We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

1

Image Acquisition, Storage and Retrieval

Hui Ding, Wei Pan and Yong Guan
Capital Normal University
China

1. Introduction

In many areas of commerce, government, academia, hospitals, and homes, large collections of digital images are being created. However, in order to make use of it, the data should be organized for efficient searching and retrieval. An image retrieval system is a computer system for browsing, searching and retrieving images from a large database of digital images. Due to diversity in content and increase in the size of the image collections, annotation became both ambiguous and laborious. With this, the focus shifted to Content Based Image Retrieval (CBIR), in which images are indexed according to their visual content.
The chapter will provide mathematical foundations and practical techniques for digital manipulation of images; image acquisition; image storage and image retrieval.
Image databases have particular requirements and characteristics, the most important of which will be outlined in this Section.

1.1 The description of CBIR

Content Based Image Retrieval or CBIR is the retrieval of images based on visual features such as colour, texture and shape (Michael et al., 2006). Reasons for its development are that in many large image databases, traditional methods of image indexing have proven to be insufficient, laborious, and extremely time consuming. These old methods of image indexing, ranging from storing an image in the database and associating it with a keyword or number, to associating it with a categorized description, have become obsolete. This is not CBIR. In CBIR, each image that is stored in the database has its features extracted and compared to the features of the query image. It involves two steps (Khalid et al., 2006):

- Feature Extraction: The first step in the process is extracting image features to a distinguishable extent.
- Matching: The second step involves matching these features to yield a result that is visually similar.

Many image retrieval systems can be conceptually described by the framework depicted in Fig. 1.
The user interface typically consists of a query formulation part and a result presentation part. Specification of which images to retrieve from the database can be done in many ways. One way is to browse through the database one by one. Another way is to specify the image in terms of keywords, or in terms of image features that are extracted from the image, such as a color histogram. Yet another way is to provide an image or sketch from which features

Source: Image Processing, Book edited by: Yung-Sheng Chen,
ISBN 978-953-307-026-1, pp. 572, December 2009, INTECH, Croatia, downloaded from SCIYO.COM

of the same type must be extracted as for the database images, in order to match these features. A nice taxonomy of interaction models is given in (Vendrig, 1997). Relevance feedback is about providing positive or negative feedback about the retrieval result, so that the systems can refine the search.

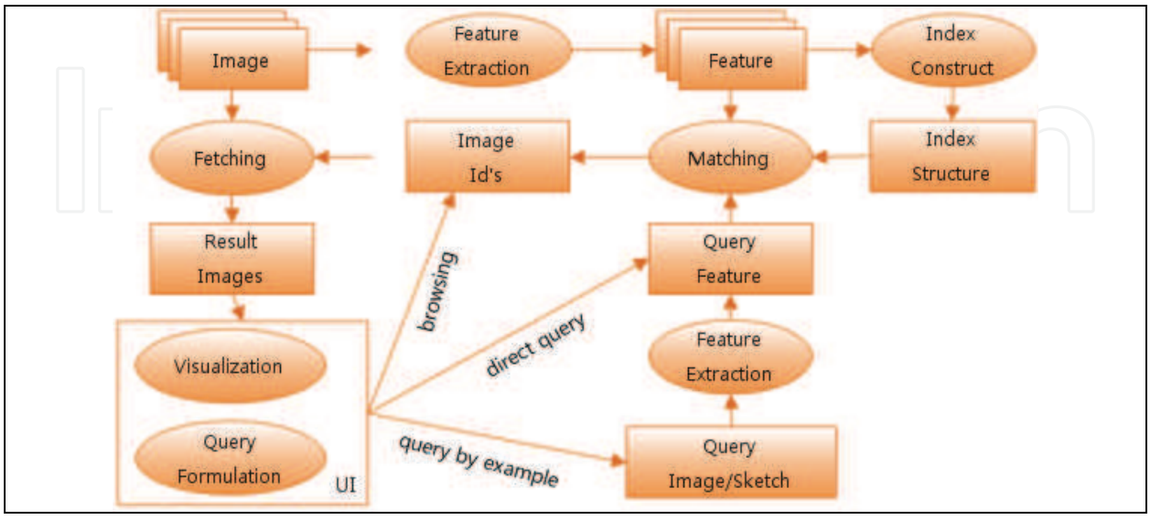

Fig. 1. Content-based image retrieval framework

1.2 A short overview

Early reports of the performance of Content based image retrieval (CBIR) systems were often restricted simply to printing the results of one or more example queries (Flickner et al., 1995). This is easily tailored to give a positive impression, since developers can chooses queries which give good results. It is neither an objective performance measure, nor a means of comparing different systems. MIR (1996) gives a further survey. However, few standard methods exist which are used by large numbers of researchers. Many of the measures used in CBIR (such as precision, recall and their graphical representation) have long been used in IR. Several other standard IR tools have recently been imported into CBIR. In order to avoid reinventing pre-existing techniques, it seems logical to make a systematic review of evaluation methods used in IR and their suitability for CBIR.

CBIR inherited its early methodological focus from the by then already mature field of text retrieval. The primary role of the user is that of formulating a query, while the system is given the task of finding relevant matches. The spirit of the time is well captured in Gupta and Jain's classic review paper from 1997 (Gupta & Jain, 1997) in which they remark that "an information retrieval system is expected to help a user specify an expressive query to locate relevant information." By far the most commonly adopted method for specifying a query is to supply an example image (known as query by example or QBE), but other ways have been explored. Recent progress in automated image annotation, for example, reduces the problem of image retrieval to that of standard text retrieval with users merely entering search terms. Whether this makes query formulation more intuitive for the user remains to be seen. In other systems, users are able to draw rough sketches possibly by selecting and combining visual primitives (Feng et al., 2004; Jacobs et al., 1995; Smith & Chang, 1996).

Content-based image retrieval has been an active research area since the early 1990's. Many image retrieval systems both commercial and research have been built.

