CHAPTER 3

Basics of Content based Image Retrieval

In the present chapter we have given a brief overview of image features such as color, texture and shape and a description on similarity measures used in content based image retrieval. An overview of image segmentation is also given.

3.1 Introduction

Content-based Image Retrieval (CBIR) is a technology that in principle helps to retrieve digital picture archives by their visual content. Low-level image features like texture, color and shape are extracted from the images of the database to define them in terms of their features. Images of the same category are expected to have similar characteristics. Therefore, when similarity measurement is performed on the basis of image features, the output set achieves a high level of retrieval performance as compared to text based retrieval. CBIR has several advantages over the traditional text based retrieval. The use of the visual contents of the query image in CBIR is more efficient and effective way of finding the relevant images than searching based on text annotations. Also CBIR does not consume the time wasted in manual annotation process of text based approach. Even though the low level image features, color, texture and shape have been shown to be effective and objective in CBIR, they are not sufficient to represent the image semantics and the semantic-based image retrieval is still an open ended research problem. A typical image retrieval block diagram is shown in Figure 3.1.

![Figure 3.1 Block diagram of Content based Image Retrieval (CBIR)](image-url)
3.2 Image Features

One of the major challenges in CBIR is to describe semantic content (automatically) since “content” is a profound word in relationship with computers and data processing. Describing the content directly in a semantic (human like) way is complex and difficult due to the lack of knowledge and a sufficient level of intelligence. Therefore, CBIR for images and video mainly exploits the properties of human visual perception via content descriptors, which are also called visual features. These features can be divided into two general types, low-level and high-level. The low-level visual features are color, texture, and shape. The extraction of low-level visual features is an essential part in CBIR systems. The traditional approaches, as explained by different authors like Faloutsos et al., (1994); Gevers and Smeulders, (2000); Ortega et al., (1997) and Pentland et al., (1996) compute the global features of the images. Here, the low-level image content is described as a mixture of different colors, textures, and edge directions.

3.2.1 Color Feature

Color plays an important role in Human Visual Perception, and it is probably the most dominant cue to be recognized in an image. Therefore the color also plays a significant role as a feature in CBIR as various existing systems (Faloutsos et al., 1994; Pentland et al., 1996; Gevers and Smeulders, 2000; Wang and Wiederhold, 2000; Carson et al., 2002) use color for content retrieval. Color can be represented in various domains (Foley et al., 1996) such as HSV, RGB, YUV, CIE-Lab, CIE-Luv, etc., each of which has its own advantages and disadvantages for certain applications. Since in CBIR, the human visual perception plays an eminent role, a perceptually uniform color space may prove useful. Colors are best feature for initial level of filtering. Before selecting an appropriate color description, color space must be determined first (Long et al., 2003). Color features can be signified in many ways like color histogram, color moment, color correlogram, color coherence vector, color codebooks, dominant color descriptor etc. In the following section a brief description of the color features is given.
a) **Color Histogram**

A color histogram represents the distribution of colors in an image, where each histogram bin corresponds to a color in the quantized color space. It represents the number of pixels that have colors in each of a fixed list of color ranges that span the image color spaces, the set of all possible colors. Color histograms are a set of bins where each bin denotes image being of a particular color. A color histogram for a given image is defined as a vector:

\[
H = \{H[0], H[1], H[2], \ldots \ldots H[i] \ldots \ldots H[n]\}
\]  

(3.1)

The main advantage of a color histogram is its small sensitivity to variations in scale, rotation and translation of an image. In order to compare images of different sizes, color histograms should be normalized. The normalized color histogram \(H'\) is defined as:

\[
H' = \{H'[0], H'[1], H'[2], \ldots \ldots H'[i] \ldots \ldots H'[n]\}
\]  

(3.2)

Where \(H'[i] = \frac{H[i]}{p}\), \(p\) is the total number of pixels of an image (Smith et al., 1996b).

b) **Color Moments**

Another color descriptor in CBIR is the color moment and the color moments are the statistical moments of the probability distributions of colors (Stricker and Orengo, 1995). Color moments used especially when images contain just the objects. The first order (mean), the second (variance) and the third order (skewness) color moments have been proved to be effective and efficient in representing color distribution of images. Color moments have been efficiently and effectively used in color distribution.

