Content Based Copy Detection with Coarse Audio-Visual Fingerprints

Ahmet Saracoğlu[1,2], Ersin Esen[1,2], Tuğrul K. Ateş[1,2], Banu Oskay Acar[1], Ünal Zubari[1], Ezgi C. Ozan[1,2], Egemen Özalp[1]
A. Aydın Alatan[2], Tolga Çiloğlu[2]
[1]*TÜBİTAK Space Technologies Research Institute*
[2]*Department of Electrical and Electronics Engineering, M.E.T.U.*
{ahmet.saracoglu,ersin.esen,tugrul.ates,banu.oskay,unal.zubari,ezgican.ozan, egemen.ozalp}@uzay.tubitak.gov.tr
{alatan,ciloglu}@eee.metu.edu.tr

Abstract

Content Based Copy Detection (CBCD) emerges as a viable choice against active detection methodology of watermarking. The very first reason is that the media already under circulation cannot be marked and secondly, CBCD inherently can endure various severe attacks, which watermarking cannot. Although in general, media content is handled independently as visual and audio in this work both information sources are utilized in a unified framework, in which coarse representation of fundamental features are employed. From the copy detection perspective, number of attacks on audio content is limited with respect to visual case. Therefore audio, if present, is an indispensable part of a robust video copy detection system. In this study, the validity of this statement is presented through various experiments on a large data set.

1. Introduction

Content Based Copy Detection (CBCD) is an emerging and active research area due to various improvements witnessed in multimedia and communication technologies, such as adoption of more efficient multimedia coding standards and astounding increase in data transfer rates. These improvements and many more generated an even more catalyzing force: "video hosting service". YouTube, Google Video, Metacafe and similar services are parts of our daily lives. As the amount of digital media in these sources increase exponentially (in August 2006 YouTube was hosting about 6.1 million videos [1] and as of April 2008, a YouTube search returns about 83.4 million videos [2]) two crucial and unavoidable problems arise; management of the copyrights and numerous duplicates.

For the solution of these problems there are two main approaches; passive methods and active methods i.e.: watermarking. However, watermarking has two significant limitations. First, since watermarks must be introduced into the original content before copies/duplicates are made, it cannot be applied to content which is already in circulation. Second, the degree of robustness is not adequate for some of the attacks that we encounter frequently. Passive detection methods, on the other hand, try to directly detect copyright infringements and duplicate videos by comparing questioned data against a database. This approach can be thought as a complementary technology to watermarking which provides a solution to the two problems mentioned above. The primary idea of this approach can be interpreted as the media being the watermark itself. That is, the media (image, video, audio) contains enough unique information to be able to detect copies.

The main difficulty of passive detection methods is that the videos are not supposed to be identical. Brightness or contrast enhancement, compression, noise, bandwidth limitation, mixing with unrelated audio, overlay text or geometric transformations can be observed on videos which yield highly modified duplicate video signals. As a result copies can be considered less similar compared to the reference video which can be considered similar for Content-Based Video Retrieval (CBVR) applications, thus the name Content-Based Copy Detection (CBCD).

Although in general video content is considered as an image sequence, in case of availability of audio the correspondence of audiovisual content constitutes an indispensable information source. In some copy/duplication cases audio content is preserved unaffected, in others there may be additional modifications to audio such as bandwidth limitation, coding related distortion, mixing with unrelated audio and in some other cases replaced by entirely different audio stream. This said; it is obvious from the perspective of a practical application that audio component of a video is an indispensible additional information source for the detection of duplicates.

In this work audio and visual content of a video are utilized jointly for a robust solution of the CBCD

problem. Although audio and visual content can be combined at different stages, in our approach we have fused decisions of individual results obtained from different methods specific to audio or visual content. There are various works that jointly utilize audio and video for CBVR, whereas to our knowledge there is no method that exploits both content types in a unified framework specifically for the more recent problem of CBCD. However there are numerous works that focus on one of the contents.

A detailed review on audio fingerprinting can be found in [6] where design criteria, fingerprint models and search techniques are extensively examined with references to the literature. In one of the key works on audio fingerprinting [8], Haitsma et al. used a semi-unique hash value corresponding to each time unit of data by utilizing the human auditory system. Moreover, in their approach Bit-Error Rate (BER) is used as the similarity measure between query and reference audio which is calculated as the ratio of number of different bits of hash values and the total number of bits.

In the literature, video copy detection methods that utilize only visual content of the video can be categorized according to features used as *local* [3] and *global* [4] and [5]. Kim et al. in [5] present a copy detection method that utilizes ordinal features that are computed from the partitioned frames. In their method, spatial matching of ordinal signatures is combined by temporal matching of temporal signatures from the temporal trails of the frame partitions. It has been reported that method is robust against color/brightness variations and format changes. In [4], CBCD has been thought as a partial matching problem that utilizes Markov models for the probabilistic nature of the problem. Joly et al. proposed a method that employs local interest point based features and incorporates a distortion-based probabilistic similarity search in [3]. It should be noted that in both [3] and [4] for the copy detection features are extracted from the keyframes obtained from the video. In this regard, they represent the video as a temporal accumulation of 2D features.

This paper is organized as follows. In Section 2 we describe our approach. Section 3 contains the experiment results and their interpretations. Finally, we state the conclusive remarks and future directions in Section 4.

2. Proposed Method

In our approach, two separate detection decisions that are obtained from visual and audio domain are fused for final decision. For visual detection, both reference video anthology and query videos are represented by feature vectors and according to the similarity of these feature vectors a matching is performed. For the audio side of the system, similar to visual case, a hash value is computed and similarity between hash values is used for the audio-based copy detection. In the end, by selecting the best matching result in terms of confidences obtained from separate audio and visual content matching, the final detection result is obtained.

2.1 Visual Fingerprints

In general video indexing maps the huge amount of content to a lower dimensional space for effective representation of the information conveyed. In this respect, index values can be characterized as short and descriptive. The shortness of the index values facilitates and accelerates the query process and decreases the required index storage. The descriptiveness of the index consists of discriminative power, which enables the discrimination between different contents, and robustness under certain attacks. We claim that these fundamental requirements can also be represented by long and coarse index values inspired by the structure of gene databases [9]. The discriminative power resides in the length and robustness in the coarse structure of the index values, whereas fast queries are possible with the utilization of special indexing that involves multi-resolution and hierarchic structures [10].

In summary our approach is mainly a feature matching between query and the reference videos in which features are extracted from spatio-temporal units of the videos. These aforementioned units are formed by a uniform grid structure which enables spatial and temporal overlapping between separate units. Furthermore, multi-resolution property of the whole method is introduced by incorporating downsampled frames into the equation. Additionally, temporality is achieved by extending each spatial grid element in time, yielding a rectangular prism. And for each *prism*, a feature vector is computed by a set of feature extractors that spans three fundamental visual information sources that are *color*, *texture* and *motion*. Our set of feature extraction methods are derived from some of the MPEG-7 visual descriptors [11] by some modifications. It should be noted that these modifications are introduced in order to decrease the computational complexity of feature extraction and introduce the ability of coarse representation to the descriptors. Moreover, coarse representation is further accentuated by quantization of the feature vectors. Note that, these pseudo-MPEG-7 features are extracted from each grid element and concatenated to form a

single *long* and *coarse* feature vector for a single prism that extends through time and space on the video. Finally, matching query segments are identified by searching query features on a database that is constructed by the reference video features.

As the first step of describing a video by the feature vectors, video is segmented into non-overlapping equal time intervals. For each segment, a single *long* feature vector is computed thus allowing subsets of reference videos to be included and searched in the database. Each segment is also divided by a multi-resolution grid structure in the spatial domain. First level grid structure represents whole frame area whereas second level structure divides frame into 5 uniform regions including a center region overlapping with the corner regions. And the third level grid structure partitions frame into 25 overlapping regions. Pixel values in level i and region j can be represented as $Y_{i,j}(x, y, t)$, $U_{i,j}(x, y, t)$, $V_{i,j}(x, y, t)$, where x, y, t are the spatio-temporal coordinate system variables and Y, U, V are the luminance and color channels of YUV color space. For a given temporal segment, a complete feature vector F_T is obtained by concatenating and quantizing features computed from each aforementioned region.

Although low-level feature extraction methods can be tailored for specific attacks, we have used following features; variants of Color Frequency and Structured Color Frequency for representation of color content, Discrete Cosine Transform and simplistic edge energy for representing texture content and finally motion activity features for representing temporal content.

Color features require the definitions of color histogram (1) and structured color histogram (2). These histograms are formed from 256 bins and for other color channels, namely U and V, are computed in the same manner.

$$h_{ij}^{Y}(c) = \sum_{x,y,t} \delta\big(Y_{ij}(x,y,t) - c\big) \tag{1}$$

$$sh_{ij}^{Y}(c) = \sum_{x,y,t} \delta\big(Y_{ij}(x,y,t) - c\big)\, \alpha^{Y}(x,y,t) \tag{2}$$

In (2), binary parameter α^{Y} takes the value 1 when in the given video volume pixel, neighboring values are in the range that is determined by a threshold otherwise it takes the value 0. Moreover in our work, coarse histograms (3) and (4) are used which are computed by using pre-determined color levels (r_1, r_2, r_3 and r_4).

$$\hat{h}_{ij}^{Y}(n) = \sum_{x,y,t} \delta\big(\hat{Y}_{ij}(x,y,t) - r_n^{Y}\big) \tag{3}$$

$$\widehat{sh}_{ij}^{Y}(n) = \sum_{x,y,t} \delta\big(\hat{Y}_{ij}(x,y,t) - r_n^{Y}\big)\, \alpha^{Y}(x,y,t) \tag{4}$$

In (3) and (4), $\hat{Y}_{ij}$ represents value of the closest pre-determined color value to the actual value which is described by (5). For U and V channels similar computation methods are used.

$$\hat{Y}_{ij}(x,y,t) = \arg\min_{r_m^{Y}} \left\| Y_{ij}(x,y,t) - r_m^{Y} \right\| \tag{5}$$

Edge and motion features are calculated by the help of edge energy $e_{ij}(x, y, t)$ and two-dimensional motion vector components $m^{X}_{ij}(x, y, t)$ and $m^{Y}_{ij}(x, y, t)$.

Color Frequency Feature (6) is computed as the frequency of color values in a given 3D video volume for every color channel. It is determined around the aforementioned pre-determined color values.

$$f_{ij}^{Y,CF} = \begin{bmatrix} \hat{h}_{ij}^{Y}(1) \\ \hat{h}_{ij}^{Y}(2) \\ \hat{h}_{ij}^{Y}(3) \\ \hat{h}_{ij}^{Y}(4) \end{bmatrix} \Bigg/ \sum_{n=1}^{4} \hat{h}_{ij}^{Y}(n) \tag{6}$$

Structured Color Frequency Feature (7) is determined similar to the Color Frequency Feature in which instead of traditional histogram a structured histogram is utilized.

$$f_{ij}^{Y,SCF} = \begin{bmatrix} \widehat{sh}_{ij}^{Y}(1) \\ \widehat{sh}_{ij}^{Y}(2) \\ \widehat{sh}_{ij}^{Y}(3) \\ \widehat{sh}_{ij}^{Y}(4) \end{bmatrix} \Bigg/ \sum_{n=1}^{4} \widehat{sh}_{ij}^{Y}(n) \tag{7}$$

Discrete Cosine Transform Feature (f_{ij}^{DCT}) is computed as the transform coefficients at the lowest four frequencies which are calculated on a 3D luminance video volume.

Edge Energy Feature (8) is computed as the average of edge energies calculated on the luminance of the 3D video volume by the help of the 2D spatial Sobel operator.

$$f_{ij}^{EE} = \frac{\sum_{x,y,t}^{N} e_{ij}(x,y,t)}{N_x \cdot N_y \cdot N_t} \tag{8}$$

Motion Activity Feature (10) is computed as the average of the magnitudes of the motion vectors on a given video prism.

$$f_{ij}^{MA} = \frac{\sum_{x,y,t}^{N} \sqrt{m_{ij}^{X}(x,y,t)^2 + m_{ij}^{Y}(x,y,t)^2}}{N_x \cdot N_y \cdot N_t} \tag{9}$$

For a given segment of a video, a feature vector F_T, in other words a fingerprint is obtained by using combinations of features presented on the 3D grid units discussed in the previous section. In this work, in the

Table 1. Fingerprint dimensions and ingredients in the unit interval.

	Visual Fingerprint
Color Frequency	3 Levels (372 values)
SCF	2 Levels (72 values)
DCT	2 Levels (24 values)
Edge Energy	2 Levels (6 values)
Motion Activity	2 Levels (6 values)
Total	480 dimensions (Fingerprint 960 bit)

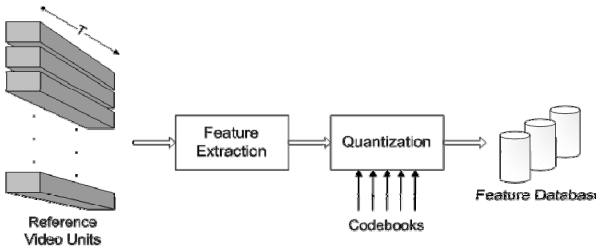

Figure 1. Method of constructing visual feature database from reference videos.

overall fingerprint that is depicted in Table 1 is used. The feature values are quantized to four levels by Lloyd's Method [7] resulting with the coarse feature vector D_T. The quantized features are stored in the reference video database as a fingerprint for the given temporal range of the reference video. The overall block diagram of the system is depicted in Figure 1.

2.2 Visual Fingerprint Matching

For the detection of the query video, firstly query fingerprint, Q_T, is computed as described in the previous section. Afterwards, similarities between Q_T and every fingerprint, D_T, in the reference database are computed. This comparison is achieved by sliding Q_T over entire length of individual reference fingerprints. At this point, unique video locations that are exceeding a predetermined similarity value are combined, sorted and presented as the search result.

Although similarity measures can be tailored for specific requirements, in this study *Euclidean Distance* (10) and *Cosine of the Angle* (11) between two fingerprints are used and compared as a measure.

$$s^{EUC} = \frac{1}{1 + \sqrt{\sum_i \left(D_{T_i} - Q_{T_i}\right)^2}} \tag{10}$$

$$s^{COS} = \frac{\sum_i \left(D_{T_i} \cdot Q_{T_i}\right)}{\|D_T\| \cdot \|Q_T\|} \tag{11}$$

2.3 Audio Fingerprint

The first step of a standalone audio copy detection algorithm is to extract effective fingerprints or in other words *hash values*. In this study, the fingerprint extraction method used is similar to the method that is introduced in [8]. However, unlike [8], fingerprints are extracted in the form of 15 bits instead of 32 in order to overcome strong attack types such as band width limitation or irrelevant speech addition. Another advantage of the 15-bit form is that, it requires a much smaller hash table and thus occupying less space in main memory, (2^{15} = 32KB) than the 32-bit form (2^{32} = 4GB).

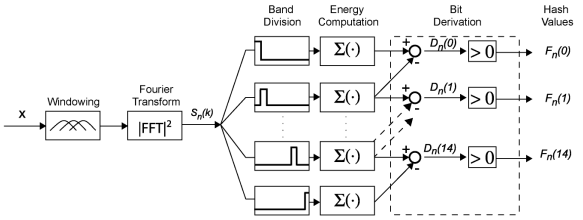

Figure 2. Audio fingerprint extraction diagram.

The hash values are calculated using the power spectra of 25 ms frames separated by 10ms. Each signal frame is multiplied by Hamming window before its Fourier Transform is computed. The spectrum over 300Hz – 3000Hz is divided into 16 sub-bands according to the Bark scale. The energy differences between these sub-bands are used to calculate the hash values according to (12).

$$F(n,m) = \begin{cases} 1, & EB(n,m) - EB(n,m+1) > 0 \\ 0, & EB(n,m) - EB(n,m+1) \le 0 \end{cases} \tag{12}$$

In (12), *EB(n, m)* represents the energy value of the n^{th} frame at the m^{th} sub-band. In Figure 2, a detailed diagram of audio fingerprint extraction method is provided.

2.4 Audio Fingerprint Matching

As a first step of audio fingerprint matching, a hash database containing all possible hash values is created to reach out quickly the exact match points. This hash database contains 2^{15} different hash values, each holding linked lists, pointing to the locations of these hash values in the reference audio files inserted to the database. The hash values of the query are matched with the hash values of the reference data using the previously formed hash database without any sequence scan.

The query file is represented as a sequence of hash values as shown in Figure 3. For every query file, a voting table is created. This voting table holds a vote that is calculated by counting the number of the equal time differences between the matching points of query and reference data. For example, the 3rd and 5th hash values of the query, matches exactly with the 10th and 12th hash values of the reference file. So the voting table holds a value of 2 for the difference 7. Then, if there are more exact matches with the same difference of 7, this value of 2 is increased. So, the sequential exact match points are searched within the reference data to locate the query. The voting table also holds the first and last time indices of the corresponding difference value. This shows where the query data located within the reference file. The voting function, V that calculates the value obtained for the time differences between the query and the reference file is given in (13).

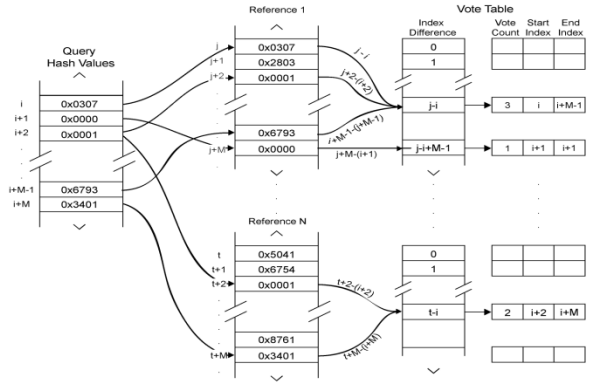

Figure 3. Voting-based Audio Fingerprint Matching Method. Audio fingerprint extraction diagram.

$$V(\tau) = \sum_{\{r,q\}\epsilon R} \delta(\tau - |r - q|) \quad (13)$$

In (13), q and r show the time indices of the matching locations of the query and reference fingerprints whereas, τ is the difference between the time indices. The similarity for every difference value τ is calculated by dividing $V(\tau)$ by the difference of the first and last time index of the corresponding difference in seconds. The point with the highest similarity gives the most similar area for the reference and query data. In other words, similarity is calculated as the number of exact matches per second. This matching method is also elaborated in Figure 3.

2.5 Decision Fusion

At the decision fusion stage, individual matching results obtained from previously explained audio and video processing stages are combined. Combination rule is to choose the best matching result in terms of confidences obtained from separate audio and visual content matching. For each query a single best matching temporal segment from the reference database is returned, if the resultant confidence value exceeds a certain threshold.

3. Experiments

In this study, 14000 query files of 300-hour length, each varying between 3 seconds and 3 minutes, are searched over a 200-hour video database. Reference database and queries are provided by NIST through the TRECVID initiative [12]. The original part of the reference file copied into the query files is at least 3 seconds length, if it exists. This leads to 720,000 feature vectors for all time segments for video fingerprints and 72,000,000 hash values to be searched

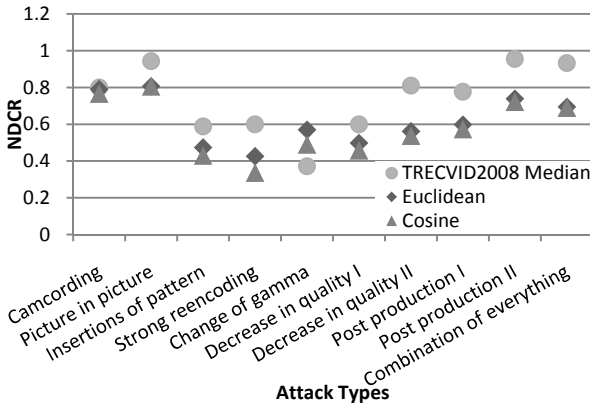

Figure 4. Comparison of visual similarity measures.

for every part longer than 3 seconds for audio fingerprints in the reference database.

Each query is created by applying one or more transformations to a randomly selected portion of an original video, which may or may not be indexed in the reference video database, and some of the queries are padded with unrelated clips, which are not in the database. Transformations are designed to imitate real life attacks and can be in 10 different forms for video, including but not limited to color transformations, spatial transformations, pattern insertion, re-encoding and different combinations of these. For audio, there are 7 different types of transformations, including but not limited to bandwidth limitation and quantization noise. Experiment results are collected over unique combinations of video and audio attacks, which sum up to 70 different transformation pairs. More detail on query generation can be found in [13].

The algorithm is tested with three different setups. Experiments are carried out by utilizing only audio fingerprints, only video fingerprints and finally by decision fusion. Normalized Detection Cost Rate (NDCR) [14] has been used for the evaluation of the proposed system, which is calculated as a weighted sum of *probability of miss* and *rate of false alarm* that are computed over the query results. It should be noted that smaller values of NDCR represent better performance.

At the first stage of experiments, different similarity measures used at visual fingerprint matching have been compared. From Figure 4 it can be seen that cosine based similarity measure performs better than the Euclidean based. This said, in the second part of experiments cosine has been used for visual fingerprint matching. Furthermore, median performance of the TRECVID 2008 participants on Video-only CBCD Task is shown on the graph.

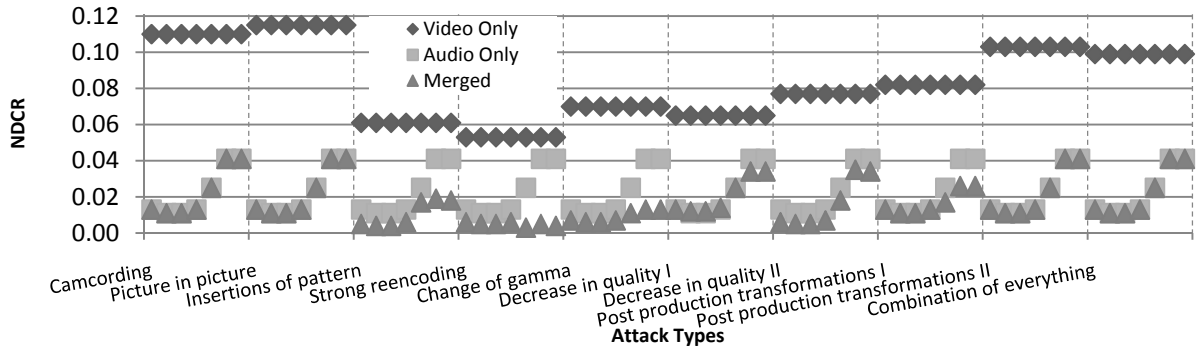

Figure 5. Performance of our framework with respect to 70 different attack combinations. Horizontal axis ticks show visual attack types.

Finally, performance of each setup with respect to transformation pairs is shown in Figure 5. Ten types of visual attacks are given in the horizontal axis. Within each group there are seven audio attack types composed of compression, superposition, filtering and combination of these.

From the results it can be seen that audio features are more robust than visual features due to the severity of visual attacks. Especially for cam-cording, picture in picture and post production attacks visual copy detection performs poorly and has no contribution in the merged case. On the other hand, we observe that for the rest of the attacks visual copy detection performs better and has a reasonable effect in the performance increase of the merged case.

4. Conclusion

We propose to use long and coarse representations of visual and audio content for CBCD task. Through various experiments we observe the validity of this approach for certain attacks. Audio copy detection appears to be more robust due to relatively simple nature of attacks. Furthermore, it is possible to have significant improvements by joint usage of visual and audio features. However, as seen in the experiments visual features cannot withstand some of the attacks e.g.: picture-in-picture attack. For such attacks local features should be utilized in addition to the currently used global features.

5. References

[1] Gomes, Lee. "Will All of Us Get Our 15 Minutes On a YouTube Video?" [Online] The Wall Street Journal, Aug. 30, 2006. [Accessed: Nov. 24, 2007].

[2] "YouTube - Broadcast Yourself," [Online]. Available: http://www.youtube.com/results?search_query=*. [Accessed: June 15, 2008].

[3] Alexis Joly; Olivier Buisson; Carl Frelicot, "Content-Based Copy Retrieval Using Distortion-Based Probabilistic Similarity Search," IEEE Transactions on Multimedia, vol.9, no.2, pp.293-306, Feb. 2007.

[4] Chih-Yi Chiu; Chu-Song Chen; Lee-Feng Chien, "A Framework for Handling Spatiotemporal Variations in Video Copy Detection," IEEE Transactions on CSVT, vol.18, no.3, pp.412-417, 2008.

[5] Changick Kim; Vasudev, B., "Spatiotemporal sequence matching for efficient video copy detection," IEEE Transactions on CSVT, vol.15, no.1, pp. 127-132, Jan. 200.

[6] P. Cano, E. Badle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," in Proc. IEEE Workshop on Multimedia Signal Processing, 2002.

[7] Lloyd, S., "Least squares quantization in PCM," IEEE Transactions on Information Theory, vol.28, no.2, pp. 129-137, 1982

[8] Haitsma, J., Kalker, T., "Robust Audio Hashing for Content Identification," In CBMI, 2001.

[9] Tamer Kahveci, Ambuj K. Singh, "MAP: Searching Large Genome Databases," PSB 2003, pages 303-314.

[10] Tamer Kahveci and Ambuj K. Singh, "An Efficient Index Structure for String Databases," International Conf. on VLDB, 2001, pages 351-360.

[11] Manjunath, BS. Introduction to MPEG-7: Multimedia Content Description Interface, 2002, ISBN:0471486787.

[12] Smeaton, A. F., Over, P., and Kraaij, W. 2006. Evaluation campaigns and TRECVid. In Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval MIR '06.

[13] "Building video queries for TRECVID2008 copy detection task," June 3, 2008. [Online]. Available: http://www-nlpir.nist.gov/projects/tv2008/TrecVid2008CopyQueries.pdf. [Accessed: Jan. 13, 2009].

[14] "Final CBCD Evaluation Plan TRECVID 2008," June 3, 2008. [Online]. Available http://www-nlpir.nist.gov/projects/tv2008/Evaluation-cbcd-v1.3.htm. [Accessed: Jan 13, 2009].