The best known are Query by Image Content (QBIC) (Flickner et al., 1995) and Photo-book (Rui et al., 1997) and its new version Four-Eyes. Other well-known systems are the search engine family Visual-SEEk, Meta-SEEk and Web-SEEk (Bach et al., 1996), NETRA, Multimedia Analysis and Retrieval System (MARS) (Honkela et al., 1997).
All these methods have in common that at some point users issue an explicit query, be it textual or pictorial. This division of roles between the human and the computer system as exemplified by many early CBIR systems seems warranted on the grounds that search is not only computationally expensive for large collections but also amenable to automation.
However, when one considers that humans are still far better at judging relevance, and can do so rapidly, the role of the user seems unduly curtailed. The introduction of relevance feedback into image retrieval has been an attempt to involve the user more actively and has turned the problem of learning feature weights into a supervised learning problem. Although the incorporation of relevance feedback techniques can result in substantial performance gains, such methods fail to address a number of important issues. Users may, for example, not have a well-defined information need in the first place andmay simply wish to explore the image collection. Should a concrete information need exist, users are unlikely to have a query image at their disposal to express it. Moreover, nearest neighbour search requires efficient indexing structures that do not degrade to linear complexity with a large number of dimensions (Weber et al., 1998).
As processors become increasingly powerful, and memories become increasingly cheaper, the deployment of large image databases for a variety of applications have now become realisable. Databases of art works, satellite and medical imagery have been attracting more and more users in various professional fields. Examples of CBIR applications are:

- Crime prevention: Automatic face recognition systems, used by police forces.
- Security Check: Finger print or retina scanning for access privileges.
- Medical Diagnosis: Using CBIR in a medical database of medical images to aid diagnosis by identifying similar past cases.
- Intellectual Property: Trademark image registration, where a new candidate mark is compared with existing marks to ensure no risk of confusing property ownership.

2. Techniques of image acquire

Digital image consists of discrete picture elements called pixels. Associated with each pixel is a number represented as digital number, which depicts the average radiance of relatively small area within a scene. Image capture takes us from the continuous-parameter real world in which we live to the discrete parameter, amplitude quantized domain of the digital devices that comprise an electronic imaging system.

2.1 Representations for the sampled image

Traditional image representation employs a straightforward regular sampling strategy, which facilitates most of the tasks involved. The regular structuring of the samples in a matrix is conveniently simple, having given rise to the raster display paradigm, which makes this representation especially efficient due to the tight relationship with typical hardware.
The regular sampling strategy, however, does not necessarily match the information contents of the image. If high precision is required, the global sampling resolution must be increased, often resulting in excessive sampling in some areas. Needless to say, this can

become very inefficient, especially if the fine/coarse detail ratio is low. Many image representation schemes address this problem, most notably frequency domain codifications (Penuebaker & Mitchell, 1993; Froment & Mallat, 1992), quad-tree based image models (Samet, 1984) and fractal image compression (Barnsley & Hurd, 1993).
Sampling a continuous-space image $g_c(x,y)$ yields a discretespace image:

$$g_d(m,n) = g_c(mX,nY) \tag{1}$$

where the subscripts c and d denote, respectively, continuous space and discrete space, and (X,Y) is the spacing between sample points, also called the pitch. However, it is also convenient to represent the sampling process by using the 2-D Dirac delta function $\delta(x,y)$. In particular, we have from the sifting property of the delta function that multiplication of $g_c(x,y)$ by a delta function centered at the fixed point (x_0,y_0) followed by integration will yield the sample value $g_c(x_0,y_0)$, i.e.,

$$g_c(x_0,y_0) = \iint g_c(x,y)\delta(x-x_0,y-y_0)dxdy \tag{2}$$

Provided $g_c(x,y)$ is continuous at (x_0,y_0). It follows that:

$$g_c(x,y)\delta(x-x_0,y-y_0) \equiv g_c(x_0,y_0)\delta(x-x_0,y-y_0) \tag{3}$$

that is, multiplication of an impulse centered at (x_0,y_0) by the continuous-space image $g_c(x,y)$ is equivalent to multiplication of the impulse by the constant $g_c(x_0,y_0)$. It will also be useful to note from the sifting property that:

$$g_c(x,y) * \delta(x-x_0,y-y_0) = g_c(x-x_0,y-y_0) \tag{4}$$

That is, convolution of a continuous-space function with an impulse located at (x_0,y_0) shifts the function to (x_0,y_0).
To get all the samples of the image, we define the comb function:

$$comb_{X,Y}(x,y) = \sum_m \sum_n \delta(x-mX,y-nY) \tag{5}$$

Then we define the continuous-parameter sampled image, denoted with the subscript s, as

$$\begin{aligned} g_s(x,y) &= g_c(x,y)comb_{X,Y}(x,y) \\ &= \sum_m \sum_n g_d(x,y)\delta(x-mX,y-nY) \end{aligned} \tag{6}$$

We see from Eq. (6) that the continuous- and discrete-space representations for the sampled image contain the same information about its sample values. In the sequel, we shall only use the subscripts c and d when necessary to provide additional clarity. In general, we can distinguish between functions that are continuous space and those that are discrete space on the basis of their arguments. We will usually denote continuous-space independent variables by (x,y) and discrete-space independent variables by (m,n).

2.2 General model for the image capture process

Despite the diversity of technologies and architectures for image capture devices, it is possible to cast the sampling process for all of these systems within a common framework.

Since feature points are commonly used for alignment between successive images, it is important to be aware of the image blur introduced by resampling. This manifests itself and is conveniently analysed in the frequency

$$Y(z) = \frac{1}{N}\sum_{m=0}^{N-1} X(W_N^m z^{1/N}) \tag{10}$$

Fig. 3. Illustration of $ALLAS_2$ in frequency domain

2.3.2 Upsampling and Interpolation

The process of increasing the sampling rate is called interpolation. Interpolation is upsampling followed by appropriate filtering. $y(n)$ obtained by interpolating $x(n)$, is generally represented as:

Fig. 4. Upsampling by the factor N

$$y(n) = STRETCH_N(x) \triangleq x(Nn), \quad n \in Z \tag{11}$$

Fig. 4 shows the graphical symbol for a digital upsampler by the factor N . To upsample by the integer factor $N-1$, we simply insert zeros between $x(n)$ and $x(n+1)$ for all n . In other words, the upsampler implements the stretch operator defined as扦

$$y(n) = STRETCH_{N,n}(x) \triangleq \begin{cases} x(n/N), & \frac{n}{N} \\ 0, & otherwize \end{cases} \tag{12}$$

In the frequency domain, we have, by the stretch (repeat) theorem for DTFTs:

$$Y(z) = REPEAT_{N,n}(X) \triangleq X(z^N), \quad z \in C \tag{13}$$

Plugging in $z = e^{jw}$, we see that the spectrum on $[-\pi, \pi)$ contracts by the factor N , and N images appear around the unit circle. For $N = 2$, this is depicted in Fig. 5.