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

(3.3)

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

(3.4)
\[
    s_i = \sqrt[3]{\frac{1}{N} \sum_{j=1}^{N} (f_{ij} - \mu_i)^3}
\]

In which \( f_{ij} \) presents the probability of the color component of pixel \( j \) is \( i \), \( N \) presents the number of pixels in the image. However, color moment is only the initial color characteristics extraction of image, and the effect of extraction is very rough.

c) **Color Coherence Vector (CCV)**

One way of incorporating the spatial information into color histograms was introduced by Pass and Zabith, (1996). They proposed the CCV, where each histogram bin is partitioned into two types: coherent if it belongs to a uniformly-colored region or incoherent otherwise. A CCV for an image \( I \) can be defined as

\[
    CCV(I) = \{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)\}
\]

where \( \alpha_i \) and \( \beta_i \) denote the number of coherent and incoherent pixels for the \( i^{th} \) color bin and \( N \) is the number of histogram bins.

d) **Color Correlogram**

Another color descriptor utilizing spatial information of the image pixels is color correlogram (Huang et al., 1997a). Besides describing the color distribution in an image, it also characterizes the spatial correlation of color pairs. Therefore, a simplified version is usually called color auto-correlogram, which only considers spatial correlation between identical colors.

In practical terms a correlogram of an image corresponds to a table indexed by color pairs \( (c_i, c_j) \) so that the \( d^{th} \) entry for row \( (i, j) \) designates the probability of finding a pixel of color \( c_j \) at a distance \( d \) from a pixel of color \( c_i \) in the image. Very much for computational reasons Huang et al., (1997) concluded that the auto-correlogram of an image (a subset of the correlogram capturing the spatial correlation between identical colors only) is sufficient for the purpose of image retrieval.
e) **Color Codebooks**

Color codebooks (Ma and Manjunath, 1997b) are another type of descriptors, where a certain number of colors are either manually selected e.g. the ones that are most distinguishable for human eyes (Ravishankar *et al.*, 1999) or determined by dynamic quantization (Mojsilovic and Soljanin, 2001) to limit the amount of color information. This approach brings color description closer to human perception by specifying important colors. However, the manual method needs to define a certain amount of colors to avoid erroneous color mapping.

In NeTra (Ma and Manjunath, 1997b), each image region is represented by a subset of colors from a color codebook. The color codebook itself could be context dependent, and a different codebook can exist for different applications. From a training dataset of image samples, the codebook is constructed using the generalized Lloyd Algorithm (GLA) to vector quantize colors in the RGB color space.

f) **Dominant Color Descriptor**

A compact color set descriptor also known as dominant color descriptor (DCD) was employed by researchers (Deng *et al.*, 1999a; Fauqueur and Boujemaa, 2002; Manjunath *et al.*, 2001; Mojsilovic and Soljanin, 2001; Mojsilovic *et al.*, 2002) in their studies. The main idea behind this is to describe the prominent colors in the ROI (region of interest) where colors are dynamically clustered (i.e. by color distortion and area until a certain number of clusters are reached). Moreover, it is further consistent to HVP, as HVS mainly perceives dominant colors and discards the rest (Mojsilovic *et al.*, 2002). Due to this fact, it is sufficient to represent the color content of an ROI by the few DCs present in the visual scenery.

3.2.2 **Texture Feature**

Texture analysis has been extensively used in computer vision and pattern recognition applications due to its potential in extracting the prominent features. Texture also indicates the spatial distribution of color and gray levels. Therefore the two-dimensional histograms or co-occurrence matrices are reasonable good texture analysis tools. There are several properties, such as coarseness, contrast, and directionality, which play an important role in describing texture. Coarseness measures the texture
scale (average size of regions that have the same intensity), contrast measure vividness of the texture (depends on the variance of the gray-level histogram) and directionality gives the eventual main direction of the image texture. Formulas to calculate these attributes can be found in (Del Bimbo, 1999). Texture analysis can be helpful when objects in an image are more characterized by their texture than by intensity and traditional thresholding techniques cannot be used effectively. Although there is no formal definition for texture, intuitively this descriptor provides measures of properties such as smoothness, coarseness, and regularity. Texture has qualities such as periodicity and scale; it can be described in terms of direction, coarseness, contrast and so on (Tamura et al., 1978). Textures can be roughly classified into three categories, namely, statistical, structural, and spectral (Gonzalez and Woods, 2002). The statistical approach characterizes texture by the statistical properties of the gray-levels of the pixels in an image. The structural approach assumes that texture is formed with simple primitives called “texels” (texture elements) by following some placement rules. The spectral approach is based on the analysis of power spectral density function and filtering theory in the frequency domain. Statistical methods include Tamura feature, Wold decomposition, and multiresolution filtering techniques such as Gabor and wavelet transform, characterize texture by statistical distribution of image intensity.

