Pattern Recognition Methods in Image and Video Databases: Past, Present and Future

Sameer Antani[1], Rangachar Kasturi[1] and Ramesh Jain[2]

[1] Department of Computer Science and Engineering,
The Pennsylvania State University, University Park, PA 16802, USA
(antani,kasturi)@cse.psu.edu
[2] Department of Electrical and Computer Engineering,
University of California at San Diego, La Jolla, CA 92093, USA
jain@ece.ucsd.edu

Abstract. Image and video (multimedia) database systems have been on an increase in recent years. Several applications demand the retrieval of multimedia data from these database systems based on their content. The users of these systems and applications perceive the data in different ways and demand the ability to query the data based on their perception of the content. This need has spurred an interest to develop pattern recognition methods which can capture the visual information content and place them in a suitable form for database indexing. This paper describes some of the image and video database systems and the various pattern recognition methods used therein.

1 Introduction

Image and video data has been on an increase in recent years. The significant improvement in processing technology in recent years coupled with decrease in the cost of memory and storage devices have led to the development of large multimedia database systems. This growth in visual data has spurred a significant interest in the research community to develop methods to query and retrieve this data based on their content. Such systems, called Content Based Image (and Video) Retrieval (CBIR) Systems, employ many pattern recognition methods developed over the years. This paper studies the significant contributions published in the literature that use these methods. In this paper, the term *pattern recognition* extends beyond its classical definition of a method to cluster and classify data to aid recognition. Many methods applied in feature extraction and content matching subsystems have their roots in the fundamental work done in pattern recognition. Hence, when we refer to the term in this paper we are referring to the applicability in feature extraction, feature clustering and generation of database indices, and determining similarity in content of the query and database elements.

Research in retrieving non-geometric pictorial information began in the earlier part of the last decade. A relational database system that used pattern recognition and image manipulation was developed for retrieving LANDSAT

images [5]. This system also introduced the preliminary concepts of Query-by-Pictorial-Example (QPE). Since then, the research has come a long way in employing various methods to retrieve images by their content. There are two main streams of study that have been followed since. In the first, image contents are modeled as a set of attributes extracted manually and managed with the framework of conventional database-management systems [13]. These systems entail a high level of image abstraction. Higher the abstraction, lesser is the scope for posing ad-hoc queries to the system. The second approach depends on an integrated feature extraction/object recognition subsystem. These features which define the image are used by the pattern recognition subsystem to select a good match in response to the user query.

Some of the features used are color, sketch, shape and texture. These features have been used very frequently by researchers in developing methods for defining images to aid their retrieval. The Query-By-Image-(and Video)-Content (QBIC) System developed at the IBM Almaden Research Center uses example images, user constructed sketches and drawings and selected color and texture patterns for image retrieval [8]. This system is used for retrieval from both image and video databases. In addition to the above features the video retrieval system also uses camera and object motion in defining local properties in the video data. Similar applications have also been developed by Virage[1]. Webseek[2] developed at Columbia University has cataloged a large number of images and videos for retrieval by content. MIT's Photobook makes extensive use of image texture features [31]. This system also incorporates the use of eigenface features for face recognition. In their article, the authors of the QBIC system have identified an important issue in the development of visual information databases. The database systems that use the above features can at best define the image content with partial semantics. In essence these features are various forms of quantifiable measurements of image features. Humans, on the other hand, attach semantic meaning to image content. This has introduced a recent interest in understanding images as visual information rather than merely as a set of features. This is the basis of an emerging field of Visual Information Retrieval (VIR) Systems [14].

Visual information comprises of visual data coupled with semantic association. While the methods for extracting/matching images remain statistical in nature [15], CBIR systems which incorporate this concept attach the annotations to the extracted features to aid in a semantic retrieval. Recent work in this area has extended to including similarity matching methods published in the psychological literature to develop more robust content annotation subsystems.

Video information is also included in the CBIR and VIR systems. Video data is being produced in large quantities on a daily basis and maintaining this enormous amount of video data is a challenging task. CBIR systems look for significant events in video data, both in the spatial and temporal domains. Events such as camera shots, scene cuts, gradual transitions and camera motion

[1] `http://www.virage.com`

[2] `http://disney.ctr.columbia.edu/webseek/`

such as pans, zooms etc. form the additional set of features which are necessary to be detected to make a query system effective.

Applications where CBIR systems can be effectively utilized are numerous and diverse. They include scientific database management, picture archiving and communication systems, law enforcement and criminal investigation, geographic information systems and video-on-demand systems among numerous others. The list is endless and growing at a high rate. With the improved processing speeds, communication data rates and development of large multimedia servers, the research in this field has spread beyond the domain of computer vision, pattern recognition and image analysis.

1.1 Importance of Pattern Recognition Methods in Image and Video Databases

A typical present day image and video database system block diagram is shown in Figure 1. The figure shows that the database system has two main points of interface, the database generation subsystem and the database query subsystem. Pattern recognition methods are applied in both these subsystems. The inputs

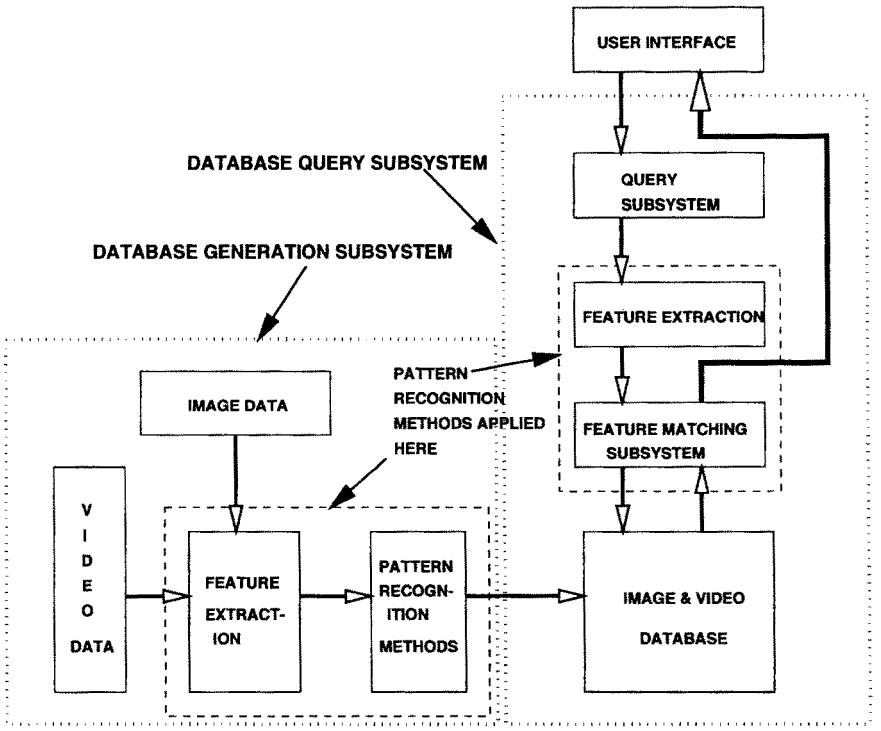

Fig. 1. *Typical present day Image and Video Database System*

to the database generation subsystem are image and video data. The necessary features are extracted from this data and then appropriately clustered to generate indices to be used by the database query subsystem. Pattern recognition methods play an important role in both the feature extraction stage and the feature clustering stage. The database query subsystem gets a query generated by the user from the user interface. This query is mapped to appropriate features by the query subsystem. If the user query has an example image or sketch, then the necessary features are extracted by the feature extraction stage. These features are then matched to the indexed elements of the database. Both the feature extraction and the feature matching stages make extensive use of the pattern recognition techniques. The techniques used for feature extraction and feature matching have been described in this paper.

Image and video databases are different in many aspects from the traditional databases containing textual information. Querying textual information is comparatively straightforward and strict. Unlike image and video data which can be perceived differently by different users, the text data has very few variables attached to its meaning. The computer vision and image processing community has woken to this fact only very recently. Although many methods have been developed for querying images that make use of exotic features extracted from the visual information, the most successful methods have been found to be simple features like color histograms and its variants. This is almost akin to the principle followed in the engineering design community "*a highly complex and seemingly efficient method is so only for the application for which it has been designed. It may fail to perform adequately when applied to generalized situations*".

The difference between the traditional database systems and the image (and video) databases starts from the way the queries are specified. While standardized query languages like SQL can efficiently express the intent of the user, it is very difficult for a user to express the content of an image. The queries tend to be in natural languages and are in themselves quite complex. Aside from this problem, the focus for the computer vision and pattern recognition community is to develop methods which can convert the query specification into meaningful set of query features which can be applied to retrieve the visual data. It has also been found that humans tend to specify global image features more than specifics of the image. This differentiates traditional object recognition and tracking methods from the methods that could be used to query images. In traditional object recognition the expected geometry, shape, texture, and color of the object are known aforehand. But, with a *real* image and video database system, the data being added to the system is possibly coming from diverse environments. This leads us to face the reality that a fixed and predecided set of features will not work well. Also, typical users of such systems cannot comprehend the complex methods used to implement the image retrieval. Hence a more intuitive interface needs to be given to the users, so that the query can then be mapped to appropriate parameters for the retrieval subsystem.

All of the above lead to the need for a middle layer between the user query and the retrieval subsystem that will translate the query into the use of an

appropriate feature to query the image. This feature should be easily computable in real-time and also be able to capture the essence of the human query. Thus, the modified block diagram of the *enhanced* database system would have only one block for doing all the pattern recognition. There will be no need for indexing the features of the image and video data, since appropriate feature extraction and mapping will be done to this data *on-the-fly*. Thus, the work done by the pattern recognition community in developing such methods is of extreme importance to the development of the image and video database systems.

This paper describes some of the significant contributions that use pattern recognition methods for content retrieval from image and video databases. The remainder of the paper is organized as follows. Section 2 describes some of the successful image/video database systems briefly. Section 3 describes the features used and pattern recognition methods employed for image databases. Section 4, on the other hand, describes the classification methods used for features extracted from video data. Section 5 discusses some similarity matching methods. We have separated the discussion on the multimedia database systems into image database and video database systems. We describe our projection of the future of visual information systems in Section 6 and conclude with Section 7.

2 Image and Video Database Systems: Examples

Several successful multimedia systems which have been developed in recent years are described in the literature. Amongst these, the notable systems include The QBIC Project [28], The Photobook System [31], The MARCO Project [37], The Manchester Multimedia Information System [11], VisualGREP [23] and Informedia Digital Library [44]. In this section we shall highlight the features of some of these systems, the features they use and the primary classification method(s) utilized by them. A book by Gong [12] discusses some of the image database systems and describes the Advanced Region Based Image Retrieval System (ARBIRS) developed by the author.

2.1 Query-By-Image-Content: QBIC

The Query-By-Image-(and Video)-Content (QBIC) System developed at IBM Almaden Research Center uses a variety of features for retrieving images from the image/video database. The system is described as a set of technologies and associated software that allows a user to search, browse and retrieve image, graphic and video data from large on-line collections [28]. The system allows image and video databases to be queried using visual features such as color, layout and texture. In QBIC the queries are matched pictorially so that users can match their perception of the visual features without using words. The query is matched against a database of precomputed features clustered meaningfully. A special feature in QBIC is the use of gray level images. With the lack of color in these images, it is impossible to use one of the most powerful cues for matching images. For such images the spatial distribution of gray level texture

and edge information are used. The feature vector thus generated has a very high dimension (sometimes over a 100 dimensions). In QBIC the queries are posed in several ways. In Query-by-Example, the user can select an image from the thumbnails of images within the database or specify an image and request similar images. The user can also sketch an image or parts thereof for describing the query image.

Recently, the feature of generating video storyboards has been added to QBIC. A storyboard consists of representative frames selected from subsequences within the video. Each subsequence is separated from the other by significant changes such as scene cuts or gradual transitions. Once the storyboard has been generated for the MPEG-1 compressed video sequences, the methods discussed above can be applied to these representative frames to retrieve video clips by content.

2.2 Photobook

The Photobook System developed at the MIT's Media Labs is described as a set of interactive tools for browsing and searching images and image sequences [31]. Direct search in image content is made possible through *semantics preserving image compression.* Photobook allows search based on 2-D shape, gray level appearance and textural properties. The focus of this system is the semantics preserving image compression which replaces the image in the database with a set of parameters which can be used to reconstruct the image in its entirety. This differs from the other methods which find features from the image which can be used to perform similarity matching. These features are from selected parts of the image and cannot be used to reconstruct the image. Thus, the Photobook System database has *perceptually complete* and *semantically meaningful* data. To generate this from the images it uses the Karhunen-Loeve Transform (KLT) and the Wold Decomposition Methods. The Appearance Photobook uses the KLT, the Shape Photobook uses the Finite Element Methods while the Texture Photobook uses the Wold Decomposition Methods to extract the features from the image. The Wold Decomposition Model developed by Picard and Liu [25] is for regular stationary stochastic process in 2-D images. The Wold Decomposition produces a compact texture description that preserves most of the texture's perceptual attributes. If the image is assumed to be a homogeneous 2-D discrete random field, then the 2-D Wold Decomposition is a superposition of three mutually orthogonal components; a harmonic field, a generalized evanescent (transient) field and a purely indeterministic field. Qualitatively these fields appear as periodicity, directionality and randomness in textures.

