Chapter 2

Related Works

2.1 Introduction

Content-based image retrieval, a technique which uses visual contents to search images from large scale image databases according to user's interests, has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development. However, there remain many challenging research problems that continue to attract researchers from multiple disciplines.

Before introducing the fundamental theory of content-based retrieval, we will take a brief look at its development. Early work on image retrieval can be traced back to the late 1970s. In 1979, a conference on Database Techniques for Pictorial Applications [ABD79] was held in Florence. Since then, the application potential of image database management techniques has attracted the attention of researchers [NSC79, NSC80, SKC81, SKC88]. Early techniques were not generally based on visual features but on the textual annotation of images. In other words, images were first annotated with text and then searched using a text-based approach from traditional database management systems. Comprehensive surveys of early text-based image retrieval methods can be found in [SKC92, HTN84]. Text-based image retrieval uses traditional database techniques to manage images. Through text descriptions, images can be organized by topical or semantic hierarchies to facilitate easy navigation and browsing based on standard Boolean queries. However, since automatically generating descriptive texts for a wide spectrum of images is not feasible, most text-based image retrieval systems require manual annotation of images. Obviously, annotating images manually is a cumbersome and expensive task for large image databases, and is often subjective, context-sensitive and incomplete. As a result, it is difficult for the traditional text-based methods to support a variety of task-dependent queries.
CHAPTER 2. RELATED WORKS

In the early 1990s, as a result of advances in the Internet and new digital image sensor technologies, the volume of digital images produced by scientific, educational, medical, industrial, and other applications available to users increased dramatically. The difficulties faced by text-based retrieval became more and more severe. The efficient management of the rapidly expanding visual information became an urgent problem. This need formed the driving force behind the emergence of content-based image retrieval techniques. In 1992, the National Science Foundation of the United States organized a workshop on visual information management systems [RJP92] to identify new directions in image database management systems. It was widely recognized that a more efficient and intuitive way to represent and index visual information would be based on properties that are inherent in the images themselves. Researchers from the communities of computer vision, database management, human-computer interface, and information retrieval were attracted to this field. Since then, research on content-based image retrieval has developed rapidly [AEC93, JDC93, CFE94, YGH94, RJP92, RJA95, HJF95]. Since 1997, the number of research publications on the techniques of visual information extraction, organization, indexing, user query and interaction, and database management has increased enormously. Similarly, a large number of academic and commercial retrieval systems have been developed by universities, government organizations, companies, and hospitals. Comprehensive surveys of these techniques and systems can be found in [BFS95, YRT99, AMW00].

Content-based image retrieval, uses the visual contents of an image such as color, shape, texture, and spatial layout to represent and index the image. In typical content-based image retrieval systems, the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketched figures. The system then changes these examples into its internal representation of feature vectors. The similarities/distances between the feature vectors of the query example or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme. The indexing scheme provides an efficient way to search for the image database. Recent retrieval systems have incorporated users' relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results. In this chapter, we introduce these fundamental techniques for content-based image retrieval.
CHAPTER 2. RELATED WORKS

2.2 Image Content Descriptors

Generally speaking, image content may include both visual and semantic content. Visual content can be very general or domain specific. General visual content include color, texture, shape, spatial relationship, etc. Domain specific visual content, like human faces, is application dependent and may involve domain knowledge. Semantic content is obtained either by textual annotation or by complex inference procedures based on visual content.

A good visual content descriptor should be invariant to the accidental variance introduced by the imaging process (e.g., the variation of the illuminant of the scene). However, there is a tradeoff between the invariance and the discriminative power of visual features, since a very wide class of invariance loses the ability to discriminate between essential differences. Invariant description has been largely investigated in computer vision (like object recognition), but is relatively new in image retrieval [HBS00].

A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of regions or objects to describe the image content. To obtain the local visual descriptors, an image is often divided into parts first. The simplest way of dividing an image is to use a partition, which cuts the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but is a way of representing the global features of the image at a finer resolution. A better method is to divide the image into homogenous regions according to some criterion using region segmentation algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image, is to undertake a complete object segmentation to obtain semantically meaningful objects (like ball, car, horse). Currently, automatic object segmentation for broad domains of general images is unlikely to succeed.

In this section, we will introduce some widely used techniques for extracting color, texture, shape and spatial relationship from images.

2.2.1 Color

Color is the most extensively used visual content for image retrieval [JDF90, JHS99, JHI97, MIA89, AKJ89, EMC98, GPR99, MSM95, MJS91, HJO95]. Its three-dimensional values make its discrimination potentiality superior to the single dimensional gray values of images. Before selecting an appropriate color description, color space must be determined first.
CHAPTER 2. RELATED WORKS

2.2.1.1 Color Space

Each pixel of the image can be represented as a point in a 3D color space. Commonly used color space for image retrieval include RGB, Munsell, CIE L*a*b*, CIE L*u*v*, HSV (or HSL, HSB), and opponent color space. There is no agreement on which is the best. However, one of the desirable characteristics of an appropriate color space for image retrieval is its uniformity [EMC98]. Uniformity means that two color pairs that are equal in similarity distance in a color space are perceived as equal by viewers. In other words, the measured proximity among the colors must be directly related to the psychological similarity among them.

RGB space is a widely used color space for image display. It is composed of three color components red, green, and blue. These components are called "additive primaries" since a color in RGB space is produced by adding them together. In contrast, CMY space is a color space primarily used for printing. The three color components are cyan, magenta, and yellow. These three components are called "subtractive primaries" since a color in CMY space is produced through light absorption. Both RGB and CMY space are device-dependent and perceptually non-uniform.

The CIE L*a*b* and CIE L*u*v* spaces are device independent and considered to be perceptually uniform. They consist of a luminance or lightness component (L) and two chromatic components a and b or u and v. CIE L*a*b* is designed to deal with subtractive colorant mixtures, while CIE L*u*v* is designed to deal with additive colorant mixtures. The transformation of RGB space to CIE L*u*v* or CIE L*a*b* space can be found in [AKJ89].

In HSV (or HSL, or HSB) space is widely used in computer graphics and is a more intuitive way of describing color. The three color components are hue, saturation (lightness) and value (brightness). The hue is invariant to the changes in illumination and camera direction and hence more suited to object retrieval. RGB coordinates can be easily translated to the HSV (or HLS, or HSB) coordinates by a simple formula [JDF90].

The opponent color space uses the opponent color axes (R-G, 2B-R-G, R+G+B). This representation has the advantage of isolating the brightness information on the third axis. With this solution, the first two chromaticity axes, which are invariant to the changes in illumination intensity and shadows, can be down-sampled since humans are more sensitive to brightness than they are to chromatic information. In the following sections, we will introduce some commonly
used color descriptors: the color histogram, color coherence vector, color correlogram, and color moments.

2.2.1.2 Color Moments
Color moments have been successfully used in many retrieval systems (like QBIC [MFH95, WNQ93]), especially when the image contains just the object. The first order (mean), the second (variance) and the third order (skewness) color moments have been proved to be efficient and effective in representing color distributions of images [MSM95]. Mathematically, the first three moments are defined as:

\[
\mu_i = \frac{1}{N} \sum_{j=1}^{N} f_{ij}
\]

(2-1)

\[
\sigma_i = \left( \frac{1}{N} \sum_{j=1}^{N} (f_{ij} - \mu_i)^2 \right)^{\frac{1}{2}}
\]

(2-2)

\[
s_i = \left( \frac{1}{N} \sum_{j=1}^{N} (f_{ij} - \mu_i)^3 \right)^{\frac{1}{3}}
\]

(2-3)

where \(f_{ij}\) is the value of the \(i\)-th color component of the image pixel \(j\), and \(N\) is the number of pixels in the image.

Since only 9 (three moments for each of the three color components) numbers are used to represent the color content of each image, color moments are a very compact representation compared to other color features. Due to this compactness, it may also lower the discrimination power. Usually, color moments can be used as the first pass to narrow down the search space before other sophisticated color features are used for retrieval.

2.2.1.3 Color Histogram
The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterizing the global distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle.

Since any pixel in the image can be described by three components in a certain color space (for instance, red, green, and blue components in RGB space, or hue, saturation, and value in HSV
CHAPTER 2. RELATED WORKS

space), a histogram, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has. However, a histogram with a large number of bins will not only increase the computational cost, but will also be inappropriate for building efficient indexes for image databases.