The common known texture descriptors used in CBIR are 1. Gabor-filter (Wu et al., 2000), 2. Co-occurrence matrices (Haralick, 1979), 3. Auto-correlation function, (Tuceryan and Jain, 1993) 4. Tamura features (Tamura et al., 1978), 5. Wavelet Transformation (WT), (Stollnitz et al., 1996), 6. Contourlet Transformation (CT), (Vo et al., 2007) and are explained in the following section.

a) Gabor Wavelet

The Gabor filter is one of the popular and powerful texture descriptors, presenting a multi-resolution approach. The main idea is to process an image by a bank of filters at different scales and orientations. Filtering can be applied in either spatial or frequency domain. Turner (1986), first implemented this method by using a bank of Gabor filters for the analyses of texture. A bank of filters at different scales and orientations allows multichannel filtering of an image to extract frequency and orientation information. This can be used to decompose the image into texture features. According to Wu et al.,
(2000), an image $I(x, y)$ filtered with Gabor filter $g_{mn}$ results in its Gabor wavelet transform $W_{mn}$, which captures different frequency and orientation information about the texture, can be formulated as,

$$W_{mn}(x, y) = \int I(x_1, y_1)g^*_{mn}(x - x_1, y - y_1)dx_1dy_1$$

(3.7)

For each scale and orientation the magnitude response $|W_{mn}|$ is calculated as an output from which the first and second order moments (mean and standard deviation) are computed as the texture features. Thus, the feature vector is rather small and is formed per scale and orientation. However, a significant drawback is the computation of the filter coefficients, which is a complex process especially when a higher number of scales and orientations are applied.

b) **Co-Occurrence Matrix**

An earlier statistical approach was the grey-level co-occurrence matrix (GLCM) by Harlick, (1979). Gray Level Co-occurrence Matrix (GLCM), one of the most known texture analysis methods, estimates image properties related to second-order statistics. Each entry $(i, j)$ in GLCM corresponds to the number of occurrences of the pair of gray levels $i$ and $j$ which are a distance $d$ apart in original image. GLCMs have been used very successfully for texture classification and evaluation (Ohanian and Dubes, 1992). Haralick defined the GLCM as a matrix of frequencies at which two pixels, separated by a certain vector, occur in the image. The distribution in the matrix will depend on the angular and distance relationship between pixels. Varying the vector used allows the capturing of different texture characteristics. Once the GLCM has been created, various features can be computed from it. These have been classified into four groups: visual texture characteristics, statistics, information theory and information measures of correlation (Gotlieb and Kreyszig, 1990; Haralick, 1979). Based on the GLCM, features can be computed such as energy, entropy, contrast, and homogeneity, all of which describe the underlying texture properties (Howarth and Ruger, 2005). The feature vector size, therefore, depends on the range of distance vectors and the amount of properties calculated from the co-occurrence matrix. The texture description power of GLCM depends on the combination of selected distance vectors where too few will
provide a rather poor description and too many will increase the computational costs during feature extraction.

c) **Auto-correlation Function**

An important property of texture is the repetitive nature of the placement of texture elements. The auto-correlation function of an image can be used to assess the amount of regularity as well as the fineness and coarseness of the texture. If the texture is coarse, then the auto-correlation function will drop slowly with distance; otherwise it will drop very rapidly. Formally, the autocorrelation function of an image $I(x, y)$ is defined as follows (Tuceryan and Jain, 1993):

$$\rho(x, y) = \frac{\sum_{u=0}^{M} \sum_{v=0}^{N} I(u,v)I(u+x,v+y)}{\sum_{u=0}^{M} \sum_{v=0}^{N} I^2(u,v)} \quad (3.8)$$

where $x, y$ is the position difference in $u, v$ direction and $M, N$ are the image dimensions.

