz, Story tracking in video news broadcasts, Ph.D. Dissertation, Electrical Engineering and Computer Science, University of Kansas, 2004.

[13] R. Kasturi, S. Strayer, U. Gargi, S. Antani. An evaluation of motion and MPEG based methods for temporal segmentation of video, Technical Report CSE-98-014, Department of Computer Science and Engineering, Penn State University, 1998.

[14] A.F. Smeaton, J. Gilvarry, G. Gormley, B. Tobin, S. Marlow, N. Murphy, An evaluation of alternative techniques for automatic detection of shot boundaries in digital video, in: Proc. of the Third Irish Machine Vision and Information Processing Conf., Dublin, 1999.

[15] K. Shen, E.J. Delp, A fast algorithm for video parsing using MPEG compressed sequences, in: Proc. of the IEEE Int. Conf. on Image Processing, Washington, DC, 1995, pp. 252–255.

[16] C. Taskiran, E.J. Delp, Video scene change detection using the generalized sequence trace, in: IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Seattle, WA, 1998, 2961–2964.

[17] A. Amir, M. Berg, S.F. Chang, W. Hsu, G. Iyengar, C.Y. Lin, M. Naphade, A. Natsev, C. Neti, H. Nock, J.R. Smith, B. Tseng, Y. Wu, D. Zhang, IBM research TRECVID-2003 video retrieval system, in: Proceedings of the TRECVID 2003 Workshop, 2003.

[18] K.M. Pua, J.M. Gauch, S.E. Gauch, J.Z. Miadowicz, Real time repeated video sequence identification, Comput. Vis. Image Understand. 93 (2004) 310–327.

[19] A. Guttman, R-Trees: a dynamic index structure for spatial searching, in: Proc. of the 1984 ACM SIGMOD Conf., Minneapolis, 1984, pp. 47–57.

[20] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, R. Barber, Efficient and effective querying by image content, J. Intell. Informat. Syst. 3 (1994) 231–262.

[21] T. Bozkaya, M. Ozsoyoglu, Indexing large metric spaces for similarity search queries, ACM Trans. Database Syst. 24 (3) (1999) 361–404.

[22] S. Park, W.W. Chu, Similarity-based subsequence search in image sequence databases, Int. J. Image Graph. 3 (1) (2003) 31–53.

[23] H. Pan, P. van Beek, M.I. Sezan, Detection of slow-motion replay segments in sports video for highlights generation, in: IEEE Int. Conf. on Acoustic, Speech and Signal Processing, 2001, pp. 1649–1652.

[24] S. Satoh, News video analysis based on identical shot detection, in: Proc. 2002 IEEE Intl. Conf. on Multimedia and Expo, 2002, pp. 69–72.

[25] I. Ide, N. Katayama, S. Satoh, Visualizing the structure of a large scale news video corpus based on topic segmentation and tracking, in: Proc. ACM Multimedia 2002 Workshop on Multimedia Information Retrieval, Juan Les Pins, France, 2002.

[26] V. Chitkara. Color-based image retrieval using compact binary signatures. Master's thesis, Dept. of Computing Science, University of Alberta, 2001.

[27] T.C. Hoad, J. Zobel, Fast video matching with signature alignment, in: Proc. of the 5th ACM SIGMM Int. Workshop on Multimedia Information Retrieval, Berkeley, 2003, pp. 262–269.

[28] J.C. Oostveen, A.A.C. Kalker, J.A. Haitsma, Visual hashing of digital video: applications and techniques, in: SPIE Applications of Digital Image Processing XXIV, San Diego, 2001, pp. 121–131.

[29] C.L. Sabharwal, S.K. Bhatia, Perfect hash table algorithm for image databases using negative associated values, Pattern Recognit. 28 (7) (1995) 1091–1101.

[30] S.K. Bhatia, C.L. Sabharwal, Near perfect hash table for image databases, in: Proc. of the 1996 ACM Symposium on Applied Computing, Philadelphia, 1996, pp. 442–446.

[31] A.G. Hauptmann, M.J. Witbrock, Story segmentation and detection of commercials in broadcast news video, Adv. Dig. Libraries, Santa Barbara (1998) 168–179.

[32] S. Marlow, D.A. Sadlier, K. McGeough, N. O'Connor, N. Murphy, Audio and video processing for automatic TV Advertisement detection, in: Irish Signals and Systems Conf., Maynooth, Ireland, 2001, pp. 25–27.

[33] R. Lienhart, C. Kuhmunch, W. Effelsberg, On the detection and recognition of television commercials, in: Int. Conf. Multimedia Comput. Syst., 1997, pp. 509–516.
[34] J.M. Sanchez, X. Binefa, Automatic digital TV commercials recognition, in: VIII National Symposium on Pattern Recognition and Image Analysis, Bilbao, Spain, 1999, pp. 223–242.
[35] J.M. Sanchez, X. Binefa, J. Vitria, P. Radeva, Local color analysis for scene break detection applied to TV commercials detection, in: Proc. of the Third Int. Conf. on Visual Information and Information Systems, Amsterdam, 1999, pp. 237–244.
[36] A. Albiol, M.J. Ch. Fulla, A. Albiol, L. Torres, Commercials detection using HMMs, in: Proc. of the Int. Workshop on Image Analysis for Multimedia Interactive Services, Lisboa, Portugal, 2004.
[37] A. Albiol, M.J. Ch. Fulla, A. Albiol, L. Torres, Detection of TV commercials, in: Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing, Montreal, 2004, pp. 541–544.
[38] L. Agnihotre, N. Dimitrova, T. McGee, S. Jeannin, S. Schaffer, J. Nesvadba, Evolvable visual commercial detector, Comput. Vis. Pattern Recognit., Madison, WI 2 (2003) 79–84.
[39] K.S. Goh, K. Miyahara, R. Radhakrishnan, Z. Xiong, A. Divakaran, Audio-visual event detection based on mining of semantic audio-visual labels, Mitsubishi Electric Research Laboratory, TR-2004-008, 2004.
[40] T.Y. Liu, T. Qin, H.J. Zhang, Time-constraint boost for TV commercials detection, in: Int. Conf. on Image Processing, Singapore, 2004.
[41] P. Duygulu, J.Y. Pan, D.A. Forsyth, Towards auto-documentary: tracking the evolution of news stories, in: Proc. of ACM Multimedia 2004, New York, 2004.
[42] P. Duygulu, M.Y. Chen, A. Hauptmann, Comparison and combination of two novel commercial detection methods, in: Proc. of the 2004 IEEE Int. Conf. on Multimedia and Expo, Taipei, Taiwan, 2004.