1 FUNDAMENTALS OF CONTENT-BASED IMAGE RETRIEVAL

Dr. Fuhui Long, Dr. Hongjiang Zhang and Prof. David Dagan Feng

We introduce in this chapter some fundamental theories for content-based image retrieval. Section 1.1, looks at the development of content-based image retrieval techniques. Then, as the emphasis of this chapter, we introduce in detail in Section 1.2 some widely used methods for visual content descriptions. After that, we briefly address similarity/distances measures between visual features, the indexing schemes, query formation, relevance feedback, and system performance evaluation in Sections 1.3, 1.4 and 1.5. Details of these techniques are discussed in subsequent chapters. Finally, we draw a conclusion in Section 1.6.

1.1 Introduction

Content-based image retrieval, a technique which uses visual contents to search images from large scale image databases according to users' 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 [6] was held in Florence. Since then, the application potential of image database management techniques has attracted the attention of researchers [12, 13, 16, 18]. 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 [14, 93]. 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.

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. Theefficient 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 [49] 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 [11, 23, 24, 35, 49, 50, 102]. 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 enormoulsy. 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 [31, 77, 87].

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 (Figure 1-1), 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.

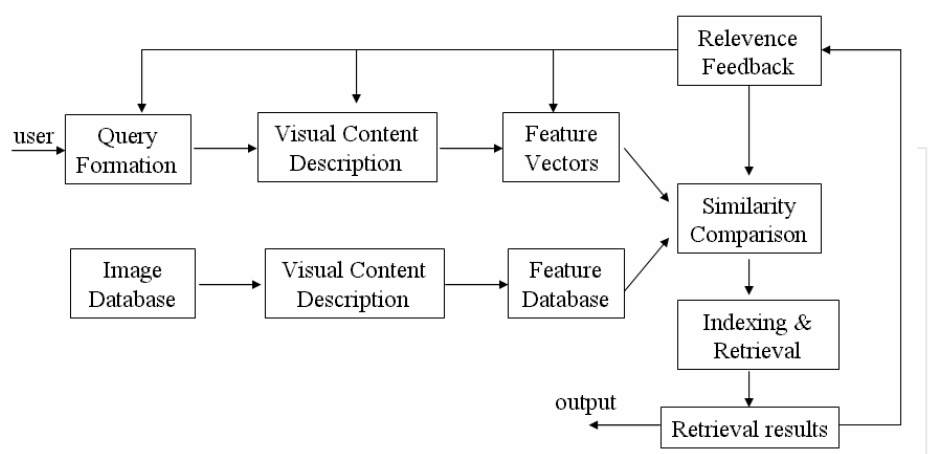

Figure 1-1. Diagram for content-based image retrieval system

1.2 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. This chapter concentrates on general visual contents descriptions. Later chapters discuss domain specific and semantic contents.

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 [8].

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.

COLOR

Color is the most extensively used visual content for image retrieval [27, 43, 44, 45, 47, 65, 71, 89, 91, 103]. 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.

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* [65]. 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 [47].

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 [27].

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.

Color Moments

Color moments have been successfully used in many retrieval systems (like *QBIC* [26, 67]), 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 [89]. Mathematically, the first three moments are defined as:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} f_{ij} \tag{1-2}$$

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^2\right)^{\frac{1}{2}} \tag{1-3}$$

$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^3\right)^{\frac{1}{3}} \tag{1-4}$$

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.

Usually the color moment performs better if it is defined by both the L*u*v* and L*a*b* color spaces as opposed to solely by the HSV space Using the additional third-order moment improves the overall retrieval performance compared to using only the first and second order moments. However, this third-order moment sometimes makes the feature representation more sensitive to scene changes and thus may decrease the performance.

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.

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 both the global and local 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 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 [35] 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 [71]. 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.

Color Coherence Vector

In [72] 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 α_i denote the number of coherent pixels in the *i*th color bin and β_i denote the number of incoherent pixels in an image. Then, the CCV of the image is defined as the vector $<(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_N, \beta_N)>$. Note that $<\alpha_1+\beta_1, \alpha_2+\beta_2, \ldots, \alpha_N+\beta_N>$ 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.

Color Correlogram

The *color correlogram* [44] 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 $I_{c(i)}$ represent the set of pixels whose colors are *c(i)*. Then, the color correlogram is defined as:

$$\gamma_{i,j}^{(k)} = \Pr_{p_1 \in I_{c(i)}, p_2 \in I} \left[p_2 \in I_{c(j)} \,\middle|\, |p_1 - p_2| = k \right] \quad (1\text{-}1)$$

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 autocorrelogram provides the best retrieval results, but is also the most computational expensive due to its high dimensionality.

Invariant Color Features

Color not only reflectsthe 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.

Invariant color representation has been introduced to content-based image retrieval recently. In [33], a set of color invariants for object retrieval was derived based on the Schafer model of object reflection. In [25], specular reflection, shape and illumination invariant representation based on blue ratio vector (r/b, g/b,1) is given. In [34], 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.

TEXTURE

Texture is another important property of images. Various texture representations have been investigated in pattern recognition and computer vision. Basically, texture representation methods can be classified into two categories: *structural* and *statistical*. Structural methods, including *morphological operator* and *adjacency graph*, describe texture by identifying structural primitives and their placement rules. They tend to be most effective when applied to textures that are very regular. Statistical methods, including *Fourier power spectra*, *co-occurrence matrices*, *shift-invariant principal component analysis* (*SPCA*), *Tamura feature*, *Wold decomposition*, *Markov random field*, *fractal model*, and *multi-resolution filtering* techniques such as *Gabor and wavelet transform*, characterize texture by the statistical distribution of the image intensity. In this section, we introduce a number of texture representations [7, 19, 21, 28, 29, 30, 33, 48, 51, 53, 54, 57, 58, 62, 63, 64, 70, 75, 92, 99], which have been used frequently and have proved to be effective in content-based image retrieval

systems.

Tamura Features

The Tamura features [92], including *coarseness*, *contrast*, *directionality*, *linelikeness*, *regularity*, and *roughness*, are designed in accordance with psychological studies on the human perception of texture. The first three components of Tamura features have been used in some early well-known image retrieval systems, such as *QBIC* [26, 67] and *Photobook* [73]. The computations of these three features are given as follows.