d) **Tamura Features**

Tamura et al., (1978) proposed a texture description by developing features close to the human visual system. In his work, he defined six features, namely coarseness, contrast, directionality, line-likeness, regularity, and roughness, compared to psychological measurements for human subjects where the first three features result in the best performance observed in (Howarth and Ruger, 2004). The most fundamental attribute used by Tamura was coarseness, which represents the relation between scale and repetition rates. Thus, coarseness tries to recognize the largest texture scale that exists in an image by measuring notable variations of grey-levels in non-overlapping windows of different sizes. Contrast measure the variation of grey levels in an image and describes to what extend their distribution is biased to black and white. It is modelled by the ratio of standard deviation and kurtosis over the image grey levels. The main idea behind directionality is to capture distribution of oriented local edges against their directional angles. In order to accomplish this, edge detection with simple masks (e.g. Sobel-operator) is applied where edge orientation (angle) and edge strength
(magnitude) are calculated for each pixel. Then, a histogram is generated by thresholding magnitude and further quantizing it by edge angles. The histogram will reflect information about the degree of directionality. Due to its psychological approach, Tamura properties represent texture close to HVP but extraction of those features might be computationally complex.

e) Wavelet Transform

Wavelet transformations are based on small waves, called wavelets, of varying frequency and limited duration (Gonzalez and Woods, 1992). It is a mathematical tool used for the hierarchical decomposition of an image and to transform an image from spatial domain to frequency domain. They allow certain functions in terms of a coarse overall shape, plus details that range from broad to narrow for an image, curve or a surface (Stollnitz et al., 1996). In wavelet analysis the signal to be analyzed is multiplied with a wavelet function and then the transform is computed for each segment generated. The Wavelet Transform, at high frequencies, gives good time resolution and poor frequency resolution, while at low frequencies, the Wavelet Transform gives good frequency resolution and poor time resolution. Wavelet analysis is a tool that has emerged relatively recently from the mathematical community and has found a variety of applications in areas such as signal processing, numerical analysis, music synthesis, and computer graphics (Stollnitz et al., 1996). There are a host of wavelet transforms in the literature (Stollnitz et al., 1996), each bearing different properties such as smoothness, symmetry, number of vanishing moments, and compactness of support. Wavelets have many favorable properties, such as vanishing moments, hierarchical and multiresolution decomposition structure, linear time and space complexity of the transformations, decorrelated coefficients, and a wide variety of basis functions.

f) Contourlet Transform (CT)

The contourlet transform provides a multi-scale, multi-directional decomposition of an image. It is a combination of a Laplacian pyramid and a directional filter bank (DFB). Bandpass images from the Laplacian pyramid are fed into the DFB so that directional information can be captured. The low frequency components are separated from the directional components. After decimation, the decomposition is iterated using the same
DFB. Its redundancy ratio is less than 4/3 because the directional sub-bands are also decimated (Vo et al., 2007). Contourlet transform is a multi scale and directional image representation that uses first a wavelet like structure for edge detection, and then a local directional transform for contour segment detection.

g) **Wold Features**

Wold decomposition describes textures in terms of perceptual properties. The three Wold components, harmonic, evanescent, and indeterministic, correspond to periodicity, directionality, and randomness of texture respectively. Picard and Liu, (1994) have presented a technique based on Wold decomposition which provides a description of textures in terms of periodicity, directionality and randomness.

**3.2.3 Shape Features**

Among other generic image features that are used to achieve this objective, like color and texture, shape is considered a very important visual feature in object recognition and retrieval (Zhang and Lu, 2002). Using the shape of an object, for object recognition and image retrieval, is an important topic of computer vision and multimedia processing. Finding good shape descriptors and similarity measures are the central issues in these applications (Xu et al., 2009). However, shape representation and description is still a difficult task. According to Kim and Kim, (2000), a good shape representation should be compact and retains the essential characteristics of the shape. Meanwhile, invariant to rotation, scale, and translation are also required since such invariance is consistent with the human vision system. Besides, a good method should deal with the challenges like noise, distortion and occlusion since they change a shape in a more complex way.