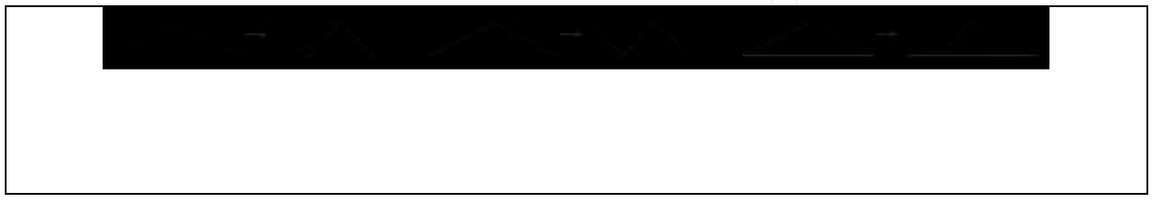

Fig. 5. Illustration of $ALLAS_2$ in frequency domain

For example, the down sampling procedure keeps the scaling parameter constant (n=1/2) throughout successive wavelet transforms so that it benefits for simple computer implementation. In the case of an image, the filtering is implemented in a separable way by

filtering the lines and columns. The progressing of wavelet decompose has shown in Fig. 6. (Ding et al., 2008)

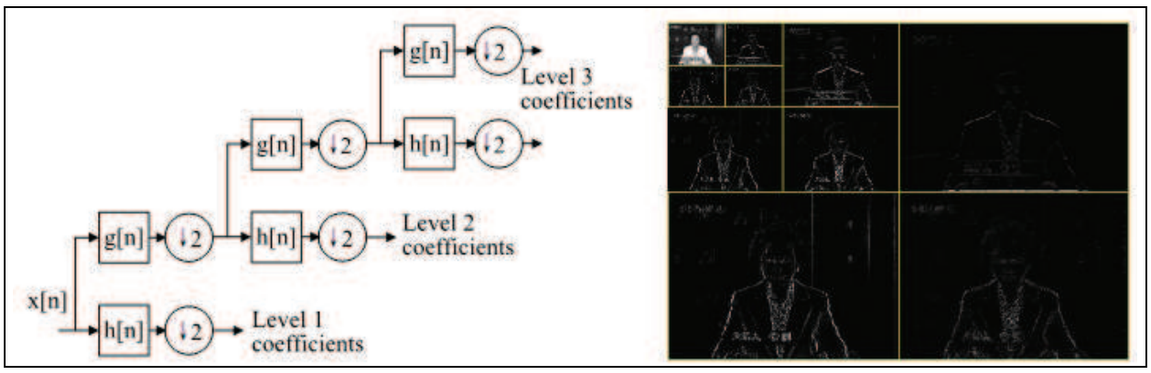

Fig. 6. 2-D wavelet decomposition. (a) : 3 level filter bank, (b): the example of wavelet decomposition

2.4 Basic enhancement and restoration techniques

The Moving Picture Experts Group (MPEG) is a working group of ISO/IEC in charge of the development of international standards for compression, decompression, processing, and coded representation of moving pictures, audio and their combination.

The process of image acquisition frequently leads (inadvertently) to image degradation. Due to mechanical problems, out-of-focus blur, motion, inappropriate illumination, and noise the quality of the digitized image can be inferior to the original. The goal of enhancement is - starting from a recorded image $c[m,n]$ to produce the most visually pleasing image $\hat{a}[m,n]$. The goal of enhancement is beauty; the goal of restoration is truth.

The measure of success in restoration is usually an error measure between the original $a[m,n]$ and the estimate $\hat{a}[m,n]: E\{\hat{a}[m,n], a[m,n]\}$. No mathematical error function is known that corresponds to human perceptual assessment of error. The mean-square error function is commonly used because:

- It is easy to compute;
- It is differentiable implying that a minimum can be sought;
- It corresponds to "signal energy" in the total error;
- It has nice properties vis à vis Parseva's theorem.

The mean-square error is defined by:

$$E\{\hat{a}, a\} = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|\hat{a}[m,n] - a[m,n]\right|^2 \tag{14}$$

In some techniques an error measure will not be necessary; in others it will be essential for evaluation and comparative purposes.

Image restoration and enhancement techniques offer a powerful tool to extract information on the small-scale structure stored in the space- and ground-based solar observations. We will describe several deconvolution techniques that can be used to improve the resolution in the images. These include techniques that can be applied when the Point Spread Functions (PSFs) are well known, as well as techniques that allow both the high resolution information, and the degrading PSF to be recovered from a single high signal-to-noise

image. I will also discuss several algorithms used to enhance low-contrast small-scale structures in the solar atmosphere, particularly when they are embedded in large bright structures, or located at or above the solar limb. Although strictly speaking these methods do not improve the resolution in the images, the enhancement of the fine structures allows detailed study of their spatial characteristics and temporal variability. Finally, I will demonstrate the potential of image post-processing for probing the fine scale and temporal variability of the solar atmosphere, by highlighting some recent examples resulting from the application of these techniques to a sample of Solar observations from the ground and from space.

3. Image storage and database

With increased computing power and electronic storage capacity, the potential for large digital video libraries is growing rapidly. In the analysis of digital video, compression schemes offer increased storage capacity and statistical image characteristics, such as filtering coefficients and motion compensation data. 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.

3.1 Statistical features

In pattern recognition and in image processing feature extraction is a special form of dimensionality reduction. Features that are extracted from image or video sequence without regard to content are described as statistical features. These include parameters derived from such algorithms as image difference and camera motion. Certain features may be extracted from image or video without regard to content. These features include such analytical features as scene changes, motion flow and video structure in the image domain, and sound discrimination in the audio domain.

3.1.1 Gaussian statistics

To understand the role of statistics in image segmentation, let us examine some preliminary functions that operate on images. Given an image f_0 that is observed over the lattice Ω, suppose that $\Omega_1 \subseteq \Omega_2$ and f_1 is a restriction of f_0 to only those pixels that belong to Ω_1. Then, one can define a variety of statistics that capture the spatial continuity of the pixels that comprise f_1.

$$T_{f_1}(p,q) = \sum_{(m,n)\in\Omega_1} \left[f_1(m,n) - f_1(m+p,n+q) \right]^2 \tag{15}$$

where $(p,q) \in [(0,1),(1,0),(1,1),(1,-1),\ldots]$, measures the amount of variability in the pixels that comprise f_1 along the $(p,q)th$ direction. For a certain f_1, if $T_{f_1}(0,1)$ is very small, for example, then that implies that f_1 has a little or no variability along the $(0,1)th$ (i.e., horizontal) direction. Computation of this statistic is straightforward, as it is merely a quadratic operation on the difference between intensity values of adjacent (neighboring) pixels. $T_{f_1}(p,q)$ and minor variation thereof is referred to as the Gaussian statistic and is widely used in statistical methods for segmentation of gray-tone images; see [6,7].