Furthermore, a very fine bin quantization does not necessarily improve the retrieval performance in many applications. One way to reduce the number of bins is to use the opponent color space which enables the brightness of the histogram to be down sampled. Another way is to use clustering methods to determine the $K$ best colors in a given space for a given set of images. Each of these best colors will be taken as a histogram bin. Since that clustering process takes the color distribution of images over the entire database into consideration, the likelihood of histogram bins in which no or very few pixels fall will be minimized. Another option is to use the bins that have the largest pixel numbers since a small number of histogram bins capture the majority of pixels of an image [YGH94] Such a reduction does not degrade the performance of histogram matching, but may even enhance it since small histogram bins are likely to be noisy.

When an image database contains a large number of images, histogram comparison will saturate the discrimination. To solve this problem, the joint histogram technique is introduced [GPR99]. In addition, color histogram does not take the spatial information of pixels into consideration, thus very different images can have similar color distributions. This problem becomes especially acute for large scale databases. To increase discrimination power, several improvements have been proposed to incorporate spatial information. A simple approach is to divide an image into sub-areas and calculate a histogram for each of those sub-areas. As introduced above, the division can be as simple as a rectangular partition, or as complex as a region or even object segmentation. Increasing the number of sub-areas increases the information about location, but also increases the memory and computational time.

2.2.1.4 Color Coherence Vector

In [GPR96] a different way of incorporating spatial information into the color histogram, color coherence vectors (CCV), was proposed. Each histogram bin is partitioned into two types, i.e., coherent, if it belongs to a large uniformly-colored region, or incoherent, if it does not. Let $\alpha_i$ denote the number of coherent pixels in the $i$th color bin and $\beta$, denote the number of incoherent
CHAPTER 2. RELATED WORKS

pixels in an image. Then, the CCV of the image is defined as the vector \( \langle \alpha_1, \beta_1 \rangle, \langle \alpha_2, \beta_2 \rangle, \ldots, \langle \alpha_N, \beta_N \rangle \). Note that \( \langle \alpha_1 + \beta_1, \alpha_2 + \beta_2, \ldots, \alpha_N + \beta_N \rangle \) is the color histogram of the image.

Due to its additional spatial information, it has been shown that CCV provides better retrieval results than the color histogram, especially for those images which have either mostly uniform color or mostly texture regions. In addition, for both the color histogram and color coherence vector representation, the HSV color space provides better results than CIE \( L^*u^*v^* \) and CIE \( L^*a^*b^* \) space.

2.2.1.5 Color Correlogram

The color correlogram [JHI97] was proposed to characterize not only the color distributions of pixels, but also the spatial correlation of pairs of colors. The first and the second dimension of the three-dimensional histogram are the colors of any pixel pair and the third dimension is their spatial distance. A color correlogram is a table indexed by color pairs, where the \( k \)-th entry for \( (i, j) \) specifies the probability of finding a pixel of color \( j \) at a distance \( k \) from a pixel of color \( i \) in the image. Let \( I \) represent the entire set of image pixels and \( lc(i) \) represent the set of pixels whose colors are \( c(i) \). Then, the color correlogram is defined as:

\[
\gamma_{i,j}^{(k)} = \Pr_{p_1 \in I_{c(i)}, p_2 \in I_{c(j)}} [p_2 \in I_{c(j)} \mid p_1 - p_2 = k]
\]

(2-4)

where \( i, j \in \{1, 2, \ldots, N\} \), \( k \in \{1, 2, \ldots, d\} \), and \( |p_1 - p_2| \) is the distance between pixels \( p_1 \) and \( p_2 \).

If we consider all the possible combinations of color pairs the size of the color correlogram will be very large \( O(N^2d) \), therefore a simplified version of the feature called the color autocorrelogram is often used instead. The color autocorrelogram only captures the spatial correlation between identical colors and thus reduces the dimension to \( O(Nd) \). Compared to the color histogram and CCV, the color auto-correlogram provides the best retrieval results, but is also the most computationally expensive due to its high dimensionality.

2.2.1.6 Invariant Color Features

Color not only reflects the material of surface, but also varies considerably with the change of illumination, the orientation of the surface, and the viewing geometry of the camera. This variability must be taken into account. However, invariance to these environmental factors is not considered in most of the color features introduced above.
CHAPTER 2. RELATED WORKS

Invariant color representation has been introduced to content-based image retrieval recently. In [TGA00], a set of color invariants for object retrieval was derived based on the Schaefer model of object reflection. In [GDF96], specular reflection, shape and illumination invariant representation based on blue ratio vector \((r/b, g/b, 1)\) is given. In [TGA99], a surface geometry invariant color feature is provided.

These invariant color features, when applied to image retrieval, may yield illumination, scene geometry and viewing geometry independent representation of color contents of images, but may also lead to some loss in discrimination power among images.

2.2.2 Texture

Texture refers to the visual patterns that have the properties of homogeneity that do not result from the presence of only a single color or intensity [JRS96]. It is an innate property of virtually all surfaces, including clouds, trees, bricks, hair, fabric, etc. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment [RMH73]. Because of its importance and usefulness in Pattern Recognition and Computer Vision, there existed rich research results in the past three decades.

In the early 70's, Haralick et al. proposed the co-occurrence matrix representation of texture feature [RMH73]. This approach explored the gray level spatial dependence of texture. It first constructed a co-occurrence matrix based on the orientation and distance between image pixels and then extracted meaningful statistics from the matrix as the texture representation. Many other researchers followed the same line and further proposed enhanced versions. For example, Gotlieb and Kreyszig studied the statistics originally proposed in [RMH73] and experimentally found that contrast, inverse deference moment and entropy had the biggest discriminatory power [CCG90].

Motivated by the psychological studies in human visual perception of texture, Tamura et al. explored the texture representation from a different angle [HTM78]. They developed computational approximations to the visual texture properties found to be important in psychology studies. The six visual texture properties were coarseness, contrast, directionality, linelikeness, regularity, and roughness. One major distinction between the Tamura texture representation and the co-occurrence matrix representation is that all the texture properties in Tamura representation are visually meaningful whereas some of the texture properties used in co-
occurrence matrix representation may not (for example, entropy). This characteristic makes the Tamura texture representation very attractive in Image Retrieval.

In early 90's after Wavelet transform was introduced and its theoretical framework established, many researchers began to study the use of Wavelet transform in texture representation [JRS94, TCC93, ALJ93, MHG94, AKJ92, KST94]. In [JRS94, JRS96], Smith and Chang used the statistics (mean and variance) extracted from the Wavelet sub-bands as the texture representation. To explore the middle-band characteristics, tree structured Wavelet transform was used by Chang and Kuo in [TCC93] to further improve the classification accuracy. Wavelet transform was also combined with other techniques to achieve better performance. Gross et al. used Wavelet transform, together with Kohonen maps, to perform texture analysis in [MHG94]. Thyagarajan et al. [KST94] and Kundu et al. [AKJ92] combined Wavelet transform with co-occurrence matrix to take advantage of both the statistics based and transform based texture analysis.

2.2.3 Shape
Shape can be defined as a “Spatial arrangement of something distinguished from its surroundings by its outline”. Shape features of objects or regions have been used in many content-based image retrieval systems [JEG92, WIG90, HVJ91, DTS94]. Compared with color and texture features, shape features are usually described after images have been segmented into regions or objects. Since robust and accurate image segmentation is difficult to achieve, the use of shape features for image retrieval has been limited to special applications where objects or regions are readily available. The state-of-art methods for shape description can be categorized into either boundary-based (rectilinear shapes [HVJ91], polygonal approximation [EMA91], finite element models [SSA95], and Fourier-based shape descriptors [KAWS90, HKT95, EPK77]) or region-based methods (statistical moments [MKH77, LYF94]). A good shape representation feature for an object should be invariant to translation, rotation and scaling. In this section, we briefly describe some of these shape features that have been commonly used in image retrieval applications.

The main idea of Fourier Descriptor is to use the Fourier transformed boundary as the shape feature. Some early work can be found in [CTZ72, EPK77]. To take into account the digitization noise in the image domain, Rui et al. proposed a modified Fourier Descriptor which is both robust to noise and invariant to geometric transformations [YRA96].
CHAPTER 2. RELATED WORKS

The main idea of Moment Invariants is to use region based moments, which are invariant to transformations, as the shape feature. In [MKH62], Hu identified seven such moments. Based on his work, many improved versions emerged. In [LYF94], based on the discrete versions of Green's theorem, Yang and Albregtsen proposed a fast method of computing moments in binary images. Motivated by the fact that most useful invariants were found by extensive experience and trial-and-error method, Kapur et al. developed algorithms to systematically generate and search for a given geometry's invariants [DKY95]. Gross and Latecki developed an approach which preserved the qualitative differential geometry of the object boundary, even after an image was digitized [DKY95]. In [DCZ95, ZLD00], a framework of algebraic curves and invariants is proposed to represent complex objects in cluttered scene by parts or patches. Polynomial fitting is done to represent local geometric information, from which geometric invariants are used in object matching and recognition.