Shape information as a property of objects or images is not directly used by humans to describe the content due to their learning capabilities, long-term memory, and intelligence where objects are rather distinguished by their semantic meaning. Nevertheless, shape is an important feature in image analysis to describe objects or image regions. Before applying a shape descriptor, an object extraction or a highly accurate strong segmentation is required in order to extract meaningful regions or objects, which is hard to achieve due to the severe limitations and infeasibilities of such
processes over general purpose images. Therefore, shape descriptors are rarely used in CBIR systems. There are few exceptions especially in region-based CBIR systems, due to Carson et al., (2002); Ma and Manjunath, (1997b) and Wang et al., (2000), where shape is used to describe image region properties. Generally speaking, the existing shape descriptors are mainly applied either on binary image databases or image databases where objects are manually extracted. There are two types of shape descriptors: contour-based and region-based. As the names imply, the contour-based methods extract shape properties based on the object outline (contour), and region-based methods utilize the pixel distribution of the 2D object region.

Contour-based methods, such as chain code (Freeman and Saghri, 1978), shape signature (Zhang and Lu, 2001a,b), polygonal approximation (Gu, 1995), autoregressive models (Kauppinen et al., 1995), Fourier descriptors (Persoon and Fu, 1977) and CSS (Mokhtarian et al., 1997), exploit shape boundary information which is crucial to human perception in judging shape similarity. Region-based methods, such as geometric moments (Hu, 1962), Zernike moments (Teague, 1980; Mehtre et al., 1997), exploit only shape interior information, therefore can be applied to more general shapes.

a) Fourier Descriptors (FD)

Fourier descriptors (Charles et al., 1972) are obtained by applying Fourier transform on shape boundary (usually represented by a shape signature), the Fourier transformed coefficients are called the Fourier descriptors of the shape. For good shape description, an appropriate shape signature is essential to obtain Fourier descriptors. Fourier descriptors characterize the object shape in a frequency domain. The descriptors can be formed for the complex-valued boundary function using the discrete Fourier transform (DFT) (Gonzalez and Woods, 1992). The Fourier transform of z(k) is

\[ F(n) = \frac{1}{N} \sum_{k=0}^{N-1} z(k)e^{-j2\pi nk/N} \]  

for \( n = 0, 1, 2, \ldots, N-1 \)and F(n) are the transform coefficients of z(k). The translational invariance is based on the shape signature. Furthermore, the coefficients have also to be normalized to achieve invariance to rotation and scaling. The descriptors can be made
rotation invariant by ignoring the phase information and using only the magnitudes of
the transform coefficients $|F(n)|$. In the case of complex-valued boundary function, the
scale can be normalized by dividing the magnitudes of the transform coefficients by
$|F(1)|$ (Kauppinen et al., 1995).

b) **Curvature Scale Space (CSS) Descriptors**

CSS descriptors (Mokhtarian and Mackworth, 1986) are essentially the descriptors of
key local shape features. By dealing shape in scale space, not only the locations, but
also the degree of convexities (or concavities) of shape boundaries are detected. Since
curvature is a very important local measure of how fast a planar contour is turning,
therefore, curvature scale space is exploited. The CSS descriptors are obtained by first
calculating the CSS contour map, the map is a multi-scale organization of the inflection
points (or curvature zero-crossing points). To calculate the CSS contour map, curvature
is derived from the shape boundary points $(x_i, y_i) i = 1, 2... l$:

$$k_i = \frac{(\dot{x}_i \dot{y}_i - \ddot{x}_i \ddot{y}_i)}{((\dot{x}_i^2 + \dot{y}_i^2)^{3/2})}$$

where $\dot{x}_i, \ddot{x}_i, \dot{y}_i, \ddot{y}_i$ are the first and the second derivatives at location $i$
respectively. Curvature zero-crossing points are then located in the shape boundary. The
shape is then evolved into next scale by applying the Gaussian smoothing function:

$$x'_i = x_i \otimes g(i, \sigma), \quad y'_i = y_i \otimes g(i, \sigma)$$

where $\otimes$ means convolution, and $g(i, \sigma)$ is Gaussian function. As $\sigma$ increases, the
evolving shape becomes smoother and smoother. New curvature zero-crossing points
are located at the new scale. This process continues until no curvature zero-crossing
points are found. The final CSS contour map is composed of all zero-crossing points $zc$
$(i, \sigma)$, where $i$ is the location and $\sigma$ is the scale at which the $zc$ is obtained. The peaks or
the maxima of the CSS contour map (only those peaks higher than the threshold are
considered) are then extracted out and sorted in descending order as CSS descriptors to
index the shape.
c) **Zernike Moments**

Teague, (1980) has proposed the use of orthogonal moments to recover the image from moments based on the theory of orthogonal polynomials, and has introduced Zernike moments, which allow independent moment invariants to be constructed to an arbitrarily high order. The complex Zernike moments are derived from Zernike polynomials:

\[ V_{nm}(x, y) = V_{nm}(\rho \cos \theta, \rho \sin \theta) = R_{nm}(\rho) \exp(jm\theta) \]  

(3.12)

and

\[ R_{nm}(\rho) = \sum_{s=0}^{(n+|m|)/2} (-1)^s \frac{(n-s)!}{s! \left( \frac{n+|m|}{2} \right) ! \left( \frac{n-|m|}{2} - s \right)!} \rho^{n-2s} \]  

(3.13)

where \( n \) and \( m \) are subject to \( n|m| = \text{even}, |m| \leq n \). Zernike polynomials are a complete set of complex-valued function orthogonal over the unit disk, i.e., \( x^2 + y^2 = 1 \). Then the complex Zernike moments of order \( n \) with repetition \( m \) are defined as:

\[ A_{nm} = \frac{1}{\pi} \sum_{x} \sum_{y} f(x, y)V_{nm}^*(x, y), x^2 + y^2 \leq 1 \]  

(3.14)

The theory of Zernike moments is similar to that of Fourier transform, to expand a signal into series of orthogonal basis.

The general shape of the object is represented by the low frequency coefficients, where as high frequency coefficients represent the fine details of the object shape. A common approach to shape representation is to use a subset of the low frequency coefficients as a shape descriptor. This way the shape can be effectively presented using a relatively short feature vector the feature vector is formed using Contour Fourier method (Kauppinen et al., 1995), which applies the complex coordinate function. In this method the descriptors are taken from positive and negative frequency axis. The feature vector for this method is

\[ x = \begin{bmatrix} F_{-L/2} \cdots F_{-1} \cdots F_{1} \cdots F_{L/2} \end{bmatrix}^T \]  

(3.15)

In which \( L \) is a constant value that defines the dimensionality of the feature vector.
d) Chain Code

Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. Usually this representation is based on 4 or 8 connectivity of the segments. Some examples of chain codes can be found in Gonzalez and Woods (2002).

Producing the chain code using all the pixel pairs in the image would lead to two major disadvantages. First the resulting chain code would be long, and secondly any disturbances along the boundary could cause changes in the code. Therefore, a common approach to avoid these problems is to resample the boundary by selecting larger grid spacing. The chain code of a boundary depends on the starting point. However, the code can be normalized easily using the following procedure. The chain code is treated as a circular sequence of direction numbers and the starting point is redefined so that the resulting sequence forms an integer of minimum magnitude. However, the normalization is exact, only when the boundary is invariant to rotation and scaling, Gonzalez and Woods, (2002).

3.3 Image Segmentation

Image segmentation is an essential preliminary step in most automatic image recognition and scene analysis operations. The main goal for segmentation is to partition an image into homogeneous, constituent, non-intersecting regions so that the union of two adjacent regions will not meet a homogeneity criterion. Here, homogeneity is achieved with respect to some predefined properties such as color and/or texture. Smeulders et al., (2000) have discussed the concepts of strong and weak segmentation. Strong segmentation is defined as the ideal object segmentation where each region would correspond to one single semantic object. Weak segmentation is reported, whenever strong segmentation cannot be achieved. Here, homogeneous regions are obtained, which do not necessarily describe objects. Weak segmentation may also result into two erroneous outcomes: under and over segmentation. Under segmentation means two regions with different properties have been merged. Over-segmentation means that a larger homogeneous region is partitioned into more than one segment. So far there is no segmentation method suitable for all image types such as general purpose images, aerial or satellite images and medical images. Moreover, each
method is not applicable to one particular image type. Hence, segmentation is mostly application driven and mainly depends on what the user wants to achieve. The weak segmentation scheme is utilized in most image segmentation ideas and various methods exist in the literature, applying this approach. One of the most classical approaches is the so-called pixel based segmentation. It is also known as segmentation by threshold or mode method. Different threshold detection methods are described by authors like Mardia and Hainsworth, (1998); Althouse and Chang, (1995). This image segmentation approach is mainly applied on grey level images using the image histogram representation (Ballard and Brown, 1982) and segmentation is, therefore, achieved by partitioning this histogram using some threshold values. Therefore, regions should be characterized by peaks in the histogram separated by valleys. Another approach is edge-based segmentation, such as the work presented by Iannizzotto and Vita, (2000). It usually consists of two main steps. The first one is to detect the edges. Within the image and the second one is to link those edges to continuous contours.

A significant progress has been achieved, due to many researchers, in texture image segmentation (Laws, 1980; Buf et al., 1990; Jain and Farrokhnia, 1991; Pappas, 1992; Unser, 1995; Randen and Husoy, 1999), color image segmentation (Chang et al., 1994; Comaniciu and Meer, 1997; Deng et al., 1999b; Luccheseyz and Mitray, 2001; Shih and Cheng, 2005; Tao et al., 2007; Singha and Hemachandran, 2011c) and the combination of color and texture image segmentation (Panjwani and Healey, 1995; Shafarenko et al., 1997; Belongie et al., 1998; Deng and Manjunath, 2001; Chen et al., 2005; Kato and Pong, 2006).

3.4 Similarity Measures

This section describes the similarity measures used for matching visual information and the approaches taken to improve similarity results. The similarity is determined as a distance between some extracted feature or a vector that combines these. Similarity matching is the process of approximating a solution based on the computation of a similarity function between a pair of images, and the result is a set of likely values. Exactness, however, is a precise concept. We have used different distance measuring methods such as Manhattan, Euclidean, Quadratic and Intersection distance.
During indexing and retrieval, similarity/dissimilarity between two items is expressed via distance calculation. This can be mainly achieved by using distance metrics. Thus, a metric space is a pair of \((X, d)\) where \(X\) is a set of two entities \(x\) and \(y\), and \(d\) is distance function. A metric should have the following properties:

1. **Identity** \(d(x, y) = 0\) if and only if \(x = y\)
2. **Positive definiteness** \(d(x, y) = 0\) if and only if \(x = y\)
3. **Symmetry** \(d(x, y) = d(y, x)\)
4. **Triangle inequality** \(d(x, z) \leq d(x, y) + d(y, z)\)

\[a) \textbf{Minkowski Distance (MD), Also Known As The L1 Norm:}\]

In general, the distance function \(d\) between any two points in \(n\)-dimensional space may be expressed via the equation given by Minkowski, also referred as \(L_p\)-norm. This generic equation form is defined as,

\[D_p (Q, D) = \left( \sum_{i=0}^{N-1} |Q_i - D_i|^p \right)^{\frac{1}{p}} \quad (3.16)\]

where \(N\) defines the data dimension and \(p\) determines the degree of distance, \(Q = \{Q_1, Q_2, \ldots, Q_n\}\), and \(D = \{D_1, D_2, \ldots, D_n\}\) are the query and an image from image database respectively. The most frequently used Minkowski's distances are the norms of degree one, two and \(\infty\), also called Manhattan distance or city block distance (\(L_1\)-norm), Euclidean distance (\(L_2\)-norm), and Chebyshev distance (\(L_\infty\)-norm), respectively. The \(L_1\) and \(L_2\)-norm can be directly derived from equation \(D_p\).

**Manhattan distance (MD) or (L1):**

\[D_{MD} (Q, D) = \sum_{i=0}^{N-1} |Q_i - D_i| \quad (3.17)\]

**Euclidean distance (ED) or (L2):**

\[D_{ED} (Q, D) = \left( \sum_{i=0}^{N-1} (Q_i - D_i)^2 \right)^{\frac{1}{2}} \quad (3.18)\]
Their role in CBIR is mainly calculating the similarity distance between two image items (query and database item) using their feature vectors and to rank the database items thereafter according to the similarity distances to the query item. The $L^p$-norms are commonly used due to their low complexity even for higher dimensional data. However, a drawback of those norms is that it only takes into account the computation between the same data locations (i.e. distance of histograms is calculated bin by bin). Here, possible correlations across dimensions are not taken into account during the distance computation.

**b) Quadratic Distance (QD):**

In order to address this problem, the quadratic-distance was introduced in QBIC (Niblack et al., 1994) which considers all bin distances within the overall similarity distance.

$$D_{QD} = \sqrt{\langle Q[i] - D[i] \rangle^T A \langle q[i] - D[i] \rangle}$$  \hspace{1cm} (3.19)

The distance formula considers the cross-correlation between histogram bins based on the perceptual similarity of the colors represented by the bins. The set of all cross-correlation values are represented by a matrix $A$, which is called a similarity matrix. This allows comparison between different histogram bins having a certain degree of cross similarity. Moreover, the quadratic distance can be applied to color sets (Manjunath et al., 2001) as well.

Smith and Chang (1996b) have used HSV color space and a $(i, j)^{th}$ element in the similarity matrix $A$ is given by:

$$A(i, j) = 1 - \frac{1}{\sqrt{5}} \sqrt{((v_i - v_j)^2 + (s_i \cos h_i - s_j \cos h_j)^2 + (s_i \sin h_i - s_j \sin h_j)^2)}$$  \hspace{1cm} (3.20)

**c) Intersection Distance (ID):**

Swain and Ballard (1991) proposed histogram intersection for color image retrieval. The intersection of histograms was originally defined as:

$$D_{ID}(Q,D) = \frac{\sum_{i=1}^{i=n} \min \left[ Q[i], D[i] \right]}{\left[ D[i] \right]}$$  \hspace{1cm} (3.21)
Smith and Chang (1996a) extended the idea, by modifying the denominator of the original definition, to include the case when the cardinalities of the two histograms are different.

\[
D_{id_s}(Q, D) = \frac{\sum_{i=1}^{m} \min \left[ Q[i], D[i] \right]}{\min \left[ |Q[i]|, |D[i]| \right]} \tag{3.22}
\]

Where \(|Q|\) and \(|D|\) gives the magnitude of each histogram, which is equal to the number of samples.

### 3.5 Performance Evaluation

To measure the retrieval performance of a conducted query, the query results need to be evaluated. Generally, multimedia databases are divided into predefined categories (classes), which are generated by subjective classification and usually represent similar semantic content. These categories are also used for evaluating retrieval performance where all category items are the so-called ground-truth data, which are considered relevant for a particular category. After obtaining the results for a query, there are different methods for evaluating them. Automatic evaluation can be done without human interaction where items are considered relevant for the retrieval if they belong to the same predefined category as the query and irrelevant otherwise. In manual evaluation items are judged relevant if they present similar (semantic) content to a human observer as the query. Here, personal assessment plays an important role because what might be relevant to one person may be irrelevant to another.

Performance measures provide numeric evaluation of retrieval results. Normally, the automatic evaluation is preferred due to its fast calculation. However, its performance will depend on the subjective category classification in the database. On the one hand, if a database contents coarse category classification then it might become difficult to retrieve those semantic category elements by low-level features and eventually that will result to a poor performance. The reason for this is that such categories will contain similar semantic objects represented by different low-level features. On the other hand, if the category content is well-separated by its low-level features then a meaningful evaluation for descriptors may be provided. However, it can be noted that different visual and aural features may have different ground truth data in the same database.
Moreover, there might be also a difference due to semantics where items may be considered the same category but have nothing in common feature-wise. Therefore, an appropriate category classification is a necessity for evaluating different visual features. During retrieval, conducting only one query per class or feature does not give enough statistics for the evaluation of the retrieval performance by numeric measurements. Hence, multiple queries are carried out to obtain a healthy statistics about the performance using some numeric methodologies, as detailed next.

The performance of retrieval system is measured using the standard procedure in terms of the precision and recall values (Banerjee et al., 2009). Recall measures the ability of the system to retrieve all the models that are relevant, while precision measures the ability of the system to retrieve only the models that are relevant. They are defined as:

\[
\text{Precision (P)} = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}} \quad (3.23)
\]

\[
\text{Recall (R)} = \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images}} \quad (3.24)
\]

The number of relevant items retrieved is the number of the returned images that are similar to the query image in this case. The number of relevant items in the collection is the number of images that are in the same particular category with the query image. The total number of items retrieved is the number of images that are returned by the search engine.

The weighted average precision value within the k retrieved images are computed as

\[
AP = \frac{1}{100} \sum_{k=1}^{100} \frac{n_k}{100} \quad (3.25)
\]

Where \( k = 1, \ldots, 100 \) and \( n_k \) is the number of matches within the first k retrieved images.

The total average precision is defined as:

\[
TAP = \frac{1}{10} \sum_{q=1}^{10} AP \quad (3.26)
\]

Here q represent category of an image.