3.1.2 Fourier statistics

$$F_{f_1}(\alpha,\beta)=\sum_{(m,n)\in\Omega_1}\left[f_1(m,n)e^{-\sqrt{-1}(m\alpha+n\beta)}\right]\times\left[f_1(m,n)e^{\sqrt{-1}(m\alpha+n\beta)}\right] \tag{16}$$

where $(\alpha,\beta)\in[-\pi,\pi]^2$, measures the amount of energy in frequency bin (α,β) that the pixels that comprise f_1 possess. For acertain f_1, if $F_{f_1}(0,20\pi/N)$ has a large value, for example, then that implies that f_1 has a significant cyclical variation of the $(0,20\pi/N)$ (i.e., horizontally every 10 pixels) frequency. Computation of this statistic is more complicated that the Gaussian one. The use of fast Fourier transform algorithms, however, can significantly reduce the associated burden. $F_{f_1}(\alpha,\beta)$, called the period gram statistic, is also used in statistical methods for segmentation of textured images; see [8,91.

3.1.3 Covariance statistics

$$K_{f_1}=\sum_{(m,n)\in\Omega_1}\left(f_1(m,n)-\mu f_1\right)^T\left(f_1(m,n)-\mu f_1\right) \tag{17}$$

where $\mu f_1=\sum_{(m,n)\in\Omega_1}f_1(m,n)$, measures the correlation between the various components that comprise each pixel of f_1. If K_{f_1} is a 3×3 matrix and $K_{f_1}(1,2)$ has large value, for example, then that means that components 1 and 2 (could be the red and green channels) of the pixels that make up f_1 are highly correlated. Computation of this statistic is very time consuming, even more so than the Fourier one, and there are no known methods to alleviate this burden. K_{f_1} is called the covariance matrix of f_1, and this too has played a substantial role in statistical methods for segmentation of color images; see [10,111.
Computation of image statistics of the type that motioned before tremendously facilitates the task of image segmentation.

3.2 Compressed domain feature

Processing video data is problematic due to the high data rates involved. Television quality video requires approximately 100 GBytes for each hour, or about 27 MBytes for each second. Such data sizes and rates severely stress storage systems and networks and make even the most trivial real-time processing impossible without special purpose hardware. Consequently, most video data is stored in a compressed format.

3.2.1 JPEG Image

The name "JPEG" stands for Joint Photographic Experts Group, the name of the committee that created the standard. The JPEG compression algorithm is at its best on photographs and paintings of realistic scenes with smooth variations of tone and color.
Because the feature image of a raw image is composed of the mean value of each 8×8 block, the mean value of each block in the JPEG image is then directly extracted from its DC coefficient as the feature. The result can be easily inferred as follows:

$$\begin{aligned}J(0,0)&=\frac{c(0)c(0)}{4}\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\left(\frac{(2x+1)\times c(0)\times\pi}{16}\right)\cos\left(\frac{(2y+1)\times c(0)\times\pi}{16}\right)\\&=\frac{1}{16}\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)=4\times\frac{1}{64}\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)=4\times M\end{aligned} \tag{18}$$

where $J(0,0)$ and M are the DC coefficient and mean value of the corresponding block. For the reason that a level shifting by –128 gray levels in the JPEG encoding, the real mean value of the block is $\left[\frac{1}{4}J(0,0)+128\right]$. The real mean values of all blocks are assigned to be the pixel values of the feature image. The size of the feature image is still 1/64 of original JPEG image size because the DCT block size is 8×8.

3.2.2 Wavelet-compressed Images

For a wavelet-compressed image, feature image is extracted from the low-low band of the wavelet-compressed. If the one-level wavelet decomposition is used in the wavelet-compressed image, the low-low band subimage will approximate to the scaled original image. Thus, the mean value of each 4×4 block in the low-low band subimage is assigned to be the pixel value of the feature image. The pixel value of the feature image is:

$$WI_{x,y}=\frac{1}{16}\sum_{j=0}^{3}\sum_{i=0}^{3}LL_{4x+i,4y+j} \tag{19}$$

where $WI_{x,y}$ is the pixel value of feature image with coordinate (x,y), and $LL_{x,y}$ is the pixel value of low-low 8×8 band image with coordinate (x,y). The size of feature image here is 1/64 of the original wavelet-compressed image size. If the wavelet-compressed image is compressed by three-level wavelet decomposition, then the image should be reconstructed back to the one-level wavelet decomposition first.

The feature images will be the same if they are extracted from the raw image and the JPEG image of the same image. Moreover, the mean squared error (MSE) between feature images generated from the raw image and from the wavelet-compressed image is quite small.

3.3 Image content descriptor

"Content-based" means that the search will analyze the actual contents of the image. The term 'content' in this context might refer to colors, shapes, textures, or any other information that can be derived from the frame image itself.

- *Color* represents the distribution of colors within the entire image. This distribution includes the amounts of each color.
- *Texture* represents the low-level patterns and textures within the image, such as graininess or smoothness. Unlike shape, texture is very sensitive to features that appear with great frequency in the image.
- *Shape* represents the shapes that appear in the image, as determined by color-based segmentation techniques. A shape is characterized by a region of uniform color.

3.2.1 Color

Color reflects the distribution of colors within the entire frame image. A color space is a mathematical representation of a set of colors. The three most popular color models are RGB (used in computer graphics); YIQ, YUV or YCbCr (used in video systems); and CMYK (used in color printing). However, none of these color spaces are directly related to the intuitive notions of hue, saturation, and brightness. This resulted in the temporary pursuit of other models, such as HIS and HSV, to simplify programming, processing, and end-user manipulation.

- RGB Color Space

The red, green, and blue (RGB) color space is widely used throughout computer graphics. Red, green, and blue are three primary additive colors (individual components are added together to form a desired color) and are represented by a three-dimensional, Cartesian coordinate system. The indicated diagonal of the cube, with equal amounts of each primary component, represents various gray levels. Table 1 contains the RGB values for 100% amplitude, 100% saturated color bars, a common video test signal.