Some recent work in shape representation and matching include Finite Element Method (FEM) [APW96], Turning Functions [EML91], and Wavelet Descriptor [CGH96]. FEM defines a stiffness matrix, which describes how each point on the object is connected to other points. The eigen-vectors of the stiffness matrix are called modes and span a feature space. All the shapes are first mapped into this space and similarity is then computed based on the eigen-values. Along a similar line of Fourier Descriptor, Arkin et al. developed a Turning Function based method for computing both convex and concave polygons [EML91]. In [CGH96], Chuang and Kuo used Wavelet transform to describe object shape. It embraced the desirable properties such as multi-resolution representation, invariance, uniqueness, stability, and spatial localization. Barrow et al. first proposed the Chamfer matching technique, which compared two collections of shape fragments at a cost proportional to linear dimension, rather than area [HGB77].

In [BLS95], Li and Ma showed that the Geometric Moments method (region-based) and the Fourier Descriptor (boundary-based) were related by a simple linear transformation. In [BMM97], Babu et al. compared the performance of boundary based representations (Chain code, Fourier Descriptor, UNL Fourier Descriptor), region based representations (Moment Invariants, Zernike moments, Pseudo-Zernike moments), and combined representations (Moment Invariants and Fourier Descriptor, Moment Invariants and UNL Fourier Descriptor). Their experiments showed that the combined representations outperformed the simple representations.
CHAPTER 2. RELATED WORKS

2.2.4 Spatial Information

Regions or objects with similar color and texture properties can be easily distinguished by imposing spatial constraints. For instance, regions of blue sky and ocean may have similar color histograms, but their spatial locations in images are different. Therefore, the spatial location of regions (or objects) or the spatial relationship between multiple regions (or objects) in an image is very useful for searching images.

The most widely used representation of spatial relationship is the 2D strings proposed by Chang et al [SKC87]. It is constructed by projecting images along the $x$ and $y$ directions. Two sets of symbols, $V$ and $A$, are defined on the projection. Each symbol in $V$ represents an object in the image. Each symbol in $A$ represents a type of spatial relationship between objects. As its variant, the 2D G-string [SKE88] cuts all the objects along their minimum bounding box and extends the spatial relationships into two sets of spatial operators. One defines local spatial relationships. The other defines the global spatial relationships, indicating that the projection of two objects are disjoint, adjoin or located at the same position. In addition, 2D C-string [SYL90] is proposed to minimize the number of cutting objects. 2D-B string [SYL92] represents an object by two symbols, standing for the beginning and ending boundary of the object. All these methods can facilitate three types of query. Type 0 query finds all images containing object $O_1, O_2, ..., O_n$. Type 1 finds all images containing objects that have certain relationship between each other, but the distance between them is insignificant. Type 2 finds all images that have certain distance relationship with each other.

In addition to the 2D string, spatial quad-tree [HST84], and symbolic image [VNG95] are also used for spatial information representation. However, searching images based on spatial relationships of regions remains a difficult research problem in content-based image retrieval, because reliable segmentation of objects or regions is often not feasible except in very limited applications. Although some systems simply divide the images into regular sub-blocks [MSM96], only limited success has been achieved with such spatial division schemes since most natural images are not spatially constrained to regular sub-blocks. To solve this problem, a method based on the radon transform, which exploits the spatial distribution of visual features without a sophisticated segmentation is proposed in [FGJ98, HWF98].
CHAPTER 2. RELATED WORKS

2.2.5 Segmentation
Segmentation is very important to Image Retrieval. Both the shape feature and the layout feature depend on good segmentation. Segmentation is the low-level operation for partitioning images by finding disjoint and homogeneous regions or, equivalently, by finding edges or boundaries. The homogeneous regions, or the edges, are supposed to correspond to actual objects, or parts of them, within the images. Thus, in a large number of applications in image processing and computer vision, segmentation plays a fundamental role as the first step before applying to images higher-level operations such as recognition, semantic interpretation, and representation. A detailed survey of segmentation methods has been reported in Chapter 4.

2.3 Similarity Measures and Indexing
2.3.1 Similarity/Distance Measures
Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image. Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different similarity/distance measures will affect retrieval performances of an image retrieval system significantly. In this section, we will introduce some commonly used similarity measures. We denote \( D(I,J) \) as the distance measure between the query image \( I \) and the image \( J \) in the database; and \( f_i(I) \) as the number of pixels in bin \( i \) of \( I \).

Minkowski-Form Distance
If each dimension of image feature vector is independent of each other and is of equal importance, the Minkowski-form distance \( L_p \) is appropriate for calculating the distance between two images. This distance is defined as:

\[
D(I,J) = \left( \sum_i \left| f_i(I) - f_i(J) \right|^p \right)^{1/p}
\]

(2-5)

when \( p=1, 2, \) and \( \infty \), \( D(I,J) \) is the \( L_1 \), \( L_2 \) (also called Euclidean distance), and \( L_\infty \) distance respectively. Minkowski-form distance is the most widely used metric for image retrieval. For instance, MARS system [YRT97] used Euclidean distance to compute the similarity between texture features; Netra [WYB97, WYM99] used Euclidean distance for color and shape feature, and \( L_1 \) distance for texture feature; Blobworld [CCM99] used Euclidean distance for texture and shape feature. In addition, Voorhees and Poggio [HVT88] used \( L_\infty \) distance to compute the similarity between texture images. The Histogram intersection can be taken as a special case of
CHAPTER 2. RELATED WORKS

$L_i$ distance, which is used by Swain and Ballard [MJS91] to compute the similarity between color images. The intersection of the two histograms of $I$ and $J$ is defined as:

$$S(I,J) = \frac{\sum_{i=1}^{N} \min(f_i(I), f_i(J))}{\sum_{i=1}^{N} f_i(J)}$$  (2-6)

It has been shown that histogram intersection is fairly insensitive to changes in image resolution, histogram size, occlusion, depth, and viewing point.

Quadratic Form (QF) Distance

The Minkowski distance treats all bins of the feature histogram entirely independently and does not account for the fact that certain pairs of bins correspond to features which are perceptually more similar than other pairs. To solve this problem, quadratic form distance is introduced:

$$D(I,J) = \sqrt{(F_i - F_j)^t A (F_i - F_j)}$$  (2-7)

where $A=[a_{ij}]$ is a similarity matrix, and $a_{ij}$ denotes the similarity between bin $i$ and $j$. $F_i$ and $F_j$ are vectors that list all the entries in $f_i(I)$ and $f_i(J)$.

Quadratic form distance has been used in many retrieval systems [JHE95, WNQ93] for color histogram-based image retrieval. It has been shown that quadratic form distance can lead to perceptually more desirable results than Euclidean distance and histogram intersection method as it considers the cross similarity between colors.

Mahalanobis Distance

The Mahalanobis distance metric is appropriate when each dimension of image feature vector is dependent of each other and is of different importance. It is defined as:

$$D(I,J) = \sqrt{(F_i - F_j)^t C^{-1} (F_i - F_j)}$$  (2-8)

where $C$ is the covariance matrix of the feature vectors.

The Mahalanobis distance can be simplified if feature dimensions are independent. In this case, only a variance of each feature component, $c_i$, is needed.

$$D(I,J) = \sum_{i=1}^{N} \frac{(F_i - F_j)^2}{c_i}$$  (2-9)
CHAPTER 2. RELATED WORKS

Cosine measure

The Cosine measure is a very important similarity measure for points in the plane (and in higher dimensions). The cosine measure assigns a high similarity to points that are in the same direction from the origin, zero similarity to points that are perpendicular to one another, and negative similarity for those that are pointing in opposing directions to one another.

\[
D(I,J) = \frac{\sum (IJ)}{(\sum I^2)(\sum J^2)}
\]  

(2-10)

2.3.2 Indexing Scheme

Another important issue in content-based image retrieval is effective indexing and fast searching of images based on visual features. Because the feature vectors of images tend to have high dimensionality and therefore are not well suited to traditional indexing structures, dimension reduction is usually used before setting up an efficient indexing scheme.