2.3 Informedia Digital Library

The Informedia Digital Library [44] developed at Carnegie Mellon University is a full fledged digital video library under development. The developers of the system project that a typical digital library user will differ from the video-on-demand user who generally browses the entire length of the video. The digital

library user's interests will lie in short video clips and content-specific segments. These segments have been called *skims* by the authors. The library developers have designed methods to create a short synopsis of each video. Language understanding is applied to the audio track to extract meaningful keywords. Each video in the database is then represented as a group of representative frames extracted from the video at points of significant activity. This activity may be abrupt scene breaks, some form of rapid camera movement, gradual changes from one scene to another, and points in the video where the keywords appear. Caption text[3] is also extracted from these frames which add to the set of indices for the video.

2.4 Other Systems

The Manchester Multimedia System [11] uses geometric features such as area, perimeter, length, bounding box, longest chord etc. of the image objects as features. Work by Petrakis and Orphanoudakis [33] demonstrates the application of CBIR to Medical MRI databases. In this system the images are classified on the basis of the relationship between its component objects. VisualGREP [23] is a system for comparing and retrieving video sequences. The features used by this system are the color atmosphere represented as Color Coherence Vector (CCV), the motion intensity represented by the Edge Change Ratio (ECR), and presence of frontal face in the image. The face detector is a neural network based system and uses the Euclidean distance between eigenfaces of the training and test images. The query can be placed in terms of the above features and is tested on aggregated and partially aggregated sequences.

Commerical systems have also been developed by Virage Inc. The developers of WebSeek have placed the system on the web. This system can be tested by an Internet user to get a first hand experience of content based image retrieval. Other than the Informedia project, there has been no significant development of a complete and commercially viable video database system.

3 Pattern Recognition Methods for Image Databases

This section describes work done for retrieval by content from image databases. The methods described in the literature have been found to use three types of features extracted from the image. These features are color based, shape based and texture based. Some systems, as described in earlier sections use two or all three features to index the image database.

3.1 Color Based Features

Color has been the most widely used feature in CBIR systems. It is a strong cue for retrieval of images and also is computationally least intensive. Extensions of

[3] Text added in video post-production by graphic editing machines

the color feature have also been successfully applied to video indexing. This has been discussed in detail in the next section. Color indexing methods have been studied using many color spaces. To name a few, the RGB space, the HSV space, the CIE $L^*u^*v^*$ space, the Munsell space etc. The effectiveness of using color is that it is an identifying feature that is local to the image and largely independent of view and resolution [50]. The major obstacles that should be overcome to find a good *image* match are

- variation in viewpoint,
- occlusion, and
- varying image resolution.

Swain and Ballard [50] use histogram intersection to match the query image and the database image. Histogram intersection is robust against these problems and they describe a preprocessing method to overcome change in lighting conditions. The authors use the HSV color space and the opponent color space in their experiments. Other systems use the L_1, L_2 and the L_∞ measures [48]. The QBIC system [3] computes the Euclidean distance between the user specified average color of an object or an image in Munsell color space. If the histogram based query is selected, then the user can specify a histogram or specify an image or a part of it. The query histogram and the database histogram are compared using the distance measure

$$d = (r - q)^T A (r - q) \tag{1}$$

where, r and q are the histograms specified as k-dimensional vectors wherein the i^{th} bucket is the percent of color-i in the image. Element a_{ij} of matrix A is the color similarity of color i to color j.

Other color based features are derived from block based segmentation of the image. This has been used in [45]. The image is represented using a 5-dimensional color feature vector. The features are the mean, standard deviation, RMS, skew and kurtosis of the image pixel $I(p)$ in a window. The image similarity is then a weighted Euclidean distance between the corresponding feature vectors. If $df(f, I)$ denotes the value of distribution of feature f in image I, F denotes the number of features used to describe the color distribution and w_f is the weight of the f^{th} feature vector, then the similarity score using these statistical feature vectors is given by

$$S_{stat}(A, B) = \sqrt{\sum_{f=1}^{F} w_f (df(f, A) - df(f, B))^2} \tag{2}$$

Weighting factor w_f is larger for features of smaller variance. In [47] each image with is tessellated with five partially overlapping, fuzzy regions. For each image the average color and color covariance for each fuzzy region are matched by summing the correlation of the features between the images. In the PICASSO system [6] the image is hierarchically organized into non overlapping blocks. The

clustering of these blocks is then done on the basis of color energy in the CIE $L^*u^*v^*$ color space.

Other systems use 2D - pseudo-hidden Markov model (PHMM) for color based CBIR [24]. The reason for using 2D-PHMM is its flexibility in image specification, ease of image matching and the ability to handle *not cared* regions in the image. Also shown is that using a subset of the color moments in the HSV space results in much better image retrieval than the other histogram based measures. Idris and Panchanathan [18] have used the LBG algorithm to generate codewords for the images in the database. The matching is done using histogram intersection and the L_2 metric.

Gevers and Smeulders [10] take a different approach to color image indexing. They state that the color based image indexing systems published in the literature do not take into account the camera position, geometry and illumination when building the histogram. In their system PicToSeek, the authors use the color variation, color saturation, color transition strength, color background and grayness as the indexing features. The color transition strength refers to the number of hue changes in the image. An image with a lot of detail will have more transitions than an image with few changes. The background is the *supposed* background of the image. It is the histogram bin with the highest color or grey value count. The HSV color system is selected since it has the value parameter which is due to the intensity in the image which can be extended to the luminosity in the scene. A reflection model for the object is built in the sensor space and the photometric color invariant features are computed. In their experiments, histogram cross-correlation provided the best results. The system is described in great detail [9].

Pass *et. al* [29] have developed color coherence histograms to overcome the matching problems with standard color histograms. The authors explain the need for color coherence vectors (CCV), a vector representation of the color coherence histograms, with the example of an image with scattered red points, as in a foliage, being matched with an image with a large region of red. Since the both image histograms will have a large number of red pixels, the histograms will be deemed similar, though the content may be quite different. CCVs include spatial information along with the color density information. Each bin j is replaced with a tuple (α_j, β_j), where α is the number of pixels of the color in bin j that are coherent, or belong to a contiguous region of largely that color. The β number indicates the number of incoherent such pixels. It is shown that substituting the tuple in a standard L_1 norm histogram distance measure, images which had been earlier deemed as similar are now marked as dissimilar. This improves the matching performance and quality of retrieval of the image database system. Another approach by Huang *et. al* [17] is the use of color correlograms. A color correlogram expresses the spatial correlation of pairs of colors with distance. An index entry for (i, j) gives the probability that the color of the pixel k distance away from the current pixel will be c_j. This includes the information of the spatial correlation between colors in an image, thus making it robust against change of viewpoint, zoom-in, zoom out etc.

The color indexing methods discussed above are by no means complete. There has been much work published in the literature which discusses different approaches to using color as a feature. This goes to strengthen the case for using color as a feature, but very little variety is observed on the whole in the approach to using color. These methods only use different ways of looking at the color histograms. The discussion above has tried to present the few important contributions that attempt to take a different direction in color based indexing of images.

3.2 Shape Based Features

Different approaches are taken for matching shapes by the CBIR systems. Some researchers have projected their use as a matching tool in QPE type queries. Others have projected its use for Query-by-User-Sketch type queries. The argument for the latter being that in a user sketch the human perception of image similarity is inherent and the image matching subsystem does not need to develop models of human measures of similarity. Several methods have been published in the literature which address this method. One method adopts the use of deformable image templates to match user sketches to the database images [7]. Since the user sketch cannot be an exact match of the shape in the database, the method elastically deform the user template to match the image contours. An image for which the template has to undergo minimal deformation, or, loses minimum energy, is considered as the best match. It is also necessary for the match to follow the edge of the image as closely as possible. If the deformation lies entirely on the image areas where the gradient is maximum then the match is good. while a low match means that the template is lying in areas where the image gradient is 0. Then, by maximizing the matching function and minimizing the elastic deformation energy, a match can be found. Similar work on energy based deformation has been done in [40]. The author deforms image prototype templates to match the images.

A different approach to CBIR based on shape has been through use of implicit polynomials for effective representation of geometric shape structures. [21]. Implicit polynomials are robust, stable and exhibit invariant properties. The method is based on fixing a d degree polynomial to a curve patch of length l. By applying certain transformations the curve representation is made invariant to affine transformations. A vector v consisting of the parameters of this curve is used to match the image to the query. A typical database would contain the boundary curve vectors at various resolutions to make the matching robust.

Rui *et. al* [36] propose multiple matching methods to make the retrieval robust. They define the requirements of the parameter as invariance and compact form of representation. The method selected by them is the Fourier Descriptor (FD) of a shape. Since Discrete Fourier Transform (DFT) suffers from the staircase effect, it is not invariant to affine transformations. The authors define a Modified Fourier Descriptor (MFD) which is an interpolated form of the low frequency coefficients of the FD normalized to unit arc-length. They also calculate the orientation of the major axis. The matching of the images is then done

using the Euclidean distance, MFD matching, Chamfer distance and Hausdorff distance. Although these matching tools have been used in this system, they can be used to match shapes which have been specified using appropriate descriptors.

Jain and Vailaya [19] define shape based matching based on the histogram of the directions of its significant edges. The edges are generated by using the Canny edge detector and the histograms are compared using the histogram intersection technique. This technique is fast but has its limitations. A histogram of edge directions is invariant to translation. It is also invariant to scaling if it is normalized to the number of edges in the image. A smoothed histogram is invariant to rotation, if the values in the bins can be appropriately shifted.

3.3 Texture Based Features

The visual characteristics of homogeneous regions of real-world images are often identified as texture. These regions may contain unique visual patterns or spatial arrangements of pixels which gray-level or color in a region alone may not sufficiently describe. Typically, textures have been found to have strong statistical, structural or their combined properties. The textures have been expressed using several methods. One system uses the Quadrature Mirror Filter (QMF) representation of the textures on a Quad-tree segmentation of the image [43]. Fisher's Discriminant Analysis is the used to determine a good discriminant function for the texture features. The distance used to classify the features is the Mahalanobis Distance. Several other features based on the local distribution of the pixel gray levels have been used [4]. An image can be described by means of features of different orders of statistics of the gray values of the pixels inside a neighborhood. The features extracted from the image histogram, called the first order features, are mean, standard deviation, third moment and entropy. The second order features are homogeneity, contrast, entropy, correlation, directionality and uniformity of the gray-level pixels derived from the gray-level co-occurrence matrices. Also included is the use of several other third order statistics from run-length matrices. A vector composed of these features is then classified based on the Euclidean Distance.

The Gabor Filter and Wavelets can also be used to generate the feature vector for the texture [26]. The assumption is that the texture regions are locally homogeneous. The feature vector is then constructed from multiple scales and orientations comprising of the means and standard deviations at each.

Strobel *et al.* [49] have developed a modified maximum a posteriori (MMAP) algorithm to discriminate between hard to separate textures. The authors use the normalized first order features, two circular Moran autocorrelation features and Pentland's fractal dimension to which local mean and variance has been added. The Euclidean Distance metric is used to determine the match. To decorrelate the features the Singular Value Decomposition (SVD) algorithm is applied. To discriminate hard-to-differentiate textures, a sliding window is used. The features are extracted within this window. Then they build a *raw* map using the Tree Structured Vector Quantizer (TSVQ) which is faster than training classical algorithms like k-means. Since the windows overlap each other the MMAP

algorithm is applied as a post processing step. The probability mass functions (histograms) of labels derived from the entire map and this region are calculated. Then, the estimate of a label which maximizes the probability that it belongs to a certain class is calculated. The distribution is assumed uniform and the texture classes have equal probability.

In the Texture Photobook due to Pentand et. al [32] the authors use the model developed by Liu and Picard [25] based on the Wold decomposition for regular stationary stochastic processes in 2-D images. This model generates parameters that are close if the images are similar. If the image is treated as a homogeneous 2-D discrete random field, then the 2-D Wold like decomposition is a sum of three mutually orthogonal components which qualitatively appear as periodicity, directionality and randomness respectively. The Photobook consist of three stages. The first stage determines if there is a strong periodic structure. The second stage of processing occurs for periodic images on the peaks of their Fourier Transform magnitudes. The third stage of processing is applied when the image is not highly structural. The complexity component is modeled by use of a multiscale simultaneous autoregressive (SAR) model. The SAR parameters of different textures are compared using the Mahalanobis distance measure.

Other work done by Puzicha et. al [35] uses Gabor Filters to generate the image coefficients. Then the probability distribution function of the coefficients is used in several distance measures to determine image similarity. The distances used are the Kolmogorov-Smirnov distance, the χ^2-statistic, the Jeffery divergence, weighted mean and variance among others.

4 Pattern Recognition Methods for Video Databases

Pattern Recognition Methods are applied at several stages in the generation and the use of a video database. Thus far, we have seen the use of pattern recognition methods to determine the similarity between the query image (picture, sketch) and the images in the database. Video data has the added feature of time added to the spatial information contained in each *frame*. A series of frames make a video *sequence* (or clip). Typically a video sequence is made up of several *subsequences*, which are uniform in content between their start and end points marked by *cuts*. Some times these end points may also be gradual changes, called by the generic name *gradual transitions*. Gradual Transitions are slow changes from the scene in one subsequence to the next. These slow changes are further classified as *blends*, *wipes*, *dissolves* etc.

In the development of a video database it is necessary to structure the video data into a compact and meaningful representation to aid search by content. Pattern recognition methods are hence applied in this structuring process and then subsequently in the retrieving process. It has been observed that if the subsequences in the video clips can be identified and the scene change points marked, then mosaicing the data within a subsequence can provide the necessary representation for efficient retrieval. The retrieval methods are the same as those studied for image databases. Thus, an important step in the generation of a video

database is the marking of the cut (or transition) points. Some methods collect all *similar* frames and find places where the similarity is at the lowest while others take the approach of finding differences between video frames and find peaks in the difference plot. This process of finding scene changes in a video sequence is also called *indexing*. Some methods go beyond this step to find the *semantics* of the video subsequence by marking the camera motion parameters and classifying it as *zooms*, *pans* etc.

Ahanger and Little [1] have surveyed the technologies for indexing digital video. The simplest, and probably the most noisy, method for detecting scene changes is performing a pixel level differencing between two succeeding video frames. If the percentage change in the pixels is greater than a threshold T, then a scene break is declared. Another simple method is to perform the Likelihood Ratio test. In this, each frame is divided into k equally sized blocks. Then, the mean (μ) and variance(σ^2) for each block is calculated. If the likelihood λ, defined as

$$\lambda = \frac{[((\sigma_i^2 + \sigma_{i+1}^2)/2) + ((\mu_i - \mu_{i+1})/2)^2]^2}{\sigma_i^2 * \sigma_{i+1}^2} \tag{3}$$

is greater than a threshold T, then the count of the blocks changed is incremented. The process is repeated for all the blocks. If the number of blocks changed is greater than another threshold T_B, then a scene break is declared. This method is a little more robust than the pixel differencing method but is insensitive to two blocks being different yet having the same density functions.

A feature based approach to video segmentation has been developed by Zabih *et al.* [53]. The segmentation process analyzes the intensity edges between two consecutive frames. The hypothesis is that during scene changes new edges will appear replacing the old ones. By counting the new and old edge pixels cuts, fades and dissolves are recognized. Hampapur *et. al* [16] have developed a different model for detecting edit points in a video. The use chromatic scaling to determine the locations of cuts, dissolves, fades etc. Other methods have been classified as color based or motion based methods and are included in the following subsections.

4.1 Color Based Video Classification

Many methods have been developed that use the color and/or intensity content of the video frame to classify scene changes. The Informedia Digital Library Project[4] at Carnegie Mellon University (CMU) uses one of these methods. Smith and Kanade [44] describe the use of comparative histogram difference measure for marking scene breaks. The peaks, representing scene breaks, in the difference plot are detected using a threshold.

Sethi and Patel [30][41] have developed methods to operate on MPEG[5] compressed video sequences. Intensity histograms of the frames are built and the hypotheses that the two frames belong to the same class (subsequence) or different

[4] `http://www.informedia.cs.cmu.edu`

[5] MPEG: Motion Pictures Experts Group (`http://www.mpeg.org`)

is tested using the Yakimovsky Likelihood Ratio, χ^2 and Kolmogorov-Smirnov Tests. All these tests are done for dissimilarity, that is a peak in the values would indicate a scene change.

Yeo and Liu [52] use DC images recovered from MPEG compressed sequences to detect scene changes and classify gradual dissolves. Scene changes are detected by two methods. The first using the sum of histogram differences for each color band (R, G and B). This is computationally more intensive. The second method detects cuts as well as gradual transition. A sliding window is moved temporally across the video sequence. At each instance of the window differences between successive frames are calculated using DC intensity histograms. A peak in the difference plot indicates a cut. Since the window is overlapping, a gradual transition will be represented by a slow rise followed by a plateau and then a slow drop. By detecting this pattern a gradual scene change can be marked.

4.2 Motion Based Video Classification

Akutsu *et al.* [2] have developed a motion vector based video indexing method. Motion vectors refer to the vector generated by a feature moving from one location in a frame to another location in the succeeding frame. In their method a motion vector is determined by minimizing

$$mb = \sum_{r \in b} \{I(r + m(r, f), f + k) - I(r, f)\}^2 \tag{4}$$

where mb is the motion vector, $I(r, f)$ represents the image frame where r denotes the spatial coordinates and f is the frame number (time). The mapping to reconstruct this frame from the $(f + k)^{th}$ frame is $m(r, f)$. Each frame is divided into b non-overlapping blocks. Cut detection is based on the average inter-frame correlation coefficient based on the motion vectors. Two methods are proposed to determine the coefficient. The first, given by

$$\frac{\sum_{r \in b}[I(r + m(r, f), f + k)' - I(r + m(r, f), f + k)][I(r, f)' - I(r, f)]}{\sqrt{\sum_{r \in b}[I(r + m(r, f), f + k)' - I(r + m(r, f), f + k)]^2[I(r, f)' - I(r, f)]^2}} \tag{5}$$

is to maximize the correlation coefficient. In the equation $I(r+m(r, f), f+k)'$ and $I(r, f)'$ are the average values of $I(r + m(r, f), f + k)$ and $I(r, f)$. The average inter-frame correlation coefficient is calculated from the maximum correlation coefficients. This value represents the inter-frame similarity based on motion. The other value is the ratio of velocity to motion in each frame. This value, denoted by W is given by

$$\begin{aligned} W_1 &= \sum w1, w1 = \begin{cases} 1 \; m(r, f) \neq 0 \\ 0 \text{ otherwise} \end{cases} \\ W_2 &= \sum w2, w2 = \begin{cases} 1 \; m(r, f + k) - m(r, f) \neq 0 \\ 0 \text{ otherwise} \end{cases} \\ W &= W_1/W_2 \end{aligned} \tag{6}$$

W represents motion smoothness and its value is infinity at a cut. Camera operations can also be detected by applying motion analysis to the motion vectors using the Hough transform. A spatial line in the image space is represented as a point in the Hough space. Conversely, a sinusoid in the Hough space represents a group of lines in the image space intersecting at the same point. In this case, the lines intersect at the convergence/divergence point of the motion vectors. Thus, if the Hough transform of the motion vectors is least squares fit to a sinusoid, the point of divergence/convergence of vectors indicates the type of camera motion. The paper describes 7 possible camera operations using the convergence/divergence point and whether the vector magnitudes are constant, changing or zero.