	Nominal Range	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
R	0 to 255	255	255	0	0	255	255	0	0
G	0 to 255	255	255	255	255	0	0	0	0
B	0 to 255	255	0	255	0	255	0	255	0

Table 1. 100% RGB Color Bars

The RGB color space is the most prevalent choice for computer graphics because color displays use red, green, and blue to create the desired color. However, RGB is not very efficient when dealing with "real-world" images. All three RGB components need to be of equal band-width to generate any color within the RGB color cube. The result of this is a frame buffer that has the same pixel depth and display resolution for each RGB component. Also, processing an image in the RGB color space is usually not the most efficient method. For these and other reasons, many video standards use luma and two color dif ference signals. The most common are the YUV, YIQ, and YCbCr color spaces. Although all are related, there are some differences.

- YCbCr Color Space

The YCbCr color space was developed as part of ITU-R BT.601 during the development of a world-wide digital component video standard. YCbCr is a scaled and offset version of the YUV color space. Y is defined to have a nominal 8-bit range of 16–235; Cb and Cr are defined to have a nominal range of 16–240. There are several YCbCr sampling formats, such as 4:4:4, 4:2:2, 4:1:1, and 4:2:0.

RGB - YCbCr Equations: SDTV

The basic equations to convert between 8-bit digital R´G´B´ data with a 16–235 nominal range and YCbCr are:

$$\begin{aligned} Y_{601} &= 0.2999R' + 0.587G' + 0.114B' \\ Cb &= -0.172R' - 0.399G' + 0.511B' + 128 \\ Cr &= 0.511R' - 0.428G' - 0.083B' + 128 \end{aligned} \tag{20}$$

$$\begin{aligned} R' &= Y_{601} + 1.371(Cr - 128) \\ G' &= Y_{601} - 0.698(Cr - 128) - 0.336(Cb - 128) \\ B' &= Y_{601} + 1.732(Cb - 128) \end{aligned}$$

When performing YCbCr to R´G´B´ conversion, the resulting R´G´B´ values have a nominal range of 16–235, with possible occasional excursions into the 0–15 and 236–255 values. This is due to Y and CbCr occasionally going outside the 16–235 and 16–240 ranges, respectively, due to video processing and noise. Note that 8-bit YCbCr and R´G´B´ data should be saturated at the 0 and 255 levels to avoid underflow and overflow wrap-around problems.

Table 2 lists the YCbCr values for 75% amplitude, 100% saturated color bars, a common video test signal.

	Nominal Range	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
SDTV									
Y	16 to 235	180	162	131	112	84	65	35	16
Cb	16 to 240	128	44	156	72	184	100	212	128
Cr	16 to 240	128	142	44	58	198	212	114	128
HDTV									
Y	16 to 235	180	168	145	133	63	51	28	16
Cb	16 to 240	128	44	147	63	193	109	212	128
Cr	16 to 240	128	136	44	52	204	212	120	128

Table 2. 75% YCbCr Color Bars.

RGB - YCbCr Equations: HDTV

The basic equations to convert between 8-bit digital R´G´B´ data with a 16–235 nominal range and YCbCr are:

$$\begin{aligned} Y_{709} &= 0.213R' + 0.751G' + 0.072B' \\ Cb &= -0.117R' - 0.394G' + 0.511B' + 128 \\ Cr &= 0.511R' - 0.464G' - 0.047B' + 128 \end{aligned} \tag{21}$$

$$\begin{aligned} R' &= Y_{709} + 1.540(Cr - 128) \\ G' &= Y_{709} - 0.459(Cr - 128) - 0.183(Cb - 128) \\ B' &= Y_{709} + 1.816(Cb - 128) \end{aligned}$$

When performing YCbCr to R´G´B´ conversion, the resulting R´G´B´ values have a nominal range of 16–235, with possible occasional excursions into the 0–15 and 236–255 values. This is due to Y and CbCr occasionally going outside the 16–235 and 16–240 ranges, respectively, due to video processing and noise. Note that 8-bit YCbCr and R´G´B´ data should be saturated at the 0 and 255 levels to avoid underflow and overflow wrap-around problems.

Table 2 lists the YCbCr values for 75% amplitude, 100% saturated color bars, a common video test signal.

- HSI, HLS, and HSV Color Spaces

The HSI (hue, saturation, intensity) and HSV (hue, saturation, value) color spaces were developed to be more "intuitive" in manipulating color and were designed to approximate the way humans perceive and interpret color. They were developed when colors had to be specified manually, and are rarely used now that users can select colors visually or specify Pantone colors. These color spaces are discussed for "historic" interest. HLS (hue, lightness, saturation) is similar to HSI; the term lightness is used rather than intensity. The difference between HSI and HSV is the computation of the brightness component (I or V), which determines the distribution and dynamic range of both the brightness (I or V) and saturation(S). The HSI color space is best for traditional image processing functions such as convolution, equalization, histograms, and so on, which operate by manipulation of the brightness values since I is equally dependent on R, G, and B. The HSV color space is

preferred for manipulation of hue and saturation (to shift colors or adjust the amount of color) since it yields a greater dynamic range of saturation.

3.2.2 Texture

Texture reflects the texture of the entire image. Texture is most useful for full images of textures, such as catalogs of wood grains, marble, sand, or stones. A variety of techniques have been developed for measuring texture similarity. Most techniques rely on comparing values of what are known as second-order statistics calculated from query and stored images (John et al.). These methods calculate measures of image texture such as the degree of contrast, coarseness, directionality and regularity (Tamur et al., 1976; Niblace et al., 1993); or periodicity, directionality and randomness (Liu & Picard, 1996). Alternative methods of texture analysis for image retrieval include the use of Gabor filters (Manjunath & Ma, 1996) and fractals. Gabor filter (or Gabor wavelet) is widely adopted to extract texture features from the images for image retrieval, and has been shown to be very efficient. Manjunath and Ma have shown that image retrieval using Gabor features outperforms that using pyramid-structured wavelet transform (PWT) features, tree-structured wavelet transform (TWT) features and multiresolution simultaneous autoregressive model (MR-SAR) features.
Haralick (Haralick, 1979) and Van Gool (Gool et al., 1985) divide the techniques for texture description into two main categories: statistical and structural. Most natural textures can not be described by any structural placement rule, therefore the statistical methods are usually the methods of choice. One possible approach to reveal many of the statistical texture properties is by modelling the texture as an autoregressive (AR) stochastic process, using least squares parameter estimation. Letting s and r be coordinates in the 2-D coordinate system, a general causal or non-causal auto-regressive model may be written:

$$y(s) = \sum_{r \in N} \theta_r y(s - r) + e(s) \tag{22}$$

Where $y(s)$ is the image, θ_r are the model parameters, $e(s)$ is the prediction error process, and N is a neighbour set. The usefulness of this modelling is demonstrated with experiments showing that it is possible to create synthetic textures with visual properties similar to natural textures.