One of the techniques commonly used for dimension reduction is principal component analysis (PCA). It is an optimal technique that linearly maps input data to a coordinate space such that the axes are aligned to reflect the maximum variations in the data. The QBIC system uses PCA to reduce a 20-dimensional shape feature vector to two or three dimensions [MFH95] [WNQ93]. In addition to PCA, many researchers have used Karhunen-Loeve (KL) transform to reduce the dimensions of the feature space. Although the KL transform has some useful properties such as the ability to locate the most important sub-space, the feature properties that are important for identifying the pattern similarity may be destroyed during blind dimensionality reduction [WJK95]. Apart from PCA and KL transformation, neural network has also been demonstrated to be a useful tool for dimension reduction of features [JAC00].

After dimension reduction, the multi-dimensional data are indexed. A number of approaches have been proposed for this purpose, including R-tree (particularly, R*-tree [NBT90]), linear quad-trees [JVM99], K-d-B tree [JTR81] and grid files [JNH84]. Most of these multi-dimensional indexing methods have reasonable performance for a small number of dimensions (up to 20), but explore exponentially with the increasing of the dimensionality and eventually reduce to sequential searching. Furthermore, these indexing schemes assume that the underlying feature comparison is based on the Euclidean distance, which is not necessarily true for many image retrieval applications. One attempt to solve the indexing problems is to use hierarchical indexing scheme based on the Self-Organization Map (SOM) proposed in [HJF95].
CHAPTER 2. RELATED WORKS

2.4 User Interaction
For content-based image retrieval, user interaction with the retrieval system is crucial since flexible formation and modification of queries can only be obtained by involving the user in the retrieval procedure. User interfaces in image retrieval systems typically consist of a query formulation part and a result presentation part.

2.4.1 Query Specification
Specifying what kind of images a user wishes to retrieve from the database can be done in many ways. Commonly used query formations are: category browsing, query by concept, query by sketch, and query by example. Category browsing is to browse through the database according to the category of the image. For this purpose, images in the database are classified into different categories according to their semantic or visual content [AVM01]. Query by concept is to retrieve images according to the conceptual description associated with each image in the database. Query by sketch [GDF96] and query by example [JAA00] is to draw a sketch or provide an example image from which images with similar visual features will be extracted from the database.

Query by sketch allows user to draw a sketch of an image with a graphic editing tool provided either by the retrieval system or by some other software. Queries may be formed by drawing several objects with certain properties like color, texture, shape, sizes and locations. In most cases, a coarse sketch is sufficient, as the query can be refined based on retrieval results.

Query by example allows the user to formulate a query by providing an example image. The system converts the example image into an internal representation of features. Images stored in the database with similar features are then searched. Query by example can be further classified into query by external image example, if the query image is not in the database, and query by internal image example, if otherwise. For query by internal image, all relationships between images can be pre-computed. The main advantage of query by example is that the user is not required to provide an explicit description of the target, which is instead computed by the system. It is suitable for applications where the target is an image of the same object or set of objects under different viewing conditions. Most of the current systems provide this form of querying.

Query by group example allows user to select multiple images. The system will then find the images that best match the common characteristics of the group of examples. In this way, a target can be defined more precisely by specifying the relevant feature variations and removing
CHAPTER 2. RELATED WORKS

irrelevant variations in the query. In addition, group properties can be refined by adding negative examples. Many recently developed systems provide both query by positive and negative examples.

2.4.2 Relevance Feedback

Human perception of image similarity is subjective, semantic, and task-dependent. Although content-based methods provide promising directions for image retrieval, generally, the retrieval results based on the similarities of pure visual features are not necessarily perceptually and semantically meaningful. In addition, each type of visual feature tends to capture only one aspect of image property and it is usually hard for a user to specify clearly how different aspects are combined. To address these problems, interactive relevance feedback, a technique in traditional text-based information retrieval systems, was introduced. With relevance feedback [YRT98] [TPM96] [YRA97] [JHS97], it is possible to establish the link between high-level concepts and low-level features.

Relevance feedback is a supervised active learning technique used to improve the effectiveness of information systems. The main idea is to use positive and negative examples from the user to improve system performance. For a given query, the system first retrieves a list of ranked images according to a predefined similarity metrics. Then, the user marks the retrieved images as relevant (positive examples) to the query or not relevant (negative examples). The system will refine the retrieval results based on the feedback and present a new list of images to the user. Hence, the key issue in relevance feedback is how to incorporate positive and negative examples to refine the query and/or to adjust the similarity measure.

2.5 Performance Evaluation

To evaluate the performance of retrieval system, two measurements, namely, recall and precision [AMW00], are borrowed from traditional information retrieval. For a query, let the retrieved number of images be $R$. $R$ will contain relevant set of images $r$ and irrelevant set of images $i$. i.e $R = r + i$. Database contains total $N$ number of images and let $n$ number of relevant images as for the query.

Precision is defined as a measure of the usefulness of the retrieved list of images; i.e. it is the percentage of retrieved images that are relevant.
CHAPTER 2 RELATED WORKS

\[
\text{Precision} = \frac{\# \text{of relevant images retrieved}}{\# \text{of images retrieved}} = \frac{r}{R}
\]

(2-11)

Recall is defined as a measure of the completeness of the retrieved list of images, i.e. it is the percentage of all relevant images that are found by the search operation.

\[
\text{Recall} = \frac{\# \text{of relevant images retrieved}}{\# \text{of relevant images in Collection}} = \frac{r}{n}
\]

(2-12)

To quantify the performance of retrieval systems normally average precision recall (APR) curves are used. APR curves are plotted as

\[
\frac{1}{N} \sum_{i=1}^{N} (x_i - x_{i+1})(y_{i+1} + y_i)
\]

(2-13)

where \((x_i, y_i)\) is the (recall, precision) pair when the number of retrieved images is \(i\) and \(N\) is the total number of top matches defines the performance area.

2.6 Some Content Based Image Retrieval Systems

Below we describe some Content Based Image Retrieval Systems which are in use

2.6.1 Amore (Advanced Multimedia Oriented Retrieval Engine)

Developer C & C Research Laboratories NEC USA, Inc.

URL http://www.ccrl.com/amore/.

References [SMK97], [SMK99].

Features The image is segmented into at most eight regions of homogeneous color, and downsized to 24x24 pixels. The regions in this picture are directly used for matching [KHY93].

Querying The user first selects a category of images. An initial set of images can be selected at random or by keyword. Of these images, visually similar images can be retrieved. The query image can also be specified by its URL. In a research version of the system, sketching a query image was possible. The user can indicate the relative importance of color and shape.

Matching First a correspondence between regions in the query and target image is found. Regions corresponding to the same regions in the other image are merged. The shape similarity between two regions is based on the number of pixels of overlap, a kind of template matching. The color similarity between two regions is the distance in HLS space between the uniform region colors [KHY93].

Indexing Indexing is performed by the COIR (Content-Oriented Image Retrieval) system, refer [KHS97].

21
CHAPTER 2. RELATED WORKS

Result presentation The retrieved images are shown as thumbnails, without an explicit order, see figure 2-1. In a research version of the system, result images were displayed as a scatter plot, with shape and color similarity values at the axes, or on a perspective wall [SMK97].

Applications Amore is used in the art retrieval system Arthur (Art Media and Text Hub and Retrieval System, http://www.isi.edu/cct/arthur/, developed at the Center for Cultural Technology within the Information Sciences Institute of the University of Southern California.

![Figure 2.1: Amore result of similarity retrieval on shape.](image)

2.6.2 BDLP (Berkeley Digital Library Project)

Developer University of California, Berkeley.

URL The homepage of project is at http://elib.cs.berkeley.edu/, a demo of retrieval from all photos in the collection is at http://elib.cs.berkeley.edu/photos/all.shtml.

References [CCV96].

Features There is a number of alphanumerical keys available for querying: the collection, key words, location, county, and photographer. The colors of each image are quantized into 13 colors bins. Six values are associated with each color bin: the percentage of the image with colors in that bin, and the number of ‘very small’, ‘small’, ‘medium’, ‘large’, and ‘very large’ dots of that color found.

Querying For content-based search, the user can select 13 colors, and indicate the amount (‘any’, ‘partly’, ‘mostly’) of that color in the picture. Also, for colored regions the user can indicate the size (‘any’, ‘small’, ‘medium’, ‘large’) and the quantity of regions with that color (‘any’, ‘few’, ‘some’, ‘many’).

Matching Image features are stored as text strings. For example, a picture of a sky with clouds might have a few large white regions, and a large amount of blue, and would have a feature text string “mostly blue large white few”. Matching is done by substring matching, e.g. with query string “large white%”.

22
CHAPTER 2. RELATED WORKS

Indexing All features are put into a relational database (the Informix Universal Server database management system).

Result presentation The retrieved photos are presented unordered, with id-number, photographer, and collection keys.

Applications The collections consist of 23195 images of plants, animals, people, and landscapes, 17000 images from the California Department of Water Resources, Corel Stock photos, and aerial photos of the Sacramento river delta region.

2.6.3 Blobworld

Developer Computer Science Division, University of California, Berkeley.


References [CCM00].

Features The features used for querying are the color, texture, location, and shape of regions (blobs) and of the background. The color is described by a histogram of 218 bins of the color coordinates in Lab-space. Texture is represented by mean contrast and anisotropy over the region, as the 2D coordinate \((\text{contrast; contrast } \times \text{ anisotropy})\). Shape is represented by (approximate) area, eccentricity, and orientation.

Querying The user first selects a category, which already limits the search space. In an initial image, the user selects a region (blob), and indicates the importance of the blob (‘somewhat’, ‘very’). Next, the user indicates the importance of the blob’s color, texture, location, and shape (‘not’, ‘somewhat’, ‘very’). More than one regions can be used for querying.

Matching To match two color histograms \(h_1\) and \(h_2\), the quadratic form distance is used: 
\[
d(h_1, h_2) = (h_1 - h_2)^T A (h_1 - h_2),
\] 
where \(A = (a_{ij})\) is a symmetric matrix of weights representing the similarity between color bins \(i\) and \(j\). The distance between two texture descriptors is the Euclidean distance between their coordinates in representation space. The distance between centroids is the Euclidean distance. The distances are combined into a single final distance.

Indexing Rather than actually computing the distances between the full color histogram vectors of length 218 as 
\[
d(h_1, h_2) = (h_1 - h_2)^T A (h_1 - h_2),
\] 
singular value decomposition (SVD) is used to project the histogram vectors onto a lower-dimensional subspace. The resulting points are indexed by an R’-tree [NBH90].

Result presentation The retrieved images are ranked in linear order, and presented together with the segmented version showing the regions, see Figure 2.2.
CHAPTER 2. RELATED WORKS

Applications The demo on the web provides retrieval from a collection of 10000 Corel stock photos.

![Figure 2.2: Blobworld query result.](image)

2.6.4 C-bird (Content-Based Image Retrieval from Digital libraries)

Developer School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada.


References [ZNL98], [ZNL99].

Features For each collected image, a feature descriptor and a layout descriptor are computed. A feature descriptor is a set of four vectors: a color vector, a most frequent color (MFC) vector, a most frequent orientation (MFO) vector, and a chromaticity vector. A 512-bin RGB histogram is stored in the color vector. The centroids of the regions associated with the 5 most frequent colors form the MFC vector and the centroids of regions of the 5 most frequent edge orientations form the MFO vector. The 36-dimensional chromaticity vector is computed as follows: first, a normalization of each RGB channel is made to obtain illumination invariance, then the 3D color histogram is replaced by a 2D chromaticity histogram. Treating this chromaticity histogram as an image, first a wavelet-based image reduction is applied, then the Discrete Cosine Transform coefficient matrix is built. The chromaticity vector is made of the 36 values of the upper left corner of the DCT matrix. For search by object model, some geometric data such as the area, the centroid and the eccentricity are computed from color regions associated with each of the MFCs. The layout descriptor contains a color layout vector and an edge layout vector. To construct these vectors the image is divided into 64 cells, and for each cell the most frequent colors and the number of edges for each orientation are determined. Also, for images at half and quarter resolution, a feature descriptor like the one described above is stored.

Querying The user is presented a grid of consecutive images from the database starting at a random position. To start a query by color histogram or color similarity with illumination
CHAPTER 2. RELATED WORKS

invariance, one of the buttons under the selected query image is pressed (see Figure 2.3). For a query by color or texture layout, grids are presented for drawing color, texture density and edge orientation layout (see Figure 2.3). For a query by color percentage, 5 colors and their percentages are indicated by the user. For a query by object model, the user browses through a selection of query images and makes a choice.

**Matching** The distance between two chromaticity vectors in an illumination invariant color query is the L2 distance. Texture orientation histograms, as well as color histograms for the full image, are matched by histogram intersection.

The first step in a query by object model is a color localization: color regions for each MFC are extracted and for each region, some geometric data such as the area, the centroid and the eccentricity are computed. After selecting the images in the database that share a number of color regions with the query image, a number of vectors are produced by connecting the centroid of the first MFC region with the centroids of the other MFCs. Analyzing the length of these vectors and the angles between them, a hypothesis regarding the existence of an object at a certain scale and orientation (the difference of angles between centroids of the regions corresponding to the MFCs in the query and database image) is made. This hypothesis is tested in a second step by comparing the texture histogram for each pair of matching regions in the two images. The 2D texture histogram measures orientation (the gradient direction of the edge pixels) and edge separation from the grey level image. Finally, if there is sufficient similarity in their texture between the query object and the area in the database image where the supposed similar object was identified, a shape verification based on the Generalized Hough Transform is performed [DHB81].

**Result presentation** The user can choose the number of rows and columns of the displayed images grid. By clicking on a thumbnail image the user can see some color and texture characteristics of the image (color percentage and layout, texture layout).

![Figure 2.3: C-bird query interface and query result.](image)
CHAPTER 2. RELATED WORKS

2.6.5 Chabot

**Developer** Department of Computer Science, University of California, Berkeley, CA, USA.

**URL** http://http.cs.berkeley.edu/~ginger/chabot.html. Chabot has evolved into Cypress (which, surprisingly, seems not to have inherited content based query capability). For a demo of Cypress, see http://elib.cs.berkeley.edu/photos/.

**References** [VEO95].

**Features** One of the early systems, Chabot aimed at combining text based descriptions with image analysis in retrieving images from a collection of photographs of the California Department of Water Resources. The system made use of an existing text description database of the collection, adding other types of textual information for querying such as the shooting date, the picture location, the perspective of the photo. For each image a color histogram containing only 20 bins is computed.

**Querying** The user is presented with a list of search criteria (such as keywords, photographer, film format, shooting date, perspective, location, and colors). The color criterion offers limited options for the user to choose from, such as ‘mostly red’ or ‘some yellow’. The user has the possibility to define concepts, which are combinations of search criteria that the concept satisfies. For example, the concept of ‘sunset’ is defined as a combination of keyword (‘sunset’) and color (‘mostly red’ or ‘mostly orange’) criteria.

**Matching** To match a ‘mostly ...’ color criterion, more than 50% of the pixels in an image must have the requested color. For the ‘some ...’ color criterion, one or two of the 20 colors in the histogram must be qualified as the requested color.

**Indexing** The images and associated data are stored in the database POSTGRES, developed at the University of California, Berkely.

**Result presentation** Images are shown without specific order.

**Applications** The database contains 11643 images of California natural resources.

2.6.6 Circus (Content-based Image Retrieval and Consultation User-centered System)


**References** [ZPF98], [ZPI97]

**Features** The color feature is a global histogram in Lab space. The texture features are the mean, standard deviation, angular second moment, inverse difference moment, sum average, contrast, correlation, and sum variance, all derived from the co-occurrence matrix of gray values in the direction of $\pi/4$. These feature vectors are projected onto a lower dimensional space, derived from the ‘feature by
CHAPTER 2. RELATED WORKS

image' occurrence matrix $A = (a_{ij})$, with $a_{ij} = l_i g_j$, where $l_i$ is the local weighting of feature $i$ in image $j$, and $g_j$ is the global weighting of feature $i$. This is analogous to the 'term by document' matrix used in Latent Semantic Indexing for text retrieval. Using singular value decomposition (SVD), an orthogonal base for features and images is computed, in which $AA$ is expressed as a linear combination. Matrix $A$ is then approximated by the first few terms of this sum. To speed up the computations, a fast approximation to the decomposition is computed using wavelet packets.

**Querying** From a query image, features are extracted and projected onto the lower-dimensional feature space. Alternatively, color queries can be specified by giving the percentages of each color in the desired image.

**Matching** The distance between two projected feature vectors is the cosine of the angle between the two vectors.

**Relevance feedback** Users can specify for each query a set of positive and negative examples. The new query consist of the intersection of features from the positive examples, minus the features from the negative examples.

**Result presentation** The images are shown in linear decreasing order of scoring value.

### 2.6.7 Compass (Computer Aided Search System)

**Developer** Centre for Scientific and Technological Research, Trento, Italy.

**URL** [http://compass.itc.it/](http://compass.itc.it/)

**References** [RBOO00]

**Features** The color features used are the hue and luminance histograms, and the 2D hue and luminance cooccurrence matrices. The texture feature is the histogram of the magnitude of the luminance gradient.

**Querying** The system works with query by example. However, rather than the usual single query image, the query consists of a set of images, see figure 2-4, which is then sent to possibly more than one image database server. An alternative is to browse through the database. For this purpose the database is clustered. Key images of all the clusters are projected using multidimensional scaling onto a line.

**Matching** The distance between two feature histograms is the $L_1$ distance. The distance between two images is a weighted sum of the individual feature distances. The cloud of query feature vectors in feature space is clustered into a number of query sets $Q_i$. The distance $d(I, Q)$ between a database image $I$ and a query set of images $Q_i$ is the minimum over all images from the query set.
CHAPTER 2. RELATED WORKS

Relevance feedback The user can indicate which result images are relevant, and which are not. The relevant images are added to the query set. When a set of irrelevant images \( Q' \) has been specified, the distance \( d(I, Q, Q') \) between a database image \( I \) and a query set of images \( Q \), given a set of irrelevant images \( Q' \) is given by \( d(I, Q) / (d(I, Q) / d(I, Q')) ^ 2 \).

Result presentation The answers from the multiple servers are then merged and proposed to the user as a single result, shown in decreasing order of similarity score.

Figure 2.4: Compass interface.

2.6.8 Excalibur Visual RetrievalWare

The Visual RetrievalWare is a software developers kit for building applications for manipulating digital image files and their visual content. The toolkit contains C++ and Java API’s for image processing, feature extraction, indexing and content-based retrieval. It also includes sample programs which might be used directly or can serve as templates for building more complex applications. One of these sample programs is the CST (Color, Shape, and Texture) demo.

Developer Excalibur Technologies.


References [EVR00].

Features The CST demo allows queries by example based on HSV color histograms, relative orientation, curvature and contrast of lines in the image, and texture attributes, that measure the flow and roughness in the image.

Querying The user first defines the desired visual similarity by specifying the relative importance of the above image attributes, and then selects one of the displayed images as query, see figure 2-5.

Result presentation The images are shown without an explicit ordering.
CHAPTER 2. RELATED WORKS

Applications The software has been used in the Image Surfer system, which was used by the Yahoo! and Infoseek WWW search engines.

Figure 2.5: Excalibur: result of querying with upper left image based on shape alone.

2.6.9 FOCUS (Fast Object Color-based Query System)

Developer Department of Computer Science, University of Massachusetts, Amherst, MA.


References [MDE97].

Querying The user can select as query one of the displayed template images, or create a new template by marking a sub-image which contains the region of interest.

Features Each image is divided in cells of 100×100 pixels and for each cell a color histogram in the HSV space, coarsely quantized along the saturation and value axes (64×10×10), is computed. The peaks of all local histograms are determined and combined in a list of unique peaks for the whole image by merging multiple copies of the same peak. Also, a frequency table is constructed which, for each color in the HSV space, gives the number of images that have a peak of that color.

The spatial relationships between colored regions are represented by means of a spatial proximity graph (SPG) constructed in two phases. First an intermediate SPG is generated, with one node corresponding to each color peak computed for the image cells. Two nodes in this graph are connected if their corresponding peaks are located in the same cell or are located in neighboring cells and have the same color. This graph is then simplified, by unifying all connected nodes of the same color in a single node, and stored using an adjacency matrix representation.
CHAPTER 2. RELATED WORKS

For the query image, a global color histogram is computed and color region relationships are determined at pixel level.

**Matching** The peaks of a query image are subjected to approximate range queries in the increasing order of their corresponding entries in the frequency table. From the resulting lists, the set of images which have peaks matching all query peaks are determined. For each image in this set, a matching score is computed as the sum of the $L_1$ distances between each query peak and the matched candidate peak.

To match the SPG of the query image with that of a candidate image, first the candidate SPG is reduced by removing any node whose corresponding peak does not match a query peak. Then it is checked if the query graph appears as a sub-graph in the candidate SPG.

**Indexing** The color peaks of the database images are stored in a B-tree [HFK91] sorted with hue as the primary key, followed by saturation and value.

**Result presentation** When the user submits a query by clicking on an image, the images are retrieved using the first phase of matching (the images displayed are the images that have all the colors of the query image). By clicking on the ‘Refine Results’ button, the retrieved images are subjected to the second phase of matching, where the spatial relationships of the matched color regions is analyzed in order to detect a query object in a candidate image.

**Applications** The database consists of 400 advertisements and 800 color natural images.

2.6.10 ImageScape

**Developer** Department of Computer Science, Leiden University, The Netherlands.


**References** [MLK97], [JMB00].

**Querying** Using the sketch interface, the user can draw an outline of the desired image. For semantic querying, the user brings icons on a canvas that represent the objects/concepts he is looking for, at the desired position in the image. Examples of object/concept categories include human faces, stone or sand, water, sky, tree or grass, points and lines (see Figure 2.6).

**Features** Edge maps of the images collected by Web crawlers are obtained using the Sobel operator and a Gaussian blurring filter. A frequency histogram of the $3 \times 3$ binary pixel patterns occurring in the edge image, which is called trigram vector, is computed for all images. This vector is subjected to a dimensionality reduction using a band-pass filter. Various other features, used in object matching, are taken at pixel level: color, Laplacian, gradient magnitude, local binary patterns, invariant moments and Fourier descriptors.
Matching  The first step of the object matching process uses the $L_1$ distance on the trigram vectors to retrieve the top 1% matches from the entire database. Among these, 20 matches are selected in a second step, a 20x20 template matching, using the most informative pixels to minimize the misdetection rate. These pixels are found as follows. For each object, a large set of positive and negative examples are used in finding the set of 256 pixels with the greatest discriminatory power, by maximizing the Kullback relative information combined with a Markov random field.

![ImageScape query built with icons.](image)

**Figure 2.6: ImageScape query built with icons.**

### 2.6.11 MARS (Multimedia Analysis and Retrieval System)

**Developer** Department of Computer Science, University of Illinois at Urbana-Champaign, further developed at Department of Information and Computer Science, University of California at Irvine, CA.


**References** [MOY97].

**Features** The system supports queries on combinations of low-level features (color, texture, shape) and textual descriptions. Color is represented using a 2D histogram over the HS coordinates of the HSV space. Texture is represented by two histograms, one measuring the coarseness and the other one the directionality of the image, and one scalar defining the contrast. In order to extract the color/texture layout, the image is divided into $5 \times 5$ sub-images. For each sub-image a color histogram is computed. For the texture of a sub-image, a vector based on wavelet coefficients is used. The object in an image is segmented out in two phases. First, a $k$-means clustering method in the color-texture space is applied, than the regions detected are grouped by an attraction based method. This consists of choosing a number of attractor regions and associating each region with the attractor that has the largest attraction to it. The attraction
between two regions, \( i \) and \( j \), is defined as 
\[
F_{ij} = \frac{M_i M_j}{d_{ij}^2},
\]
where \( M_i, M_j \) are the sizes of the two regions and \( d_{ij} \) is the Euclidean distance between the two regions in the spatial-color-texture space. In the MARS system, five attractors are used: one for each corner of the image (background attractors) and one in the center of the image (the objects attractor). This is consistent with the fact that their database consists of images of single objects. The shape of the boundary of the extracted object is represented by means of Fourier Descriptors (FD).

**Querying** Complex queries can be formulated using boolean operators. The desired features can be specified either by example (pointing an image database that has such a property) or direct (for example, by choosing colors from a palette or textures from an available set of patterns).

**Matching** The similarity distance between two color histograms is computed by histogram intersection. The similarity between two textures of the whole image is determined by a weighted sum of the Euclidean distance between contrasts and the histogram intersection distances of the other two components, after a normalization of the three similarities. For computing the texture similarity between two corresponding sub-images, the Euclidean distance between the vector representations is used. A weighted sum of the \( 5 \times 5 \) color/texture similarities is used to compute the color/texture layout distance between two images. The similarity measure between two FD shape representations is a weighted sum of the standard deviations of 
\[
\text{ratio}(k) = M_i(k)/M_j(k) \quad \text{and} \quad \text{shift}(k) = \theta_i(k) - \theta_j(k) - \Psi, \quad k = -N_c, \ldots, N_c,
\]
where \( M_i(k) \) and \( \theta_i(k) \) are the magnitude and the phase angle of the FD coefficients, \( \Psi \) is the difference of the major axis orientations of the two shapes and \( N_c \) is the number of FD coefficients.

Each query has a query tree associated. In a query tree, the leaves represent the feature vectors (the terms of the boolean expression defining the query) while the internal nodes correspond to boolean operators or more complex terms indicating a query by object. Individual queries on each of the query terms are made. The tree is evaluated bottom-up: each internal node receives from each child a list of ranked images and combines these lists, after a normalization process, according to the weights on the parent-child links.

**Indexing** There is no information about indexing data structures used for queries. A new version of the system, WebMARS, is developed where the feature vectors are indexed using hybrid trees, which combine aspects of several indexing trees.

**Result presentation** Images are listed in order of decreasing similarity.

**Relevance feedback** Initially, the weights of the edges in the query tree are equal for the children of the same parent and their sum is 1. Based on the relevant images chosen by the user from the query result list, a tree re-weighting process takes place.

**Applications** The database consists of images of ancient African artifacts from the
CHAPTER 2. RELATED WORKS

Fowler Museum of Cultural History at UCLA.

Figure 2.7: Mars shape query with drawn polygon.

2.6.12 NETRA

Developer Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA.

URL http://maya.ece.ucsb.edu/Netra/. A demo of the system is available at web address http://maya.ece.ucsb.edu/Netra/netra.html.

References [WYM97], [WYM99].

Features Images in the database are segmented into regions of homogeneous color. Of those regions, the following features are extracted: color, texture, shape, and spatial location. On the
CHAPTER 2. RELATED WORKS

basis of a training set of images, the RGB color space is quantized, and represented by a color
codebook of 256 colors, the centroids of the quantization cells. The colors of an image region are
also quantized, giving a color feature vector \( f_c = (c_0p_0, ..., c_np_n) \), with \( c \), the index into the color
code book, and \( p \) the fraction of that color in the region, \( p_0 + ... + p_n = 1 \). The number \( n \) is the
number of colors used to represent the region, which is different for each region. Texture is
represented by a feature vector \( f_t \) containing the normalized mean and standard deviation of a
series of Gabor wavelet transforms of the image: \( f_t = (\mu_0, ..., \mu_s, \sigma_0, ..., \sigma_s) \), with \( s \) the number
of scales, and \( k \) the number of directions.

There are three feature vectors used to represent the shape of regions. The first, \( f_k \), is based on the
curvature function of the contour, giving the curvature at each point on the contour. The second,
\( f_k \) is based on the centroid distance function, giving at each contour point the distance to the
centroid of the region. The third, \( f_z \) is the complex coordinate function, representing each contour
point as a complex number with real component equal to the x-coordinate, and the imaginary
component equal to the y-coordinate. On 64 samples of each of these functions, the fast Fourier
transform (FFT) is applied, of which the real (amplitude) component of the coefficients is used,
the numbers \( F_31, ..., F_32 \). The feature vectors are as follows:

\[
\begin{align*}
  f_k &= (|F_1|, ..., |F_32|), \\
  f_k &= (|F_1|, ..., |F_32|) / |F_0|, \\
  f_z &= (|F_31|, ..., |F_1|, |F_2|, ..., |F_32|) / |F_1|.
\end{align*}
\]

**Querying** There are 2,500 images from the Corel photo collection, organized in 25 categories,
with 100 images in each category. You can select any one of them as the query image. All images
in the database have been segmented into homogeneous regions. You can click on one of the
regions and select one of the four image attributes: color, spatial location, texture, and shape.
Instead of using an image example, you can also directly specify the color and spatial location.
The spatial location querying tool utilizes two bounding boxes to define the area of interest. The
inner box is used to define the preferred area, and the box outside is used to constrain the objects
to be within this area. Thus, if the object has any of its bodies exceeding this outside box, they will
not be considered.

**Matching** Consider two color feature vectors, \( f_c^A \) of region \( A \), and \( f_c^B \) of region \( B \). For each color
\( c \) in \( f_c^A \), the closest color \( c^B \) in \( f_c^B \) is found, and the distance \( d(c^A, f_c^B) \) is calculated as the
weighted Euclidean distance in RGB space: \( d(c^A, f_c^B) = |p^A - p^B| \). The distance
between two texture feature vectors is the \( L_1 \) distance. The distance between two shape feature
vectors is the Euclidean distance.

34
CHAPTER 2. RELATED WORKS

Indexing is based on the SS-tree [DWR96]. Color, texture, and shape are indexed separately. The first feature the user specifies is used to retrieve about 100 candidates. Then this feature and the possible other features together are used to order the retrieval result.

Result presentation The matched images are linearly ordered, see figure 2-8.

Applications An initial prototype of NETRA is used in ADL (see above) to search on texture.

2.6.13 QBIC (Query By Image Content)

Developer IBM Almaden Research Center, San Jose, CA.


References [WNR93].

Features Color features computed are: the 3D average color vector of an object or the whole image in RGB, YIQ, Lab, and Munsell color space and a 256-dimensional RGB color histogram. If \( x \) is an \( n \)-dimensional color histogram and \( C = [c_1, c_2, \ldots, c_n] \) is a \( 3 \times n \) matrix whose columns represent the RGB values of the \( n \) quantized colors, the average color vector \( x_{avg} \) is \( C x \). The texture features used in QBIC are modified versions of the coarseness, contrast, and directionality features proposed by Tamura [HTM78].

The shape features consist of shape area, circularity, eccentricity, major axis orientation and a set of Algebraic moment invariants. The major axis orientation and the eccentricity are computed from the second order covariance matrix of the boundary pixels: the major axis orientation as the
CHAPTER 2. RELATED WORKS

direction of the largest eigenvector and eccentricity as the ratio of the smallest eigen value to the largest one. For the database images, these shape features are extracted for all the object contours, semi-automatically computed in the database population step. In this process, the user enters an approximate object outline, which is automatically aligned with the nearby image edges, using the active contours technique. In this object detection step, the user can also associate text to the outlined objects.

The 18 algebraic moment invariants are the eigen values of the matrices $M_{[2,2]}$, $M_{[2,3]} \times M_{[3,2]}$, $M_{[3,3]}$, $M_{[3,4]} \times M_{[4,3]}$, $M_{[4,4]} \times M_{[5,4]}$, where the elements of $M_{[ij]}$ are scaled factors of the central moments.

QBIC also implemented a method of retrieving images based on a rough user sketch. For this purpose, images in the database are represented by a reduced binary map of edge points. This is obtained as follows: first, the color image is converted to a single band luminance; using a Canny edge detector, the binary edge image is computed and is next reduced to size $64 \times 64$. Finally this reduced image is thinned.

**Querying** QBIC allows queries based on example images, user-constructed sketches or selected color and texture patterns. In the last case, the user chooses colors or textures from a sampler. The percentage of a desired color in an image is adjusted by moving sliders.

**Matching** For the average color, the distance between a query object and database object is a weighted Euclidean distance, where the weights are the inverse standard deviation for each component over the samples in the database. In matching two color histograms, two distance measures are used: one low dimensional, easy to compute (the average color distance) and one much more computationally expensive (the quadratic histogram distance). The first one (which is computed for all the images in the database) acts as a filter, limiting the expensive matching computation to the small set of images retrieved by the first matching. The average color distance is $d_{\text{avg}}(x,y) = (x_{\text{avg}} - y_{\text{avg}})(x_{\text{avg}} - y_{\text{avg}})$. The histogram quadratic distance is given by $d_{\text{hist}}(x,y) = (x - y)^T A(x - y)$, where the symmetric color similarity matrix $A$ is given by $a_{ij} = 1 - d_i/d_{\text{max}}$, with $d_i$ being the $L_2$ distance between the colors $i$ and $j$ in the RGB space and $d_{\text{max}} = \max_i d_i$. The texture distance is a weighted Euclidean distance, with the weighting factors being the inverse variances for each of the three texture components over the entire database. Two shapes are matched also by a similar weighted Euclidean distance between shape feature vectors.
CHAPTER 2. RELATED WORKS

In a query by sketch, after reducing the binary sketch image drawn by the user to size 64×64 a correlation based matching is performed, a kind of template matching. This is done by partitioning the user sketch into 8×8 blocks of 8×8 pixels and finding the maximum correlation of each block of the sketch within a search area of 16×16 pixels in the image database (this is done by shifting the 8×8 block in the search area). This local correlation score is computed on the pixel level using logical operations. The matching score of a database image is the sum of the correlation scores of all local blocks.

**Indexing** QBIC was one of the first systems that applied multidimensional indexing to enhance the speed performance of the system. The average color and the texture features (both 3D vectors) are indexed using $R^*$-trees. The 18-dimensional moment-based shape feature vector is first reduced using the KL transform and then indexed by using $R^*$-trees.

**Result presentation** The best matches are presented in decreasing similarity order with (optionally) the matching score aside.

**Relevance feedback** Any retrieved image can be used as a seed for a subsequent query by example.

**Applications** At [http://www.qbic.almaden.ibm.com/tmdemo/](http://www.qbic.almaden.ibm.com/tmdemo/) is a demonstration of the QBIC system as trademark server.

![QBIC Figure](image.png)

**Figure 2.9: QBIC.**
CHAPTER 2. RELATED WORKS

2.6.14 VIR Image Engine

The VIR Image Engine is an extensible framework for building content based image retrieval systems.

Developer Virage Inc.


References [JBC96].

Features A basic concept is that of a primitive, which denotes a feature's type, computation and matching distance. Five abstract data types are defined: global values and histograms, local values and histograms, and graphs. The VIR Image Engine provides a set of general primitives, such as global color, local color, texture and shapes. Apart from these, various domain specific primitives can be created when developing an application. When defining such a primitive, the developer supplies a function for computing the primitive's feature data from the raw image.

Querying and Result presentation The VIR Image Engine provides a set of GUI tools necessary for the development of a user interface. These include facilities for image insertion, image query, weight adjustment for re-query, inclusion of keywords, and support for several popular image file formats. Another available component, the query canvas, allows queries-by-sketch; it consists of a bitmap editor where the user can sketch a picture with drawing tools and color it using the colors from a palette. Also, the user can bring onto the canvas an image from an existing collection and modify it using the same drawing tools. Queries can be performed on various user-defined combinations of primitives.

Matching When defining a new primitive, a function for computing the similarity between two sets of feature data previously extracted must also be supplied by the developer. When comparing two images, for each primitive in the current query combination, a similarity score is computed using the distance function defined within the primitive. These individual scores are combined in an overall score using a set of weights in a way characteristic to the application. This score is then stored in a score structure, which contains also the individual similarity scores for each primitive. This allows a quick recomputation of the overall score for a new set of weights.

Relevance feedback A sort of relevance feedback is obtained by searching for matches to a fixed query image for different sets of weights.

Indexing The storage of feature data for all primitives is the responsibility of the application developer.
CHAPTER 2. RELATED WORKS

Applications The system was integrated into databases from Sybase, Object Design, and Objectivity, and added as a component to the Oracle DBMS. The AltaVista Photofinder and Illustra’s Visual Intelligence system are two applications of Virage technology.

2.6.15 VisualSEEk

Developer Image and Advanced Television Lab, Columbia University, NY.

URL http://www.ctr.columbia.edu/VisualSEEk/

References [JRS97].

Features In the database population step, each image is automatically decomposed into regions of equally dominant colors. For each region, feature properties and spatial properties are retained for the subsequent queries. A query consists of finding the images that contain the most similar arrangements of similar regions. The color region extraction uses the back-projection technique. The first step is the selection of a color set. This is a 166-dimensional binary vector \( \mathbf{c} \), which defines a selection of 166 colors in the HSV color space. From a given image \( I \), another image \( B \) is generated by

\[
B[x, y] = \max_k a[k, j] \mathbf{c}[j],
\]

where \( k \in \{0, ..., 165\} \) is the index of the color of the pixel \((x, y)\) in the image \( I \) and \( a[k, j] \) is the similarity between two colors (with the indices \( k \) and \( j \)) in the HSV space. Next the image \( B \) is filtered and spatially localized color regions are extracted. Along with the color set used for back-projection, the region centroid, area (defined as the number of pixels in the region) and the width and height of the minimum bounding rectangle are also stored.

Querying To start a query, the user sketches a number of regions, positions and dimensions them on the grid (see Figure 2.10) and selects a color for each region. Also, the user can indicate boundaries for location and size and/or spatial relationships between regions. After the system returns the thumbnail images of the best matches, the user is allowed to search by example using the returned images.

Matching To find the matches of a query image with a single region, queries on color set, region absolute location, area and spatial extent are first done independently. The color set similarity is computed by

\[
d(\mathbf{c}_q, \mathbf{c}_t) = (\mathbf{c}_q - \mathbf{c}_t)^T \mathsf{A}(\mathbf{c}_q - \mathbf{c}_t),
\]

where \( \mathbf{c}_q, \mathbf{c}_t \) are two color sets and \( \mathsf{A} = (a[i, j]) \) is the color similarity matrix. If the user has defined spatial boundaries for the query region, then its distance to a target region is 0 if the target region centroid falls inside this boundaries, and is given by the Euclidean distance between the centroids otherwise. The distance in area between two regions is the absolute value of the difference, while the distance between the minimum bounding rectangles, \((w_q, h_q)\) and \((w_t, h_t)\) of two regions is the \( L_2 \) metric. The results of these
CHAPTER 2. RELATED WORKS

queries are intersected and from the obtained candidate set, the best matching images are taken by minimizing a total distance given by the weighted sum of the four distances mentioned. If the query image consists of a number of regions, in absolute or relative location, then for each region positioned in absolute location, a query like that described above is made, and for regions positioned by relative location individual queries on all attributes except location are performed. For the intersection of all this query results, the relative spatial relations specified by the user are evaluated using 2D string representation [SKC87].

**Indexing** The color set distance can be written as \( d(c_q, c_t) = \mu_q + \mu_t - 2c_q r_{qrt} \), where \( \mu_q = c_q A c_q, \mu_t = c_t A c_t \), and \( r_t = A c_t \). For each target region, \( \mu_t \) and \( r_t[m], m \in \{0, ..., 165\} \), are indexed individually. The centroids of the image regions are indexed using a quadtree. For the indexing of the minimum bounding rectangles, R-tree are used.

**Result presentation** The results of a query are displayed in decreasing similarity order. Under each retrieved image, the distance to the query image is indicated.

![Figure 2.10: VisualSEEK query interface.](image)

2.7 Discussion

In this chapter, we introduced some fundamental techniques for content-based image retrieval, including visual content description, similarity/distance measures, indexing scheme, user interaction and system performance evaluation. Our emphasis is on visual feature description techniques.

General visual features most widely used in content-based image retrieval are color, texture, shape, and spatial information. Color is usually represented by the color histogram, color
correlogram, color coherence vector, and color moment under a certain color space. Texture can be represented by co-occurrence matrix, inverse deference moment, entropy, tamura feature, wavelet transform etc. Shape can be boundary-based and region-based and are represented by Fourier Descriptors and Moment invariants, polynomial fitting, Turning Function, Finite Element Method, Chain code, Zernike moments. The spatial relationship between regions or objects is usually represented by a 2D string. In addition, the general visual features on each pixel can be used to segment each image into homogenous regions or objects. Local features of these regions or objects can be extracted to facilitate region-based image retrieval.

There are various ways to calculate the similarity distances between visual features. In this chapter we have discussed some basic metrics, including the Minkowski-form distance, quadratic form distance, Mahalanobis distance, and Cosine Measure.

Efficient indexing of visual feature vectors is important for image retrieval. To set up an indexing scheme, dimensionality reduction is usually performed first to reduce the dimensionality of the visual feature vector. Commonly used dimension reduction methods are PCA, ICA, Karhunen-Loeve (KL) transform, and neural network methods. After dimension reduction, an indexing tree is built up. The most commonly used tree structures are R-tree, R*-tree, quad-tree, K-d-B tree, etc.

Image retrieval systems rely heavily on user interaction. On the one hand, images to be retrieved are determined by the user’s specification of the query. On the other hand, query results can be refined to include more relevant candidates through the relevance feedback of users. Updating the retrieval results based on the user’s feedback can be achieved by updating the images, the feature models, the weights of features in similarity distance, and select different similarity measures.

Although content-based retrieval provides an intelligent and automatic solution for efficient searching of images, the majority of current techniques are based on low level features or current techniques are primarily based on low level features. In general, each of these low level features tends to capture only one aspect of an image property. Neither a single feature nor a combination of multiple features has explicit semantic meaning. In addition, the similarity measures between visual features do not necessarily match human perception. Users are interested in are semantically and perceptually similar images, the retrieval results of low-level feature based retrieval approaches are generally unsatisfactory and often unpredictable. Although relevance
CHAPTER 2. RELATED WORKS

feedback provides a way of filling the gap between semantic searching and low-level data processing, this problem remains unsolved and more research is required.