Shahraray [42] has also developed a block matching technique. The matching is done on the image divided into 12 blocks. The image intensity is used by the matching process and the match values are normalized between 0 and 1. Here 0 indicates a perfect match. The image match coefficient IM is given by

$$IM = \sum_{i=1}^{K} c_i l_i \tag{7}$$

where i is the block number, K is the total number of blocks, l_i the ordered set of match values and c_i is a predetermined constant value for each block.

The Informedia Project [44] at CMU also defines camera motion within a scene. Trackable features of the object are used to form flow vectors. The mean length, mean phase and phase variance of the flow vectors determine if the scene is a static scene, pan or a zoom. Sethi and Patel [30] have applied a decision tree based classifier to detect the camera motion. The feature vectors is generated from the motion predicted frames of MPEG compressed video data.

5 Similarity Measures

The systems discussed above use pattern recognition and classification methods to determine image similarity. The example systems discussed use color, texture and shape as the primary features with which the images are matched. This section discusses some of the classification methods. Also, discussed are some studies done on these methods and their application to CBIR.

The most common indexing scheme adopted in image databases is the use of color histograms. A metric on the color histogram space is used to determine the similarity of the images. In this technique, the colors in an image are mapped into a color space containing n discrete colors. A color histogram $H(M)$ is a vector $(h_1, h_2, ..., h_n)$ in a n-dimensional vector space, where each element h_j represents the number of pixels of color j in the image M. A metric on the histogram space is used to measure the distance between histograms. For a given distance t, we say that the histograms are *t-similar* if their distance is less than or equal to t and *t-different* otherwise.

Stricker [46] has done a study on the discrimination power of the histogram based color indexing techniques. In his work he makes the observation that histogram based techniques would work effectively only if the histograms are sparse. He also determines the lower and upper bounds on the number of histograms that fit in the defined color space and suggests ways to determine the threshold based on these bounds.

Some of the common histogram comparison tests are histogram intersection, histogram comparison, Yakimovsky likelihood ratio test, χ^2 test, Kolmogorov-Smirnov test, the L_1-,L_2- and L_∞- measures.

Histogram Comparison If H_I and H_J represent the histograms then the difference between these histograms is given by $D(t)$

$$D = \sum_{i=0}^{N} |H_I(i) - H_J(i)| \tag{8}$$

where i and j are indexes into the histograms. A high value of D represents high dissimilarity.

Histogram Intersection Histogram intersection is defined as

$$H(I, M) = \frac{\sum_{j=1} n \min(I_j, M_j)}{\sum_{j=1} n M_j} \tag{9}$$

where I and M represent the image histograms having n bins and i and j are indexes into the histograms.

Yakimovsky Likelihood Ratio Test: The Yakimovsky test was proposed to detect the presence of an edge at the boundary of two regions. The expression for Yakimovsky Likelihood Ratio is given by

$$y = \frac{(\sigma_0^2)^{m+n}}{(\sigma_1^2)^m (\sigma_2^2)^n} \tag{10}$$

where σ_1^2 and σ_2^2 are the individual variances of the histograms while σ_0^2 is the variance of the histogram generated from the pooled data. The numbers m and n are the number of elements in the histogram. A low value of y indicates a high similarity.

The Chi-Square Test: The χ^2 test is given by

$$\chi^2 = \sum_j \frac{(HP_j - HC_j)^2}{(HP_j + HC_j)^2} \tag{11}$$

where HP_j and HC_j represent the number of entities in the j^{th} bin of the histograms. A low value for χ^2 indicates a good match.

Kolmogorov-Smirnov Test: This test is based on the cumulative distribution of the two sets of da'a. If CHP_j and CHC_j represent the cumulated number of entities up to the j^{th} bin of the histograms of the previous and current frame, the Kolomogorov-Smirnov statistic is given by

$$D = \max_j |CHP_j - CHC_j| \tag{12}$$

Again, a low value of D results for a good match.

Soffer [45] uses $N \times M$-grams to develop a similarity measure for image matching. A $N \times M$-gram is a $N \times M$ portion of the database image that repeats frequently in the test image. The similarity between image is gauged in terms of the frequency of a $N \times M$-gram and the number of common $N \times M$-grams. Lew *et. al* [22] apply the Fisher Linear Discriminant and the Karhunen-Loeve Transform to match a query image. The KLT determines the optimal linear features for describing a data set. Fisher's Linear Discriminants determine the optimal linear features for classification.

Minka and Picard [27] state that it is often difficult to select the right set of features to extract to make the image database robust. They use multiple feature models in a system that *learns* from the user queries and responses about the type of feature that would best serve the user. The combinatronics of using multiple features is reduced by a multistage clustering and weight generation process. The stages closest to the user are trained the fastest and the adaptations are propagated to earlier stages improving overall performance with each use.

Other important feature used in CBIR systems is shape. Shape is represented in forms of Fourier Descriptors, d dimensional implicit polynomials, set of gradients etc. These are matched using Euclidean distance, Chamfer distance, Hausdorff distance etc. Texture features are also matched using the similarity matching methods described above in addition some methods use the Mahalanobis distance.

5.1 Similarity Matching

Santini and Jain [38] have studied similarity measures for selecting images based on characteristics of human similarity measurement. The authors apply results from human psychological studies to develop a metric for qualifying image similarity. The basis for their work is the need to find a similarity measure relatively independent of the feature space used. They state that if S_A and S_B are two stimuli, represented as vectors in a space of suitable dimension, then the similarity between the two can be measured by a psychological distance function $d(S_A, S_B)$. This introduces the difference between *perceived dissimilarity* d and *judged dissimilarity* δ. The two are related by a monotonically decreasing function g given by

$$\delta(S_A, S_B) = g[d(S_A, S_B)] \tag{13}$$

where the requirements of the distance function are constancy of self-similarity, minimality (satisfied by the monotonicity of Eq. 13), and symmetrical distance between the stimuli.

The authors lay the foundation for Set-Theoretic and Fuzzy Set-Theoretic Similarity Measures. They also introduce three operators which are similar in nature to the *and*, *or* and *not* operators used in relational databases but are able to powerfully express complex similarity queries. In [39] the authors show that the assumption made by most similarity based methods that the feature space is Euclidean is incorrect. They analyze some similarity measures proposed in the psychological literature to model human similarity perception, and show that all of them challenge the Euclidean assumption in non trivial ways. The suggestion is that similarity criteria must work not only for images *very* similar to the query, as would be the case for matching, but for all the images *reasonably* similar to the query. The global characteristics of the distance measure used are much more important in this case.

The authors describe some similarity examples [39] to which the matching method can be applied. Silhouettes are used to evaluate the method. The propositions used to assess the similarity are the predicates: *The vertical sides are complex*, *The horizontal sides are complex*, *The figure is complex*, *The figure is slender*, and *The figure is equilibrated.* The truths of these predicates determine the similarity. For textures, a wavelet transform of the image is taken, and the energy (sum of the squares of the coefficients) in each sub-band is computed and included in the feature vector. To apply the similarity theory, a limited number of predicate-like features such as *luminosity*, *scale*, *verticality* and *horizontality* are computed. For color histograms the predicate used was of the form 'color x is present in the image". The distance between two histograms h_1 and h_2 is given by

$$\begin{aligned} D(h_1, h_2) = &-\theta \sum_{i=1}^{n_b} \min\{\mu(h_1[i]), \mu(h_2[i])\} \\ &+ \alpha \sum_{i=1}^{n_b} \max\{\mu(h_1[i]) - \mu(h_2[i]), 0\} \\ &+ \beta \sum_{i=1}^{n_b} \max\{\mu(h_1[i]) - \mu(h_2[i]), 0\} \end{aligned} \tag{14}$$

where n_b is the number of histogram bins, α and β are constants which decide the importance of the number of colors specified in the query. The distance function is defined according to the Fuzzy Tversky model [39].

All methods discussed above are effective only if the features used by them to match images are stored in efficient structures. Commonly, the image databases cluster the features to enhance the similarity search. Similarity searches often are generalized k-nearest neighbor searches where performance and accuracy can be easily traded. The features in this case are often stored on tree structures called Similarity Search (SS)-Trees. The problem with tree structures when applied to high dimensional data is that the search engine has to search a significant portion of the data before it can arrive at a decision. Kurniawati et. al [20]

have developed an improved tree structure, called SS^+-Tree, for searches in a high-dimensional feature space. They employ the widely used k-means clustering algorithm. A clustering of the nodes at a particular level in the search tree is done and the centroid of this cluster is used to determine if a new level should be created. Pun and Squire [34] use Correspondence Analysis and the χ^2 metric to determine the independence between the features and form a suitable index structure to use this independence effectively. White and Jain [51] propose a variant of the $k-d$ tree called the VAM $k-d$ tree. This is so named because its split orientation is based on its *variance* and the split position is *approximately* the *median*. The authors also describe a method to construct an R-tree based on the VAM $k-d$ tree called the VAMSplit R-tree.

6 Future Directions

The image and video database systems are becoming more important everyday. In the light of this growth the future seems limitless for them. Yet, as we have highlighted above, many improvements are necessary to these systems before they can be used effectively by the average user. The future for these systems demands a more conceptually simple interface coupled with natural language query system and a middle level subsystem which could map these queries into suitable features and pattern recognition methods *on-the-fly*. These changes would then demand simple, yet effective methods for capturing the visual features of the image and video data.

Once the above enhancement is incorporated into the image and video database systems, the next step in the development of the database systems is content based interactivity on live, dynamic objects. The systems would have interactive ability beyond the content-based query operations on stored, mono-stream media objects offered by today's systems. This has been dubbed as *Presence Technology* [6]. This technology stems from the basis that every bit of information that is captured in various environments through cameras, microphones, infra-red sensors, etc. is in some sense visual information that is meant to be searched and interacted with. The technology would allow individuals to be a part of a remote, live environment through a reception-device. The system would blend component technologies like heterogeneous sensor fusion, live media delivery, telepresence and information systems into a novel set of functionality that enables the user to perceive, move around, enquire about, and interact with the remote environment through her reception and control devices. The system also would assimilate information from multiple sources over the environment of interest to prepare a Gestalt information model of the environment, so that the user can have a comprehensive perception of the environment.

[6] White paper on The Presence Technology at Praja Inc. `http://www.praja.com`

7 Conclusions

This paper has detailed various systems that have been developed to query image and video databases by their content. These database systems use pattern recognition methods for feature extraction, clustering and grouping the features, and matching the features to the query image. Pattern recognition thus plays a significant role in content based recognition and has applications in more than one subsystem in the database. Yet, very little work has been done on addressing the issue of human perception of visual data content. The approaches taken by the computer vision, image analysis and pattern recognition community have been *bottom up*. It is not only necessary to develop better pattern recognition methods to capture the visually important features from the image, but also to develop them such that they are simple, efficient and easily mapped to human queries. Techniques such as color histograms have been immensely successful in retrieving the queried data because of their global approach to images. This is as opposed to local approaches taken by systems that try to identify objects and other image components. The Pattern Recognition community with its rich tradition of making fundamental contributions to diverse applications has a great opportunity to make break-through advances in content based image and video information systems.

References

1. G. Ahanger and T. D. C. Little. A Survey of Technologies for Parsing and Indexing Digital Video. *Journal of Visual Communication and Image Representation, special issue on Digital Libraries*, 7(1):28–43, 1996.
2. A. Akutsu et al. Video Indexing using Motion Vectors. In *Proceedings of SPIE Visual Communications and Image Processing*, volume 1818, pages 1522–1530, 1992.
3. J. Ashley, R. Barber, M. D. Flickner, J. L. Hafner, D. Lee, W. Niblack, and D. Petkovic. Automatic and semiautomatic methods for image annotation and retrieval in QBIC. *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases III, Vol. SPIE 2420*, pages 24–35, 1995.
4. M. Borchani and G. Stammon. Use of texture features for image classification and retrieval. In *Proceedings of IS&T/SPIE Conference on Multimedia Storage and Archiving Systems II, Vol. SPIE 3229*, pages 401–406, 1997.
5. N.-S. Chang and K.-S. Fu. Query-by-pictorial-example. *IEEE Transactions on Software Engineering*, 6(6):519–524, 1980.
6. A. Del Bimbo, M. Mugnaini, P. Pala, and F. Turco. PICASSO: Visual querying by color perceptive regions. In *Second International Conference on Visual Information Systems (VISUAL'97)*, pages 125–131, 1997.
7. A. DelBimbo and P. Pala. Effective image retrieval using deformable templates. In *Proc. International Conference on Pattern Recognition*, pages 120–124, 1996.
8. M. Flickner, Sawhney H., W. Niblack, et al. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–31, 1995.
9. T. Gevers. *Color Image Invariant Segmentation And Retrieval.* PhD thesis, Department of WINS, University of Amsterdam, 1996.

10. T. Gevers and A. W. M. Smeulders. Pictoseek: A content-based image search system for the world wide web. In *Second International Conference on Visual Information Systems (VISUAL'97)*, pages 93–100, 1997.
11. C. Goble, M. O'Docherty, P. Crowther, M. Ireton, J. Oakley, and C. Xydeas. The manchester multimedia information system. *Proc. E. D. B. T.'92 Conf. on Advances in Database Technology*, 580:39–55, 1994.
12. Y. Gong. *Intelligent Image Databases - Towards Advanced Image Retrieval.* Kluwer Academic Publishers, Boston, 1998.
13. V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *IEEE Computer*, 28(9):18–22, 1995.
14. A. Gupta and R. Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, May 1997.
15. A. Gupta, S. Santini, and R. Jain. In search of information in visual media. *Communications of the ACM*, 40(12):34–42, 1997.
16. A. Hampapur, R. Jain, and T. Weymouth. Production Model based Digital Video Segmentation. *Journal of Multimedia Tools and Applications*, 1(1):9–46, 1995.
17. J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 762–768, 1997.
18. F. Idris and S. Panchanathan. Image and video indexing using vector quantization. *Machine Vision and Applications*, 10:43–50, 1997.
19. A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.
20. R. Kurniawati, J. S. Jin, and J. A. Sheperd. The SS^+-tree: An improved index structure for similarity searches in a high-dimensional feature space. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases V, Vol. SPIE 3022*, pages 110–120, 1997.
21. Z. Lei, T. Tasdizen, and D. Cooper. Object signature curve and invariant shape patches for geometric indexing into pictorial databases. In *Proceedings of IS&T/SPIE Conference on Multimedia Storage and Archiving Systems II, Vol. SPIE 3229*, pages 232–243, 1997.
22. M. S. Lew, D. P. Huijsmans, and D. Denteneer. Content based image retrieval: KLT, projections, or templates. In *Proc. of the First International Workshop on Image Databases and Multi-Media Search*, pages 27–34, 1996.
23. R. Lienhart, W. Effelsberg, and Jain R. VisualGREP: A systematic method to compare and retrieve video sequences. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases VI, Vol. SPIE 3312*, pages 271–282, 1997.
24. H. C. Lin, L. L. Wang, and S. N. Yang. Color image retrieval based on hidden markov-models. *IEEE Transactions on Image Processing*, 6(2):332–339, 1997.
25. F. Liu and R. W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, 1996.
26. B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
27. T. P. Minka and R. W. Picard. Interactive learning with a society of models. *Pattern Recognition*, 30(4):565–581, 1997.
28. W. Niblack, X. Zhu, J. L. Hafner, T. Breuel, et al. Updates to the QBIC system. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases VI, Vol. SPIE 3312*, pages 150–161, 1997.

29. R. Pass, G. Zabih and J. Miller. Comparing images using color coherence vectors. In *4th ACM Conference on Multimedia*, 1996.
30. N. V. Patel and I. K. Sethi. Video shot detection and characterization for video databases. In *To appear in Pattern Recognition, Special Issue on Multimedia*, 1997.
31. A. Pentland, R. W. Picard, and S. Scarloff. Photobook: Tools for content-based manipulation of image databases. In *Proceedings of IS&T/SPIE 23rd IAPR Workshop on Image and Information Systems, Vol. SPIE 2368*, pages 37–50, 1994.
32. A. P. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.
33. E. G. M. Petrakis and S. C. Orphanoudakis. Methodology for the representation, indexing and retrieval of images by content. *Image and Vision Computing*, 11(8):504–521, 1993.
34. T. Pun and D. Squire. Statistical structuring of pictorial databases for content-based image retrieval-systems. *Pattern Recognition Letters*, 17(12):1299–1310, 1996.
35. J. Puzicha, T. Hofmann, and J. M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 267–272, 1997.
36. Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega. Automatic matching tool selection using relevance feedback in MARS. In *Second International Conference on Visual Information Systems (VISUAL'97)*, pages 109–116, 1997.
37. H. Samet and A. Soffer. MARCO: MAp Retrieval by COntent. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):783–798, 1996.
38. S. Santini and R. Gupta. Similarity queries in image databases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1996.
39. S. Santini and R. Jain. Similarity matching. Personal communications with the authors.
40. S. Scarloff. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, 30(4):627–641, 1997.
41. I. K. Sethi and N. V. Patel. A Statistical Approach to Scene Change Detection. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases III, Vol. SPIE 2420*, 1995.
42. B. Shahraray. Scene Change Detection and Content-based Sampling of Video Sequences. In *SPIE/IS&T Symposium on Electronic Imaging Science and Technology: Digital Video Compression: Algorithms and Technologies*, volume 2419, 1995.
43. J. R. Smith and Chang. S-F. Quad-tree segmentation for texture-based image query. In *ACM International Conference on Multimedia*, pages 279–286, 1994.
44. M. A. Smith and T. Kanade. Video Skimming for Quick Browsing based on Audio and Image Characterization. Technical Report CMU-CS-95-186, Carnegie Mellon University, 1995.
45. A. Soffer. Image categorization using N × M-grams. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases V, Vol. SPIE 3022*, pages 121–132, 1997.
46. M. Stricker. Bounds of the discrimination power of colorindexing techniques. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases II, Vol. SPIE 2185*, pages 15–24, 1994.
47. M. Stricker and A. Dimai. Spectral covariance and fuzzy regions for image indexing. *Machine Vision and Applications*, 10(2):66–73, 1997.

48. M. Stricker and M. Orengo. Similarity of color images. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases III, Vol. SPIE 2420*, pages 381–392, 1995.
49. N. Strobel, C. S. Li, and V. Castelli. MMAP: Modified Maximum a Posteriori algorithm for image segmentation in large image/video databases. In *Proc. IEEE International Conference on Image Processing*, pages 196–199, 1997.
50. M. J. Swain and D. H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
51. D. White and Jain R. Similarity indexing: Algorithms and performance. In *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases IV Vol. SPIE 2670*, pages 62–73, 1996.
52. B. Yeo and B. Liu. Rapid Scene Analysis on Compressed Video. *IEEE Transactions on Circuits and Systems for Video technology*, 5(6), 1995.
53. R. Zabih, R. Miller, and K. Mai. A feature-based algorithm for detecting and classifying scene breaks. In *ACM International Conference on Multimedia*, pages 189–200, 1995.