3.2.3 Shape

Shape represents the shapes that appear in the image. Shapes are determined by identifying regions of uniform color. In the absence of color information or in the presence of images with similar colors, it becomes imperative to use additional image attributes for an efficient retrieval. Shape is useful to capture objects such as horizon lines in landscapes, rectangular shapes in buildings, and organic shapes such as trees. Shape is very useful for querying on simple shapes (like circles, polygons, or diagonal lines) especially when the query image is drawn by hand. Incorporating rotation invariance in shape matching generally increases the computational requirements.

4. Image indexing and retrieval

Content-based indexing and retrieval based on information contained in the pixel data of images is expected to have a great impact on image databases. The ideal CBIR system from a user perspective would involve what is referred to as semantic retrieval.

4.1 Feature-based retrieval

Object segmentation and tracking is a key component for new generation of digital video representation, transmission and manipulations. The schema provides a general framework for video object extraction, indexing, and classification. By video objects, here we refer to objects of interest including salient low-level image regions (uniform color/texture regions), moving foreground objects, and group of primitive objects satisfying spatio-temporal constraints (e.g., different regions of a car or a person). Automatic extraction of video objects at different levels can be used to generate a library of video data units, from which various functionalities can be developed. For example, video objects can be searched according to their visual features, including spatio-temporal attributes. High-level semantic concepts can be associated with groups of low-level objects through the use of domain knowledge or user interaction.

As mentioned above, in general, it is hard to track a meaningful object (e.g., a person) due to its dynamic complexity and ambiguity over space and time. Objects usually do not correspond to simple partitions based on single features like color or motion. Furthermore, definition of high-level objects tends to be domain dependent. On the other hand, objects can usually be divided into several spatial homogeneous regions according to image features. These features are relatively stable for each region over time. For example, color is a good candidate for low-level region tracking. It does not change significantly under varying image conditions, such as change in orientation, shift of view, partial occlusion or change of shape. Some texture features like coarseness and contrast also have nice invariance properties. Thus, homogenous color or texture regions are suitable candidates for primitive region segmentation. Further grouping of objects and semantic abstraction can be developed based on these basic feature regions and their spatio-temporal relationship. Based on these observations, we proposed the following model for video object tracking and indexing (Fig. 7).

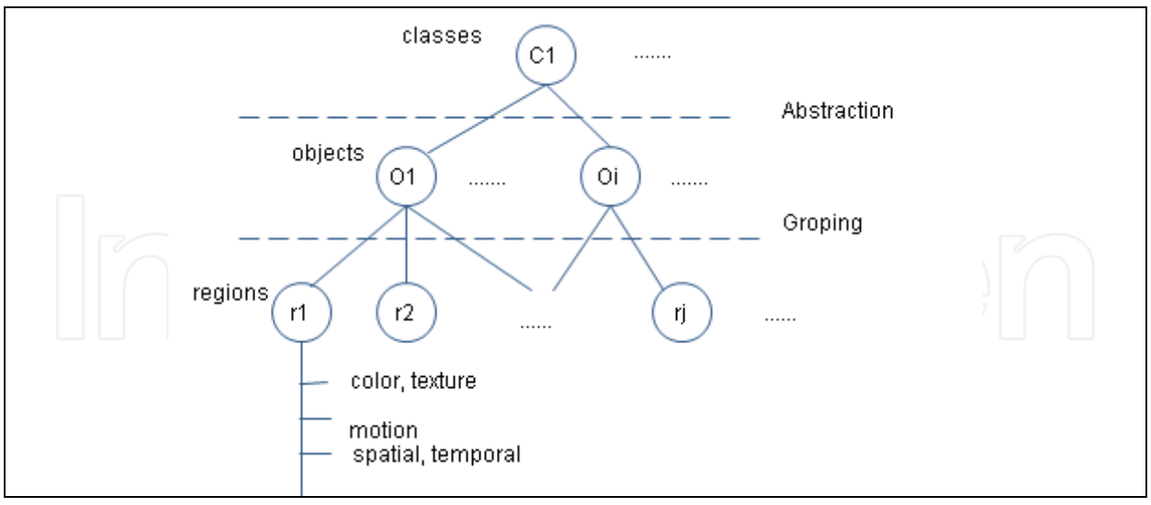

Fig. 7. Hierarchical representation of video objects

At the bottom level are primitive regions segmented according to color, texture, or motion measures. As these regions are tracked over time, temporal attributes such as trajectory, motion pattern, and life span can be obtained. The top level includes links to conceptual abstraction of video objects. For example, a group of video objects may be classified to

moving human figure by identifying color regions (skin tone), spatial relationships (geometrical symmetry in the human models), and motion pattern of component regions. We propose the above hierarchical video object schema for content-based video indexing. One challenging issue here is to maximize the extent of useful information obtained from automatic image analysis tasks. A library of low-level regions and mid-level video objects can be constructed to be used in high-level semantic concept mapping. This general schema can be adapted to different specific domains efficiently and achieve higher performance.

4.2 Content-based retrieval

In this section, we will construct such a signature by using semantic information, namely information about the appearance of faces of distinct individuals. We will not concern ourselves with the extraction of face-related information, since ample work has been performed on the subject. Instead we will try to solve the problems of consistency and robustness with regards to face-based indexing, to represent face information with minimal redundancy, and also to find a fast (logarithmic-time) search method. All works on face-related information for video indexing until now have focused on the extraction of the face-related information and not on its organization and efficient indexing. In effect, they are works on face recognition with a view to application on indexing.

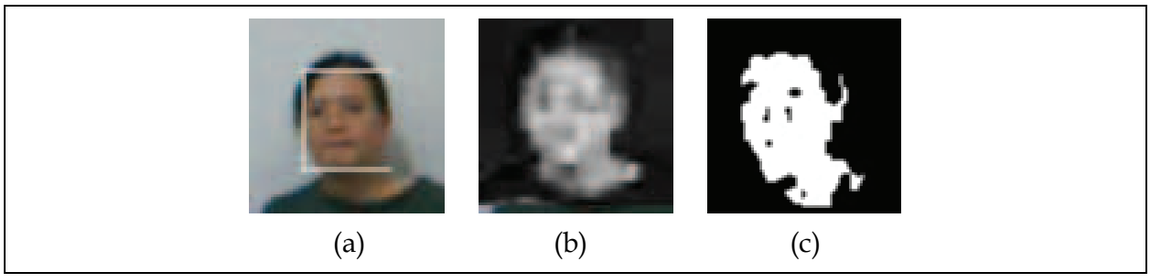

Fig. 8. Results of face detection: (a) the capture frame image; (b) result of similarity; (c) the binary result

The research on CBVIR has already a history of more than a dozen years. It has been started by using low-level features such as color, texture, shape, structure and space relationship, as well as (global and local) motion to represent the information content. Research on feature-based visual information retrieval has made quite a bit but limited success. Due to the considerable difference between the users' concerts on the semantic meaning and the appearances described by the above low-level features, the problem of semantic gap arises. One has to shift the research toward some high levels, and especially the semantic level. So, semantic-based visual information retrieval (CBVIR) begins in few years' ago and soon becomes a notable theme of CBVIR.