Coarseness

Coarseness is a measure of the granularity of the texture. To calculate the coarseness, moving averages $A_k(x,y)$ are computed first using $2^k \times 2^k$ (k = 0, 1, …, 5) size windows at each pixel (x, y), i.e.,

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} g(i,j) / 2^{2k} \tag{1-5}$$

where $g(i, j)$ is the pixel intensity at (i, j).

Then, the differences between pairs of non-overlapping moving averages in the horizontal and vertical directions for each pixel are computed, i.e.,

$$\begin{aligned} E_{k,h}(x,y) &= \left| A_k(x+2^{k-1}, y) - A_k(x-2^{k-1}, y) \right| \\ E_{k,v}(x,y) &= \left| A_k(x, y+2^{k-1}) - A_k(x, y-2^{k-1}) \right| \end{aligned} \tag{1-6}$$

After that, the value of k that maximizes E in either direction is used to set the best size for each pixel, i.e.,

$$S_{best}(x,y) = 2^k \tag{1-7}$$

The coarseness is then computed by averaging S_{best} over the entire image, i.e.,

$$F_{crs} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{best}(i,j) \tag{1-8}$$

Instead of taking the average of S_{best}, an improved version of the coarseness feature can be obtained by using a histogram to characterize the distribution of S_{best}. Compared with using a single value to represent coarseness, using histogram-based coarseness representation can greatly increase the retrieval performance. This modification makes the feature capable of dealing with an image or region which has multiple texture properties, and thus is more useful to image retrieval applications.

Contrast

The formula for the contrast is as follows:

$$F_{con} = \frac{\sigma}{\alpha_4^{1/4}} \tag{1-9}$$

where the kurtosis $\alpha_4 = \mu_4/\sigma^4$, μ_4 is the fourth moment about the mean, and σ^2 is the variance. This formula can be used for both the entire image and a region of the image.

Directionality

To compute the directionality, image is convoluted with two 3x3 arrays (i.e., $\begin{matrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{matrix}$ and $\begin{matrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{matrix}$) and a gradient vector at each pixel is computed.

The magnitude and angle of this vector are defined as:

$$\begin{aligned}|\Delta G| &= \left(|\Delta_H| + |\Delta_V|\right)/2\\ \theta &= \tan^{-1}\left(\Delta_V/\Delta_H\right) + \pi/2\end{aligned} \tag{1-10}$$

where Δ_H and Δ_V are the horizontal and vertical differences of the convolution.

Then, by quantizing θ and counting the pixels with the corresponding magnitude $|\Delta G|$ larger than a threshold, a histogram of θ, denoted as H_D, can be constructed. This histogram will exhibit strong peaks for highly directional images and will be relatively flat for images without strong orientation. The entire histogram is then summarized to obtain an overall directionality measure based on the sharpness of the peaks:

$$F_{dir} = \sum_{p}^{n_p} \sum_{\phi \in w_p} \left(\phi - \phi_p\right)^2 H_D(\phi) \tag{1-11}$$

In this sum p ranges over n_p peaks; and for each peak p, w_p is the set of bins distributed over it; while ϕ_p is the bin that takes the peak value.

Wold Features

Wold decomposition [28, 57] provides another approach to describing textures in terms of perceptual properties. The three Wold components, *harmonic*, *evanescent*, and *indeterministic*, correspond to *periodicity*, *directionality*, and *randomness* of texture respectively. Periodic textures have a strong harmonic component, highly directional textures have a strong evanescent component, and less structured textures tend to have a stronger indeterministic component.

For a homogeneous regular random field $\{y(m,n),\ (m,n) \in Z^2\}$, 2D Wold decomposition allows the field to be decomposed into three mutually orthogonal components:

$$y(m,n) = u(m,n) + d(m,n) = u(m,n) + h(m,n) + e(m,n) \tag{1-12}$$

where $u(m,n)$ is the indeterministic component; and $d(m,n)$ is the deterministic component which can be further decomposed into the harmonic component $h(m,n)$ and evanescent component $e(m,n)$. In the frequency domain, a similar expression exists:

$$F_y(\xi,\eta) = F_u(\xi,\eta) + F_d(\xi,\eta) = F_u(\xi,\eta) + F_h(\xi,\eta) + F_e(\xi,\eta) \tag{1-13}$$

where $F_y(\xi,\eta), F_u(\xi,\eta), F_d(\xi,\eta), F_h(\xi,\eta), F_e(\xi,\eta)$ are the spectral distribution functions (SDF) of $\{y(m,n)\}$, $\{u(m,n)\}$, $\{d(m,n)\}$, $\{h(m,n)\}$ and

$\{e(m,n)\}$ respectively.

In the spatial domain, the three orthogonal components can be obtained by the maximum likelihood estimation (MLE), which involves fitting a high-order AR process, minimizing a cost function, and solving a set of linear equations. In the frequency domain, Wold components can be obtained by global thresholding of Fourier spectral magnitudes of the image. In [57], a method using harmonic peak extraction and MRSAR modeling without an actual decomposition of the image is presented. This method is designed to tolerate a variety of inhomogeneities in natural texture patterns.

Simultaneous Auto-Regressive (SAR) Model

The *SAR model* is an instance of *Markov random field* (*MRF*) models, which have been very successful in texture modeling in the past decades. Compared with other MRF models, SAR uses fewer parameters. In the SAR model, pixel intensities are taken as random variables. The intensity $g(x,y)$ at pixel (x,y) can be estimated as a linear combination of the neighboring pixel values $g(x',y')$ and an additive noise term $\varepsilon(x,y)$, i.e.,

$$g(x,y) = \mu + \sum_{(x',y')\in D} \theta(x',y')g(x',y') + \varepsilon(x,y) \tag{1-14}$$

where μ is a bias value determined by the mean of the entire image; D is the neighbor set of (x, y); $\theta(x',y')$ is a set of weights associated with each of the neighboring pixels; ε(x,y) is an independent Gaussian random variable with zero mean and variance σ^2. The parameters θ and σ are used to measure texture. For instance, a higher σ value implies a finer granularity or less coarseness; a higher $\theta(x, y+1)$ and $\theta(x, y-1)$ values indicate that the texture is vertically oriented. The least square error (LSE) technique or the maximum likelihood estimation (MLE) method is usually used to estimate the parameters of the SAR model.

The SAR model is not rotation invariant. To derive a *rotation-invariant SAR model* (*RISAR*), pixels lying on circles of different radii centered at each pixel (x,y) serve as its neighbor set D. Thus the intensity $g(x,y)$ at pixel (x,y) can be estimated as

$$g(x,y) = \mu + \sum_{i=1}^{p} \theta_i(x,y) l_i(x,y) + \varepsilon(x,y) \tag{1-15}$$

where p is the number of circular neighborhood. To make the computational cost inexpensive and to achieve rotation invariance at the same time, p can neither be too large nor too small. Usually $p = 2$. $l(x,y)$ can be computed by:

$$l_i(x,y) = \frac{1}{8i} \sum_{(x',y')\in N_i} w_i(x',y')g(x',y') \tag{1-16}$$

where N_i is the ith circular neighborhood of (x,y); $w_i(x',y')$ is a set of pre-computed weights indicating the contribution of the pixel (x',y') in the ith circle.

To describe textures of different granularities, the *multi-resolution simultaneous auto-regressive model* (*MRSAR*) [64] has been proposed to enable multi-scale texture analysis. An image is represented by a multi-resolution Gaussian pyramid with low-pass filtering and sub-sampling applied at several successive levels. Either the

SAR or RISAR model may then be applied to each level of the pyramid.

MRSAR has been proved [63, 75]to have better performance on the *Brodatz texture database* [7] than many other texture features, such as principal component analysis, Wold decomposition, and wavelet transform.

Gabor Filter Features

The *Gabor filter* has been widely used to extract image features, especially texture features [22] [48]. It is optimal in terms of minimizing the joint uncertainty in space and frequency, and is often used as an orientation and scale tunable edge and line (bar) detector. There have been many approaches proposed to characterize textures of images based on Gabor filters. The basic idea of using Gabor filters to extract texture features is as follows.

A two dimensional Gabor function $g(x, y)$ is defined as:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{1-17}$$

where, σ_x and σ_y are the standard deviations of the Guassian envelopes along the x and y direction.

Then a set of Gabor filters can be obtained by appropriate dilations and rotations of $g(x, y)$:

$$\begin{aligned} g_{mn}(x,y) &= a^{-m}g(x',y') \\ x' &= a^{-m}(x\cos\theta + y\sin\theta) \\ y' &= a^{-m}(-x\sin\theta + y\cos\theta) \end{aligned} \tag{1-18}$$

where $a > 1$, $\theta = n\pi/K$, $n = 0, 1, \ldots, K-1$, and $m = 0, 1, \ldots, S-1$. K and S are the number of orientations and scales. The scale factor a^{-m} is to ensure that energy is independent of m.

Given an image $I(x, y)$, its Gabor transform is defined as:

$$W_{mn}(x,y) = \int I(x,y)g_{mn}^{*}(x-x_1, y-y_1)dx_1dy_1 \tag{1-19}$$

where * indicates the complex conjugate. Then the mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of $W_{mn}(x, y)$, i.e., $f = [\mu_{00}, \sigma_{00}, \ldots, \mu_{mn}, \sigma_{mn}, \Lambda, \mu_{S-1k-1}, \sigma_{S-1K-1}]$ can be used to represent the texture feature of a homogenous texture region.

Wavelet Transform Features

Similar to the Gabor filtering, the *wavelet transform* [21, 62] provides a multi-resolution approach to texture analysis and classification [19, 54]. Wavelet transforms decompose a signal with a family of basis functions $\psi_{mn}(x)$ obtained through translation and dilation of a mother wavelet $\psi(x)$, i.e.,

$$\psi_{mn}(x) = 2^{-m/2}\psi(2^{-m}x - n) \tag{1-20}$$

where m and n are dilation and translation parameters. A signal $f(x)$ can be represented as:

$$f(x) = \sum_{m,n} c_{mn} \psi_{mn}(x) \tag{1-21}$$

The computation of the wavelet transforms of a 2D signal involves recursive filtering and sub-sampling. At each level, the signal is decomposed into four frequency sub-bands, LL, LH, HL, and HH, where L denotes low frequency and H denotes high frequency. Two major types of wavelet transforms used for texture analysis are the *pyramid-structured wavelet transform* (*PWT*) and the *tree-structured wavelet transform* (*TWT*). The PWT recursively decomposes the LL band. However, for some textures the most important information often appears in the middle frequency channels. To overcome this drawback, the TWT decomposes other bands such as LH, HL or HH when needed.

After the decomposition, feature vectors can be constructed using the mean and standard deviation of the energy distribution of each sub-band at each level. For three-level decomposition, PWT results in a feature vector of 3x4x2 components. For TWT, the feature will depend on how sub-bands at each level are decomposed. A fixed decomposition tree can be obtained by sequentially decomposing the LL, LH, and HL bands, and thus results in a feature vector of 52x2 components. Note that in this example, the feature obtained by PWT can be considered as a subset of the feature obtained by TWT. Furthermore, according to the comparison of different wavelet transform features [58], the particular choice of wavelet filter is not critical for texture analysis.

SHAPE

Shape features of objects or regions have been used in many content-based image retrieval systems [32, 36, 46, 94]. 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 [46], polygonal approximation [2], finite element models [84], and Fourier-based shape descriptors [1, 52, 74]) or *region-based* methods (statistical moments [41, 101]). 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. For a concise comprehensive introductory overview of the shape matching techniques, see [97].

Moment Invariants

Classical shape representation uses a set of *moment invariants*. If the object R is represented as a binary image, then the central moments of order $p+q$ for the shape of object R are defined as:

$$\mu_{p,q} = \sum_{(x,y)\in R} (x - x_c)^p (y - y_c)^q \tag{1-22}$$

where (x_c, y_c) is the center of object. This central moment can be normalized to be scale invariant [[48]]:

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}, \quad \gamma = \frac{p+q+2}{2} \tag{1-23}$$

Based on these moments, a set of moment invariants to translation, rotation, and scale can be derived [41] [101]:

$$\begin{aligned}
\phi_1 &= \mu_{2,0} + \mu_{0,2} \\
\phi_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \\
\phi_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2 \\
\phi_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2 \\
\phi_5 &= (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})\left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2\right] \\
&\quad + (\mu_{0,3} - 3\mu_{2,1})(\mu_{0,3} + \mu_{2,1})\left[(\mu_{0,3} + \mu_{2,1})^2 - 3(\mu_{3,0} + \mu_{1,2})^2\right] \\
\phi_6 &= (\mu_{2,0} - \mu_{0,2})[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{0,3} + \mu_{2,1})^2] + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1}) \\
\phi_7 &= (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2})\left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2\right]
\end{aligned} \tag{1-24}$$

Turning Angles

The contour of a 2D object can be represented as a closed sequence of successive boundary pixels (x_s, y_s), where $0 \le s \le N\text{-}1$ and N is the total number of pixels on the boundary. The *turning function* or *turning angle* $\theta(s)$, which measures the angle of the counterclockwise tangents as a function of the arc-length s according to a reference point on the object's contour, can be defined as:

$$\begin{aligned}
\theta(s) &= \tan^{-1}\left(\frac{y'_s}{x'_s}\right) \\
y'_s &= \frac{dy_s}{ds} \\
x'_s &= \frac{dx_s}{ds}
\end{aligned} \tag{1-25}$$

One major problem with this representation is that it is variant to the rotation of object and the choice of the reference point. If we shift the reference point along the boundary of the object by an amount t, then the new turning function becomes $\theta(s+t)$. If we rotate the object by angle ω, then the new function becomes $\theta(s)+\omega$.

Therefore, to compare the shape similarity between objects A and B with their turning functions, the minimum distance needs to be calculated over all possible shifts t and rotations ω, i.e.,

$$d_p(A,B)=\left(\min_{\omega\in R,t\in[0,1]}\int_0^1\left|\theta_A(s+t)-\theta_B(s)+\omega\right|^p ds\right)^{\frac{1}{p}} \tag{1-26}$$

Here we assume that each object has been re-scaled so that the total perimeter length is 1. This measure is invariant under translation, rotation, and change of scale.

Fourier Descriptors

Fourier descriptors describe the shape of an object with the Fourier transform of its boundary. Again, consider the contour of a 2D object as a closed sequence of successive boundary pixels (x_s, y_s), where $0 \le s \le N$-1 and N is the total number of pixels on the boundary. Then three types of contour representations, i.e., *curvature*, *centroid distance*, and *complex coordinate function*, can be defined.

The curvature $K(s)$ at a point s along the contour is defined as the rate of change in tangent direction of the contour, i.e.,

$$K(s)=\frac{d}{ds}\theta(s) \tag{1-27}$$

where $\theta(s)$ is the turning function of the contour, defined as (1-25).

The centroid distance is defined as the distance function between boundary pixels and the centroid (x_c, y_c) of the object:

$$R(s)=\sqrt{(x_s-x_c)^2+(y_s-y_c)^2} \tag{1-28}$$

The complex coordinate is obtained by simply representing the coordinates of the boundary pixels as complex numbers:

$$Z(s)=(x_s-x_c)+j(y_s-y_c) \tag{1-29}$$

The Fourier transforms of these three types of contour representations generate three sets of complex coefficients, representing the shape of an object in the frequency domain. Lower frequency coefficients describe the general shape property, while higher frequency coefficients reflect shape details. To achieve rotation invariance (i.e., contour encoding is irrelevant to the choice of the reference point), only the amplitudes of the complex coefficients are used and the phase components are discarded. To achieve scale invariance, the amplitudes of the coefficients are divided by the amplitude of DC component or the first non-zero coefficient. The translation invariance is obtained directly from the contour representation.

The Fourier descriptor of the curvature is:

$$f_K=\left[|F_1|,|F_2|,...,|F_{M/2}|\right] \tag{1-30}$$

The Fourier descriptor of the centroid distance is:

$$f_R=\left[\frac{|F_1|}{|F_0|},\frac{|F_2|}{|F_0|},...,\frac{|F_{M/2}|}{|F_0|}\right] \tag{1-31}$$

where F_i in (1-30) and (1-31) denotes the ith component of Fourier transform coefficients. Here only the positive frequency axes are considered because the curvature and centroid distance functions are real and, therefore, their Fourier

transforms exhibit symmetry, i.e., $|F_{-i}| = |F_i|$.

The Fourier descriptor of the complex coordinate is:

$$f_Z = \left[\frac{|F_{-(M/2-1)}|}{|F_1|}, ..., \frac{|F_{-1}|}{|F_1|}, \frac{|F_2|}{|F_1|}, ..., \frac{|F_{M/2}|}{|F_1|} \right] \tag{1-32}$$

where F_1 is the first non-zero frequency component used for normalizing the transform coefficients. Here both negative and positive frequency components are considered. The DC coefficient is dependent on the position of a shape, and therefore, is discarded.

To ensure the resulting shape features of all objects in a database have the same length, the boundary ((x_s, y_s), $0 \le s \le N$-1) of each object is re-sampled to M samples before performing the Fourier transform. For example, M can be set to $2^m = 64$ so that the transformation can be conducted efficiently using the fast Fourier transform.

Circularity, Eccentricity, and Major Axis Orientation

Circularity is computed as:

$$\alpha = \frac{4\pi S}{P^2} \tag{1-33}$$

where S is the size and P is the perimeter of an object. This value ranges between 0 (corresponding to a perfect line segment) and 1 (corresponding to a perfect circle).

The *major axis orientation* can be defined as the direction of the largest eigenvector of the second order covariance matrix of a region or an object. The eccentricity can be defined as the ratio of the smallest eigenvalue to the largest eigenvalue.

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* [17]. 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* [15] 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 disjoin, adjoin or located at the same position. In addition, *2D C-string* [55] is proposed to minimize the number of cutting objects. *2D-B string* [56] 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* [82], and *symbolic image* [37] 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 [90], 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 [38, 100].

1.3 Similarity Measures and Indexing Schemes

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:

$$D(I,J) = (\sum_i | f_i(I) - f_i(J) |^p)^{1/p} \quad (1\text{-}34)$$

when p=1, 2, and ∞, $D(I, J)$ is the L_1, L_2 (also called Euclidean distance), and L_∞ distance respectively. Minkowski-form distance is the most widely used metric for image retrieval. For instance, MARS system [78] used Euclidean distance to compute the similarity between texture features; Netra [61, 60]used Euclidean distance for color and shape feature, and L_1 distance for texture feature; Blobworld [9] used Euclidean distance for texture and shape feature. In addition, Voorhees and Poggio [99] used L_∞ distance to compute the similarity between texture images.

The *Histogram intersection* can be taken as a special case of L_1 distance, which is used by Swain and Ballard [91] to compute the similarity between color images. The intersection of the two histograms of I and J is defined as:

$$S(I,J)=\frac{\sum_{i=1}^{N}\min(f_i(I),f_i(J))}{\sum_{i=1}^{N}f_i(J)} \tag{1-35}$$

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:

$$D(I,J)=\sqrt{(\mathbf{F}_I-\mathbf{F}_J)^T\mathbf{A}(\mathbf{F}_I-\mathbf{F}_J)} \tag{1-36}$$

where $A=[a_{ij}]$ is a similarity matrix, and a_{ij} denotes the similarity between bin i and j. $\mathbf{F}_I$ and $\mathbf{F}_J$ 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 [40, 67] 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:

$$D(I,J)=\sqrt{(\mathbf{F}_I-\mathbf{F}_J)^T C^{-1}(\mathbf{F}_I-\mathbf{F}_J)} \tag{1-37}$$

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.

$$D(I,J)=\sum_{i=1}^{N}(\mathbf{F}_I-\mathbf{F}_J)^2/c_i \tag{1-38}$$

Kullback-Leibler (KL) Divergence and Jeffrey-Divergence (JD)

The *Kullback-Leibler (KL) divergence* measures how compact one feature distribution can be coded using the other one as the codebook. The KL divergence between two images I and J is defined as:

$$D(I,J)=\sum_i f_i(I)\log\frac{f_i(I)}{f_i(J)} \tag{1-39}$$

The KL divergence is used in [66] as the similarity measure for texture.

The *Jeffrey-divergence* (*JD*) is defined by:

$$D(I,J)=\sum_i f_i(I)\log\frac{f_i(I)}{\hat{f}_i}+f_i(J)\log\frac{f_i(J)}{\hat{f}_i} \tag{1-40}$$

where $\hat{f}_i = [f_i(I) + f_i(J)]/2$. In contrast to KL-divergence, JD is symmetric and numerically more stable when comparing two empirical distributions.

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 [26] [67]. 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 [53]. Apart from PCA and KL transformation, *neural network* has also been demonstrated to be a useful tool for dimension reduction of features [10].

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* [5]), *linear quad-trees* [98], *K-d-B tree* [76] and *grid files* [68]. 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 [[102]]. In addition to benefiting indexing, SOM provides users a useful tool to browse the representative images of each type. Details of indexing techniques are given in Chapter 8.

1.4 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.

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 [95]. Query by concept is to retrieve images according to the conceptual description associated with each image in the database. Query by sketch [25] and query by example [3] is to draw a sketch or provide an example image from which images with similar visual features will be extracted from the database. The first two types of queries are related to the semantic description of images which will be introduced in the following chapters.

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 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.

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 [79] [66] [80] [42], 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. Detail discussions on various feedback approaches can be found in chapter 3.

1.5 Performance Evaluation

To evaluate the performance of retrieval system, two measurements, namely, *recall* and *precision* [87], are borrowed from traditional information retrieval. For a query q, the data set of images in the database that are relevant to the query q is denoted as $R(q)$, and the retrieval result of the query q is denoted as $Q(q)$. The precision of the retrieval is defined as the fraction of the retrieved images that are indeed relevant for the query:

$$precision = \frac{|Q(q) \cap R(q)|}{|Q(q)|} \tag{1-45}$$

The recall is the fraction of relevant images that is returned by the query:

$$recall = \frac{|Q(q) \cap R(q)|}{|R(q)|} \tag{1-46}$$

Usually, a tradeoff must be made between these two measures since improving one will sacrifice the other. In typical retrieval systems, recall tends to increase as the number of retrieved items increases; while at the same time the precision is likely to decrease. In addition, selecting a relevant data set $R(q)$ is much less stable due to various interpretations of the images. Further, when the number of relevant images is greater than the number of the retrieved images, recall is meaningless. As a result, precision and recall are only rough descriptions of the performance of the retrieval system.

Recently MPEG7 recommend a new retrieval performance evaluation measure, the *average normalized modified retrieval rank* (*ANMRR*) [104]. It combines the precision and recall to obtain a single objective measure. Denote the number of ground truth images for a given query q as $N(q)$ and the maximum number of ground truth images for all Q queries, i.e., $\max(N(q_1), N(q_2), \ldots, N(q_Q))$, as M. Then for a given query q, each ground truth image k is assigned a rank value $rank(k)$ that is equivalent to its rank in the ground truth images if it is in the first K (where $K=min[4N(q), 2M]$) query results; or a rank value $K+1$ if it is not. The *average rank* $AVR(q)$ for query q is computed as:

$$AVR(q) = \sum_{k=1}^{N(q)} \frac{rank(k)}{N(q)} \tag{1-47}$$

The modified retrieval rank MRR(q) is computed as:

$$MRR(q) = AVR(q) - 0.5 - 0.5 * N(q) \tag{1-48}$$

$MRR(q)$ takes value 0 when all the ground truth images are within the first K retrieval results.

The *normalized modified retrieval rank* $NMRR(q)$, which ranges from 0 to 1, is computed as:

$$NMRR(q) = \frac{MRR(q)}{K + 0.5 - 0.5 * N(q)} \tag{1-49}$$

Then the average normalized modified retrieval rank ANMRR over all Q queries is computed as:

$$ANMRR = \frac{1}{Q}\sum_{q=1}^{Q} NMRR(q) \tag{1-50}$$

1.6 Conclusion

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. Details of indexing of high-dimensional features, user relevance feedback, and semantic description of visual contents will be addressed in chapters 3, 4, 8 and 9.

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 Tamura feature, Wold decomposition, SAR model, Gabor and Wavelet transformation. Shape can be represented by moment invariants, turning angles, Fourier descriptors, circularity, eccentricity, and major axis orientation and radon transform. 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. This chapter introduced some basic metrics, including the Minkowski-form distance, quadratic form distance, Mahalanobis distance, Kullback-Leibler divergence and Jeffrey divergence. Up to now, the Minkowski and quadratic form distance are the most commonly used distances for image retrieval.

Efficient indexing of visual feature vectors is important for image retrieval. To set up an indexing scheme, dimension 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. Details of indexing techniques will be introduced in Chapter 8.

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. Details will

be introduced in Chapter 3.

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 unpredicable. Although relevance 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. New techniques in semantic descriptions of visual contents will be addressed in Chapters 4 and 9.

References

[1] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of affine-invariant Fourier descriptors to recognition of 3D objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 640-647, 1990.

[2] E. M. Arkin, L.P. Chew, D..P. Huttenlocher, K. Kedem, and J.S.B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 209-226, 1991.

[3] J. Assfalg, A. D. Bimbo, and P. Pala, "Using multiple examples for content-based retrieval," *Proc. Int'l Conf. Multimedia and Expo*, 2000.

[4] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. F. Shu, "The virage image search engine: An open framework for image management," *In Proc. SPIC Storage and Retrieval for Image and Video Database*, Feb. 1996.

[5] N. Beckmann, *et al*, "The R*-tree: An efficient robust access method for points and rectangles," *ACM SIGMOD Int. Conf. on Management of Data*, Atlantic City, May 1990.

[6] A. Blaser, Database Techniques for Pictorial Applications, *Lecture Notes in Computer Science*, Vol.81, Springer Verlag GmbH, 1979.

[7] P. Brodatz, "Textures: A photographic album for artists & designers," Dover, NY, 1966.

[8] H. Burkhardt, and S. Siggelkow, "Invariant features for discriminating between equivalence classes," *Nonlinear Model-based Image Video Processing and Analysis*, John Wiley and Sons, 2000.

[9] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," In D. P. Huijsmans and A. W. M. Smeulders, ed. *Visual Information and Information System, Proceedings of the Third International Conference VISUAL'99*, Amsterdam, The Netherlands, June 1999, Lecture Notes in Computer Science 1614. Springer, 1999.

[10] J.A. Catalan, and J.S. Jin, "Dimension reduction of texture features for image retrieval using hybrid associative neural networks," *IEEE International Conference on Multimedia and Expo*, Vol.2, pp. 1211 -1214, 2000.

[11] A. E. Cawkill, "The British Library's Picture Research Projects: Image, Word, and Retrieval," *Advanced Imaging*,Vol.8, No.10, pp.38-40, October 1993.

[12] N. S. Chang, and K. S. Fu, "A relational database system for images," *Technical Report TR-EE 79-82*, Purdue University, May 1979.

[13] N. S. Chang, and K. S. Fu, "Query by pictorial example," *IEEE Trans. on Software Engineering*, Vol.6, No.6, pp. 519-524, Nov.1980.

[14] S. K. Chang, and A. Hsu, "Image information systems: where do we go from here?" *IEEE Trans. on Knowledge and Data Engineering*, Vol.5, No.5, pp. 431-442, Oct.1992.

[15] S. K. Chang, E. Jungert, and Y. Li, "Representation and retrieval of symbolic pictures using generalized 2D string", *Technical Report*, University of Pittsburgh, 1988.

[16] S. K. Chang, and T. L. Kunii, "Pictorial database systems," *IEEE Computer Magazine*, Vol. 14,

No.11, pp.13-21, Nov.1981.
[17] S. K. Chang, Q. Y. Shi, and C. Y. Yan, "Iconic indexing by 2-D strings," *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol.9, No.3, pp. 413-428, May 1987.
[18] S. K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt, "An intelligent image database system," *IEEE Trans. on Software Engineering*, Vol.14, No.5, pp. 681-688, May 1988.
[19] T. Chang, and C.C.J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. on Image Processing*, vol. 2, no. 4, pp. 429-441, October 1993.
[20] I. J. Cox, M. L. Miller, T. P. Minka, T. Papathomas, and P. N. Yianilos, "The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments," *IEEE Trans. on Image Processing*, Vol.9, No.1, pp. 20-37, Jan. 2000.
[21] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. on Information Theory*, Vol. 36, pp. 961-1005, Sept. 1990.
[22] J. G. Daugman, "Complete discrete 2D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. ASSP*, vol. 36, pp. 1169-1179, July 1998.
[23] J. Dowe, "Content-based retrieval in multimedia imaging," *In Proc. SPIE Storage and Retrieval for Image and Video Database*, 1993.
[24] C. Faloutsos et al, "Efficient and effective querying by image content," *Journal of intelligent information systems*, Vol.3, pp.231-262, 1994.
[25] G. D. Finlayson, "Color in perspective," *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol.8, No. 10, pp.1034-1038, Oct. 1996.
[26] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system." *IEEE Computer*, Vol.28, No.9, pp. 23-32, Sept. 1995.
[27] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer graphics: principles and practice*, 2nd ed., Reading, Mass, Addison-Wesley, 1990.
[28] J. M. Francos. "Orthogonal decompositions of 2D random fields and their applications in 2D spectral estimation," N. K. Bose and C. R. Rao, editors, *Signal Processing and its Application*, pp.20-227. North Holland, 1993.
[29] J. M. Francos, A. A. Meiri, and B. Porat, "A unified texture model based on a 2d Wold like decomposition," *IEEE Trans on Signal Processing*, pp.2665-2678, Aug. 1993.
[30] J. M. Francos, A. Narasimhan, and J. W. Woods, "Maximum likelihood parameter estimation of textures using a Wold-decomposition based model," *IEEE Trans. on Image Processing*, pp.1655-1666, Dec.1995.
[31] B. Furht, S. W. Smoliar, and H.J. Zhang, *Video and Image Processing in Multimedia Systems*, Kluwer Academic Publishers, 1995.
[32] J. E. Gary, and R. Mehrotra, "Shape similarity-based retrieval in image database systems," *Proc. of SPIE, Image Storage and Retrieval Systems*, Vol. 1662, pp. 2-8, 1992.
[33] T. Gevers, and A.W.M.Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval," *IEEE Trans. on image processing*, Vol.9, No.1, pp102-119, 2000.
[34] T. Gevers, and A. W. M. Smeulders, "Content-based image retrieval by viewpoint-invariant image indexing," *Image and Vision Computing*, Vol.17, No.7, pp.475-488, 1999.
[35] Y. Gong, H. J. Zhang, and T. C. Chua, "An image database system with content capturing and fast image indexing abilities", *Proc. IEEE International Conference on Multimedia Computing and Systems*, Boston, pp.121-130, 14-19 May 1994.
[36] W. I. Grosky, and R. Mehrotra, "Index based object recognition in pictorial data management," *CVGIP*, Vol. 52, No. 3, pp. 416-436, 1990.
[37] V. N. Gudivada, and V. V. Raghavan, "Design and evaluation of algorithms for image retrieval by spatial similarity," *ACM Trans. on Information Systems*, Vol. 13, No. 2, pp. 115-144, April 1995.
[38] F. Guo, J. Jin, and D. Feng, "Measuring image similarity using the geometrical distribution of image contents", *Proc. of ICSP*, pp.1108-1112, 1998.
[39] A. Gupta, and R. Jain, "Visual information retrieval," *Communication of the ACM*, Vol.40, No..5, pp.71-79, May, 1997.
[40] J. Hafner, *et al*., "Efficient color histogram indexing for quadratic form distance functions," IEEE *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, pp. 729-736, July 1995.
[41] M. K. Hu, "Visual pattern recognition by moment invariants," in J. K. Aggarwal, R. O. Duda, and A. Rosenfeld, *Computer Methods in Image Analysis*, IEEE computer Society, Los Angeles, CA, 1977.
[42] J. Huang, S. R. Kumar, and M. Metra, "Combining supervised learning with color correlograms for content-based image retrieval," *Proc. of ACM Multimedia'95*, pp. 325-334, Nov. 1997.

[43] J. Huang, S.R. Kumar, M. Metra, W. J., Zhu, and R. Zabith, "Spatial color indexing and applications," *Int'l J. Computer Vision*, Vol.35, No.3, pp. 245-268, 1999.
[44] J. Huang, *et al*., "Image indexing using color correlogram," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 762-768, Puerto Rico, June 1997.
[45] M. Ioka, "A method of defining the similarity of images on the basis of color information," *Technical Report RT-0030*, IBM Tokyo Research Laboratory, Tokyo, Japan, Nov. 1989.
[46] H. V. Jagadish, "A retrieval technique for similar shapes," *Proc. of Int. Conf. on Management of Data, SIGMOID'91*, Denver, CO, pp. 208-217, May 1991.
[47] A. K. Jain, *Fundamental of Digital Image Processing*, Englewood Cliffs, Prentice Hall, 1989.
[48] A. K. Jain, and F. Farroknia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, Vo.24, No.12, pp. 1167-1186, 1991.
[49] R. Jain, *Proc. US NSF Workshop Visual Information Management Systems*, 1992.
[50] R. Jain, A. Pentland, and D. Petkovic, *Workshop Report: NSF-ARPA Workshop on Visual Information Management Systems*, Cambridge, Mass, USA, June 1995.
[51] A. Kankanhalli, H. J. Zhang, and C. Y. Low, "Using texture for image retrieval," *Third Int. Conf. on Automation, Robotics and Computer Vision*, pp. 935-939, Singapore, Nov. 1994.
[52] H. Kauppinen, T. Seppnäen, and M. Pietikäinen, "An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification," *IEEE Trans. Pattern Anal. and Machine Intell*., Vol. 17, No. 2, pp. 201-207, 1995.
[53] W. J. Krzanowski, *Recent Advances in Descriptive Multivariate Analysis*, Chapter 2, Oxford science publications, 1995.
[54] A. Laine, and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, pp. 1186-1191, Nov. 1993.
[55] S. Y. Lee, and F. H. Hsu, "2D C-string: a new spatial knowledge representation for image database systems," *Pattern Recognition*, Vol. 23, pp 1077-1087, 1990.
[56] S. Y. Lee, M.C. Yang, and J. W. Chen, "2D B-string: a spatial knowledge representation for image database system," *Proc. ICSC'92 Second Int. computer Sci. Conf.*, pp.609-615, 1992.
[57] F. Liu, and R. W. Picard, "Periodicity, directionality, and randomness: Wold features for image modeling and retrieval," *IEEE Trans. on Pattern Analysis and Machine Learning*, Vol. 18, No. 7, July 1996.
[58] W. Y. Ma, and B. S. Manjunath, "A comparison of wavelet features for texture annotation," *Proc. of IEEE Int. Conf. on Image Processing*, Vol. II, pp. 256-259, Washington D.C., Oct. 1995.
[59] W. Y. Ma, and B. S. Manjunath, "Image indexing using a texture dictionary," *Proc. of SPIE Conf. on Image Storage and Archiving System*, Vol. 2606, pp. 288-298, Philadelphia, Pennsylvania, Oct. 1995.
[60] W. Y. Ma, and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," *Multimedia Systems*, Vol.7, No.3, pp.:184-198, 1999.
[61] W. Y. Ma, and B. S. Manjunath, "Edge flow: a framework of boundary detection and image segmentation," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 744-749, Puerto Rico, June 1997.
[62] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 674-693, July 1989.
[63] B. S. Manjunath, and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 837-842, Aug. 1996.
[64] J. Mao, and A. K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognition*, Vol. 25, No. 2, pp. 173-188, 1992.
[65] E. Mathias, "Comparing the influence of color spaces and metrics in content-based image retrieval," *Proceedings of International Symposium on Computer Graphics, Image Processing, and Vision*, pp. 371 -378, 1998.
[66] T. P. Minka, and R. W. Picard, "Interactive learning using a 'society of models', " *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 447-452, 1996.
[67] W. Niblack et al., "Querying images by content, using color, texture, and shape," *SPIE Conference on Storage and Retrieval for Image and Video Database*, Vol. 1908, pp.173-187, April 1993.
[68] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "The grid file: an adaptable symmetric multikey file structure," *ACM Trans. on Database Systems*, pp. 38-71, March 1984.
[69] V. E. Ogle, and M. Stonebraker, "Chabot: Retrieval from a relational database of images," *IEEE Computer*, Vol.28, No.9, pp. 40-48, Sept. 1995.
[70] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with

classification based feature distributions," *Pattern Recognition*, Vol.29, No.1, pp.51-59, 1996.
[71] G.Pass, and R. Zabith, "Comparing images using joint histograms," *Multimedia Systems*, Vol.7, pp.234-240, 1999.
[72] G. Pass, and R. Zabith, "Histogram refinement for content-based image retrieval," *IEEE Workshop on Applications of Computer Vision*, pp. 96-102, 1996.
[73] A. Pentland, R.W. Picard and S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases," *Proc. Storage and Retrieval for Image and Video Databases II*, Vol. 2185, San Jose, CA, USA February, 1994.
[74] E. Persoon, and K. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, and Cybern*., Vol. 7, pp. 170-179, 1977.
[75] R. W. Picard, T. Kabir, and F. Liu, "Real-time recognition with the entire Brodatz texture database," *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 638-639, New York, June 1993.
[76] J. T. Robinson, "The k-d-B-tree: a search structure for large multidimensional dynamic indexes," *Proc. of SIGMOD Conference*, Ann Arbor, April 1981.
[77] Y. Rui, T. S. Huang, and S. F. Chang, "Image retrieval: current techniques, promising directions and open issues, " *Journal of Visual Communication and Image Representation*, Vol.10, pp. 39-62, 1999.
[78] Y. Rui, T.S.Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," *Proceedings of International Conference on Image Processing*, Vol.2, pp. 815 -818, 1997.
[79] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology*, 1998.
[80] Y. Rui, *et al*, "A relevance feedback architecture in content-based multimedia information retrieval systems," *Proc of IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997.
[81] G. Salton, and M. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1983.
[82] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, Vol.16, No.2, pp.187-260, 1984.
[83] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1989.
[84] S. Sclaroff, and A. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 6, pp. 545-561, June 1995.
[85] S. Sclaroff, L. Taycher, and M. L. Cascia, "ImageRover: a content-based image browser for the World Wide Web," Boston University CS Dept. *Technical Report* 97-005, 1997.
[86] A. W. M. Smeulders, S. D. Olabariagga, R. van den Boomgaard, and M. Worring, "Interactive segmentation," *Proc. Visual'97: Information Systems*, pp.5-12, 1997.
[87] A. M. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years, " *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.22, No.12, pp. 1349-1380, Dec. 2000.
[88] J. R. Smith, and S. F. Chang, "VisualSEEk: a fully automated content-based image query system," *ACM Multimedia 96*, Boston, MA, Nov. 1996.
[89] M. Stricker, and M. Orengo, "Similarity of color images," *SPIE Storage and Retrieval for Image and Video Databases III*, vol. 2185, pp.381-392, Feb. 1995.
[90] M. Stricker, and M. Orengo, "Color indexing with weak spatial constraint," *Proc. SPIE Conf. On Visual Communications*, 1996.
[91] M. J. Swain, and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, pp.11-32, 1991.
[92] H. Tamura, S. Mori, and T. Yamawaki, "Texture features corresponding to visual perception," *IEEE Trans. On Systems, Man, and Cybernetics*, vol. Smc-8, No. 6, June 1978.
[93] H. Tamura, and N.Yokoya, "Image database systems: A survey, " *Pattern Recognition*, Vol.17, No.1, pp. 29-43, 1984.
[94] D. Tegolo, "Shape analysis for image retrieval," *Proc. of SPIE, Storage and Retrieval for Image and Video Databases -II*, no. 2185, San Jose, CA, pp. 59-69, February 1994.
[95] A. Vailaya, M. A. G. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for content-based indexing," *IEEE Trans. on Image Processing*, Vol.10, No.1, Jan. 2001.
[96] N. Vasoncelos, and A. Lippman, "A probabilistic architecture for content-based image retrieval," *Proc. Computer vision and pattern recognition*, pp. 216-221, 2000.
[97] R. C. Veltkamp, and M. Hagedoorn, "State-of-the-art in shape matching," *Technical Report UU-CS-1999-27*, Utrecht University, Department of Computer Science, Sept. 1999.

[98] J. Vendrig, M. Worring, and A. W. M. Smeulders, "Filter image browsing: exploiting interaction in retrieval," *Proc. Viusl'99: Information and Information System*, 1999.
[99] H. Voorhees, and T. Poggio. "Computing texture boundaries from images," *Nature*, 333:364-367, 1988.
[100] H. Wang, F. Guo, D. Feng, and J. Jin, "A signature for content-based image retrieval using a geometrical transform," *Proc. Of ACM MM'98*, Bristol, UK, 1998.
[101] L. Yang, and F. Algregtsen, "Fast computation of invariant geometric moments: A new method giving correct results," *Proc. IEEE Int. Conf. on Image Processing*, 1994.
[102] H. J. Zhang, and D. Zhong, "A Scheme for visual feature-based image indexing," *Proc. of SPIE conf. on Storage and Retrieval for Image and Video Databases III*, pp. 36-46, San Jose, Feb. 1995.
[103] H. J. Zhang, *et al*, "Image retrieval based on color features: An evaluation study," *SPIE Conf. on Digital Storage and Archival*, Pennsylvania, Oct. 25-27, 1995.
[104] MPEG Video Group, Description of core experiments for MPEG-7 color/texture descriptors, *ISO/MPEGJTC1/SC29/WG11 MPEG98/M2819*, July 1999.