Still Image and Video Fingerprinting

Nikos Nikolaidis Ioannis Pitas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: pitas@aiia.csd.auth.gr, nikolaid@aiia.csd.auth.gr

Abstract

Multimedia fingerprinting, also know as robust/perceptual hashing and replica detection is an emerging technology that can be used as an alternative to watermarking for the efficient Digital Rights Management (DRM) of multimedia data. Two fingerprinting approaches are reviewed in this paper. The first is an image fingerprinting technique that makes use of color and texture descriptors, R-trees and Linear Discriminant Analysis (LDA). The second is a two-step, coarse-to-fine video fingerprinting method that involves color-based descriptors, R-trees and a frame-based voting procedure. Experimental performance evaluation is provided for both methods.

1. Introduction

Recent advances in the area of multimedia content distribution have resulted in a major reorganization of this trade. Valuable digital artworks can be reproduced and distributed arbitrarily without any control by the copyright holders. Thus, issues related to intellectual property rights protection and management arise.

Numerous systems addressing the issue of copyright protection can be found in the literature, the majority of them being based on watermarking. Watermarking is the technique of imperceptibly embedding information within the content of the original medium [1]. Although watermarking has attracted considerable interest, it bears certain deficiencies. The requirement of embedding information inside a multimedia document before it reaches the public, implies distortion of the data at a certain extent and automatically excludes data that are already in the public domain and need to be protected. In addition, watermarking is unable to deal with content leakages, i.e. cases where an unwatermarked copy of the original artwork is stolen and distributed.

In order to overcome these inherent watermarking deficiencies, the scientific community recently started to investigate copyright protection and digital rights management in multimedia data through content based approaches These approaches, which come under different names such as multimedia fingerprinting [2], [3], [4], [5], robust or perceptual hashing [6], [7], [8], [9], [10], and replica or copy recognition/detection [11], [12], [13], [14], [15], [16], [17], [18], [19], aim at extracting from the data a feature vector, called perceptual hash, fingerprint or signature, that characterizes them in a unique and discriminative way. This feature vector can be combined with a database of multimedia documents that need to be managed with respect to their intellectual property rights, an appropriate similarity metric and an efficient search strategy in order to devise a DRM system. More specifically, such a system can decide if a query digital item resembles a reference item in the database. If this is indeed the case, the query item is identified as being a copy (replica) of the corresponding item in the database and legal action can be pursued against its holder/distributor if he is not posessing/distributing it in a legal way. In order to be of practical use, the feature vectors and the matching procedure involved in a fingerprinting system should be robust to manipulations that multimedia data might undergo, either due to their distribution and use or due to an intentional attempt to make them unrecognizable by the fingerprinting system. The major advantage of fingerprinting is that, unlike watermarking, no information needs to be embedded within the content, thus ensuring perfect quality for the data to be protected and furthermore making the system applicable to data that are already in the public domain.

The basic hypothesis behind the aforementioned approach is that every multimedia document (and the corresponding feature vectors) shares enough information with its modified copies to allow their identification as such, and yet bears enough discriminative information with respect to other data to allow for their identification as nonrelevant. Furthermore, it is assumed that the modified data maintain sufficient quality and resemblance to the originals. Severely distorted copies are of no interest for a fingerprinting system since their commercial value is reduced.

The problem of multimedia fingerprinting bears certain similarities with that of content-based indexing and retrieval but has also important differences. The major difference between fingerprinting and retrieval is that the similarity criterion is usually looser in retrieval, since the user is often interested not only in copies of a multimedia item, but also in different items that are perceptually similar to it. Moreover, the requirement of robustness to manipulations (attacks) is not applicable to retrieval. Finally, a replica detection system should be able to return an empty results set if a ”no replica” decision has been reached whereas, in most cases, a retrieval system always returns one or more matching multimedia documents.

In this paper two fingerprinting systems are described. The first system is an image fingerprinting system that involves color and texture image descriptors, multidimensional indexing based on an R-tree constructed through an attacks-based training procedure and Linear Discriminant Analysis. A more detailed description of an earlier version of this system can be found in [16]. The second system is a novel color-based two-step, coarse to-fine-video fingerprinting system. The first step performs a coarse selection of the database (original) videos that are candidates for matching the query video. Similar to the image fingerprinting system, this step makes use of an R-tree and a training procedure that involves attacked versions of the database videos and aims at achieving robustness to attacks. A frame-based voting procedure is also involved in this step. A refinement step that processes the set of videos returned by the first step in order to select the final matching video (if any) follows.

2. Image Fingerprinting Using R-trees and Linear Discriminant Analysis

2.1. System Overview

The construction of the proposed fingerprinting - image replica detection system can be separated to two independent phases. The first phase deals with the database organization (Fig. 1). Each time a new original, copyright protected image is added into the database, the image is subjected to a series of predefined attacks selected according to the system’s design specifications. Feature vectors are extracted from each attacked version resulting in a feature table which contains samples from the feature space neighborhood of the original image. This is utilized for the calculation of an extent vector that specifies the neighborhood extent for each original image in the form of a multidimensional hyper-parallelepiped, which for brevity will, from now on, be called bounding box (BB). Finally, the original image is indexed within the R-tree structure, according to the BB.

The second phase implements the actual fingerprinting functionality, once the database has been organized. An unknown image is submitted as a query to the indexing structure. The R-tree uses the query image feature vector in order to provide a set of candidate matching images or an empty set. The next step attempts to enhance the system performance by applying LDA, preceded by Principal Components Analysis (PCA) on the set returned by the R-tree. Finally, the system returns the database image that is closest to the query image in the LDA space, provided that their distance is below a threshold. Thus, the decision that the query image is not a (possibly modified) copy of the images in the database may occur either during the R-tree traversal or after the application of LDA. Fig 2 depicts the database querying procedure.

2.2. Feature Vector Extraction

The proposed system is based on a descriptor (feature vector) proposed in MPEG-7 for capturing texture and color information of an image. This descriptor, called Scalable Color Edge Histogram (SCEH) has been selected through experimentation from a set of color-only or color and texture MPEG-7 descriptors. SCEH is a 208-dimensional descriptor, generated by concatenating the ScalableColor [20] and EdgeHistogram [21] descriptors. ScalableColor consists of a normalized color histogram initially computed in the HSV color space according to a uniform quantization and subsequently converted through Haar transform into a representation that assigns higher significance to small values. The EdgeHistogram descriptor represents the local distribution of edges in an image. It must be noted that other feature vectors can be also used along with the proposed scheme.

2.3. R-tree Indexing

An R-tree [22] is used in order to efficiently index the images in the database. Our system works under the assumption that the feature vectors of modified copies of an original image are localized around the feature vector that corresponds to the original image. The used R-tree is constructed by associating a bounding box to every original image. These bounding boxes are defined using an extent vector, evaluated through a training procedure. More specifically, in order to determine the extent vector for each image in the database, we simulate all attacks that an image may undergo and we wish the system to be able to withstand. Thus, before inserting an original image into the indexing structure, a series of predefined attacks are performed. The produced images are used for determining the extent vector.

In more detail, a feature vector is extracted from every modified (attacked) version of an image and the distances in each feature dimension between each attacked image and the original image are calculated. The maximum distance for each dimension is selected as the extent in this dimension. Thus, the aforementioned procedure derives for each original image an extent vector consisting of 208 scalar values that determines the extent of its bounding box in each feature dimension.

When queried with an unknown image, the ”trained” R-tree returns all images whose BBs include the feature vector of this query.

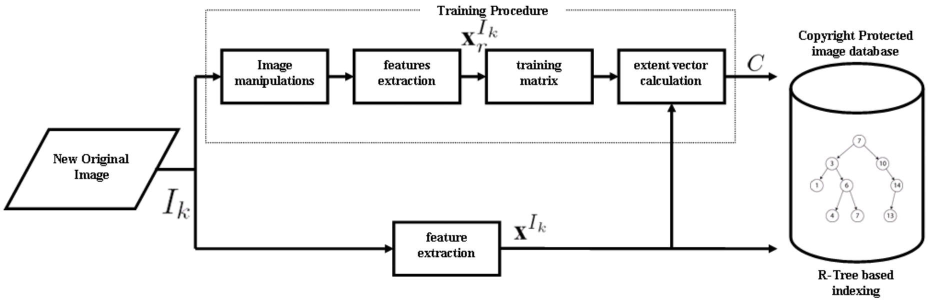

Figure 1. Database organization of the image fingerprinting system.

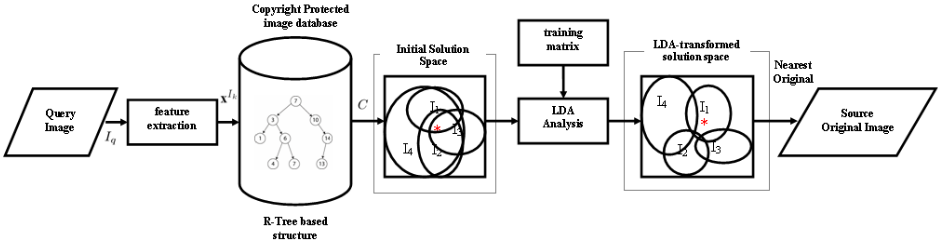

Figure 2. Database querying.

2.4. Applying Linear Discriminant Analysis

The fact that the R-tree can return more than one original images as candidates for being related to the query image does not allow the system to decide unambiguously. In order to obtain a single result and reduce the decision errors LDA was used for discriminant feature selection. Prior to the application of the LDA, a dimensionality reduction step using PCA is performed. The set of participating classes in the LDA is the set of images (classes) returned by the R-tree. Each of these classes is comprised of the feature vector of the original image along with the feature vectors of its attacked versions. These vectors are the observations used for calculating the class statistics. The LDA is applied every time a query is submitted. The result of LDA is a linear transformation $\mathbf{W}_o$ that transforms and/or reduces the dimensionality of the image feature vectors $\mathbf{x}_k$ as:

$$\acute{\mathbf{x}}_k = \mathbf{W}_o^T \mathbf{x}_k. \tag{1}$$

The goal of the linear transformation $\mathbf{W}_o$ is to maximize the between class scatter while minimizing the within class scatter. By projecting the feature vectors to the newly created space, better separation of classes is achieved. A similarity metric is then used to find the closest class (image) to the query image. A threshold on the distance of the query image from the selected original image is used to accept or reject the query image as a copy of this image.

2.5. Experimental Performance Evaluation

A sample of 2.232 color images were collected from the Internet in order to form the test set. In order to examine the system's ability to distinguish between replicas of highly similar original images, the images were selected so as to form 12 content categories, each containing different views of an architectural monument (Parthenon, Eiffel Tower, etc).

For training, we generated 40 manipulated copies for each of the images stored in the database by applying the following 40 transforms: Colorizing the Red, Green, and Blue channel by 10%, cropping by 5%, 10%, 20%, and 30%, despeckling, downsampling by 10%, 20%, 30%, 40%, 50%, 70%, 90%, mirroring along the horizontal direction, color quantization to 256 colors, adding an outer frame of 4 different colors to the image, rotation by 90^o, 180^o, and 270^o, scaling up then down by a factor of 2,4, and 8, scaling down then up by a factor of 2,4, and 8, modulation of the color saturation amplitude by 70%, 80%, 90%, 110%, and 120%, intensity change by 80%, 90%, 110%, and 120%.

For evaluating the performance of the proposed system, the false positive (FP) and false negative (FN) metrics were used. A false positive image identification occurs when a query image is erroneously considered to be a replica of a certain image. A false negative occurs when a query image that is a replica of a certain original is not evaluated as such. Let N_{org} be the number of original images in the database and N_{nrep}, N_{rep} the number of non-replicas and replicas per original image respectively that are used for constructing the query image set. Let also T be the number of cases that a replica is identified as such but is classified to a wrong original, W be the number of cases that a non-replica is evaluated as a replica and S the number of cases that a replica is considered as a non-replica. Then false positive and false negative rates are defined as:

$$FP = \frac{T+W}{N_{org} \cdot N_{rep} + N_{nrep}} \qquad FN = \frac{T+S}{N_{org} \cdot N_{rep}} \quad (2)$$

In order to test the performance of the proposed system, from the set of 2232 images we selected a set S_{org} of 2000 images for populating the image database while the remaining 232 images formed the set S_{nrep} of non-replicas. Two query sets were used in the experiments. The query set S_{Q1} comprised of original images, test replicas and non-replicas. 200 images randomly chosen from S_{org} were used to form the set of original images S_{seed} that was included in the query set S_{Q1}. These images were also used to generate the set S_{rep} of 8000 test replicas by applying the same manipulations used for the training images to the images in S_{seed} (40 transforms per original image). Finally, S_{nrep} was appended to S_{Q1} resulting in a query set containing a total of 8432 images. When queried with this set the system achieved an Equal Error Rate (EER), i.e. the operating point where $FP = FN$ equal to 1.1%.

The fact that the set of attacks used for training was exactly the same with that used to create the replicas included in the query set S_{Q1} may lead to biased performance evaluation. For this reason, a query set that included manipulated images that were not used during training was constructed. In order to produce the new query set S_{Q2} we utilized S_{seed} and S_{nrep} but instead of S_{rep} we used a different set of replicas (40 replicas per original image) that was generated by exposing the original images of S_{seed} to the same manipulations (attacks) described above but with different attack parameters. When queried with S_{Q2} the system produced an EER equal to 3%.

3. Video Fingerprinting Using R-trees and a Frame-based Voting Approach

3.1. System Overview

The video fingerprinting - replica detection system proposed in this paper operates upon a database of stored video originals. The system can be queried with a certain video and decide whether this video is a replica of a stored original or not. Replica detection is again dealt as a classification problem. Videos are represented by color feature vectors and a multidimensional indexing structure that is based on R-trees (and functions similarly to the R-tree in the image fingerprinting system) is employed. The efficiency of the R-tree in reducing retrieval time and producing accurate and robust results depends directly on the selection of optimal bounding boxes. For selecting these bounding boxes, an attack-oriented training strategy that aims at modelling potential attacks that the system is designed to encounter is used. In the implementation described in this paper, scaling, additive Gaussian noise and MPEG-4 compression are considered. The R-tree returns a set of videos that are candidates for being the original of the query video. This is achieved by a voting scheme where each frame from the query video casts a vote to a video in the database. Subsequently, a refinement step that utilizes distances between frame color histograms is applied on the videos returned by the R-tree in order to reach the final decision.

3.2. Feature Extraction and R-tree Training

The proposed approach utilizes the color histograms of the video frames in order to describe a video. More precisely, it makes use of the MacBeth palette which consists of 24 colors selected to emulate common natural colors such as skin colors, foliage and sky and includes also additive and subtractive primaries and six shades of gray. For each video that is to be inserted in the database, we evaluate and use as video descriptors the 24-bin histograms of its frames (every fifth frame is selected) with respect to the MacBeth palette. This results in N_F histograms.

The three attacks mentioned before, namely additive Gaussian noise, scaling and MPEG-4 compression are applied on each of the N_V videos that are to be inserted in the database, thus resulting in 3 attacked versions for each video. For each original video and its attacked versions, we extract a set of color histograms as described above. Thus for each video we extract a total of $N_{CH} = 4N_F$ histograms.

These histograms are subsequently used in order to build/train the R-tree structure that is used for storing information for the original videos and retrieving a set of original videos in response to a query. More specifically, the N_{CH} color histograms of an original video and its attacked versions are used to evaluate the boundaries of the multidimensional bounding box (BB) that encloses all histograms for this video in the 24-dimensional space. More specifically, the minimum and maximum values (evaluated over all frames of the original video and its attacked versions) for the i-th dimension of the histograms determine the boundaries for the i-th dimension of the BB for this video. These bounding boxes are then used to index the videos in the R-tree.

3.3. Querying the System

Querying the system with a test video in order to decide whether it is a replica of a video in the database is a two - step procedure.

3.3.1. Coarse response. When a query video enters the system, a subset comprising of N_Q frames is selected and their color histograms are evaluated as explained above. Each histogram is inserted into the R-tree and all the BBs (videos) that enclose it are found. The corresponding videos obtain one vote each. Obviously, if the frame's color histogram belongs to more than one BBs, due to BB overlaps, all the corresponding videos will obtain a vote. Videos that receive at least $a\%$ of the votes cast by the frames of the query video are selected for further processing. The above procedures aims at a quick selection of videos that are close to the query.

3.3.2. Refinement. Let W be the subset of the database videos obtained from the first step. If V_i is one of the videos in W ($i = 1, ..., N_W$, where N_W the number of videos in W), then for every one of the N_Q selected frames of the query video we calculate the minimum of the L_1 distances between its histogram and all the N_{CH} color histograms extracted from the V_i video, i.e., the histograms of all selected frames of V_i and its attacked versions. In more detail, if $\mathbf{Q}_j$, $j = 1, ..., N_Q$ is the histogram of the $j - th$ frame of the query video, we evaluate the distance:

$$D_{ij} = min_k ||\mathbf{Q}_j - \mathbf{V}_i(k)||_1, \quad k = 1, ..., N_{CH} \tag{3}$$

where $\mathbf{V}_i(k)$ is the $k - th$ histogram of video i. Consequently, the total distance of the query video from the i-th video in W is evaluated as:

$$D_i = \sum_{j=0}^{N_Q - 1} (D_{ij})^2 \tag{4}$$

The final winner is the video V_k in W with the minimum total distance D_k from the query video. This distance is subsequently compared against an experimentally evaluated threshold T_1 in order to reach the final decision.

4. Experimental Performance Evaluation

The proposed system was tested on a database of 589 short videos collected from the Internet, mainly from the popular YouTube website. These videos are of relatively poor quality, have a size of 320x240 pixels, 25 frame per second and on average they are no longer than 5min. As explained below, the performance metrics used in this case were different from the ones used for the image fingerprinting method.

The specific attacks used for training the system were: a) uniform scaling with a scaling factor of 0.8 in each dimension, b) XviD MPEG-4 compression with a target quantizer of value 4 and c) 15% additive gaussian noise.

In order to test the system, 4 sets of query videos were created. The first three sets, A, B, and C were used to test the system performance, when queried with videos that belong to the database or modified versions of these videos. In this case performance can be measured in terms of the miss-classification error (MC) i.e., the percentage of query videos that were classified to a wrong original video in the database and the false rejection error (FR) which is the percentage of query videos that were erroneously tagged as not belonging to the database. The last set, D, consisted of videos that did not belong to the database. On such a query set, the performance is measured in terms of false acceptance (FA) error i.e., percentage of query videos that are erroneously tagged as belonging to the database. The procedures and results for each query set are described below.

A: 100 videos were selected from the database and the three attacks with the same parameters as in the training phase were applied on each of them (400 videos in total). Both the miss-classification rate and the false rejection rate in this case were 0%.

B: In this case, the 100 selected database videos were modified by the same three attacks but, this time, different attack parameters from the ones used in training were used. Again both MC and FR were equal to 0%.

C: In this set, the 100 selected videos were modified by combinations of more than one of the three attacks. In more detail, three subsets (with 200 videos each) were created. The first subset consisted of scaled videos affected by additive Gaussian. Both the miss-classification rate and the false rejection rate were 0% in this case. The second subcategory consisted of videos that have been attacked by additive Gaussian noise and MPEG-4 compression. In this case MC was 0% and FR was 1.5%. The third subset consisted of videos modified by MPEG compression and scaling. In this case MC was 0% and FR 0.5%.

D: 300 videos that did not belong to the database were used in this case. The false acceptance rate (FA) was found to be 3.3%.

5. Conclusion

Multimedia fingerprinting is an efficient alternative to watermarking, having the additional advantage of being a technique that does not alter the content of the data. Two different fingerprinting approaches have been presented in this paper. The first approach utilizes color-based descriptors along with R-trees and LDA in order to achieve identification of (possibly modified) copies of images from a database of originals. The second is a two-step, coarse-to-fine video fingerprinting method that involves color-based descriptors, R-trees and a frame-based voting procedure. Experimental performance analysis shows that the proposed techniques can be used for the efficient DRM of images and video.

Acknowledgment

This work was developed within ECRYPT IST-2002-507932, European Network of Excellence in Cryptology (http://www.ecrypt.eu.org/), funded under the European Commission IST FP6 programme.

References

[1] A. Tefas, N. Nikolaidis, and I. Pitas, "Image watermarking: Techniques and applications," in *The Essential Guide to Video Processing*, A. Bovik, Ed. Academic Press, 2008.

[2] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Proc. 5th International Conference on Recent Advances in Visual Information Systems (VISUAL 2002)*, March 2002, pp. 117–128.

[3] J. Seo, J. Haitsma, T. Kalker, and C. Yoo, "Affine transform resilient image fingerprinting," in *2003 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 03)*, vol. III, April 2003, pp. 61–64.

[4] J. S. Seo, J. Haitsma, T. Kalker, and C. D. Yoo, "A robust image fingerprinting system using the radon transform," *Signal Processing: Image Communication*, vol. 19, pp. 325–339, 2004.

[5] S. Lee and C. D. Yoo, "Robust video fingerprinting based on 2D-OPCA of affine covariant regions," in *IEEE International Conference on Image Processing*, 2008.

[6] C. Y. Hsu and C. Lu, "Geometric distortion-resilient image hashing system and its application scalability," in *ACM International Conference on Multimedia (Proceedings of the 2004 workshop on Multimedia and security)*, Magdeburg, Germany, 2004, pp. 81–92.

[7] S. Yang and C. Chen, "Robust image hashing based on SPIHT," in *International Conference on Information Technology: Research and Education (ITRE 05)*, June 2005, pp. 110–114.

[8] A. Swaminathan, Y. Mao, and M. Wu, "Image hashing resilient to geometric and filtering operations," in *Proc. of IEEE Workshop on Multimedia Signal Processing (MMSP'04)*, Siena, Italy, Sept. 2004, pp. 355–358.

[9] R. Venkatesan, S. Kaon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *2000 IEEE International Conference on Image Processing (ICIP 00)*, September 2000.

[10] B. Coskun, B. Sankur, and N. Memon, "Spatio-temporal transform-based video hashing," in *IEEE Transactions on Multimedia*, vol. 8, no. 6, December 2006, pp. 1190–1208.

[11] Y. Ke, R. Sukthankar, and L. Huston, "An efficient parts-based near-duplicate and sub-image retrieval system," in *Proceedings of the 12th annual ACM international conference on Multimedia*, New York, USA, 2004, pp. 869–876.

[12] S. Roy and E.-C. Chang, "Watermarking with retrieval systems," *ACM Multimedia Systems*, vol. 9, no. 5, pp. 433–440, March 2004.

[13] S. Roy, E.-C. Chang, and K. Natarajan, "A unified framework for resolving ambiguity in copy detection," in *Proceedings of the 13th annual ACM international conference on Multimedia*, Singapore, 2005, pp. 648–655.

[14] A. Qamra, Y. Meng, and E. Chang, "Enhanced perceptual distance functions and indexing for image replica recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 379–391, March 2005.

[15] Y. Maret, S. Nikolopoulos, F. Dufaux, T. Ebrahimi, and N. Nikolaidis, "A novel replica detection system using binary classifiers, R-trees and PCA," in *2006 IEEE International Conference on Image Processing (ICIP 06)*, Atlanta, GA, 2006.

[16] S. Nikolopoulos, S. Zafeiriou, P. Sidiropoulos, N. Nikolaidis, and I. Pitas, "Image replica detection using R-trees and linear discriminant analysis," in *2006 IEEE International Conference on Multimedia and Expo (ICME 06)*, Toronto, Canada, 2006.

[17] Y. Maret, F. Dufaux, and T. Ebrahimi, "Adaptive image replica detection based on support vector classifiers," *Signal Processing : Image Communication*, vol. 21, no. 8, pp. 688–703, Sept. 2006.

[18] J. Law-To, V. Gouet-Brunet, O. Buisson, and N. Boujemaa, "Video copy detection on the internet: The challenges of copyright and multiplicity," in *IEEE International Conference on Multimedia and Expo*, July 2007, pp. 2082–2085.

[19] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 293–306, Feb. 2007.

[20] D. Messing, P. van Beek, and J. Errico, "The MPEG-7 colour structure descriptor: image description using colour and local spatial information," *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1, pp. 670–673 vol.1, 2001.

[21] M. Eom and Y. Choe, "Fast extraction of edge histogram in DCT domain based on MPEG-7," in *International Conference on Enformatika, Systems Sciences and Engineering (ESSE 2005)*, Istanbul, Turkey, Nov 2005.

[22] V. Gaede and O. Gunther, "Multidimensional access methods," *ACM Computing Surveys*, vol. 30, no. 2, pp. 170–231, 1998.