4.3 Semantic-based retrieval

How to bridge the gap between semantic meaning and perceptual feeling, which also exists between man and computer, has attracted much attention. Many efforts have been converged to SBVIR in recent years, though it is still in its commencement. As a consequence, there is a considerable requirement for books like this one, which attempts to make a summary of the past progresses and to bring together a broad selection of the latest

results from researchers involved in state-of-the-art work on semantic-based visual information retrieval.
Several types of semantic gaps can be identified, showed in Fig. 8.:

- The semantic gap between different data sources - structured or unstructured
- The semantic gap between the operational data and the human interpretation of this data
- The semantic gap between people communicating about a certain information concept.

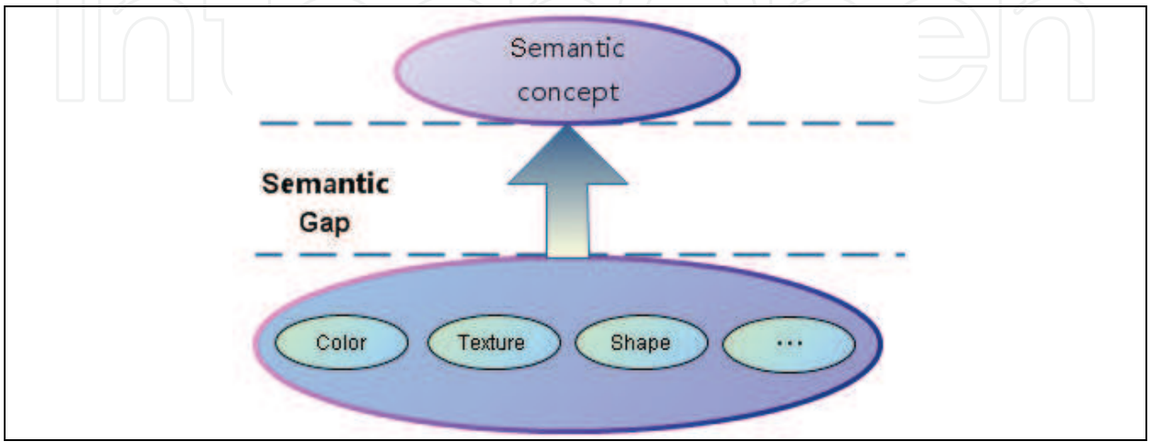

Fig. 8. Semantic Gap

Several applications aim to detect and solve different types of semantic gaps. They rage from search engines to automatic categorizers, from ETL systems to natural language interfaces, special functionality includes dashboards and text mining.

4.4 Performance evaluation

Performance evaluation is a necessary and benefical process, which provides annual feedback to staff members about job effectiveness and career guidance. The performance review is intended to be a fair and balanced assessment of an employee's performance. To assist supervisors and department heads in conducting performance reviews, the HR-Knoxville Office has introduced new Performance Review forms and procedures for use in Knoxville.
The Performance Review Summary Form is designed to record the results of the employee's annual evaluation. During the performance review meeting with the employee, use the Performance Review Summary Form to record an overall evaluation in:

- Accomplishments
- service and relationships
- dependability
- adaptability and flexibility
- and decision making or problem solving.

5. Software realization

Digital systems that process image data generally involve a mixture of software and hardware. This section describes some of the software that is available for developing image

and video processing algorithms. Once an algorithm has been developed and is ready for operational use, it is often implemented in one of the standard compiled languages.

5.1 Matlab

Matlab support images generated by a wide range of devices, including digital cameras, frame grabbers, satellite and airborne sensors, medical imaging devices, microscopes, telescopes, and other scientific instruments.

Image Processing Toolbox™ software provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can restore noisy or degraded images, enhance images for improved intelligibility, extract features, analyze shapes and textures, and register two images. Most toolbox functions are written in the open MATLAB® language, giving you the ability to inspect the algorithms, modify the source code, and create your own custom functions.

5.2 OpenCV

OpenCV is a computer vision library originally developed by Intel. It focuses mainly on real-time image processing, as such, if it find Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself. It is free for commercial and research use under a BSD license. The library is cross-platform, and runs on Windows, Mac OS X, Linux, PSP, VCRT (Real-Time OS on Smart camera) and other embedded devices. It focuses mainly on real-time image processing, as such, if it finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself. Released under the terms of the BSD license, OpenCV is open source software.

The OpenCV library is mainly written in C, which makes it portable to some specific platforms such as Digital signal processor. But wrappers for languages such as C# and Python have been developed to encourage adoption by a wider audience.

6. Future research and conclusions

As content-based retrieval techniques of multimedia objects become more effective, we believe a similar semi-automatic annotation framework can be used for other multimedia database applications. The use of image sequences to depict motion dates back nearly two centuries. One of the earlier approaches to motion picture "display" was invented in 1834 by the mathematician William George Horner. In this chapter we have reviewed the current state of the art in automatic generation of features for images.

We present a semi-automatic annotation strategy that employs available image retrieval algorithms and relevance feedback user interfaces. The semi-automatic image annotation strategy can be embedded into the image database management system and is implicit to users during the daily use of the system. The semi-automatic annotation of the images will continue to improve as the usage of the image retrieval and feedback increases. It therefore avoids tedious manual annotation and the uncertainty of fully automatic annotation. This strategy is especially useful in a dynamic database system, in which new images are continuously being imported over time.

The problem of deriving good evaluation schemes for automatically generated semantic concept is still complex and open.

7. Acknowledgments

This work is supported by the research and application of intelligent equipment based on untouched techniques for children under 8 years old of BMSTC & Beijing Municipal Education Commission (No. 2007B06 & No. KM200810028017).

8. References

Michael Lew, et al. (2006). Content-based Multimedia Information Retrieval: State of the Art and Challenges, *ACM Transactions on Multimedia Computing, Communications, and Applications,* pp. 1-19

Khalid S.; Jerzy P. & Romuald M. (2006). *Content-Based Image Retrieval - A Survey*, Biometrics, Computer Security Systems and Artificial Intelligence Applications Springer, ISBN: 978-0-387-36232-8 (Print) 978-0-387-36503-9 (Online), pp: 31-44

J. Vendrig. *Filter image browsing: a study to image retrieval in large pictorial databases.* Master's thesis, Dept. Computer Science, University of Amsterdam, The Netherlands, http://carol.wins.uva.nl/˜vendrig/thesis/, February 1997.

Flickner M.; Sawhney H. et al. (1995). Query by image and video content: The QBIC system. *IEEE Computer*, 28(9), pp: 23-32

MIR (1996). MTR A: Evaluation frameworks for interactive multimedia retrieval applications. Esprit working group 20039., http://www.dcs.gla.ac.uk/mira/.

Gupta A. & Jain R. (1997). Visual information retrieval. *Commun ACM,* 40(5), pp: 71–79

Feng S.; Manmatha R. & Lavrenko V. (2004). Multiple Bernoulli relevance models for image and video annotation. *In: Proc int'l conf computer vision and pattern recognition.* IEEE, Piscataway, pp: 1002–1009

Jacobs C.; Finkelstein A. & Salesin D. (1995). *Fast multiresolution image querying*, Technical report, University of Washington, US

Smith J. & Chang S-F. (1996). VisualSEEk: a fully automated content-based image query system, *In: Proc ACM int'l conf multimedia (SIGMM)*. ACM, New York

Flickner M.; Sawhney H.; Niblack W.; et al. (1995). Query by image and video content: The QBIC system. IEEE Computer, pp: 23-31

Rui Y.; Huang T. S. & Mehrotra S. (1997). Content-based image retrieval with relevance feedback in MARS. *In Proc. of IEEE Int. Conf. on Image Processing '97*, pp: 815-818, Santa Barbara, California, USA

Bach J. R.; Fuller C. & Gupta A.; et al. (1996). The virage image search engine: An open framework for image management. *In Sethi I. K. and Jain R. J.; editors, Storage and Retrieval for Image and video Database IV,* vol. 2670 of Proceedings of SPIE, pp: 76-87

Honkela T.; Kaski S.; Lagus K. & Kohonen T. (1997). WEBSOM-self-organizing maps of document collections. *In Proceedings of WSOM'97, Workshop on Self-Organizing Maps,*

Espoo, Finland, pp: 310-315, Helsinki University of technology, Naural Networks Rearch Centre, Espoo, Finland

Weber R.; Schek J-J. & Blott S. (1998). Aquantitative analysis and performance study for similarity search methods in high-dimensional space. *In: Proc int'l conf very large databases*, New York, 24–27 August 1998, pp 194–205

Pennebaker W. & Mitchell J. (1993). *JPEG: Still Image Data compression Standard*, The Colour Resource, San Francisco, CA

Froment J. & Mallat S. (1992). Second generation compact image coding with wavelets. *In Chui C. K., editor, Wavelets: A tutorial in Theory and Applications*, pp: 655-678, Academic Press, San Diego, CA

Barnsley M. & Hurd L. (1993). *Fractal Image Compression*, A K Peters.

Samet H. (1984). The quadtree and related hierarchical data structures. *ACM Comp*. Surv., 16: 187-260.

Bracewell R. N. (1986). *The Fourier Transform and Its Applications*. McGraw-Hill, New York

Abdou I. E. & Schowengerdt R. A. (1982). Analysis of linear interpolation schemes for bi-level image applicatio, *IBM Journal of Research and Development*, vol. 26, pp: 667-680

Ding H.; Ding X. Q.; Wang S. J. (2008). Texture Fusion Based Wavelet Transform Applied to Video Text Enhancement, *Journal of Information and Computational Science*, pp: 2083-2090

John P. Eakins & Margaret E. Graham, *Content-based Image Retrieval: A Report to the JISC Technology Applications Program,* www.unn.ac.uk/iidr/research/cbir/report.html.

Tamura H.; Mori S. & T. Yamawaki. (1976). Texture features corresponding to visual perception, I*EEE Trans. on Systems*, Man and Cybernetics. 6(4), pp: 460-473

Niblack W. et. al. (1993). The QBIC Project: Querying Images by Content Using Color, Texture and Shape, *Proc. of the Conference Storage and Retrieval for Image and Video Databases*, SPIE vol.1908, pp: 173-187

Liu F. & Picard R W. (1996). Periodicity, directionality and randomness: Wold features for image modelling and retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), pp: 722-733

Manjunath B. S. & Ma W. Y. (1996). Texture features for browsing and retrieval of large image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (Special Issue on Digital Libraries), Vol. 18 (8), pp: 837-842.

Kaplan L. M. et al. (1998). Fast texture database retrieval using extended fractal features, *In Storage and Retrieval for Image and Video Databases VI* (Sethi, I K and Jain, R C, eds), Proc SPIE 3312, 162-173, 1998.

John R. Smith. (1997). *Integrated Spatial and Feature Image System: Retrieval, Analysis and Compression*, Ph.D thesis, Columbia University, 1997.

Deng Y. (1999). *A Region Representation for Image and Video Retrieval*, Ph.D thesis, University of California, Santa Barbara, 1999.

Haralick R. M. (1979). Statistical and structural approaches to texture, *Proc. IEEE.* Vol 67, pp: 786-804

Gool L. V.; Dewaele P. & Oosterlinck A. (1985). Texture analysis anno 1983, *Computerr Vision,* Graphics and Image Processing, vol. 29, pp: 336-357.

Image Processing

Edited by Yung-Sheng Chen

ISBN 978-953-307-026-1
Hard cover, 516 pages
Publisher InTech
Published online 01, December, 2009
Published in print edition December, 2009

There are six sections in this book. The first section presents basic image processing techniques, such as image acquisition, storage, retrieval, transformation, filtering, and parallel computing. Then, some applications, such as road sign recognition, air quality monitoring, remote sensed image analysis, and diagnosis of industrial parts are considered. Subsequently, the application of image processing for the special eye examination and a newly three-dimensional digital camera are introduced. On the other hand, the section of medical imaging will show the applications of nuclear imaging, ultrasound imaging, and biology. The section of neural fuzzy presents the topics of image recognition, self-learning, image restoration, as well as evolutionary. The final section will show how to implement the hardware design based on the SoC or FPGA to accelerate image processing.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen