INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

University of Alberta

Image Indexing in the Embedded Wavelet Domain

by

Liu, Chuping ©

A thesis submitted to the Faculty of graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring of 2002

National Library of Canada

Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services

Acquisitions et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-69730-4

Canada

University of Alberta

Library Release Form

Name of Author: Liu, Chuping

Title of Thesis: Image Indexing in the Embedded Wavelet Domain

Degree: Master of Science

Year this Degree Granted: 2002

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Liu, Chuping
#12, 10730-84 Ave.
Edmonton, Alberta
Canada, T6E 2H9

Date: Jan 29, 2002

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Image Indexing in the Embedded Wavelet Domain** submitted by **Liu, Chuping** in partial fulfillment of the requirements for the degree of **Master of Science**.

Robert Fedosejevs

Dr. Robert Fedosejevs *Committee Chair*

Dr. Mrinal K. Mandal *Supervisor*

Dr. Bruce F. Cockburn *Committee Member*

Jörg Sander

Dr. Joerg Sander *Committee Member*

Date: Jan. 29, 2002

Youth

Youth is not a time of life ~ it is a state of mind. It is not a matter of rosy cheeks, red lips and supple knees; it is an attitude, a quality of the imagination, a vigor of the emotions.
Youth means a predominance of courage over timidity, a preference for adventure over love of ease.
Years wrinkle the skin, but to give up enthusiasm wrinkles the soul.
Whether seventy or sixteen, there can be in every being's heart a sense of wonder,
the sweet amazement at the stars, the challenge of new events, childlike curiosity, and the joy of living.
You are as young as your faith, as old as your doubt; as young as your self-confidence, as old as your fears; as young as your hope, as old as your despair.
So long as your heart is warmed by the message of beauty, hope, cheer, courage, and meaning in life, you will remain young.

– Anonymous

谨以此文，
献给我的父母，是他们送给我如此缤纷的世界；
献给我的兄弟姐妹，是他们充实了我生命的空间；
献给我昔日的情人，是他留给我一段美丽、忧伤而悠远的记忆；
还有他，我前尘、今生、来世的爱人，他是我的梦，我的心，我灵魂的守望者……

to my parents, who gifted me a splendid world by their love;
to my brothers and sisters, who enriched my life with their wisdom;
to my past lover, who left me a beautiful, dolourous and everlasting memory;
and to him, my beloved in the world, who were my dream, my heart, and my soul ......

ABSTRACT

Very large size image databases are being built for different application such as digital libraries, museum object management, individual picture and photograph collections. With the fast development of Internet and mass storage devices, these image archives are made publicly accessible, and there is an increasing demand on techniques for searching and retrieving the images from the databases. The classical method of using keywords to represent images is impractical due to the substantial manual work involved. Hence, content-based image indexing methods have become popular in the last decade. Current research on image indexing concentrates on the techniques in the compressed domain in order to reduce the storage cost and computational complexity arising from the large image database size.

In this thesis, the indexing techniques from the classical text-based to the latest compressed domain content-based are critically reviewed. It is widely known that wavelet-based techniques have great potential to provide superior coding and indexing performance in the compressed domain. In this thesis, several indexing techniques in the wavelet domain are proposed. First, two techniques are proposed in the embedded zerotree wavelet framework. These techniques are based on the histogram of the number of significant wavelet coefficients. Four indexing techniques in the JPEG2000 framework are then proposed. These techniques are based primarily on bit-plane and the packet header of JPEG2000 bitstreams. Experimental results show that the proposed techniques can achieve retrieval efficiency of up to 93%. These techniques can be integrated easily with the recently established MPEG-4 and JPEG2000 standards.

ACKNOWLEDGEMENTS

This work was made possible through the support of many people and organizations.

First of all. I would like to thank Dr. Mandal, my supervisor, for his constant guidance. funding. logistical support and encouragement. His patience in reading this full thesis word by word. and correcting my poor English vocabulary and grammar is highly appreciated.

Many thanks to Dr. Cockburn and Dr. Sander. As my thesis committee members. they have spent significant amount of time and energy in reading this thesis.

I am grateful to the administrative staff, especially Michelle. Carla. Terra – and many others who helped me to finish my Master program. I am also grateful to Kees, the department computer and network manager, who helped me fall in love with Linux.

I am greatly indebted to the member of the MCCL: Yufei Yuan. Hongyu Liao. Victor Sanchez. Yuan Wan. Yang Lee. and other group members. They are not only my collaborators but also good friends who shared in my happiness and sadness. I especially thank Yufei from whom I learned a lot in our idea exchanges on our projects and playing with Linux as well. I am very thankful to the couple. Yuxin Wang and Yunan Xiang, always great friends during my studies. I was fortunate to have them as group members to do a great project in the ASIC design course.

The financial support from NSERC. CITR and CSA is gratefully acknowledged.

Many thanks to my family – my parents. my brother and my sister for their love, care and support over all these years. Finally my thanks to my husband, my dearest Huang-Yin, who supports me in every circumstance, and inspires me with his wisdom and knowledge.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CMD	Combination of MNSCH and DNNSCH
CML	Combination of MNSCH and LLBM
CDL	Combination of DNSSCH and LLBM
CMDL	Combination of MNSCH, DNNSCH and LLBM
CSN	Combination of SBM and NBH
DCT	Discrete Cosine Transform
DNNSCH	Histogram of Differences in the Number of Subband Significant Coefficients
DWT	Discrete Wavelet Transform
ISC	Insignificant Coefficient
KLT	Karhunen-Loeve Transform
LLBM	Lowpass Subband Binarization Map
MNSCH	Modified Histogram of the Number of Significant Coefficients
NSCH	Histogram of the Number of Significant Coefficients
NBH	Histogram of the Number of Bits in Bit-Plane
PHMV	Vector of Moments of Packet Headers
PCA	Principle Component Analysis
RSTN	Rotation, Scaling, Translation and reflectioN
SBM	Map of Significant Bits
SC	Significant coefficient
WMV	Vector of Moments of Wavelet Coefficients

Chapter 1 INTRODUCTION

Very large digital image archives have been created and used in a number of applications, such as archives for museum objects, trademarks, and photos from daily life. Furthermore, these large image archives are made publicly accessible due to the explosion of the World Wide Web (WWW), benefiting from the availability of inexpensive mass storage devices such as large hard disks, CDROMs, and DVDs. Consequently, there is an increasing demand on the methods by which users can retrieve pictorial entities from the large image archives.

The existing image indexing and retrieval techniques for image databases are still very limited in scope. In the early image storage and collections, the representation of the image by its content was done manually. This becomes impractical with large-size image databases. The method to describe the images by keywords [1] cannot fulfill the requirement of users anymore. As we know, today's images are stored in compressed formats, *e.g.*, JPEG [2], to reduce the storage cost. This requires two main tasks for image retrieval system: (1) automatic indexing and retrieval by computer to reduce manual interaction; (2) possible approaches to indexing and retrieval in the compressed domain to enhance the retrieval efficiency. Therefore, the term "Content Based Image Retrieval (CBIR)" was coined, which means image indexing and retrieval by its content, or so-called "query by content". Research in this field involves different related areas, such as transform (*e.g.*, Fourier, Cosine and Wavelet) and compression, feature extraction and analysis and similarity. Especially, the wavelet transform has become more popular due to its many useful features for image processing.

1.1 Objectives

This thesis aims to explore the feasibility of realizing CBIR progressively in the embedded wavelet domain and the possibility of improving the retrieval performance.

This project implements indexing and retrieval algorithms based on significant wavelet coefficients in an embedded zerotree wavelet framework, and based on the wavelet-coefficients bit-plane and packet headers in JPEG2000 standard [3] framework.

More specifically, the main objectives of this thesis are as follows:

- Introduce the existing CBIR techniques and address the problems through a detailed review.
- Explore the possibility of overcoming these problems in the discrete wavelet transform domain by comparing several available techniques.
- Propose new algorithms which enhance the retrieval performance compared with the results in the literature based on the wavelet transform (in both embedded zerotree wavelet and JPEG2000 frameworks), and research into the proposed techniques to obtain good query performance with possibly lower computation complexity.

1.2 Novelty

This work considers CBIR techniques in the wavelet transform domain. With increasing image database sizes, CBIR techniques in the compressed domain dominate this field. Wavelet transform compression has become popular recently due to many advantages, such as multi-resolution and high compression efficiency [4]. It is also adapted into the JPEG2000 image format standard and the MPEG-4 [5] video coding standard. The first part of this thesis deals with the techniques based on an embedded zerotree wavelet. The novelty of this work also lies on the first implementation of indexing and retrieval directly in the JPEG2000 framework, which was finally drafted early last year.

1.3 Outline

The organization of the thesis is as follows. A review of background work is presented in Chapter 2. Different research work in text-based techniques and CBIR techniques is reviewed. The review highlights the significance of the techniques in compressed domain, especially the wavelet-based compressed domain. A brief introduction of wavelet concepts is given for a better understanding of this thesis.

In Chapter 3, the working environments for testing and evaluation are outlined.

Several existing wavelet-based indexing techniques, which are related to this research, are critically evaluated in Chapter 4.

Indexing techniques in the embedded zerotree wavelet (EZW) domain is presented in Chapter 5. The evaluation results of the proposed techniques are then compared to those of the techniques reviewed. A discussion on the best choice for CBIR based on EZW is presented.

In Chapter 6, the proposed techniques for the JPEG2000 framework are presented along with an overview of the JPEG2000 standard. The factors influencing the retrieval efficiency are analyzed to get a good performance.

Chapter 7 summarizes the overall work in this thesis. Possible future research directions for extending the proposed techniques are also discussed.

Chapter 2 Review of Wavelet and Image Indexing

This chapter presents a comprehensive introduction and review of the latest approaches proposed in the field of image indexing and retrieval. There are several methods to classify image indexing techniques. Here we would like to divide these techniques into two groups: text-based and content-based. Text-based techniques are traditional retrieval methods. They are not quite popular as before and will be briefly reviewed. For the content-based image retrieval (CBIR) techniques, substantial work has been done in two domains: pixel (non-compression) and compressed. Depending on the compression techniques used, CBIR techniques in compression domains can also be divided into several groups, *e.g.*, discreet cosine transform (DCT)-based and wavelet-based. We will focus on techniques in the wavelet-based compression domain because wavelet compression has been adapted into the JPEG2000 still image format standard and the MPEG-4 video coding standard. For better understanding of wavelet-based indexing techniques, basic knowledge on wavelet is given prior to the review of image indexing techniques.

2.1 Wavelet Transform

From Fourier theory, we know that a signal can be expressed as a linear combination of a possibly infinite series of sines and cosines, also known as the Fourier expansion. The main shortcoming in Fourier expansion is that it has only effective frequency resolution and no time resolution. For example, a raw signal in the time domain $x(t)$, its corresponding Fourier transform of a time domain signal $X(f)$ is defined as:

$$X(f)=\int_{-\infty}^{+\infty}x(t)e^{-2j\pi ft}dt \qquad (2.1)$$

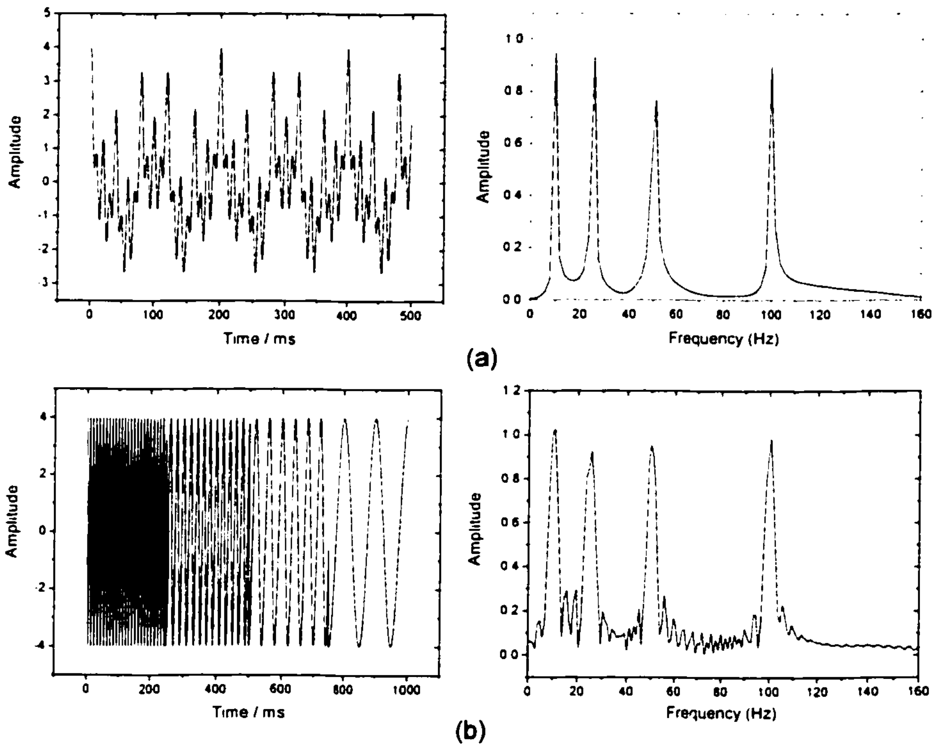


Figure 2-1 Two different signals having the same Fourier spectrum. (a) The first signal is from the function of $x(t) = \cos(20\pi t) + \cos(50\pi t) + \cos(100\pi t) + \cos(200\pi t)$; (b) the second signal is not a stationary signal. In the interval 0 to 250 ms it has a 100 Hz sinusoid, 50 Hz sinusoid in 250 to 500 ms interval, 25 Hz sinusoid in 500 to 750 interval and finally 10 Hz in the 750 to 1000 ms interval. Their Fourier spectra both have the same peaks at 10, 25, 50 and 100 Hz. Fourier transform cannot give any time information.

$X(f)$ is a function of only the frequency f, not function of time t. Therefore, it is difficult to distinguish such two signals (Figure 2-1 (a) and (b)) in the frequency domain.

In the past decades several solutions have been developed to solve this problem of representing a signal in the time and frequency domain simultaneously. The underlying idea here is to cut the signal into several parts and then analyze these parts separately, where a fully scalable modulated window is used. The window is shifted along the signal and the spectrum is calculated for every position. Then this process is repeated many

many times with a slightly shorter or longer window for every new cycle. This leads to the transform into a collection of both time and frequency representations, all with different resolutions. In general, scale instead of frequency is used in wavelet analysis. Large scale means the big picture while small scale shows details of the picture. While changing from large scale to small scale, the picture is enlarged or zoomed in.

2.1.1 *Continuous Wavelet Transform and Wavelet Properties*

Wavelet analysis as described above is known as continuous wavelet transform (CWT). Mathematically, it can be written as:

$$\begin{aligned} \gamma(s,\tau) &= \int f(t)\Psi^{*}_{s,\tau}(t)dt \\ f(t) &= \int\int \gamma(s,\tau)\Psi_{s,\tau}(t)d\tau ds \end{aligned} \qquad (2.2)$$

where * denotes complex conjugation. According to this equation, the original raw signal $f(t)$ is decomposed into a set of basis function $\Psi_{s,\tau}(t)$, and s,τ are the new dimensions: scale and translation. The wavelets, whose name suggests "small wave", are generated from a so-called "mother wavelet" $\Psi(t)$ by scaling and translation:

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right) \qquad (2.3)$$

where s is the scaling factor and τ the translation factor. $\frac{1}{\sqrt{s}}$ is used to normalize the energy to 1.

So we can see that the wavelet transform is different from the Fourier transform and other transforms in that the mother wavelet is not specified. But at the same time, the wavelets cannot be anything, they should obey some rules, which are also their properties. Based on these properties, efficient wavelets can be built. The most important property of wavelets is admissibility:

$$C_\Psi = \int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty \tag{2.4}$$

where $\Psi(\omega)$ is the Fourier transform of $\Psi(t)$. This equation was derived by Grossman and Morlet in 1984 [6].

The admissibility condition implies that the Fourier transform of $\Psi(t)$ vanishes at zero frequency. That is,

$$\Psi(\omega)\big|_{\omega=0} = \int \Psi(t)dt = 0 \tag{2.5}$$

2.1.2 Discrete Wavelet Transform

Before wavelet transform can be used in practice, three problems should be solved.

1. Redundancy in CWT. As described above, the CWT is calculated by shifting a continuous scalable function over a signal and calculating the correlation between the two. These scalable functions are far away from orthogonal and the resulted wavelet transform coefficients are quite redundant.
2. The number of wavelets basis functions derived from the mother wavelet needed to construct a raw signal is infinite and it is desirable to reduce this number to a more manageable count.
3. Fast algorithms are required because most of the wavelet transform functions do not have analytical solutions and can only be solved numerically.

The first problem can be solved by introducing discrete wavelets. This is achieved by modifying the wavelet representation in Eq. (2.3) to:

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}} \Psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right) \tag{2.6}$$

where j, k are integers and $s_0 > 1$ is a fixed dilation step. s_0 is usually chosen as 2 because it is very easy to implement by computers. This sampling of frequency is also called dyadic sampling (Figure 2-2).

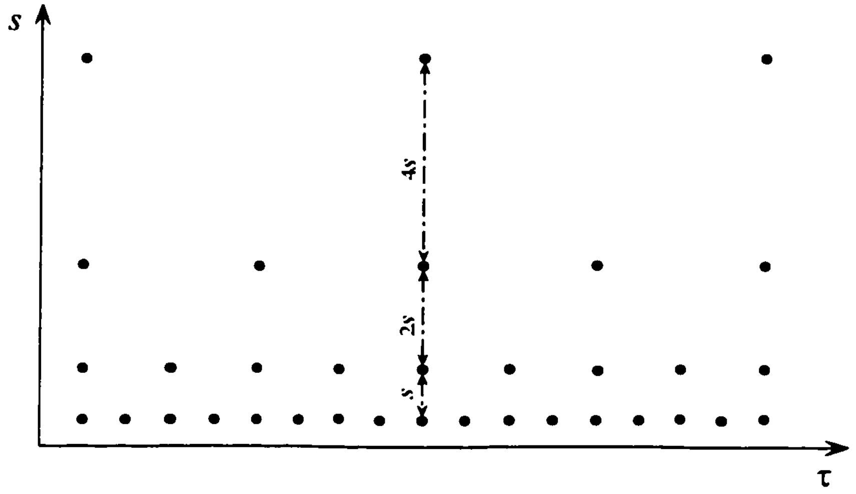


Figure 2-2 Illustration of dyadic sampling for discrete wavelets.

The problem now is whether it is possible to reconstruct the original signal from the discrete wavelet coefficients. As pointed out by Daubechies [7] in 1992, the necessary and sufficient conditions for stable reconstruction is that the energy of the wavelet must lie between two positive bounds, *i.e.*,

$$A\|f\|^2 \le \sum_{j,k} \left|\langle f, \Psi_{j,k} \rangle\right|^2 \le B\|f\|^2 \tag{2.7}$$

where $\|f\|^2$ is the energy of the raw signal $f(t)$, $0 < A \le B < \infty$ and both A and B are independent of $f(t)$. If this condition is satisfied, then the family of wavelet basis functions $\Psi_{j,k}(t)$ with integers j and k is referred to as a *frame* with frame bounds A and B. When $A = B$ the frame is tight and the discrete wavelets behave exactly like orthogonal basis functions. When $A \neq B$ exact reconstruction is still possible at the expense of a *dual frame*. In a dual frame discrete wavelet transform, the decomposition wavelet is different from the reconstruction wavelet. The next step is to make these discrete wavelets orthogonal by special selection of the mother wavelet. Finally, an arbitrary signal can be the sum of the orthogonal wavelet basis functions weighted by discrete wavelet transforms (DWT) coefficients:

$$f(t)=\sum_{j,k}\gamma(j,k)\Psi_{j,k}(t) \tag{2.8}$$

To address the second problem, we can look at the wavelet in another way: as a band-pass filter. Then a series of dilated wavelets with the same ratio (also called fidelity factor *Q*) between the center frequency and the width of the wavelet spectrum can be seen as a band-pass filter bank. But this still needs infinite wavelets to construct the filter bank in order to cover the spectrum all the way down to zero; the solution is simply not to cover the entire spectrum, but to use a cork to plug the hole when it is small enough. This cork is a low-pass spectrum and is called a *scaling function*. The scaling function is sometimes also called an *averaging filter* due to its low-pass nature. Since we only need to use the wavelets up to the scale j and use the scaling function to fill the remaining part not covered by the wavelets, only a finite number (j) of wavelets is used. Now the wavelet transform of a signal results from passing the signal through the filter bank (both the wavelet filter bank and the scaling function low-pass filter). The outputs of the different filter stages are the coefficients of the wavelet transform and the scaling function transform. This process is similar in nature to the subband coding used in computer vision applications.

The filter bank needed here can be built in several ways. One is to build many band-pass filters to split the spectrum into frequency bands. This method allows us to select the band width freely, but at the same time we have to design the filters separately and this is very time-consuming. Another way is to split the signal spectrum into two equal parts (low-pass and high-pass). The low-pass part still contains many details, which can be equally split again. Iterating this process results in an iterated filter bank. In this method we only need to design two filters, but the problem in the covered spectrum region is fixed.

2.1.3 Some Popular Wavelets

Here, some wavelet functions are introduced below. Among the four wavelet bases, the first basis set is non-orthogonal, while the other three basis sets are orthogonal [7].

1. Mexico Hat Wavelets

$$\Psi(t) = \frac{2}{\sqrt{3}}\pi^{\frac{1}{4}}(1-t^2)e^{-\frac{t^2}{2}} \tag{2.9}$$

2. Haar Wavelets

$$\Psi(t) = \begin{cases} 1, & 0 \le \mathrm{t} \le 1/2 \\ -1, & 1/2<\mathrm{t} \le 1 \\ 0, & \text{others} \end{cases} \tag{2.10}$$

3. Shannon Wavelets

$$\Psi(t) = \frac{\sin\pi(t-\frac{1}{2}) - \sin 2\pi(t-\frac{1}{2})}{\pi(t-\frac{1}{2})} \tag{2.11}$$

4. Daubechies Wavelets

Daubechies wavelets cannot be expressed in closed analytical form. A set of refinement coefficients h_n can be used to generate the scaling function:

$${}_N\varphi(t) = \sqrt{2}\sum_{n=0}^{2N-1} h_n\varphi(2t-n) \tag{2.12}$$

The wavelets functions can be generated from the following equation:

$${}_N\Psi(t) = \sqrt{2}\sum_{n=0}^{2N-1} g_n\varphi(2t-n) \tag{2.13}$$

$$g_n = (-1)^n h_{2N-n-1} \qquad (2.14)$$

Table 2-1 shows Daubechies wavelets for N=2,4,8, and 12 taps, and the corresponding mother wavelets are plotted in Figure 2-3.

Table 2-1 Minimum-phase Daubechies wavelets for N=2,4,8, and 12 taps.

Taps	n	h[n]	Taps	n	h[n]
N=2	0	0.70710678	N=12		
	1	0.70710678		0	0.111540743350000
N=4	0	0.482962913144		1	0.494623890398000
	1	0.836516303737		2	0.751133908021000
	2	0.224143868042		3	0.315250351709000
	3	-0.129409522551		4	-0.226264693965000
N=8	0	0 230377813308		5	-0.129766867567000
	1	0.714846570552		6	0.097501605587000
	2	0.630880767939		7	0.027522865530000
	3	-0.027983769416		8	-0.031582039318000
	4	-0.187034811171		9	0.000553842201000
	5	0.030841381835		10	0.004777257511000
	6	0.032883011666		11	-0.001077301085000
	7	-0.010597401785			

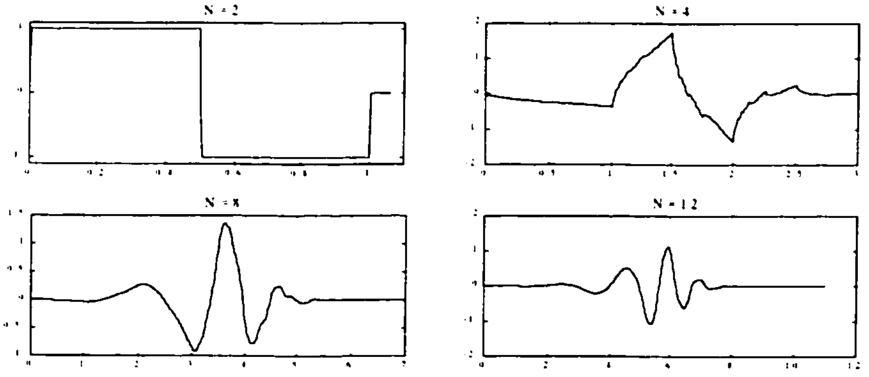


Figure 2-3 A few selected Daubechies mother wavelets. The wavelets correspond to the refinement coefficients shown in Table 2-1. N is the number of coefficients for the mother wavelet.

2.2 Text-based Indexing and Retrieval

Traditional databases use text keywords as labels to quickly search large quantities of text data. A human indexer who builds the image database describes the images according to the image content. the caption or the background information. To retrieve a desired image. a user constructs queries using keywords to match the annotations made by the indexer.

Since the annotations of the images in the database depend on the human indexer. the keywords input by a user may have poor retrieval results or cannot find the desired images. In the last decades. considerable attention has been paid to develop thesauruses of keywords as standard query keyword databases. Most of the terms required to understand the text-based indexing and retrieval techniques have been discussed by van den Berg [1]. who presented an overview of the features of ICONCLASS (a system that contains approximately 24,000 definitions of objects. events. situations and abstract ideas. see http://www.iconclass.nl/)

We note that text-based techniques are simple and efficient. However. the representation of an image with keywords requires a large amount of manual processing and entails extra storage. For example. it has been pointed out [8] that ICONCLASS is well suited for dealing with the traditional iconography of art history. but it is less satisfactory when dealing with the description of common objects. such as tables, rivers and houses. Furthermore, the retrieval results might not be satisfactory because the query is based on features that may not reflect the visual content.

2.3 Content-based Image Indexing and Retrieval

As suggested from the name. a content-based image retrieval (CBIR) system is able to retrieve relevant images based on their semantic and visual contents rather than by using keywords assigned to them [9]. Because of the extreme difficulty in image understanding, the interpretation of various features of an image is fuzzy; the emphasis of CBIR systems at present is supporting query and browsing capabilities instead of precise search, which is still a long-term research area.

CBIR techniques have been reviewed by several researchers [10-14]. A block schematic of a typical CBIR system in the pixel domain is shown in Figure 2-4. For each image in the image database, features such as color [15, 16], texture [17, 18] and shape [19, 20] are extracted and stored. The features are matched between the query image and images in the database. The processes involved in the CBIR include:

- understanding user needs and their searching behaviors
- identifying the important features of the images
- extracting such features
- matching query and candidate images by similarity measures
- efficiently accessing stored images by image content
- user-friendly interface

Obviously, the key issue in CBIR is how to represent image contents using an extracted feature vector (*i.e.*, index), and to compare the similarities of images using the feature vectors (*i.e.*, indices).

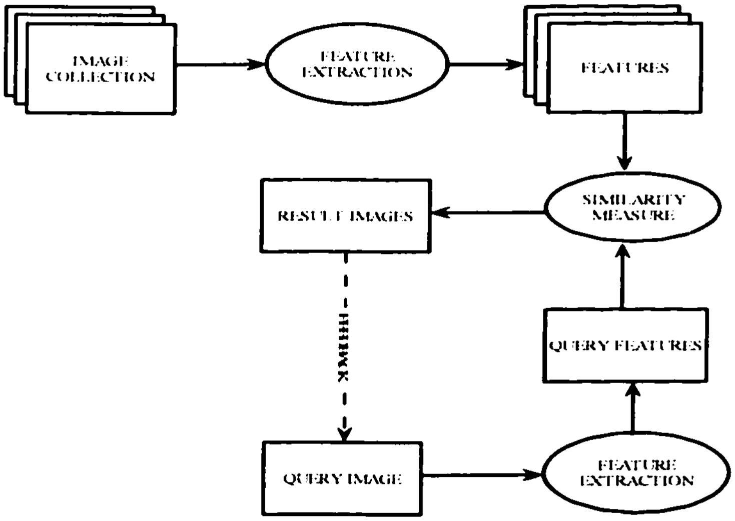


Figure 2-4 Schematic diagram of a typical CBIR system [21].

2.3.1 Image Indexing in the Pixel Domain

Classical visual image indexing techniques are based on features such as color, texture and shape, as mentioned above. In such techniques, the image features are extracted directly from the image pixels. We categorize these existing approaches on CBIR based on the types of features that have been utilized to express the content of images.

2.3.1.1 Color

The color feature is probably the most visible feature for most humans. It provides powerful information for image retrieval because a color feature is relatively robust to background complication and different image orientations. Each image in the database is analysed to compute a color histogram which shows the proportion of pixels of each color within the image.

A *histogram* is an alternative style of statistical diagram, like a probability density function (*pdf*) without normalization. It presents discrete frequency distribution for a grouped data set which has different discrete values grouped into a number of intervals. Below is an example diagram of histogram for the score distribution of a class (Figure 2-5).

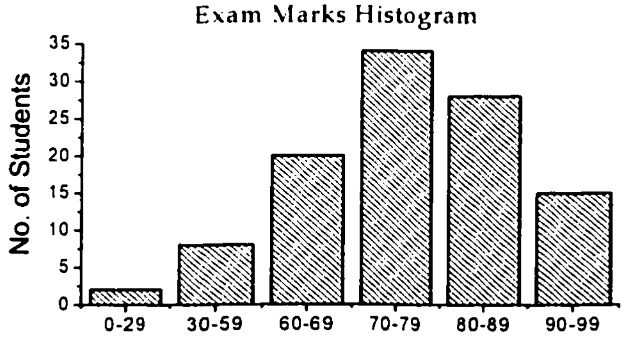


Figure 2-5 Histogram of student exam marks

In Figure 2-5 the value (*i.e.*, height) of each bin in the histogram is the count of scores within that interval. The width of each bin may be equal or unequal, depending on the application.

Assume that the color of any pixel may be represented in terms of component R(red), G(green), B(blue) values. A color histogram consists of three independent histograms, each corresponding to a color component. It denotes the joint probabilities of the intensities of RGB components. An example of color histogram is given in Figure 2-6. In the figure, the Lena image is stored in RGB color space. Hence, the color histogram for Lena image includes 3 histograms for color component R, G and B, respectively.

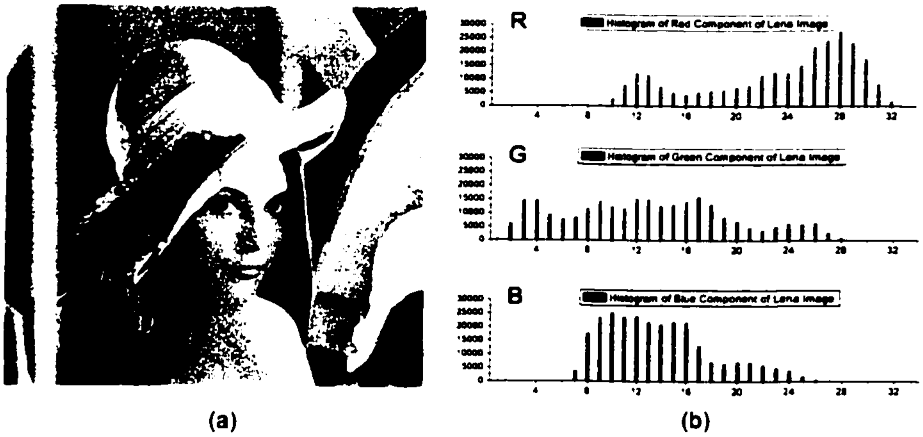


Figure 2-6 Color histograms for the Lena image. (a) Three channel color Lena image; (b) Histogram for each channel in RGB color space. The intensity value, ranging from 0 to 255, of each color component is quantized with a quantization step of 8, resulting in 32 bins in the corresponding histogram.

In color histogram based indexing, the similarity of two images is obtained by calculating the similarity of the corresponding histograms. The similarity of histograms is generally estimated using one of the three metrics discussed below.

1) *Histogram Euclidean Distance*

The L^2 Euclidean distance between two color histograms h and g is given by:

$$d(h,g) = \sum_{m=0}^{M-1} (h[m] - g[m])^2 \tag{2.15}$$

where M is the total number of bins, and m denotes the index of a bin.

In this distance formula there is only comparison between corresponding bins in the respective histograms. Two neighboring bins may represent perceptually similar colors, but this is not considered in Eq. (2.15). For example, light red and deep red are perceptually similar colors, but they may fall in different histogram bins and produce a large difference when matching two indices.

2) Histogram Intersection

The color histogram intersection was first developed by Swain and Ballard [16]. Variants of this technique are now used in a high proportion of current CBIR systems. The intersection of histograms h and g is given by:

$$d(h,g) = \frac{\sum_{m=0}^{M-1} \min(h[m], g[m])}{\min\left(\sum_{m_0=0}^{M-1} h[m_0], \sum_{m_1=0}^{M-1} g[m_1] \right)} \tag{2.16}$$

Colors not present in the user's query image do not contribute to the intersection. Compared to Euclidean distance, Eq. (2.16) has an improvement in reducing the difference of perceptual similarity and mathematical similarity, but the problem still exists.

3) Histogram Quadratic Distance

The color histogram quadratic distance is used by the QBIC (Query by Image Content) system [22]. The quadratic distance between histograms h and g is given by:

$$
\begin{aligned}
d(h,g) &= \sum_{m_0=0}^{M-1}\sum_{m_1=0}^{M-1}(h[m_0]-g[m_0])\cdot a_{m_0,m_1}\cdot(h[m_1]-g[m_1]) \\
&= -\sum_{m_0=0}^{M-1}\sum_{m_1=0}^{M-1}(h[m_0]-g[m_0])\cdot d_{m_0,m_1}\cdot(h[m_1]-g[m_1])
\end{aligned}
\tag{2.17}
$$

The quadratic distance formula uses the cross-correlation between histogram bins based on the perceptual similarity of the colors m_0 and m_1. One appropriate value for the cross-correlation is given by $a_{m_0,m_1} = 1 - d_{m_0,m_1}$, where d_{m_0,m_1} is the distance between color m_0 and m_1 normalized with respect to the maximum distance. In this case the histogram quadratic distance formula reduces to the form on the right.

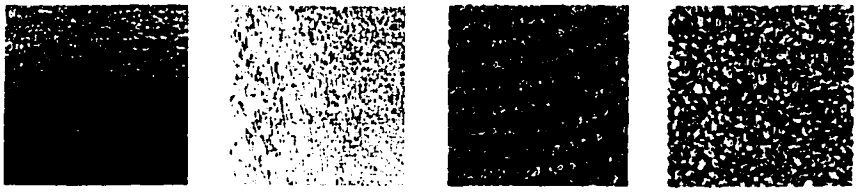
Color distribution, as a global property that does not require knowledge of how an image is composed of component objects, is more suitable in textured images and other images that are not particularly amenable to segmentation.

However, when an image is transformed into a histogram, all spatial information is lost [23]. Indexing using color histograms has significant limitations due to this lack of spatial information [24]. To overcome this problem, several methods to combine color histograms with spatial information have been proposed. Sticker and Dimai [24] proposed a method of dividing an image into "five fuzzy regions" and claimed the regions are natural for encoding of minimal spatial information. Pass *et al.* [25] presented a method called color coherence vectors (CCV) which partitions pixels based on spatial coherence. Huang *et al.* [26] presented another method to incorporate color and spatial information, called color correlogram that represents how the spatial correlation of color pairs change with distance.

Human color perception issue is another problem in the usage of color histogram to index images. The RGB color space is not perceptually uniform, and the proximity of colors in this color space does not indicate color similarity. Therefore, linear or non-linear transformation of color space is often used, such as YIQ and YUV. The IBM QBIC [22] system utilizes the Munsell color order system, which organizes the colors according to more natural attributes. In addition, the HSV color space is a natural color space and approximately perceptually uniform. Currently there is still no consensus about which color space is most suitable for color histogram-based image indexing and retrieval.

2.3.1.2 Texture

Texture is an important attribute of an image. Texture based indexing is also useful where the color of an object is not important. The texture features can be employed to distinguish between two areas with similar color, *e.g.*, blue hue in sea and sky, green hue in leaves and grass. It is difficult to define a texture precisely. However, a major characteristic of texture is the repetition of a pattern over a region [27]. The usage of texture features to retrieve images is based on the assumption that an image can be considered as a mosaic of different texture regions. Although the precise definition of texture has been elusive, the notion of texture generally refers to the presence of a spatial pattern that has some properties of homogeneity. Figure 2-7 gives some examples of textures.



Water 1 **Water 2** **Fabric 1** **Fabric 2**

Figure 2-7 Examples of texture [28].

The texture analysis techniques, which have been developed and are most extensively used, involve three main categories: structural, statistical and spectral [29].

Among various techniques, three approaches have been demonstrated to be particularly effective in image retrieval. The Tamura features [30] includes several readily perceived properties of a texture, such as contrast, directionality and coarseness. A modified set of the Tamura features has been used in the QBIC project. The simultaneous autoregressive (SAR) model [31] provides a description of each pixel in terms of its neighboring pixels when a gray-level image is given. The Wold features are based on the decomposition of a two-dimensional regular and homogeneous random field. The decomposition provides another approach to describing textures in terms of perceptual properties: periodicity, directionality and randomness [32]. Each of these properties is associated with dominance of a different Wold component: Periodic textures have a strong harmonic component, highly directional textures have a strong evanescent component, and less structured textures tend to be strongest in the indeterministic component.

Since the introduction of wavelet transform in the early 1990's, many researchers have tried to use the wavelet transform in texture representation. Smith and Chang [33] used the statistics extracted from wavelet subbands as the texture representation. This approach achieved over 90% accuracy on the 112 Brotadz texture images.

2.3.1.3 Shape

Among the direct features of images, shape may be the most important criterion for matching objects based on their profile and physical structure. The ability to retrieve images by shape is possibly the most obvious requirement at the primitive level in this field. Unlike texture, shape is a well-defined concept. It is widely believed that the humans recognize natural objects by their shape. For the present image retrieval techniques, two main types of shape features are commonly used – boundary-based and region-based. In boundary-based techniques, the outline of the objects is obtained by edge detection, thinning and shrinking algorithms, and is used as the index. In region-based techniques, the region inside the object boundary is extracted as the index. Examples for boundary-based and region-based methods are presented in Figure 2-8 and Figure 2-9, respectively.

Loncaric [34] presented a survey of shape analysis techniques. Different classifications of shape-based indexing techniques were also discussed in this paper. The goals for shape similarity retrieval are to design efficient shape descriptors, and a proper similarity measure associated with these descriptors. Sometimes these descriptors are required to rotate and translate.

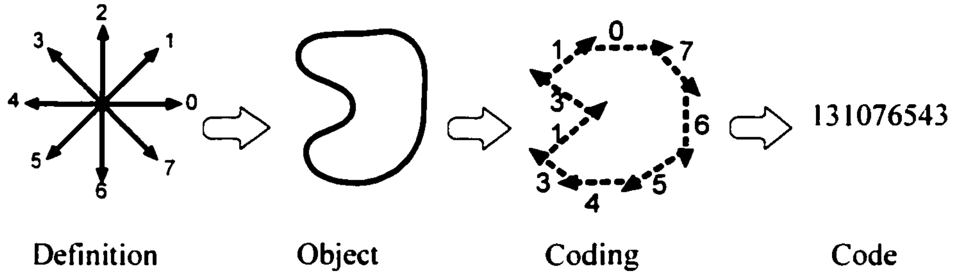


Figure 2-8 Boundary-based representation of shape features.

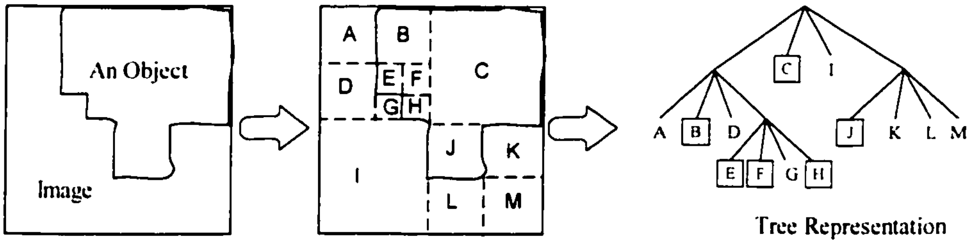


Figure 2-9 Region-based representation of shape features.

A promising approach was proposed by Sclaroff and Pentland [35] in which shapes are represented as canonical deformations of prototype objects. In this method, a "physical" model of the 2D shape is built using a new form of Galerkin's interpolation method (finite-element discretization) and the possible deformation modes are analyzed using Karhunen-Loeve (KL) transform. This yields an ordered list of deformation modes corresponding to rigid body modes, low-frequency non-rigid modes (global deformations) and high-frequency modes (localized deformations).

The most successful representatives in image feature extraction for boundary-based and region-based categories are Fourier Descriptor and Moment Invariant, respectively [14]. In the Fourier Descriptor method, the Fourier transformed boundary is used as the shape feature, whereas in the Moment Invariant method, the region-based moments that are invariant to transformations are used as the shape features. It is generally difficult to employ L^1/L^2 metric for shape index matching. Gadi and Benslimane [19] proposed a new measure based on fuzzy logic, a fuzzy similarity measure, for shapes described by

Fourier Descriptors. In this approach, images are represented by an N-dimensional feature vectors. The similarity measure is calculated for each component of the feature vectors. Hence, for each pair of images, N similarity measures will be obtained. The global similarity measure is obtained by using a fuzzy IF-THEN rule or considering each similarity measure obtained between two components as an opinion.

Experiment shows that the retrieval system is robust to the rotation, scaling, translation and reflection (RSTN), and the retrieved images can be in different positions and orientations (as shown in Figure 2-10).

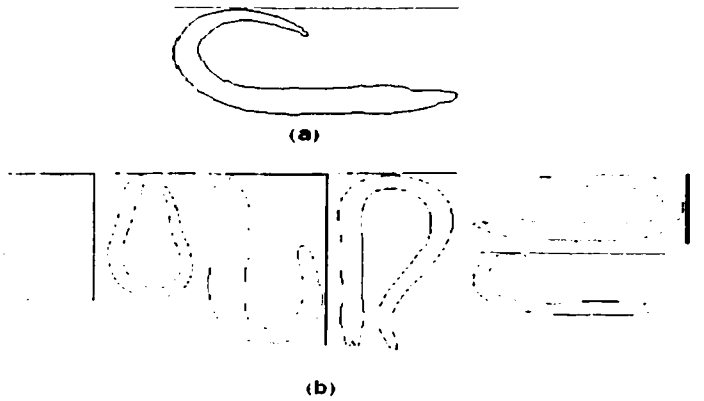


Figure 2-10 Illustration of shape retrieval (a) Target Image (b) Retrieved images [19].

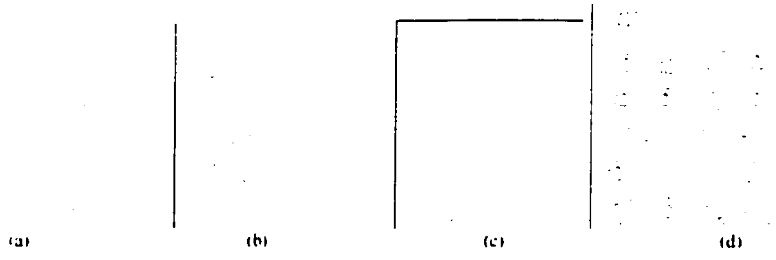


Figure 2-11 Examples of shape retrieval from the shape database of plant leaf images [19].

Nishida [36] presents another efficient, robust method for shape retrieval from image databases composed of boundary contours of objects. The method is mainly based on an indexing technique for structural features, along with a voting technique for ranking model images in terms of extracted features from the query image. In particular, shape feature generation techniques are incorporated into structural indexing to improve the accuracy and robustness of shape classification against noise and local shape transformations.

Experimental trials with large image databases of boundary contours have shown that the shape feature generation method significantly improves the robustness, and efficiency of shape retrieval (as shown in Figure 2-11).

2.3.1.4 Spatial Relationship

The positions of objects and their spatial relationships in an image can be used to represent the content of the image. Enser [37] and De Marsicoi *et al.* [38] reviewed several different methods based on spatial relationship. The objects in an image are first segmented and recognized. The image is then converted into a symbolic picture that is encoded using 2D strings. This step is computationally intensive, which is a major disadvantage.

2.3.2 Image Indexing in the Compressed Domain

With the explosive of Internet use, and increasing volume of image databases, images must be stored in compressed format. The indexing techniques in the compressed domain refer to those methods of retrieving images from the database without decoding the images back to their pixel format. The advantages of these techniques include: 1) it is unnecessary to decode the compressed images in the process of image feature extraction; 2) less computing cost due to small computation complexity. Therefore, indexing techniques in the compressed domain are believed to be the mainstream in this field.

Indexing in the compressed domain involves some challenging issues. For example, each compression technique poses addition constraints, *e.g.*, non-linear process, rigid data structure syntax, and resolution reduction. Also, there exists some unique information available in the compressed domain, such as bi-directional optical flows and bit rate. The key issue on how to explore and develop new compression techniques with maximal content accessibility for good retrieval performance is still feature extraction. Features in the compressed domain are not as visual as those in pixel domain, *e.g.*, color and shape, but rather may be possibly the coefficients of a transform, *e.g.*, Fourier transform, Karhunen-Loeve transform.

Image indexing techniques in the compressed domain have been critically reviewed by Mandal *et al.* [12]. Here we will follow the classification of these techniques based on the compression techniques: DFT (discrete Fourier transform), DCT (discrete cosine transform, used in the JPEG standard), KLT (Karhunen-Loeve transform), DWT (discrete wavelet transform, used in the JPEG2000 standard), vector quantization (VQ) and fractal compression. Wavelet-based techniques have some special advantages versus other compression techniques and the indexing techniques proposed in this thesis are also wavelet-based.

2.3.2.1 Discrete Fourier Transform

The Discrete Fourier transform is very important in image and signal processing. For 2D images, the DFT of an image $f(x,y)$ is defined as:

$$F(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}e^{-2\pi i(\frac{ux}{N}+\frac{vy}{M})}f(x,y) \tag{2.18}$$

for u = 0, 1, 2, ..., N-1 and v = 0, 1, 2, ..., M-1. Eq. (2.18) is a separable transform and properties of the 1D case are directly applicable here. If $M = N$ then the DFT power spectrum of $f(x,y)$ is also invariant with rotation.

Stone and Li [39] have proposed an image indexing algorithm in the Fourier domain. The algorithm has two thresholds that allow the user to adjust independently the closeness of a match. One threshold controls an intensity match and the other controls a texture match. The thresholds are correlations that can be computed efficiently in the Fourier transform domain of an image, and are particularly efficient to compute when the Fourier coefficients are mostly zero.

Evaluation of the image retrieval in the Fourier compressed domain has been studied by Augusteijn *et al.* [40], and Celantano and Lecce [41]. The former used several texture measures for classification of satellite images, based on the magnitude of the Fourier spectra of an image, including maximum magnitude, average magnitude, energy of magnitude and variance of magnitude of the Fourier coefficients. The author also studied the retrieval performance based on the radial and angular distribution of Fourier coefficients, which correspond to the texture coarseness and directionality, respectively. These kinds of measures gave a satisfactory performance. In the paper by Celantano and Lecce, the angular distribution of Fourier coefficients in image indexing has been evaluated. Before DFT, the images were pre-processed with a low-pass filter. An angular histogram, calculated by computing the sum of image component contributions, for each angle, is used as the feature vector for indexing.

2.3.2.2 Karhunen-Loeve Transform

The principal component analysis (PCA) or Karhunen-Loeve transform (KLT) is a mathematical way to determine the linear transformation of a sample of points in N-dimensional space which exhibits the properties of the sample most clearly along the coordinate axes. Along the new axes the sample variances are extremes (maxima and minima), and uncorrelated. For image processing, the KLT uses the eigenvectors of the autocorrelation matrix of the set of image features, *i.e.*, it uses the principal components of the distribution of image features. These eigenvectors can be thought as a set of parametric variations from the mean or prototypical appearance. Therefore, before the KLT is applied, the image data are always mean centered. Normally only a few of the eigenvectors with the largest eigenvalues are employed. The KLT has in practice been used to reduce the dimensionality of problems, and to transform interdependent coordinates into significant and independent ones. For example, an image consists of RGB (three) components and they are inter-correlated. By employing the KLT these three components can be decorrelated and then analyzed or matched separately. Since the

KLT basis functions are image adaptive, good indexing performance can be obtained by projecting the images in KL space and comparing the KLT coefficients.

The KLT was adapted into the image indexing system Photobook for face recognition [42]. A set of optimal basis images, also called eigenfaces, is created based on a randomly chosen subset of face images. A query image is then transformed into its eigenimage representation, *e.g.*, projected into eigenfaces. Similarity matching between two images are based on the Euclidean distance between the KLT coefficients. The KLT is also applied for the color representation of images to improve the performance on color histogram features. A straightforward application of the KLT to color histograms gives poor results since the KLT treats the color histogram as an ordinary vector and ignores the properties of the underlying color distribution. Early study has used coarser historgram to avoid these problems [43]. Y. Deng *et al.* [44] proposed a method to match the similarity between the query image and the database using quadratic color histogram for each dominant color and then to combine the matches from all of the query colors to obtain the final retrievals. In L.V. Tran's paper [45], two different spaces, color histogram and local differences histogram, were both considered to measure similarity with Euclidean metric and metrics based on quadratic forms between histogram bins. For each of these four types of metric spaces a set of basis vectors by PCA-based methods were computed. The expansion coefficients computed from these basis vectors are then used as compact descriptors for the color historgram.

An eigenspace approach to multiple image registration was proposed by H. Schweitzer [46]. The idea consists of the iteration of these two steps: 1) Eigenfeatures are computed from the images and 2) new images are computed by registering the images on the eigenfeatures subspace. The KLT has also been applied to reduce the dimensionality of features derived from a texture for classfication [47].

The KLT or PCA method has the potential to provide good performance on image retrieval. However, a detailed investigation on how to generate feature vectors for a large database with widely varying characteristics is still under research. Furthermore, it is worth mentioning that the KLT is generally not used in traditional image coding because of its high complexity.

2.3.2.3 Discrete Cosine Transform

The DCT is a derivative of the DFT. It employs real-valued sinusoidal basis functions and has energy compaction efficiency close to the optimal KLT for most natural images. Hence, most recent international image and video compression standards, such as JPEG, MPEG 1 and 2, H.261/H.263, adapt DCT as the compression kernel.

In JPEG, the original image is divided into 8×8 blocks, then, each block is transformed independently by the DCT. The transformed coefficients are quantized and Huffman coded. The quantization step is lossy and thus compression is achieved. The DCT is defined as

$$F_{u,v} = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)u\pi}{16} f(i,j) \tag{2.19}$$

where $$C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u,v = 0 \\ 1 & \text{otherwise} \end{cases}$$

and $F_{u,v}$ is the 2D DCT coefficient, and $f(i,j)$ the image spatial value. $F_{0,0}$ is normally called the DC component while the rest 63 are AC or high-frequency coefficients. The basis vectors of the DCT are linear and orthogonal.

As mentioned above, the JPEG still image format uses the DCT compression as its kernel. Hence, several image indexing techniques have been proposed in the DCT domain. Chang [48] reported several possible ways of extracting low level features in the DCT domain. For example, the texture feature is formed by computing the statistical measures of the DCT coefficients. To reduce the dimensionality of the feature space, one can employ the Fisher Discriminant techniques to maximize the separability among the known texture classes.

Furht and Saksobhavivat [49] proposed an algorithm to calculate the DC coefficients of this image and then to create the histogram of DC coefficients. Then, the algorithm compares the DC histogram of the submitted image with the DC histograms of the images stored in the database using a histogram similarity metric. Several histogram similarity metrics: weighted Euclidean distance, square difference, and absolute difference have been used to evaluate the retrieving results, in which the absolute difference seems to be the most reliable. This method of indexing depends largely on the number of bins used. A smaller bin number might deteriorate the performance, while a large bin number has a larger computation complexity.

Shen and Sethi [50] proposed a technique to locate areas of interest and also to detect the edge in the images using the AC coefficients of DCT. The authors tried to relate three edge parameters: edge orientation, edge strength and edge offset from center with the DCT coefficients from the JPEG/DCT standard size block (8×8). Performance was compared to the Sobel operator and it was suggested this technique is good enough for some coarse classification or feature based scene detection in video sequence, in which edge detection has to be applied on each (key) frame and fast process for each frame is desired.

In the approach by Ngo *et al.* [51], ten DCT coefficients (from $F_{0,0}$ to $F_{3,0}$) were extracted from each 8×8 JPEG image block. By applying the Mandala transformation, these coefficients were grouped to form ten blocks, each of which represents a particular frequency content of the original image. The first block conveys color information and so it was used to compute color histograms. The second and third blocks were used to extract shape information. The first nine AC coefficients were used for texture description. These result in a much smaller computing and analysis cost. The performance was found to be acceptable and very fast.

Shenier and Mottaleb [52] proposed a technique for image retrieval using JPEG. Their indexing and retrieval method is also based on quantized DCT coefficients. They used coefficients of some selected image windows to avoid the problem when the number of DCT coefficients is equal to the number of pixels. In such situations, the index or representation would not be compact. This method brought out another problem: the choice of windows will affect the performance dramatically and it is very important to choose the windows appropriately.

2.3.2.4 Discrete Wavelet Transform

The basic idea of the wavelet transform is similar to that of the Fourier transform: approximating a signal through a set of component basis functions. The main difference between wavelet and Fourier transforms is that wavelet functions are able to give a multi-resolution representation of the signal since each frequency component is analyzed at a different resolution and scale, whereas the Fourier transform divides the time-frequency domain in a homogeneous way.

Image indexing and retrieval based on the wavelet transform has become popular in the last ten years. Jacobs *et al.* [53] have proposed an algorithm based on direct comparison of the number of significant wavelet coefficients. Three color spaces, RGB, HSV and YIQ, were evaluated and YIQ was found to be the most effective. The authors demonstrated that this algorithm performs much faster than traditional algorithms, with accuracy comparable to traditional algorithms when the query is a hand sketch or a low-quality image scan.

In the Wavelet-Based Image Indexing and Searching (WBIIS) project by Wang *et al.* [54], the wavelet coefficients in the lowest few frequency bands and their variances are considered as feature vectors. Two steps of the similarity matching are employed: standard deviation comparison and a mask-weighted variation of the Euclidean distance. It was reported that the retrieval performance provided improvement over the technique proposed by Jacobs *et al.* [53]. On the other hand, WBIIS is not robust to high degrees of rotation and translation, a problem shared by Jacobs' algorithm.

You *et al.* [55] proposed a hierarchical image matching scheme. The interesting points of an image were detected at each level based on optimal thresholding via fuzzy compactness. The technique includes a guided searching strategy for the best matching from coarse to fine level. It was implemented in a parallel virtual machine environment (PVM).

Sebe *et al.* [56] proposed a wavelet-based color salient points extraction algorithm. It was shown that extracting the color information at the locations of these salient points provides significantly improved retrieval results as compared to global color feature approaches. The salient points extracted were those coefficients with the highest gradient in the tracing algorithm and color moments were used to retrieve the similar images.

Albuz *et al* [57] presented a scalable CBIR technique based on the vector wavelet coefficients of color images. Highly decorrelated wavelet coefficient planes were used to acquire a search-efficient feature space. It was claimed that the feature key of an image in this approach corresponds to not only the image itself but also how much the image is different from other images in the database. Query searching time was found to be less than 30 msec for a 5000-image database.

Regentova, Latifi *et al.* [58] proposed a method to evaluate the similarity of wavelet compressed images. Two features that describe the image structural content, edge point locations and edge density, were computed directly from multi-scale data. Depending on the image type and the feature selections for processing, the distance between two images was computed in one or two-dimensional space. The measurements were performed using blocks of image data.

Ardizzoni *et al* [59] proposed WINDSURF (Wavelet-based Indexing of Image Using Region Fragmentation), which uses the wavelet transform to extract color and texture features from an image and applies a clustering technique to partition the image into a set of "homogeneous" regions. Similarity between images is assessed by using the Bhattacharyya distance to compare region descriptors, and then combining the results at the image level.

Despite the methods described above, there are still many techniques, based on the DWT, proposed in the last few years. Three kinds of them, which have close relationship with this thesis, will be overviewed in much more detail in the next chapter.

2.3.2.5 Vector Quantization

Vector quantization (VQ) has been used for image compression for many years. It compresses images by coding vectors instead of scalars, leading to better performance. Here, a block of image pixels, called an image vector, is represented by an identification number. For image decompression, this identification number is mapped back to its corresponding image vector, which approximates the original image vector. Vector quantization can be defined as a mapping Q of K-dimensional Euclidean space R^K into a finite into a finite subset Y of R^K:

$$Q : R^K \rightarrow Y$$

where Y = (x'_i; i = 1,2,...N), x'_i is the i^{th} vector in Y and N is the number of codewords in the codebook. Y is the set of reproduction vectors and is called the VQ codebook or VQ table.

A complete VQ compression and decompression consists of these processes: (1) Codebook generation, with each entry of the codebook identified by its index number. Similar types of images may share the same codebook to achieve a high compression ratio. (2) Dividing the image into blocks → vectors. (3) For each vector, find its best matching entry in codebook and the compressed bitstream is obtained as a sequence of index numbers. (4) Transmission of the codebook and also the compressed bitstream to the decoder which can then reconstruct the original image.

Two image indexing techniques were proposed by Idris and Panchanathan [60] using vector quantization. In the first technique, for each codeword in the codebook, a histogram is generated and stored along with the codeword. The superposition of the histograms of the codewords is considered to be a close approximation of the histogram of the image and thus these histograms are used as indices to store and retrieve the images. In the second technique, the histogram of the labels of an image is used as an index. The average retrieval efficiencies are 95% and 94% at compression ratios of 16:1 and 64:1, respectively. The performance of the second technique was also compared with the histogram of the DC coefficients in JPEG-compressed images.

Vellaikal *et al.* [61] applied the VQ technique for content-based retrieval of remote sensed images. The feature representation used is formed by clustering spatial-local pixels, and the cluster features are used to process several types of queries which are expected to occur frequently in the context of remote sensing image retrieval.

Zhu [62] proposed an approach, called *keyblock*, generated by clustering algorithms. Each image is encoded as 1-D index codes of the keyblocks in the codebook. The keyblocks are selected by expanding the approaches developed in VQ for compression purpose. The keyblock-based approach is arguably the first work towards establishing a general framework of information retrieval in the image domain.

2.3.2.6 Fractal Compression

Fractal image compression is based on the mathematical results of iterated function systems (IFS) [63, 64]. The image is partitioned into a collection of non-overlapping regions, called *range blocks*. For each such image block, the encoder finds a larger block within the same image (termed the *domain block*) such that a transformation of this block is the best approximation of the range block, according to a certain criterion of similarity calculation. The fractal code contains the geometrical positions of both the range block and the domain block, and the transformation as well. The original image can be reconstructed approximately from a finite number of iterations from its fractal codes. Since the fractal codes used to represent the image pixel data are much more compact, the fraction image representation can reach a high compression ratio. The transformations of an IFS have the following general form:

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & \alpha_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ s_i \\ \beta_i \end{bmatrix} \qquad (2.20)$$

where w_i is the transformation, (x, y, z) is a region point to be encoded, with x, y being the spatial coordinates and z the grey level; the coefficient α_i represents the grayscale factor, e_i and s_i translate the point in the spatial domain, whereas β_i translates it in the grayscale domain. The coefficients a_i, b_i, c_i, and d_i represent both the geometric contraction and the isometric transformation, including identity, orthogonal reflections (about axis and diagonals), and rotations around center of block through different angles.

Vissac [65] *et al* presented a fractals-inspired approach for CBIR, in which similarity of images is measured by block matching after optimal (geometric, photometric, etc.) transformation. This block matching method consists of localized optimization and Viterbi algorithm is further employed to ensure the continuity and coherence of the localized block matching results. The preliminary evaluation on a logo subset of MPEG-7 database was performed.

Nappi *et al* [66, 67] proposed an image index based upon the fractal framework of the IFS. The image index is represented through a vector of numeric features, corresponding to Contractive Functions (CF) of the IFS framework. In order to improve performance during index construction and image retrieval, index dimensionality is reduced by using the DFT.

Zhang *et al* [68] introduced a joint coding approach between images to cluster images. The similarity between two images (M_1 and M_2) is identified by performing the fractal, texture-based coding of one image (M_1) using both two images (M_1 and M_2). They also compared the performance of wavelet and fractals in image retrieval [69]. Mean absolute value and variance of different subbands are used as the image features for the wavelet domain. Based on the simulation results, it was concluded that wavelets are more effective for images which contain strong texture features, while fractals perform relatively well for various types of images.

2.3.2.7 Hybrid Schemes

Hybrid schemes here refer to a combination of two or more basic coding schemes (*e.g.*, DCT and VQ). Idris and Panchanathan [70] proposed a new technique based on wavelet vector quantization for the storage and retrieval of compressed images. The images were first decomposed using DWT followed by vector quantization of the transform coefficients. The codebook labels corresponding to an image constitute a feature vector that can be used as an index to store and retrieve the image.

Sabharwal and Subramanya [71] have proposed a hybrid technique using the compression power of wavelet transform and RSTN invariance of Fourier transform spectrum. This hybrid scheme is shown to be faster and more robust than the separate use of the DFT and DWT.

2.3.3 Comments

It is difficult to comment and compare so many techniques in the CBIR field. Techniques in the pixel domain are not very popular because of their high computational complexity. Direct feature extraction from the compressed image contents is preferred. The compression techniques used in some degree determine the success and popularity of an indexing technique. KLT or PCA, which is statistically optimal, is computationally intensive. The DCT in JPEG and the DFT have some disadvantages (already mentioned before) compared to the DWT. DWT-based techniques are promising for indexing applications because of (1) inherent multi-resolution capability; (2) simple edge and shape detection; and (3) readily available direction information. The author believes that popularity of DWT will increase in future because of its adaptation into the JPEG2000 standard.

Chapter 3 EXPERIMENTAL SET UP

In this chapter, the experimental environment, in which image indexing performance is to be evaluated, is introduced. These include the image database used to test all the indexing techniques and the program run-time environment. Meanwhile, necessary assumptions, definitions and annotations needed for index extraction and retrieval evaluation is also stated here so that the indexing techniques can be represented clearly and intelligibly.

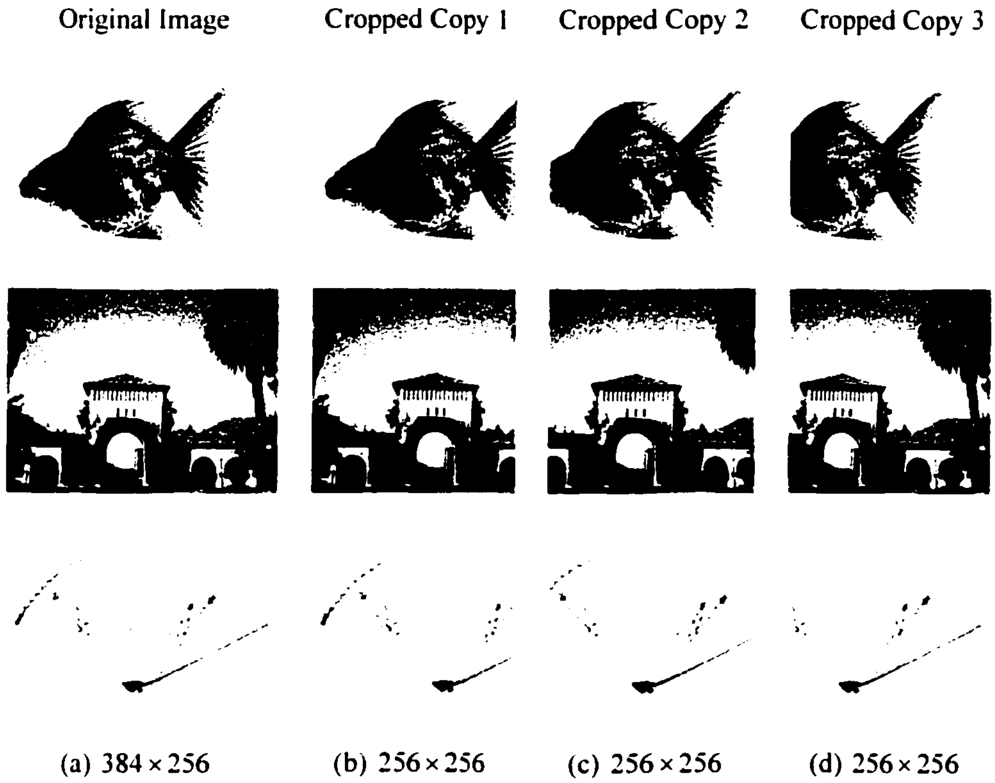
3.1 Experimental Environments

The experimental image database, which is a subset of MPEG-7 test database, contains 3000 color images. The images are in BPPM format which represents images using interleaved RGB color components.

It is difficult to set an objective criterion for performance evaluation with respect to a given indexing technique since the similarity between two images is subjective. The best way to evaluate an indexing technique is to present a query image and verify manually that the retrieved images are indeed similar to the query image. However, for a system-wide evaluation, the above method is too time consuming, and we need a better criteria to evaluate all the techniques we have.

In this thesis, a simple method is used. The database was created such that each image has two similar images in the database. This is done by cropping three smaller copies of an original image with different offsets against the upper-left corner of the original image. The three cropped images have the same dimensions in both width and height. The created database contains all the cropped images but omits the original images. Figure 3-1 illustrates the establishment of the experimental database. The images in the first column are the original images from the MPEG-7 test database, they are of size 384×256, while all their cropped copies are of size 256×256. The images in the 2nd column are cropped from those in the 1st column with offset $(0,0)$, the 3rd column with offset (64,0), and the 4th columns with offset (128,0). At last, the images in column 2, 3, and 4 are combined to construct the experimental database. Hence, in this database, each image has two similar copies, as mentioned above. When any image in this database is used as a query image, if one or both of its similar copies appear in the retrieved image list, then this set retrieval can be regarded as partial or complete success, respectively.

Meanwhile, for the evaluation, each color component of any image in the database is decomposed using Daubechies-6 orthogonal wavelets.



Original Image Cropped Copy 1 Cropped Copy 2 Cropped Copy 3

(a) 384 × 256 (b) 256 × 256 (c) 256 × 256 (d) 256 × 256

Figure 3-1 Illustration of establishment of experimental database. Original images were cropped to generate a series of cropped copies. (a) original images in size of 384 × 256 ; (b) cropped copy 1 with offset (0,0) ; (c) cropped copy 1 with offset (64,0) ; (d) cropped copy 1 with offset (128,0) ;

The indexing techniques were evaluated on two Linux machines: one is a PIII-500 MHz and the other is a PIII-2×750 MHz (dual 750 MHz), both using 256 MB memory.

3.2 Definitions and Annotations

3.2.1 Wavelet Decomposition Level and Resolution

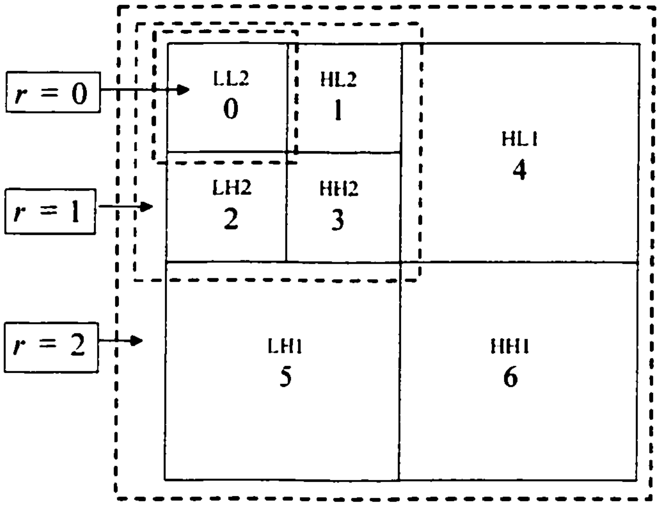


Figure 3-2 Illustration of resolution and subband. The wavelet decomposition level is n. r denotes resolution, where $r = 0$ corresponds to the lowest (coarsest) resolution (subband LL), and $r = n$ – the highest (finest) resolution.

Assume that an image is an n-level wavelet decomposed, as shown in Figure 3-2. There will be $n+1$ resolutions:

$$[0,\cdots,r,\cdots,n] \qquad (0 \le r \le n) \tag{3.1}$$

$r = 0$ – the lowest (coarsest) resolution (subband LL)

$r = n$ – the highest (finest) resolution

and $3n+1$ subbands:

$$[0,\cdots,b,\cdots,3n] \qquad (0 \le b \le 3n) \tag{3.2}$$

where the value of subband index b increases from HL→LH→HH in that order within a resolution and from lower resolution to higher resolution, $b = 0$ corresponds the subband in the lowest resolution (LL), and $b = 3n$ corresponds to the HH subband in the highest resolution. Hereafter, let DL denote the largest wavelet decomposition level for convenience, *i.e.*, $DL = n$ in this assumption.

3.2.2 Distance Metric for Image Retrieval

For image retrieval, an index of a query image is compared with the corresponding indices of all candidate images from the database.

In this work, for distance calculation between indices which are histograms or vectors, the L^1 Euclidian distance is employed, which is defined as follows. Assume a query image Q and a candidate image C, the distance metric:

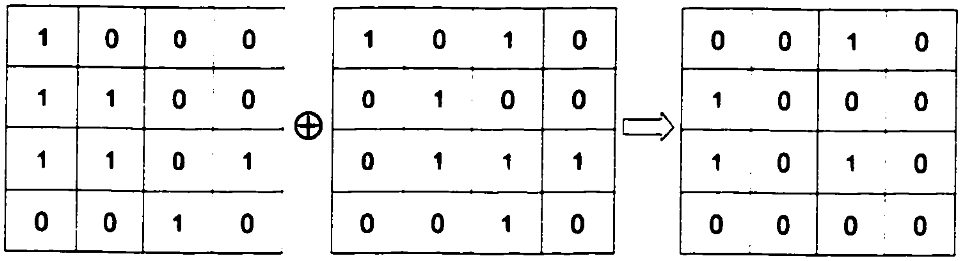
$$D(Q,C) = \sum_{k=0}^{B-1} w(k)\left|h^Q(k) - h^C(k)\right| \tag{3.3}$$

B – the number of elements in the index.

$w(k)$ – the weight for element k.

Eq. (3.3) is useful for indices such as histograms and vectors. However, when a "bit map", which is a 2-D matrix composed of '0' and '1' values, is used as an index of an image, the similarity metric is different from Eq. (3.3). It can be obtained by applying a bitwise exclusive OR (XOR) operation on two bit maps Γ_n^Q and Γ_n^C followed by counting the number of '1' from the result of XOR (see Figure 3-3). Assume a bit map is denoted as Γ^X, where X is query image Q or candidate image C, mathematically:

$$D(Q,C) = \text{count of '1's after } \left(\Gamma^Q \oplus \Gamma^C\right) \tag{3.4}$$



1	0	0	0
1	1	0	0
1	1	0	1
0	0	1	0

(a) index *Q*

1	0	1	0
0	1	0	0
0	1	1	1
0	0	1	0

(b) index *C*

0	0	1	0
1	0	0	0
1	0	1	0
0	0	0	0

(c) distance

Figure 3-3 Illustration of XOR operations of two bit maps. (a) bit map for image Q; (b) bit map for image C; (c) bit map after bit map Q exclusive OR bit map C

3.2.3 Computational Complexity for Distance Calculation

One criterion to evaluate the efficiency of an algorithm is to find out the computational complexity of the algorithm. Due to the various types and performance of current computers, it is difficult to judge an algorithm in time units. In this thesis, to compare the computational complexity and retrieval efficiency of all the proposed algorithms, we use the average number of operations in each distance calculation (*i.e.*, distance calculation between two image indices).

Because the running time for additions/subtractions multiplications and bitwise exclusive OR are different, for Index-α, we denote the number of multiplication operations as $O_\alpha(\times)$, addition/subtraction as $O_\alpha(\pm)$ (an addition operation is regarded as costly as a subtraction operation), and bitwise exclusive OR (XOR) as $O_\alpha(\oplus)$. Simultaneously, the algorithm running time for each query image with 60 images retrieved from a 3000-image database, in which 3-level wavelet decomposition is applied to all images, will be given as well for illustration.

3.2.4 Retrieval Efficiency and Tolerance

The performance of each reviewed or proposed technique is expressed in terms of the retrieval efficiency, η, as defined below. For each image x in a database of size K, let

M_x, $1 \le x \le K$, be the number of images known to be similar to image x. An image q is considered the query image, and T images are retrieved from the database. If m_q is the number of successfully retrieved images, the retrieval efficiency η is defined as:

$$\eta = \frac{\sum_{q=0}^{K} m_q}{\sum_{q=0}^{K} M_q} \tag{3.5}$$

where m_q is equal to or less than the total number of images similar to the query one.

During the experiments, the query image is chosen arbitrarily from the database, while all other images in the database are considered to be candidate images to be matched with the query image. No indexing technique is perfect. Given a query image, a specified percent T of the number of images are retrieved using the smallest distance criterion. Here, T can be considered as a tolerance factor due to the imperfection of the indexing technique. If there are two similar copies for each image, the retrieval is considered 100%, 50%, or 0% successful if the retrieved image set (consisting of T percent of the image number) contains both, only one, or none of the similar images, respectively.

Chapter 4 Review of Indexing Techniques in the Wavelet Domain

As introduced in Chapter 2, the wavelet transform provides many useful features for image processing. For example, 1) not only frequency but also spatial information can be exploited in the wavelet coefficients; 2) most of the power in the wavelet coefficients is concentrated in a few lower subbands for most of natural images; 3) the subbands of wavelet-decomposed image progressively represent the image. Meanwhile, image indexing techniques in the wavelet domain provides one solution to index images in large volume databases which must be compressed in order to save space. Therefore, many indexing techniques have been proposed by exploiting these features.

In this chapter, three indexing techniques previously proposed in the wavelet domain, which related to the proposed techniques or useful as references in evaluating the proposed ones, are discussed and reviewed below.

4.1 Histogram of the Number of Significant Coefficients

Liang and Kuo [72] proposed an indexing technique in the embedded zerotree wavelet framework, in which the histogram of the number of significant coefficients (henceforth referred to as Index-NSCH) is used as an index. Embedded zerotree wavelet coding is a popularly used compression algorithm in the wavelet domain. In this section, this algorithm will be briefly reviewed first followed by the introduction of Index-NSCH.

4.1.1 Embedded Zerotree Wavelet Coding

The embedded zerotree wavelet (EZW) technique originated from Shapiro [73] for image compression. Two main concepts in the EZW algorithm should be emphasized here: *progressive encoding* and *zerotree encoding*.

Progressive encoding, also known as embedded encoding, is used to compress an image into a bit stream with increasing accuracy. That is, when more bits are added to the stream, the decoded image contains more detail. Progressive encoding makes it possible to represent the coded image in multi-resolution.

The concept of *zerotree encoding* is based on the following two assumptions: First, natural images have a low-pass spectrum, *i.e.*, the energy of a wavelet-transformed image is concentrated in coarser scales. Second, larger wavelet coefficients are more important than the smaller coefficients irrespective of the scales where they occur. The significance of a wavelet coefficient $Coef$ is determined by comparing it with a specified threshold TH. Here, only the absolute value of $Coef$, without considering its sign, is compared to TH. If $|Coef| \geq TH$, $Coef$ is considered to be significant; otherwise insignificant (coefficients are possibly negative if the original pixel values of an image are normalized to their mean before applying the wavelet transform). Consequently, only the significant coefficients (SC) are stored, while all the other coefficients are discarded, to achieve high compression.

In EZW encoding, the choice of thresholds to determine SCs is crucial to the quality of the reconstructed image at various resolutions. Because the term *resolution* has been used for denoting wavelet decomposition level, here we use *threshold-level* to represent the quality level of the reconstructed image, resulting from using different levels of thresholds for filtering away insignificant coefficients. Each threshold-level is applied to output the corresponding one of the successive approximate representations of the reconstructed image. Higher threshold leads to fewer SCs, thus a poor quality reconstructed image, and vice versa (see Figure 4-1). Assume the threshold-level (referred to as *TL* hereafter) has the value of *i*, where *i* is an positive integer. Then the thresholds, one for each *TL*, are calculated as follows:

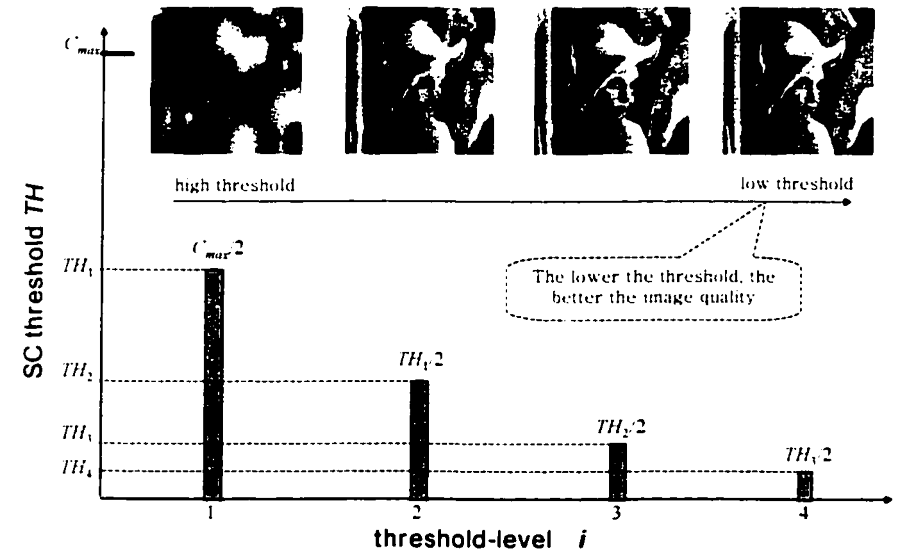


Figure 4-1 Representation of a reconstructed Lena image with respect to four successive threshold levels.

$$\begin{aligned} TH_1 &= |Coef|_{\max}/2 \\ TH_i &= TH_{i-1}/2 \qquad (i > 1) \end{aligned} \tag{4.1}$$

$|Coef|_{\max}$ – maximum value of the DWT coefficients

TH_i – threshold for i^{th} threshold-level

Due to the high proportion of insignificant coefficients in practice, image compression using EZW encoding generally provides reconstructed images of good subjective quality even at a high compression ratio.

In EZW coding, an image is represented by wavelet SC maps at various threshold-levels. These SC maps are used progressively (from lower to higher threshold-level), or jointly to reconstruct an image. Since the SC maps refer to the location of high-magnitude DWT coefficients, they provide key features of an image. Hence, an SC map can be employed to generate an index of an image. In order to exploit the multi-resolution capability of wavelets for progressive retrieval, SC maps of different successive threshold-levels are employed to generate the image index.

4.1.2 Index-NSCH Generation

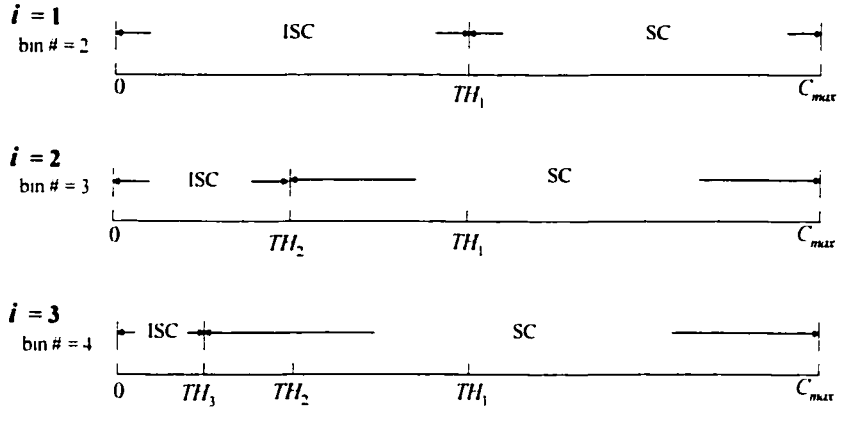


Figure 4-2 Relationship of histogram bin and threshold in Index-NSCH. There are three histograms, each corresponding to a threshold-level. For $TL = 1, 2$ and 3, **the number of bins in the histograms is 2,3,4, respectively. SC: significant coefficients; ISC: insignificant coefficients.**

Index-NSCH, proposed in [72], is a histogram of the number of significant/insignificant wavelet coefficients with respect to a series of successively decreased thresholds. The number of bins of a histogram corresponding to threshold TH_i, or equivalently, $TL = i$, is illustrated in Figure 4-2. From the figure, the number of bins is equal to $i+1$. For any bin b other than $b = i+1$ (*i.e.*, $1 \le b \le i$), its vertical value is the number of SCs which fall in the range $[TH_{i-1}, TH_i]$. For bin $b = i+1$, its vertical value is the number of total insignificant coefficients. Therefore, for threshold-level $TL = i$, the index histogram $Index_{nsch}^{(i)}$ can be expressed as follows:

$$Index_{nsch}^{(i)} = \left(\rho_{SC}^{(1)}, \cdots, \rho_{SC}^{(b)}, \cdots, \rho_{SC}^{(i)}, \rho_{ISC}\right) \tag{4.2}$$

$\rho_{SC}^{(b)}$ – number of SCs for bin b, $1 \le b \le i$

ρ_{ISC} – number of ISCs for bin $i + 1$

Table 4-1 shows an example of Index-NSCH using a series of thresholds (7 thresholds used here). As defined in Subsection 4.1.1, each threshold corresponds to an threshold-level. The index is extracted from the Lena image with RGB color components in size of 256×256, in each of which 5-level wavelet decomposition is applied. The indices shown in Table 4-1 were obtained from color component G.

This method allows the number of bins of any histogram to be set as small as down to 2 at $TL = 1$, at which the retrieval can be finished very quickly. On the other hand, from Table 4-1, we can see that the number of insignificant coefficients is far more than the number of significant coefficients (about 25 times as large as the largest number of SCs for threshold TH_7), and the large number of insignificant coefficients may interfere with the calculation of distances, resulting in a poor similarity measurement.

Table 4-1 Relationship between the number of histogram bins and the threshold-level for Index-NSCH, which was extracted from the color component G of the Lena image. Here, 7 histograms, each corresponding to a threshold-level, are illustrated in the following. SC: significant coefficient; ISC: insignificant coefficient. ($DL = 5$)

$TL = i$	# of SCs							# of ISCs
1	14							65522
2	14	39						65483
3	14	39	115					65368
4	14	39	115	221				65147
5	14	39	115	221	581			64566
6	14	39	115	221	581	1190		63376
7	14	39	115	221	581	1190	2413	60963

4.1.3 Computational Complexity

The distance for Index-NSCH, $D_{nsch}(Q,C)$, between the query image, Q, and a candidate image, C, of Index-NSCH is calculated using Eq. (3.3). The computational complexity includes subtraction, addition and multiplication operations. For threshold-level $TL = i$, it requires $i+1$ subtractions, i additions and $i+1$ multiplications for each color component. For the three color components RGB, the total number of operations comes to:

$$\begin{cases} O_{nsch}^{(i)}(\pm) = 6i+5 \\ O_{nsch}^{(i)}(\times) = 3i+3 \end{cases} \tag{4.3}$$

The NSCH indexing technique is a fast retrieval technique, though, from Eq. (4.3), the computational complexity increases linearly with the number of threshold levels. The average running time at $DL = 3$ and $TL = 3$ for a query image with 2% images retrieved in a 3000-image database is 23.09 ms.

4.1.4 Performance

The retrieval performance of Index-NSCH is shown in Figure 4-3 for different tolerance T ($T = 1\%$, 2% and 3% , and $DL = 3$) and Figure 4-4 for different wavelet decomposition levels DL ($DL = 3,4$ and 5, and $T = 2\%$).

From Figure 4-3 it is clear that the retrieval efficiency η has a same trend with the increase of threshold at a fixed decomposition level, *e.g.*, $DL = 3$. When $TL \leq 3$, η increases and the slope is steep, while with further increases in TL, *i.e.*, when $TL > 3$, η starts to decrease and drops to a low value at $TL = 7$. This trend occurs because the small number of bins in the first two threshold-levels cannot help to distinguish indices thoroughly. For example, at $TL = 1$, the number of bins for the index histogram is only two, which cannot exhibit the actual SC distribution of an image. With the increases in

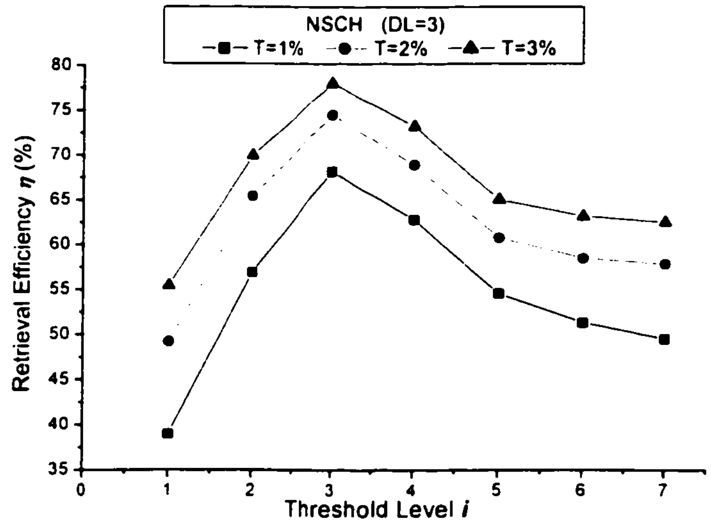


Figure 4-3 Retrieval performance of the NSCH technique. Wavelet decomposition $DL = 3$, tolerance $T = 1\%, 2\%$ and 3% respectively.

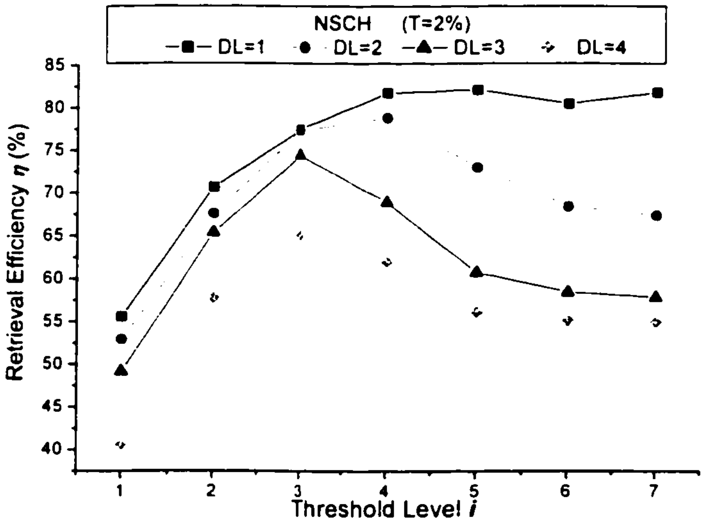


Figure 4-4 Retrieval performance of the NSCH technique. Tolerance $T = 2\%$, wavelet decomposition level $DL = 1 \sim 4$.

TL, the number of bins increases as well which enriches the features available for indexing so as to improve the retrieval efficiency. However, since $TL > 3$, from Table 4-1, the values for new coming bins are much larger values than previously. For instance, at $TL = 7$, the value for bin 7 is 1190, which is 10 times more than the value, 115, for bin 3. The of SCs in bin 7 is less significant than in bin 3. This explains why the performance degrades when $TL > 3$.

In addition, from Figure 4-3, for a certain threshold, η is enhanced about 7-8% with every 1% increase of tolerance, which is reasonable.

On the other hand, η decreases when the decomposition level increases at a fixed tolerance, *e.g.*, $T = 2\%$ (Figure 4-4).

It should be noted that when $DL = 3$ and $T = 2\%$, the maximum retrieval efficiency is about 74% at $TL = 3$. This technique still leaves room for improvement, which will be discussed in the next chapter.

4.2 Moment of Wavelet Coefficients

"Moments are commonly used in statistics to characterize the distribution of random variables, and, similarly, in mechanics to characterize bodies by their spatial distribution of mass"[74]. The method of moments provides the capability to transform an arbitrary shape into a finite set of characteristic features. It has been widely used for image analysis due to its mathematical simplicity and versatility.

Wang *et al.* [54] and Mandal *et al.* [75] have proposed techniques in which statistical properties of subbands of wavelet coefficients are used as indices. Here, the first-order moment (*i.e.*, mean) and the second-order moment (*i.e.*, standard deviation) derived from the wavelet coefficients with respect to each subband are used to build an image index. This index, which to be referred to as Index-WMV hereafter, is a moment vector based on the coefficients in every wavelet subband.

Assuming that subband b includes κ_b coefficients, and $\mu_{wmv,b}$ and $\sigma_{wmv,b}$ denote the mean and standard deviation of the coefficients in subband b, respectively, we have:

$$\mu_{wmv,b} = \sum_{i=1}^{\kappa_b} \frac{Coef_i}{\kappa_b} \tag{4.4}$$

$$\sigma_{wmv,b}^2 = \sum_{i=1}^{\kappa_b} \frac{\left(Coef_i - \mu_{wmv,b}\right)^2}{\kappa_b} \tag{4.5}$$

4.2.1 Index-WMV Generation

Because the wavelet coefficients are distributed in $3n+1$ subbands, and each subband is associated with a pair of $\mu_{wmv,b}$ and $\sigma_{wmv,b}$, a total of $3n+1$ pairs of $\mu_{wmv,b}$ and $\sigma_{wmv,b}$ can be used to construct an index. Index-WMV is actually a moment vector consisting of a mean vector and a standard deviation vector. Let $V_{wmv,\mu}$ and $V_{wmv,\sigma}$ denote vectors of means and vector of standard deviations, respectively. For decomposition level $DL = n$, they can be expressed as follows:

$$V_{wmv,\mu}^{(n)} = \left(\mu_{wmv,0}, \mu_{wmv,1}, \cdots \mu_{wmv,b}, \cdots, \mu_{wmv,3n}\right) \tag{4.6}$$

$$V_{wmv,\sigma}^{(n)} = \left(\sigma_{wmv,0}, \sigma_{wmv,1}, \cdots \sigma_{wmv,b}, \cdots, \sigma_{wmv,3n}\right) \tag{4.7}$$

The Index-WMV at $DL = n$ is then the concatenation of them:

$$Index_{wmv}^{(n)} = \left\{V_{wmv,\mu}^{(n)}, V_{wmv,\sigma}^{(n)}\right\} \tag{4.8}$$

Here we give an index example extracted from the wavelet coefficients of the Lena color image comprised of color RGB component planes of size of 256×256. 5-level wavelet decomposition is applied to each component (R, G, B) independently, leading to 16 subbands for each component. The generated Lena Index-WMVs are showed in Table 4-2. From the table, we find that both the mean and standard deviation are generally descending from lower subbands (in coarser resolutions) to higher subbands (in finer resolutions). It is likely that if multi-subbands are used to generate an index, the higher subbands could contribute less information than the lower subbands.

Table 4-2 Index-WMV extraction from the *Lena* image with 5 decomposition levels, whereas, the number of subbands: $5 \times 3 + 1 = 16$. STDEV: standard deviation; R: red, G: green, B: blue.

Subband	R		G		B	
	Mean	STDEV	Mean	STDEV	Mean	STDEV
0	71	1020	-5	988	-32	627
1	-33	574	-19	648	-12	370
2	-30	246	-3	303	-2	207
3	19	219	46	286	31	193
4	-6	234	-14	276	-6	164
5	-1	96	-3	117	-5	83
6	9	124	9	139	7	89
7	1	93	0	99	0	67
8	0	58	0	59	1	43
9	0	57	0	62	0	44
10	0	46	0	51	0	36
11	0	33	0	35	0	29
12	-1	31	-2	36	-1	26
13	0	23	0	27	0	21
14	0	20	0	21	0	18
15	0	15	0	17	0	15

4.2.2 Computational Complexity

For Index-WMV at $DL = n$, the distance $D_{wmv}(Q,C)$ is the difference between the mean and deviation of the query image Q and a candidate image C, which can be computed from Eq. (3.3).

Again, the computational complexity of Index-WMV can be analyzed. All the calculating consists of subtraction, addition and multiplication operations. Different from Index-

NSCH, the number of operations is wavelet-decomposition-level dependent. When DL = n, it requires $2(3n + 1)$ subtractions, $6n + 1$ additions and $2(3n + 1)$ multiplications for a single color component. The total number of operations for 3 color components can be obtained using the following equation:

$$\begin{cases} O_{wmv}^{(n)}(\pm) = 36n + 11 \\ O_{wmv}^{(n)}(\times) = 6n + 2 \end{cases} \tag{4.9}$$

It is observed from the Eq. (4.9) that the WMV indexing technique is still a relatively fast retrieval technique though the computational complexity increases linearly with the number of decomposition levels. The average running time at $DL = 3$ for a query image with 60 images retrieved from a 3000-image database is about 71 ms.

4.2.3 Performance

The retrieval performance of Index-WMV is shown in Table 4-3 for different tolerances T ($T = 1\%, 2\%$ and 3% , and $DL = 3$) and for different decomposition levels ($DL = 3$, 4 and 5, at $T = 2\%$).

Table 4-3 Retrieval efficiency of Index-WMV for various decomposition levels and tolerances.

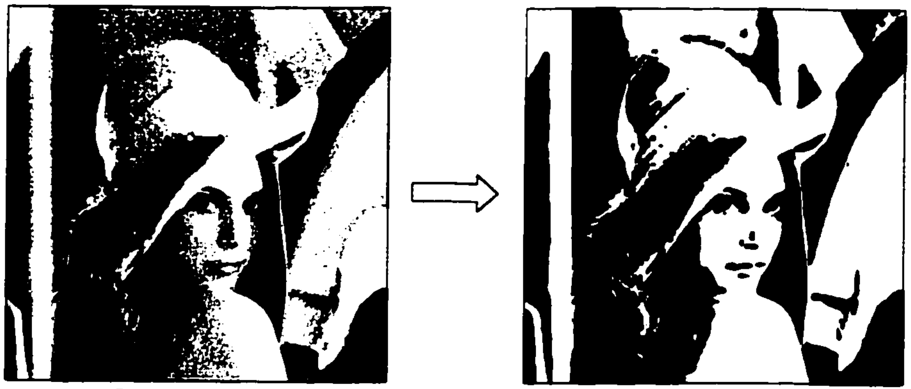
Index-WMV Retrieval Efficiency η (%)					
$DL = 3$			$T = 2\%$		
$T = 1\%$	$T = 2\%$	$T = 3\%$	$DL = 2$	$DL = 3$	$DL = 4$
93.7333	95.4833	96.1333	93.0667	95.4833	96.1

For a certain DL values, the retrieval efficiency is always enhanced with increases in tolerance. On the other hand, the retrieval performance increases when the DL increases at a fixed tolerance, *e.g.*, $T = 2\%$ as shown in Table 4-3.

It should be noted that when $DL = 3$ and $T = 2\%$, the retrieval efficiency is about 95%, which is much higher than that of Index-NSCH.

4.3 Binary Map of Low-Pass Subband

In the wavelet decomposition, the LL subband is a coarse version of the original image. For most natural images, this subband can represent the original image roughly, *i.e.*, it can be regarded as a sub-image of the original one in smaller scale and lower resolution. It is expected that if a binarization with respect to a specific threshold be applied to this sub-image, the binarized map with feature of the original image can be employed to construct an index [72]. This index, referred to as Index-LLBM, generated from LL subband is in the format of 2-D binary map, and so it is a binary representation of the lowest frequency subband LL. Figure 4-5 shows a binarization example generated from the original color Lena image. The outline the image is clear and we can still recognize it clearly.



(a) Color Lena Image

(b) Binarized Lena Image

Figure 4-5 Illustration of Image Binarization: (a) color Lena image; (b) Lena image after binarization. It is observed that the binarized image is actually a black and white image here.

4.3.1 Index-LLBM Generation

Assume the value of any coefficient $|Coef|_x$, (here $|Coef|_{\min} \le |Coef|_x \le |Coef|_{\max}$), for a given threshold TH_t, (also $|Coef|_{\min} \le TH_t \le |Coef|_{\max}$), in the LL-subband can be binarized as follows. By comparing $|Coef|_x$ within the LL-subband with threshold TH_t, if $|Coef|_r \geq TH_t$, $Coef_r$ is set to 1, otherwise, 0. Figure 4-6 shows an example of the process of binarization. In the figure, threshold $TH = 15$ is assumed to apply to a LL band, so that for those coefficients greater than 15 are to '1', otherwise, to '0'.

The binarization map of wavelet LL subband is strongly threshold-dependent. The higher threshold, the fewer SCs will be found, leading to fewer '1's after binarization. On the other hand, the lower the threshold, the more SCs, resulting in more '1's. For thresholds smaller than a certain value, which is image-dependent, the binarization map of LL subband may probably composed by nothing but '1's which means that such kind of binarization map will lose ability to distinguish the difference from others. Figure 4-7 shows an example of LL band binarization map with different thresholds applying to a 5-level decomposed Lena image. The 4 thresholds are obtained from Eq.(4.1), in the order $TH_1 > TH_2 > TH_3 > TH_4$.

For example, if an image of size 512×512 is wavelet decomposed, for 3 levels, the LL subband will have a size of 64×64. The binary map corresponding to the LL subband will be:

$$\Gamma_4 = \begin{bmatrix} b(0,0) & b(0,1) & \cdots & \cdots & b(0,63) \\ b(1,0) & \ddots & & & \vdots \\ \vdots & & b(x,y) & & \vdots \\ \vdots & & & \ddots & \vdots \\ b(63,0) & \cdots & \cdots & \cdots & b(63,63) \end{bmatrix} \tag{4.10}$$

where $b(x,y)$ is the binary value at coordinate (x,y) of the map.

Then, Γ_n is used as an index of LLBM technique:

$$Index_{llbm}^{(n)} = \Gamma_n \tag{4.11}$$

n – the number of decomposition levels

Γ_n – LL subband binarization map with respect to $DL = n$

25	10	14	2
30	22	12	8
20	25	9	16
4	8	21	6

(a) original coefficients in LL band

TH = 15

1	0	0	0
1	1	0	0
1	1	0	1
0	0	1	0

(b) Coefficients after binarizaiton

Figure 4-6 Binarization process, a threshold of 15 is used. For those original wavelet coefficients $Coef \geq 15$, they are to 1, others are to 0.

1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	1	0	0

(a) *TH*1

1	0	0	1	1	1	1	0
1	0	0	0	1	0	1	1
0	0	0	1	1	1	0	1
0	0	1	0	1	1	1	0
0	0	0	1	0	1	1	1
1	0	1	1	1	0	1	1
1	0	1	1	0	0	0	1
1	0	1	1	1	1	0	0

(b) *TH*2

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1
1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	1	1	1	1	0	1

(c) *TH*3

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

(d) *TH*4

Figure 4-7 LL band binarization map with respect to successive thresholds. Here, $TH_1 > TH_2 > TH_3 > TH_4$.

To more conveniently compare the proposed techniques to the LLBM technique in the EZW framework, we apply multiple thresholds obtained in Eq. (4.1) to wavelet coefficients, and generate LL subband binarization maps with respect to each threshold. Thus 7 sets of indices are generated corresponding to 7-level thresholds.

4.3.2 Calculation Complexity

Index-LLBM is a "bit-map", and therefore its distance can be obtained using Eq. (3.4). The calculation of index distance for Index-LLBM includes addition and bit-wise exclusive OR (XOR) operations. The number of operations depends on both wavelet decomposition level and image dimension. Assume the dimensions of an image are $X_{siz} \times Y_{siz}$. For $DL = n$, the complexity is calculated as follows for 3 color components:

$$\begin{cases} O_{llbm}^{(n)}(\pm) = 3\left(\dfrac{X_{siz} \times Y_{siz}}{(2^2)^n}\right) \\ O_{llbm}^{(n)}(\oplus) = 3\left(\dfrac{X_{siz} \times Y_{siz}}{(2^2)^n}\right) \end{cases} \tag{4.12}$$

From Eq. (4.12), the number of operations increases in the power of 2^2 with the decrease of decomposition-level DL. When $DL = 1$, the size of the LL subband sub-image is one-fourth of the original one, which takes a considerably longer time to calculate.

The average running time at $DL = 3$ for a query image with 60-image retrieved in a 3000-image database is about 2370 ms.

4.3.3 Performance

The retrieval performance of Index-LLBM is shown in Figure 4-8 for different tolerances T ($T = 1\%$, 2%, and 3% , and $DL = 3$) and Figure 4-9 for different wavelet decomposition levels DL ($DL = 3$, 4 and 5, and $T = 2\%$).

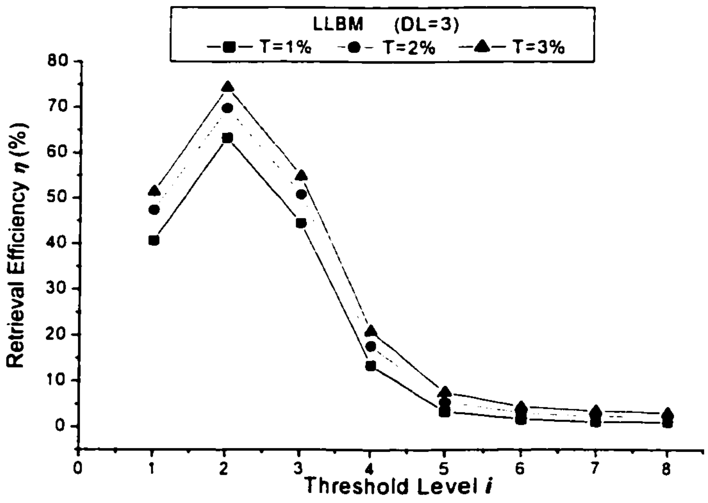


Figure 4-8 Retrieval performance of Index-LLBM. Wavelet decomposition DL=3, and tolerance T=1%, 2% and 3%, respectively.

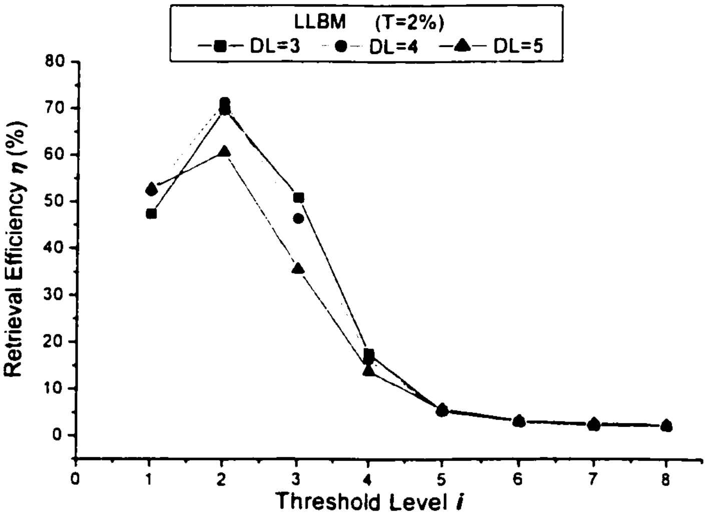


Figure 4-9 Retrieval performance of Index-LLBM. Tolerance T=2%, and DL=3, 4 and 5, respectively.

From Figure 4-8, it is clear that for a given decomposition level DL and tolerance T, with increasing threshold-level TL, the retrieval efficiency η increases when $TL \leq 2$. When $TL > 2$, η drops sharply even down to no more than 10%, which proves the analysis in Subsection 4.3.1, *i.e.* the lower the threshold, the more SCs so that all '1's occupy the binarization map. All '1' binarization maps make images indistinguishable from each other so that the retrieval performance becomes meaningless.

Again, for a certain DL and TL, the retrieval efficiency is always enhanced with the increase of tolerance (Figure 4-8). On the other hand, the retrieval performance is almost independent on the DL at a fixed tolerance, *e.g.*, $T = 2\%$ as shown in Figure 4-9.

In addition, at $DL = 3$ and $T = 2\%$, the maximum retrieval efficiency is about 70%.

4.4 Comparison and Summary

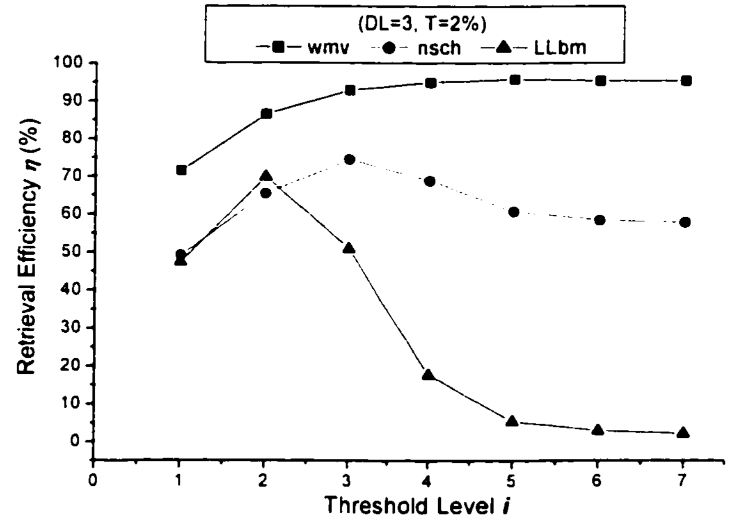


Figure 4-10 Retrieval performance comparison of Index-NSCH, Index-WMV, and Index-LLBM. Here, $DL = 3$ and $T = 2\%$.

Table 4-4 Computational complexity of Index-NSCH, Index-WMV and Index-WMV at $DL = 3$, and $TL = i$, where $i = 1 - 7$, image size is 256×256

i		1	2	3	4	5	6	7
$O_{nsch}(\ )$	±	11	17	23	29	35	41	47
$O_{wmv}(\ )$	±	119	119	119	119	119	119	119
$O_{llbm}^{(n)}(\ )$	±	3096	3096	3096	3096	3096	3096	3096
	⊕	3096	3096	3096	3096	3096	3096	3096

So far, we have introduced 3 indexing techniques in the wavelet-based framework. The indexing performance of these techniques is evaluated below.

Figure 4-10 shows the retrieval performance for the 3 indices at $DL = 3$ and $T = 2\%$. Index-WMV provides the best performance over the other two. It can be used as reference to evaluate the proposed techniques. Index-LLBM has retrieval efficiency close to Index-NSCH only at the first 2 threshold-levels. Due to the large complexity and low retrieval efficiency, Index-LLBM is not suitable to be employed independently for the purpose of retrieval. However, it might be helpful to improve the performance by using Index-LLBM jointly with other indices. On the other hand, Index-NSCH can also be improved to achieve higher performance.

For the computational complexity, we set weights for all indices to '1' so that there is no multiplication operation considered in the process of analysis for simplicity. The computational complexity, at $DL = 3$ and $T = 2\%$, and from $TL = 1$ through $TL = 7$, for all the techniques is tabulated in Table 4-4. From Table 4-4, when the decomposition level is fixed, the complexity for Index-NSCH is the smallest but increases linearly with TL. For Index-WMV, it is small enough though a little higher than that for Index-NSCH. Finally, for Index-LLBM, the complexity is considerably larger compared to the former two (about one magnitude higher at $DL = 3$). Additionally, the complexity for both Index-NSCH and Index-LLBM is independent of TL. From the viewpoint of complexity, both Index-NSCH and Index-WMV are selectable under most situations; however, Index-LLBM is practical only for large DL (how large the quantitive value of DL is determined by the actual image size).

Chapter 5 INDEXING IN THE EMBEDDED WAVELET FRAMEWORK

We have reviewed the EZW coding algorithm proposed by Shapiro [73], in which the progressive coding is implemented by setting a series of monotonically decreasing thresholds. More details are added to the coding sequence with each decrease of the threshold. A lower threshold decreases the noise and therefore increases the signal-noise-ratio (SNR). For a reconstructed image, the lower the threshold used, the better the quality that can be obtained. In another word, the EZW coding algorithm embeds an *SNR progression*.

Liang and Kuo [72] proposed the Index-NSCH for indexing in the EZW framework. Here, the number of significant coefficients with respect to a threshold is used to generate an index of the image. The generation of Index-NSCH has taken advantage of the progressive property of EZW. However, there is further scope for improvement. In this chapter, two individual indices based on the EZW framework are presented in Section 5.1 and 5.2, respectively, followed by analysis of the combination indices in Section 5.3. In Section 5.4, a comparison of all indices mentioned in this chapter will be given, followed by a summary in the end.

5.1 Modified Histogram of the Number of Significant Coefficients

The Index-NSCH proposed by Liang and Kuo [72] is a relatively coarse-resolution histogram which has only a few bins (see Section 4.1). For example, there exists only 4 bins when using a 3-level threshold TH_3, which is not enough to describe the feature of an image. In this section, a new index, a modified version of Index-NSCH (referred to as Index-MNSCH) is proposed below [76].

5.1.1 Index-MNSCH Generation

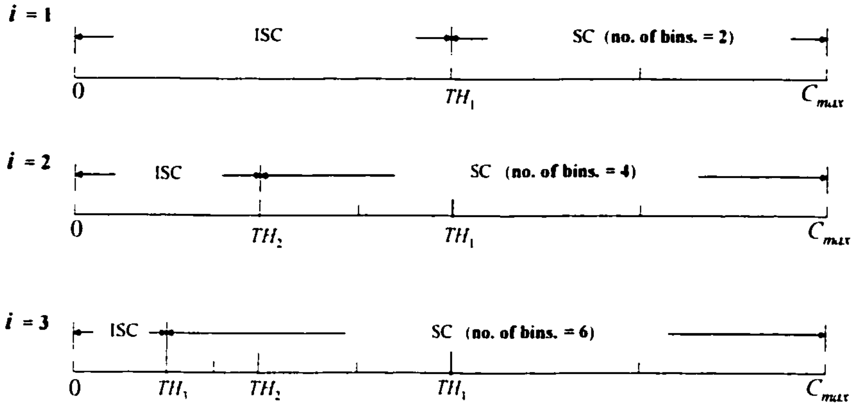


Figure 5-1 Number of histogram bins with respect to threshold-level $TL = i$. Here, $i = 1 \sim 3$. SC: significant coefficients; ISC: insignificant coefficients.

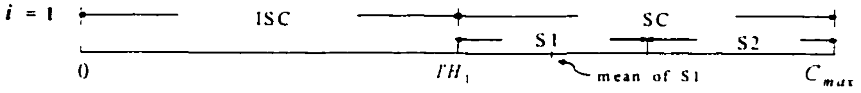


Figure 5-2 Illustration of the segmentation of SCs. For wavelet coefficients fall in two adjacent thresholds TH_1 and C_{max}, where $TH_1 = C_{max}/2$, they are divided into two groups further, i.e., the group 1 is in the range of $[TH_1, (TH_1 + C_{max})/2]$, and the group 2 is in the range of $[(TH_1 + C_{max})/2, C_{max}]$.

Index-MNSCH, derived from the overall significant wavelet coefficients, is a histogram of the number of significant coefficients. The histogram bin for insignificant coefficients in Index-MNSCH is taken away because the relatively large number of insignificant coefficients compared to SCs (especially when threshold is higher) can interfere with accuracy of the distance calculation.

For threshold-level $TL = i$, the range of absolute values of SCs, which is $[TH_i, Coef_{\max}]$, (see Section 4.1), is divided into several non-overlapping segments such that there are always two segments with equal interval between every two adjacent thresholds (Figure 5-1). Thus, the number of segments of the range of SCs is related to the value of TL. From Figure 5-1, the number of segments is always twice the threshold-level i, *i.e.*, $2i$. These segments obtained as above are used as bins of the SC histogram now so that the number of bins for the SC histogram corresponding to threshold-level i is identical to $2i$. The interval size of each segment is the width of a bin. The number of histogram bins obtained in such a way is always larger than that of Index-NSCH except at $i=1$. For example, when $i=1$, there are two histogram bins. For second threshold-level ($i=2$), there are 4 bins in Index-MNSCH instead of 3 bins in Index-NSCH. Hence, it is expected that the indexing performance could be enhanced.

Unlike the NSCH technique, in which the sign of coefficients are ignored in the course of finding ISCs/SCs, the MNSCH technique uses two parallel histograms with exactly the same division and number of bins: one for positive SCs and the other for negative ones. We call the two histograms the positive histogram $H_{mnsch,p}$ and the negative histogram $H_{mnsch,n}$, respectively. For example, in Figure 5-2, only one threshold is used to identify SCs. The both positive and negative histograms include 2 bins named S_1 and S_2. If a positive coefficient falls in bin S_1, the count for bin S_1 in $H_{mnsch,p}$ is increased by 1. If its absolute value of a negative coefficient falls in bin S_1, the count for bin S_1 in $H_{mnsch,n}$ is increased by 1.

The $H_{mnsch,p}$ and $H_{mnsch,n}$ at threshold-level $TL=i$ are defined as:

$$\begin{aligned} H_{mnsch,p}{}^{(i)} &= \left(h_{pi}(1),\cdots,h_{pi}(r),\cdots,h_{pi}(2i)\right) \\ H_{mnsch,n}{}^{(i)} &= \left(h_{ni}(1),\cdots,h_{ni}(r),\cdots,h_{ni}(2i)\right) \end{aligned} \tag{5.1}$$

where $h_i(r)$ is the number SCs in the r^{th} bin and $1 \le r \le 2i$.

Index-MNSCH at $TL=i$ is defined as:

$$Index_{mnsch}{}^{(i)} = \left\{H_{mnsch,p}{}^{(i)}, H_{mnsch,n}{}^{(i)}\right\} \tag{5.2}$$

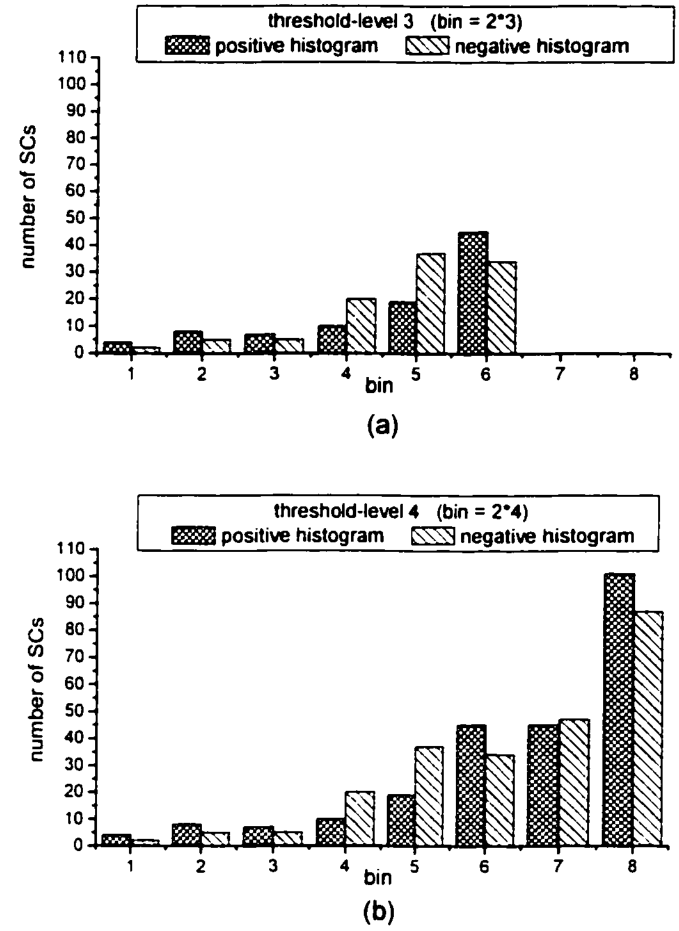


Figure 5-3 Index-MNSCH histograms for threshold-level (a) $TL = 3$ and (b) $TL = 4$.

The SC histograms for $TL = 3$ and $TL = 4$ obtained from the Lena image are illustrated in Figure 5-3. We know that the first 6 bins at $TL = 4$ are exactly the same as the bins at $TL = 3$ in order and value, and the histogram at $TL = 4$ is comprised of histogram at $TL = 3$ with 2 bins appended.

5.1.2 Computational Complexity

Considering one color component, the distance of indices between the query image Q and the candidate image C for threshold-level i is the summation of distance for $H_{mnsch,p}$ and distance for $H_{mnsch,n}$, using Eq. (3.3).

The computational complexity of retrieval depends only on the number of bins of SC histogram for Index-MNSCH, which is determined by TL. Therefore, the complexity is considered to depend on TL only. For each component, when $TL = i$, we require $2\beta_i$ subtractions, $\beta_i + \beta_i - 1$ additions and $2\beta_i$ multiplications to calculate the distance between the query image and the candidate image. For all 3 components, the total number of operations (*i.e.*, additions/subtractions and multiplications) corresponding to a threshold-level i is:

$$\begin{cases} O_{mnsch}^{(i)}(\pm) = 24i \\ O_{mnsch}^{(i)}(\times) = 12i \end{cases} \tag{5.3}$$

The computational complexity increases linearly with i, and if the weights are set to be unity, $O_{mnsch}^{(i)}(\times)$ can be reduced to 0. However, similar to Index-NSCH, the complexity even for higher TL is still not large, for example, only 168 addition operations are needed at $TL = 7$, and therefore, the computational complexity of Index-MNSCH can be considered always small. At $TL = 3$ and $DL = 3$, the running time for a query image with 60 images retrieved from a 3000-image database is approximately 44.08 ms.

5.1.3 Performance

The retrieval performance of Index-MNSCH is shown in Figure 5-4 for different tolerances T ($T = 1\%, 2\%$ and 3% at $DL = 3$) and Figure 5-5 for different wavelet decomposition levels DL ($DL = 3, 4$ and 5 at $T = 2\%$).

The retrieval efficiency η follows the same trend with increases in threshold at a fixed decomposition level. When $TL \leq 3$, η increases with a steep slope, while with further increases in TL, *i.e.*, when $TL > 3$, η starts to decrease and drops to a bottom line at $TL = 7$. The reason for this trend is similar to Index-NSCH, that is, when TL is low, the number of bins of a corresponding histogram is small as well, the difference between two images can be just compared roughly and results in a low retrieval efficiency. With increases in TL, more bins are added to the histogram such that more details are brought in. However, a higher TL means a lower threshold, and the number of SCs increases exponentially (Figure 5-3), while the increased SCs is less significant than the SCs in the previous bins. Hence, when TL increases to some extent, the predominance of more bins are overwhelmed by interference from a large number of increased SCs. This trend suggests that the threshold-level has to be selected carefully to achieve the best performance.

On the other hand, for a certain TL, η is enhanced on average by 2-3% for every one more percent in the number of retrieved images (Figure 5-4). At a fixed tolerance, η decreases when the decomposition level increases. From Figure 5-5, when $DL = 5$, the performance degrades heavily compared to $DL = 3,4$. This means that Index-MNSCH works effectively just for lower decomposition level $DL \leq 4$. The maximum retrieval efficiency is about 85% found at $DL = 3$ and $T = 2\%$.

In Figure 5-6, the retrieval performance for both Index-NSCH and Index-MNSCH at decomposition level $DL = 3$ and tolerance $T = 2\%$ is illustrated. It shows that Index-MNSCH has an average of 10% improvement in the retrieval efficiency over Index-NSCH. Both of them follow similar trends with increases in TL. Retrieval efficiency η increases when TL is small and reaches maximum at $TL = 3$ or $TL = 4$. After that, η starts to decrease monotonically.

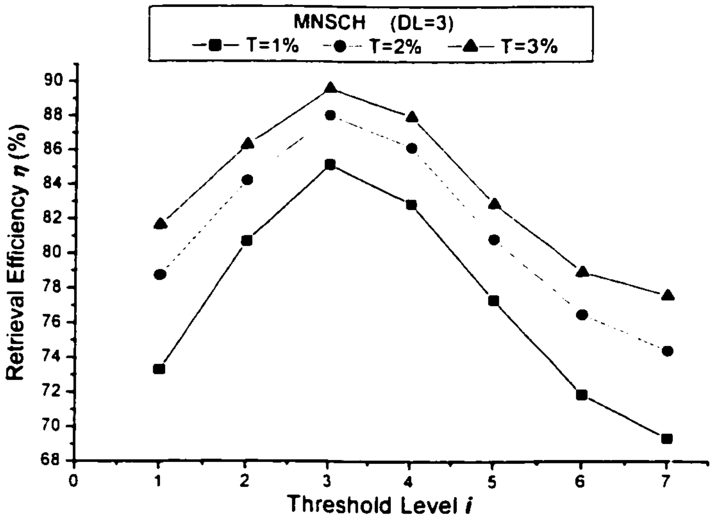


Figure 5-4 Retrieval performance of Index-MNSCH with *DL* = 3 **and** *T* = 1%, 2% and 3%

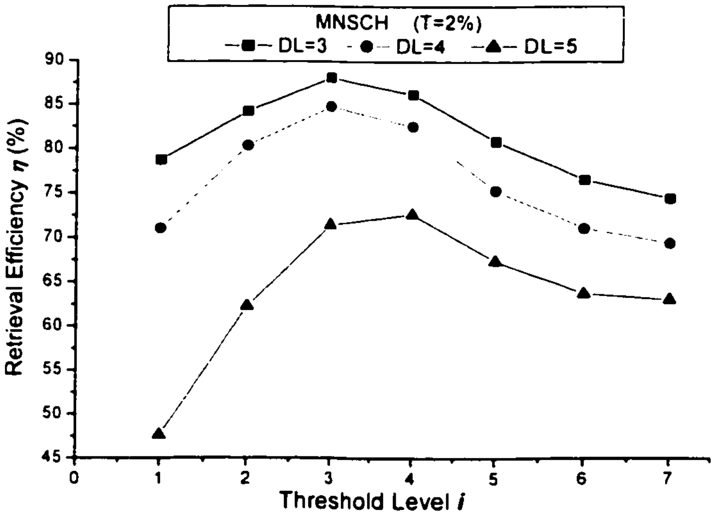


Figure 5-5 Retrieval Performance of Index-MNSCH at *T* = 2% **and** *DL* = 3, 4 and 5

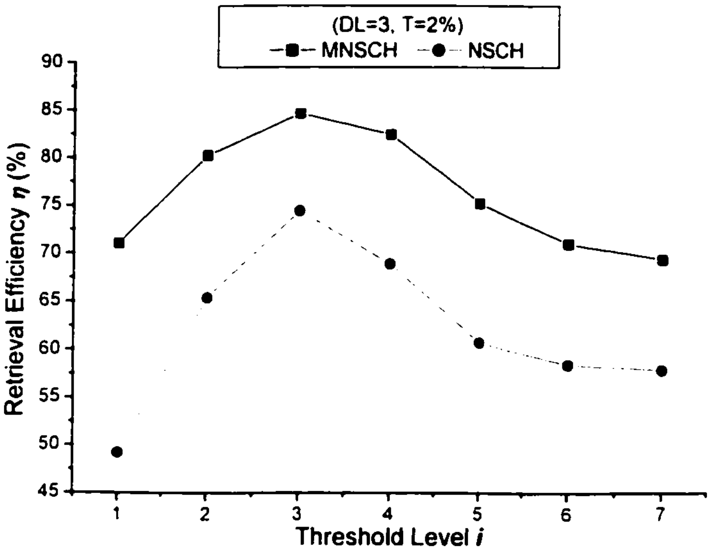


Figure 5-6 Comparison of retrieval performance of Index-MNSCH and Index-NSHC. Here, wavelet decomposition level *DL* = 3 **and** *T* = 2%.

5.2 Histogram of Differences in the Number of Subband Significant Coefficients

Subbands corresponding to different resolutions can represent an image in a progressive way, *i.e.*, subbands in lower resolutions can represent an image coarsely, and the quality of the image can be improved by adding more subbands corresponding to higher resolutions. This is called a *resolution progression*. In the EZW algorithm, when using a lower threshold, the number of SCs increases not only in all wavelet domains but also in each of subband obviously. In this section, a new index, a histogram of difference in the number of subband significant coefficients (referred as to Index-DNSSCH), is presented below.

5.2.1 Index-DNSSCH Generation

Index-DNSSCH is a histogram of differences in the number of subband-SCs (DNSSC histogram). Each bin corresponds to a wavelet subband. The value for bin k (*i.e.* subband k) is the difference in the number of SCs for subband k in current $TL = i$ from the previous $TL = i - 1$ except $TL = 1$, while for $TL = 1$, the value of bin k is the number of SCs in subband k itself. The difference instead of the number of SCs itself is used as the bin value because the number of SCs for a subband of higher resolution will increase considerably with the lowering of the threshold and result in huge interference in the distance calculation. The significance of using the difference in the number of SCs is that it can degrade, to some degree, such kinds of interference. The interference can be reduced further by introducing weighting.

Figure 5-7 is the DNSSC histogram at $TL = 3$ extracted from the Lena image. The number of bins for the DNSSC histogram is only dependent on the wavelet decomposition level. For the same DL, a higher TL provides more SCs for each bin.

Let $g_i(k)$ be the value of bin k at $TL = i$ from $TL = i - 1$ when $DL = n$:

$$g_i(k) = \left\{ \begin{array}{cc} p_1(k) & (i = 1) \\ p_i(k) - p_{i-1}(k) & (i \geq 2) \end{array} \right\} \qquad (0 \leq k \leq 3n) \tag{5.4}$$

where $p_i(k)$ is the number of SCs in subband k at $TL = i$.

Then, the Index-DNSSCH for $TL = i$ and $DL = n$ can be expressed as:

$$Index_{dnssch}^{(i,n)} = H_{dnssch}^{(i,n)} = \{g_i(0), \cdots, g_i(k), \cdots g_i(3n)\} \tag{5.5}$$

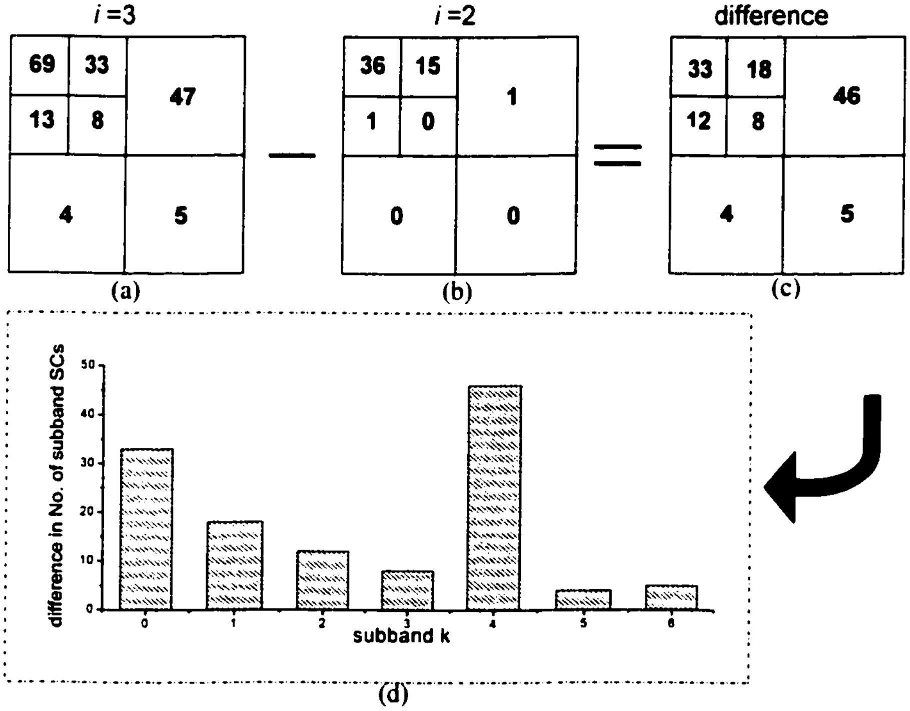


Figure 5-7 Index-DNSSCH histogram at $TL = 3$. (a) No. of SCs in each subband at $TL = 3$; (b) No. of SCs in each subband at $TL = 2$; (c) Difference no. of SCs in each subband for $TL = 3$ from $TL = 2$; (d) Histogram obtained from (c).

5.2.2 Computational Complexity

For $TL = i$ and $DL = n$, the distance of indices between the color query image Q and candidate image C can be determined using the L^1 metric as shown in Eq. (3.3). The computational complexity of retrieval depends on the number of bins in the DNSSC histogram, which is determined only by DL. We have assumed that $DL = n$, and consequently, there exists $3n+1$ subbands, *i.e.*, $3n+1$ bins in the DNSSC histogram. For each color component at each TL, we require $3n+1$ subtractions, $3n$ additions and $3n+1$ multiplications to calculate the distance $D_{dnssch}(Q,C)$. For color images including 3 components, the total number of operations (*i.e.*, additions/subtractions and multiplications) corresponding to an threshold-level is:

$$\begin{cases} O_{dnssch}^{(n)}(\pm) = 18n + 5 \\ O_{dnssch}^{(n)}(\times) = 9n + 3 \end{cases} \tag{5.6}$$

The retrieval complexity is fixed when the decomposition level is given. If the weights are set to be unity, $O_{dnssch}^{(n)}(\times)$ can be reduced to 0. From Eq. (5.6), the complexity of Index-DNSCH depends on the decomposition level rather than the threshold-level. For a fixed DL, the number of mathematical operations, which is very small (only 113 even for $DL = 6$), is the same for all threshold-levels. Hence, Index-DNSCH can achieve a fast retrieval as well as Index-MNSCH and Index-NSCH.

For $TL = 3$ and $DL = 3$, the running time for a query image with 60 images retrieved from a 3000-image database is approximately 68.21 ms.

5.2.3 Performance

The retrieval performance of Index-DNSSCH is shown in Figure 5-8 for different tolerances T ($T = 1\%$, 2% and 3% at $DL = 3$) and Figure 5-9 for different DL ($DL = 3$, 4 and 5 at $T = 2\%$).

The retrieval efficiency η also follows the same trend with increases in threshold at a fixed decomposition level. When $TL \le 4$, η increases and the slope is steep, while with further increases in TL, *i.e.*, when $TL > 4$, η starts to decrease monotonously.

This evaluation performance is reasonable. It can be explained in a similar way as the trend in Index-MNSCH. That is, when TL is low, the number of SCs in each subband (histogram bin) is small even down to zero due to the high threshold: the difference between two images can be just compared roughly and result in a low efficiency. With increasing TL, more SCs are found in each bin as more details are brought in. However, the number of total coefficients of subbands in finer resolutions are multiplied by that in coarser resolutions by the power of 2^2. With further increases in TL, *i.e.*, lowering the threshold, the number of SCs in subbands at finer resolutions

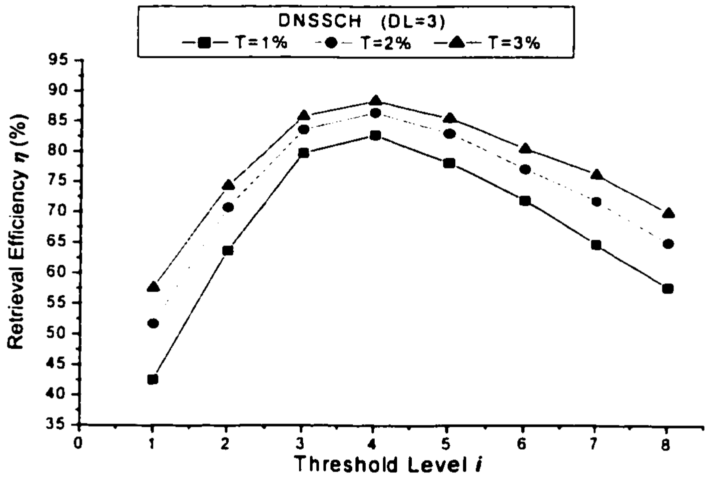


Figure 5-8 Retrieval performance of Index-DNSSCH at $DL = 3$ **and** $T = 1\%$, 2% and 3%

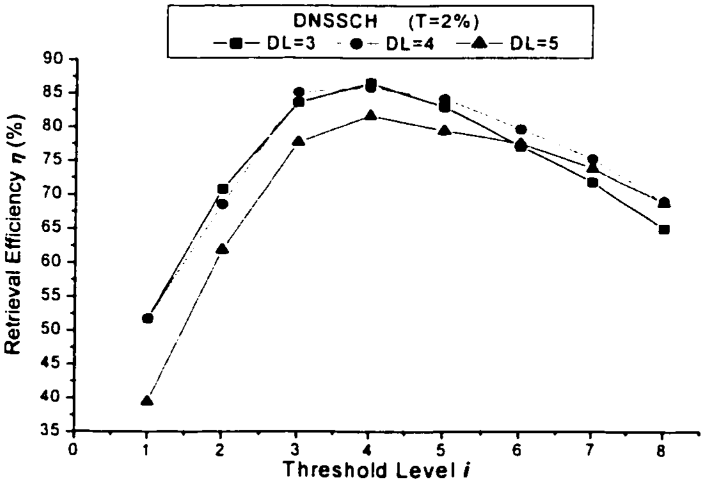


Figure 5-9 Retrieval performance of Index-DNSSCH at $T = 2\%$ **and** $DL = 3$, 4 and 5

increases exponentially in contrast with the number at coarser resolutions, while the increased SCs at finer resolutions are less significant than those at coarser resolutions. When TL increases to some extent, exponentially increasing in less significant SCs appears as noise right now to interfere with the results. Therefore, similar to Index-MNSCH, Index-DNSCH works effectively just for some threshold-levels which have to be selected carefully.

For a certain TL, η also increases with larger tolerances (Figure 5-8). While, for a fixed tolerance (Figure 5-9), η keeps approximately unchanged at $DL = 3$ and $DL = 4$, and decreases slightly at $DL = 5$. By the way, at $DL = 3$ and $T = 2\%$, the maximum retrieval efficiency reaches 86.5% at $TL = 4$.

5.3 Joint EZW-Based Indexing Techniques

In this section, the combined use of indices, including those which have been introduced in Chapter 4 and Section 5.1 and 5.2, will be also analyzed. The combinations include: 1) MNSCH and DNSSCH (Index-CMD); 2) MNSCH and LLBM (Index-CML); 3) DNSSCH and LLBM (Index-CDL); and 4) MNSCH, DNSSCH and LLBM (Index-CMDL). The computational complexity for distance for these combinations is the summation of that of the combined indices, and hence, the analysis for it will not be repeated here.

5.3.1 Combination of MNSCH and DNSSCH

Index-CML is the combination of Index-MNSCH and Index-LLBM. The retrieval performance of Index-CMD is shown in Figure 5-10 for different T ($T = 1\%$, 2% and 3% at $DL = 3$) and Figure 5-11 for different DL ($DL = 2$, 3, 4 and 5 at $T = 2\%$).

From Figure 5-10 it is clear that the retrieval efficiency η has a same trend with the increase of threshold at a fixed decomposition level. When $TL \leq 3$, η increases and the slope is steep, while with further increases in TL, *i.e.*, when $TL > 3$, η starts to decrease smoothly. The reason for this observation is similar to that of Index-MNSCH and Index-DNSSCH, *i.e.*, when TL is low, the increase in the number of SCs with respect to the lowering of threshold contributes more detail to the difference between two images so that it leads to increase in retrieval efficiency. However, when TL is high, the benefit brought by the more SCs becomes to disappear and it converts to noise in computing the distance between two images.

Again, for a certain TL, η is enhanced with the increases in tolerance (see Figure 5-10), while at a fixed tolerance, η decreases when the decomposition level increases (see Figure 5-11). When $DL = 3$ and $T = 2\%$, the maximum retrieval efficiency is about 90%.

In addition, Figure 5-12 shows the retrieval efficiency of Index-MNSCH, Index-DNSSCH and Index-CMD at decomposition level $DL = 3$ and tolerance $T = 2\%$. We find that Index-CMD provides an average 6% improvement over the former two techniques.

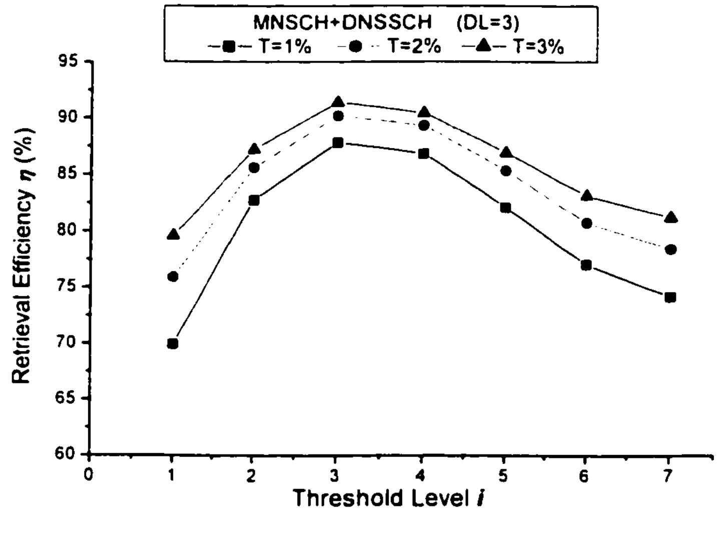


Figure 5-10 Retrieval performance of Index-CMD at $DL = 3$ **and** $T = 1\%$, 2% and 3%

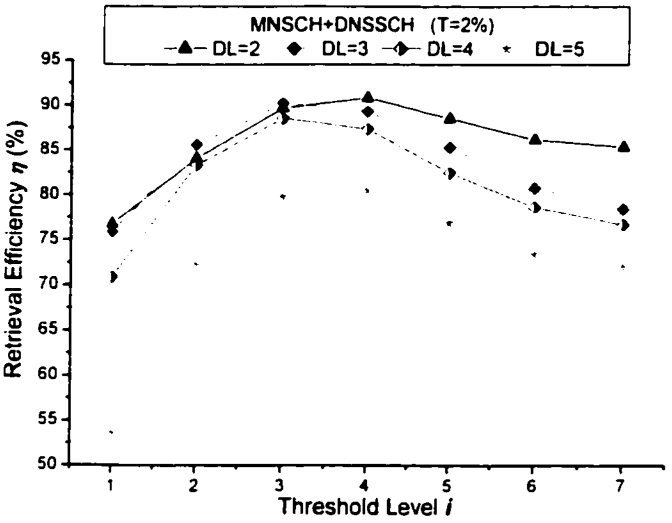


Figure 5-11 Retrieval performance of Index-CMD at $T = 2\%$ **and** $DL = 2, 3, 4$ and 5

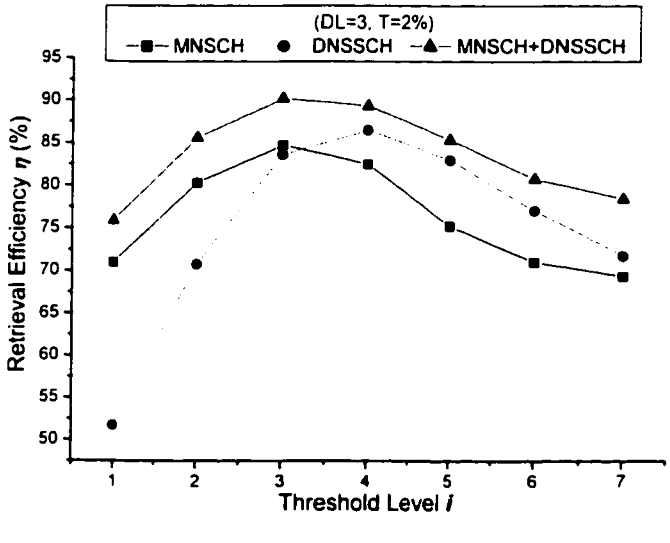


Figure 5-12 Comparison of retrieval performance among Index-NSCH, Index-MNSCH, Index-DNSSCH and Index-CMD, at $DL = 3$ **and** $T = 2\%$

5.3.2 Combination of MNSCH and LLBM

The retrieval performance of Index-CML, the combination of Index-MNSCH and Index-LLBM, is shown in Figure 5-13 for different tolerances T ($T = 1\%$, 2% and 3% at $DL = 3$) and Figure 5-14 for different DL ($DL = 3$, 4 and 5 at $T = 2\%$).

The maximum retrieval efficiency at $DL = 3$ and $T = 2\%$ is 90%. The trend is similar to that of Index-MNSCH and so does the explanation. Additionally, as described in Section 4.3, a binarization map of the LL subband helps to improve the retrieval efficiency when $TL \leq 3$, and becomes useless since $TL > 3$.

On the other hand, for a certain TL, η is enhanced with the increase of tolerance (Figure 5-13), while at a fixed tolerance, η decreases when the decomposition level increases (Figure 5-14).

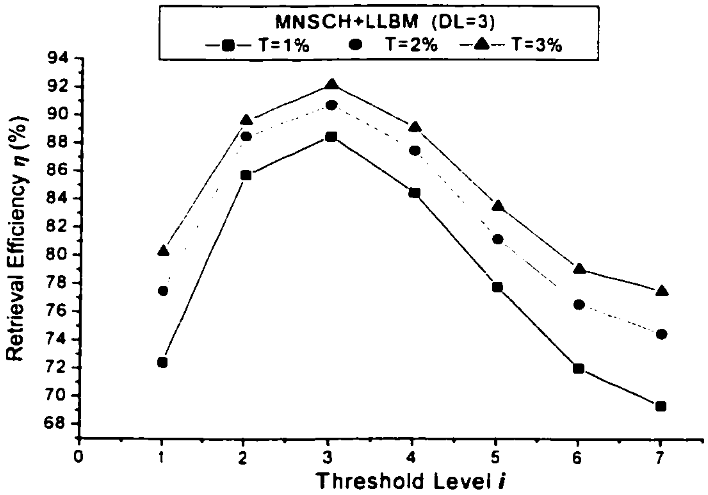


Figure 5-13 Retrieval performance of Index-CML at $DL = 3$ **and** $T = 1\%$, 2% and 3%

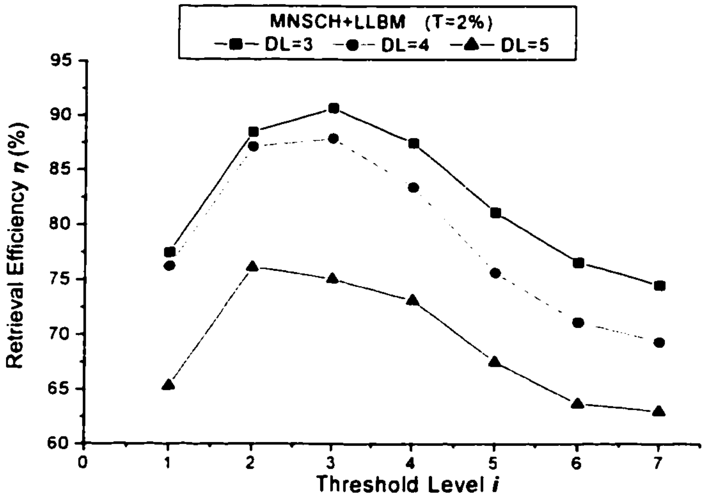


Figure 5-14 Retrieval performance of Index-CML at *T* = 2% **and** *DL* = 3, 4 and 5

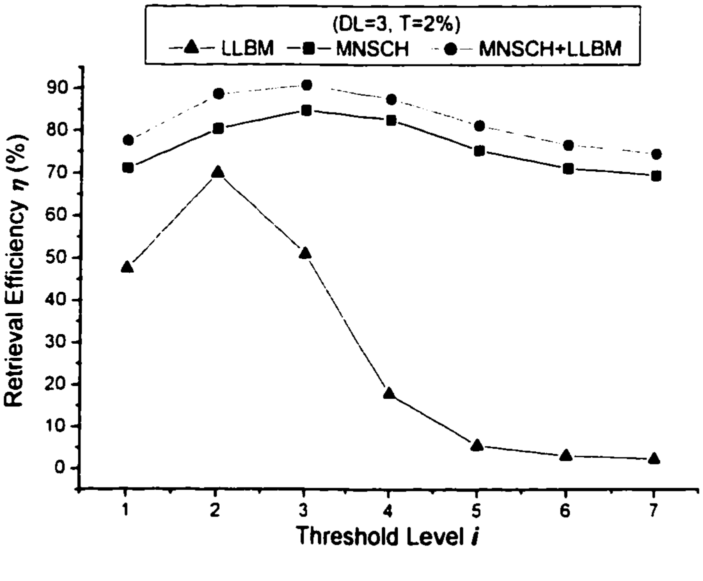


Figure 5-15 Comparison of retrieval performance among Index-LLBM, Index-MNSCH, Index-CML, at *DL* = 3 **and** *T* = 2%

Compared to Index-MNSCH (Figure 5-15), Index-CML provides an average 6% improvement over Index-MNSCH. However, the cost for this improvement is the considerably larger complexity of Index-CML over Index-MNSCH. The best performance is achieved at $TL = 3$ for both Index-MNSCH and Index-CML with $\eta = 85\%$ and $\eta = 91\%$, respectively. Between the two, which should be chosen for retrievai is determined by the retrieval conditions and requirements, *i.e.*, if the decomposition level is high, *e.g.*, $DL = 5$, and the computer system in which the retrieval program is running is fast enough, Index-CML might be a better choice and vice versus.

5.3.3 Combination of DNSSCH and LLBM

The retrieval performance of Index-CDL is shown in Figure 5-16 for different T ($T = 1\%$, 2% and 3% at $DL = 3$) and Figure 5-17 for different DL ($DL = 3$, 4 and 5 at $T = 2\%$).

The maximum retrieval efficiency is about 89%, at $DL = 3$ and $T = 2\%$. The result is similar to that of Index-DNSSCH (see Section 5.2.3) and Index-LLBM (see Section 4.3) and can be explained in the same way.

Figure 5-18 shows the comparison of Index-LLBM, Index-DNSSCH and Index-CDL. Index-CDL has average 10% improvement over Index-DNSSCH with the help of Index-LLBM when $TL \leq 4$, while the two curves overlap after $TL > 4$. The cost for this 10% improvement is the considerable large complexity introduced by Index-LLBM. Obviously, the selection between Index-DNSSCH and Index-CDL depends on which threshold-level is applied. For $TL \leq 4$, Index-CDL is a better choice, while for $TL > 4$, Index-DNSSCH is better than Index-CDL due to its lower complexity than Index-CDL.

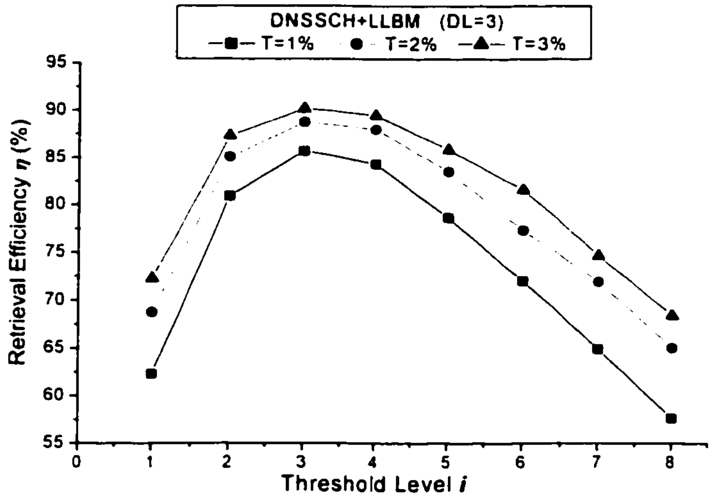


Figure 5-16 Retrieval performance of Index-CDL at $DL = 3$ **and** $T = 1\%, 2\%$ and 3%

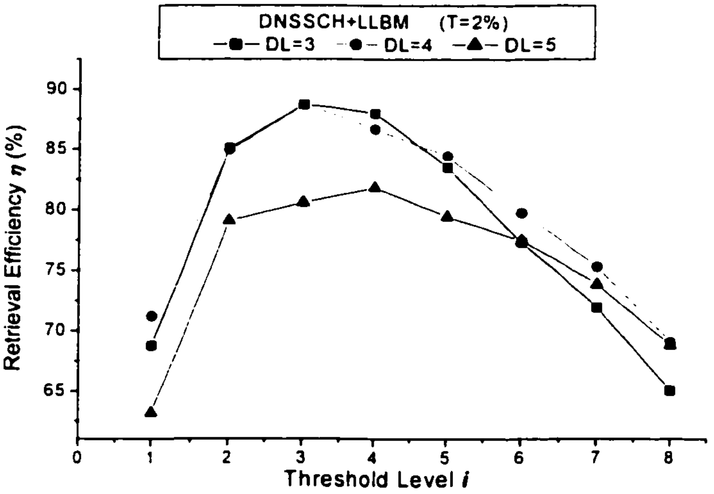


Figure 5-17 Retrieval performance of Index-CDL at $T = 2\%$ **and** $DL = 3, 4$ and 5

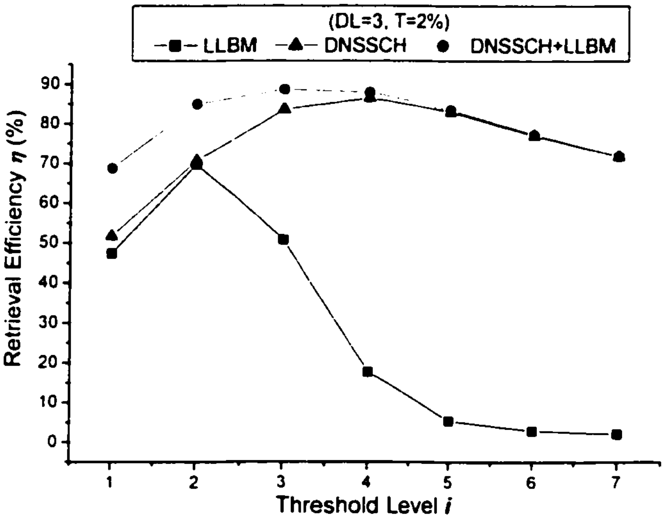


Figure 5-18 Comparison of retrieval performance among Index-LLBM, Index-DNSSCH and Index-CDL, at $DL = 3$ **and** $T = 2\%$

5.3.4 Combination of MNSCH, DNSSCH and LLBM

The retrieval performance of Index-CML is shown in Figure 5-19 for different tolerances T ($T = 1\%$, 2% and 3% at $DL = 3$) and Figure 5-20 for different DL ($DL = 3$, 4 and 5 at $T = 2\%$).

The maximum retrieval efficiency is about 92%, at $DL = 3$ and $T = 2\%$. No different evaluation trend from Index-CMD and Index-MNSCH is observed.

For retrieval performance, it is shown from Figure 5-21 that Index-CMDL has average 2% improvement over Index-CDM when $TL \leq 4$, while the two curves overlap after $TL > 4$.

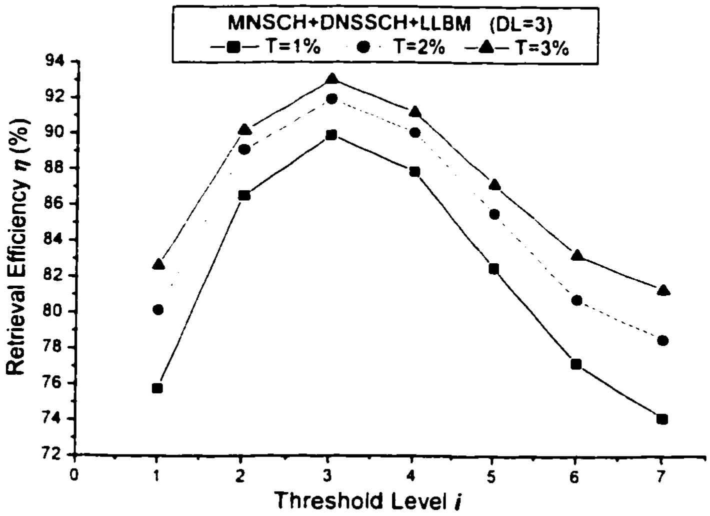


Figure 5-19 Retrieval performance of Index-CMDL at $DL = 3$ **and** T = 1%, 2% and 3%

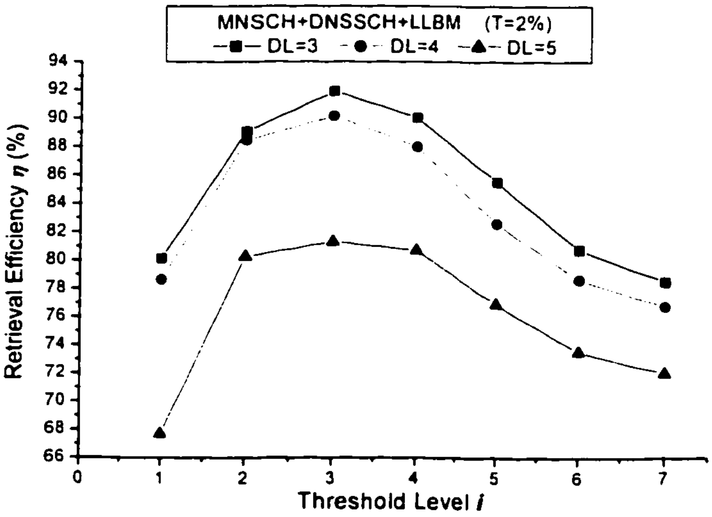


Figure 5-20 Retrieval performance of Index-CMDL at $T = 2\%$ **and** DL = 3, 4 and 5

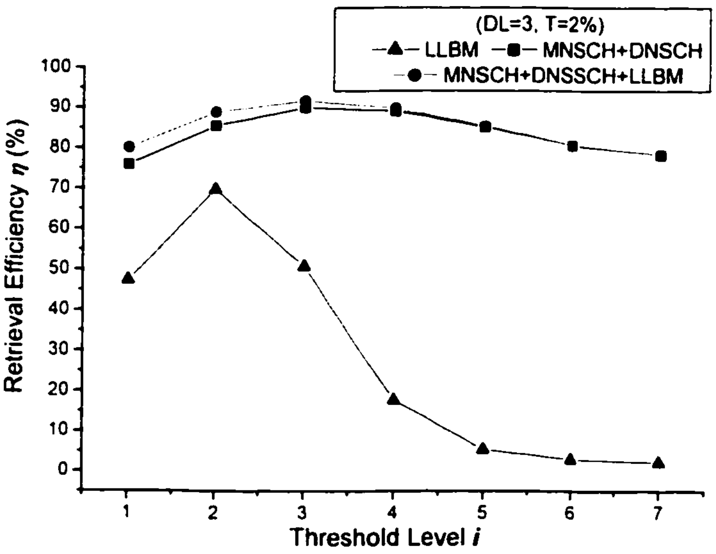


Figure 5-21 Comparison of retrieval performance among Index-LLBM, Index-CMD and Index-CMDL, at $DL = 3$ **and** $T = 2\%$

5.4 Comparison of EZW-Based Indexing Techniques

So far, we have introduced all EZW-related image indexing techniques, including the previous related works, *e.g.*, NSCH technique, and the proposed ones, such as MNSCH, DNSSCH techniques and their combinations. In the last section, the analysis of retrieval performance was done among several technique groups, *i.e.*, 1) MNSCH, DNSSCH and CMD; 2) MNSCH and CML; 3) DNSSCH and CDL; and 4) CMD and CMDL. Meanwhile, the comparisons were also done within each group. In this section, we would like to do more comparisons among techniques which have not been covered in the last section, *i.e.*, CMD, CML, CDL, CMDL and WMV, in both retrieval efficiency and computational complexity. Here, we use the results at $DL = 3$ and $T = 2\%$ for all indices.

For simplicity, we set the weights for all indices to '1' so that multiplication is not considered in the analysis of computational complexity. Meanwhile, the computational complexity, at $DL = 3$ and $T = 2\%$ and from $TL = 1$ through $TL = 7$ for the all techniques to be compared, re-tabulated in Table 5-1.

Table 5-1 Computational Complexity for all EZW-based indexing techniques introduced in Chapters 4 and 5 at $DL = 3$ and $T = 2\%$ from $TL = 1 \sim 7$

TL		1	2	3	4	5	6	7	8
$O_{nsch}^{(t)}(\ )$	±	11	17	23	29	35	41	47	53
$O_{wmv}^{(n)}(\ )$	±	119	119	119	119	119	119	119	119
$O_{llbm}^{(n)}(\ )$	±	3071	3071	3071	3071	3071	3071	3071	3071
	⊕	3072	3072	3072	3072	3072	3072	3072	3072
$O_{mnsch}^{(t)}(\ )$	±	23	47	71	95	119	143	167	191
$O_{dnsch}^{(n)}(\ )$	±	59	59	59	59	59	59	59	59
$O_{cmd}^{(t,n)}(\ )$	±	83	107	131	155	179	203	227	251
$O_{cml}^{(t,n)}(\ )$	±	3095	3119	3143	3167	3191	3215	3239	3263
	⊕	3072	3072	3072	3072	3072	3072	3072	3072
$O_{cdl}^{(n)}(\ )$	±	3131	3131	3131	3131	3131	3131	3131	3131
	⊕	3072	3072	3072	3072	3072	3072	3072	3072
$O_{cmdl}^{(t,n)}(\ )$	±	3155	3179	3203	3227	3251	3275	3299	3323
	⊕	3072	3072	3072	3072	3072	3072	3072	3072

For retrieval performance, it is shown in Figure 5-22 that the curves for Index-CML, Index-CDL, Index-CMD and Index-CDML are close for each TL, and that they all follow the same trend: with increasing TL, retrieval efficiency η increases when TL is small and climbs to the peak at $TL = 3$, after that, η starts to decrease monotonically. Among them, Index-CMDL provides marginally better retrieval efficiency over the others.

However, from Table 5-1, when $DL = 3$, the complexity cost for Index-CML, Index-CDL and Index-CMDL is almost 30 times large of that for Index-CMD. The extra complexity for each of the former three over Index-CMD is all due to the contribution of Index-LLBM. Because the complexity of Index-LLBM is strongly DL -dependent, Index-LLBM and all indices in which Index-LLBM joins have considerably larger complexity when $DL \leq 3$ (see details in Section 4.3) and their complexity is close because the complexity of other indices, such as Index-NSCH, Index-MNSCH and Index-DNSCH, can be ignored compared to that of Index-LLBM.

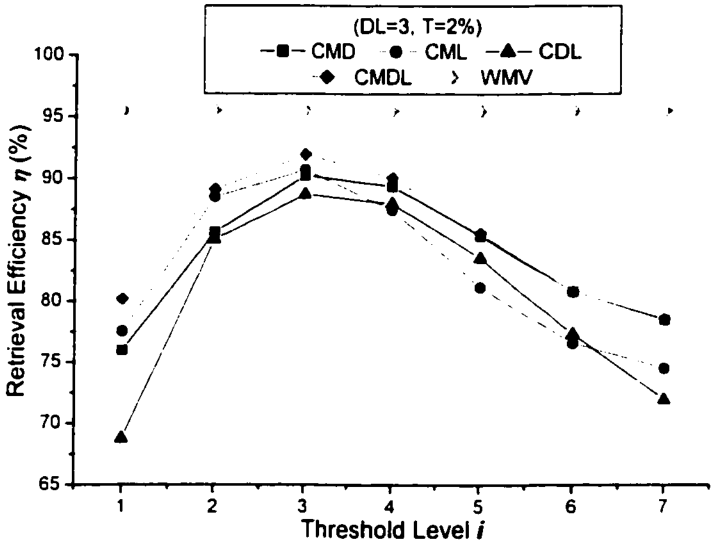


Figure 5-22 Comparison of retrieval performance among Index-CMD, Index-CML, Index-CDL, Index-CMD and Index-WMV, at $DL = 3$ and $T = 2\%$.

Considering both the retrieval efficiency and the computational complexity, obviously, Index-CMD is preferred for retrieval due to its low complexity and similar retrieval performance compared to the others. The best performance is achieved at $TL = 3$ for Index-CML with $\eta = 91\%$, for Index-CDL $\eta = 88\%$, for Index-CMD with $\eta = 91\%$, and for Index-CDML with $\eta = 93\%$.

On the other hand, Index-WMV is also compared in Figure 5-22. Index-WMV is a non-EZW-based indexing technique. This technique has good retrieval performance in the wavelet domain. Compared to Index-WMV, Index-CDML has close retrieval efficiency when $TL = 3$. The best retrieval performance is $\eta = 93\%$ for Index-CMDL and $\eta = 96\%$ for Index-WMV.

5.5 Summary

In this chapter, the proposed EZW-based indices were introduced, and all indices including those introduced in Chapter 4 were compared. It was found that for all EZW-based indexing techniques, the trend in retrieval performance is similar, *i.e.*, when TL is small, η increases with the increase of TL. On the other hand, when TL increases to some extent, η decreases with the increase of TL. In other words, the best performance of EZW-based techniques is always achieved in the middle range of TL. Among all EZW-based techniques, Index-CDML can achieve the best performance, but with considerably larger complexity. Index-CDM can achieve good performance that is marginally lower than that of Index-CDML but with a much smaller computational complexity. For fast retrieval requirement, Index-CMD is the best choice among all EZW-based indexing techniques.

Chapter 6 INDEXING IN THE JPEG2000 STANDARD FRAMEWORK

JPEG2000 is the new advanced standard for still image compression being developed (now in its final-processing) by the ISO [3]. It is intended to provide low bit-rate operation with rate-distortion optimization and subjective image quality performance superior to the current existing standards (*e.g.* JPEG) without sacrificing performance at the higher bit-rates. With the increasing use of multimedia technologies, JPEG2000 is expected to be used in many applications, *e.g.*, Internet, color facsimile, printing, scanning, digital photography, remote sensing, mobile applications, medical imagery, digital library and E-commerce. Therefore, it is important to develop indexing techniques based on the JPEG2000 standard algorithm.

In this chapter, the JPEG2000 standard algorithms will be introduced first. The proposed indexing techniques are then presented.

6.1 Review of the JPEG2000 Standard

The kernel of JPEG2000 is primarily based on the Embedded Block Coding With Optimized Truncation of the bitstream (EBCOT) [77]. The EBCOT algorithm provides superior compression performance with modest complexity while producing a bit-stream with a rich set of features, including resolution and SNR scalability together with a "random access" property.

In JPEG2000, an image to be encoded is independently partitioned into non-overlapping blocks (tiles) in the same dimension, to reduce the memory requirements and efficiently process the regions of interest in the image. A block schematic of the JPEG2000 encoder is shown in Figure 6-1. Various operations, such as wavelet transform, scanning, quantization and entropy encoding, are performed independently on all blocks of the image. For each block, the discrete wavelet transform is applied on the source image data. The wavelet coefficients are then scanned by a scanning algorithm and quantized. A rate-control mechanism is used along with the quantizer to satisfy the rate requirements by truncating the quantized coefficients. An entropy coding is then performed to generate the output bitstream.

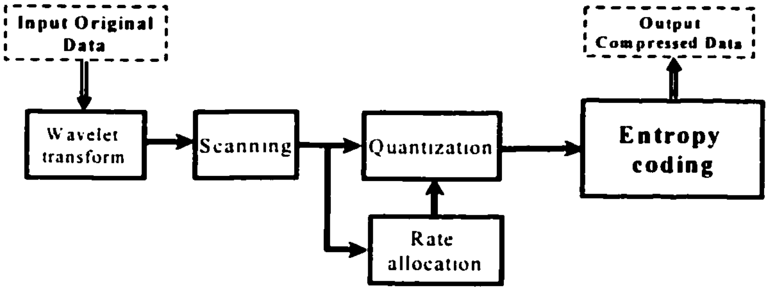


Figure 6-1 Block diagram of the JPEG2000 encoding process.

We note that the entropy coding in JPEG2000 employs a bit-modeling algorithm. Here, the wavelet coefficients are represented in the form of combinations of bits that are distributed in different bit-planes. Further more, these bits are reassembled in coding passes according to their significance status. The use of bit modeling provides a hierarchical representation of wavelet coefficients by ordering the bit-planes of wavelet coefficients from the MSB to the LSB. Hence, the formed bitstreams with inherent hierarchy can be stored or transferred at any bit rate without destroying the completeness of the content of the image.

Compared to the current JPEG still image compression standard – JPEG, JPEG2000 has the following features [78, 79]:

- Superior low bit-rate performance without sacrificing performance on the rest of the rate-distortion spectrum.
- The DWT is used as the basis instead of discrete cosine transform.

- It provides the following features: (a) embedded lossy to lossless, (b) multiple-component images, (c) static/dynamic region-of-interest, (d) error resilience, (e) spatial/quality scalability, and (f) rate-control.
- It is expected that JPEG2000 will provide more than 20% improved performance over JPEG for most images.

Because the ideas for constructing indices in the JPEG2000 framework come from the compressed JPEG2000 bitstream structure, brief explanations on necessary concepts/terms related to the JPEG2000 standard will be helpful for understanding the principle, and the process of index generation.

6.1.1 Bit-Plane

Table 6-1 Illustration of bit-planes. "bp" in the table refers to bit-plane.

Coefficients	bp5 (MSB)	bp4	bp3	bp2	bp1 (LSB)
21	1	0	1	0	1
1	0	0	0	0	1
0	0	0	0	0	0
10	0	1	0	1	0

A number can be represented in several formats, (*e.g.*, decimal and hexadecimal). The most popular format in digital signal processing is the binary representation. The order of the bits of the number descends from the most significant bit (MSB) to the least significant bit (LSB). A bit-plane is the decomposition of the binary representation for a given set of decimal numbers. An example is given in Table 6-1. The four decimal numbers (21, 1, 0, 10) are expressed in binary representations (rows). The binary representations are then arranged so that the bits corresponding to a particular weight are in the same column. The rows represent the original decimal number and the columns construct the bit-planes. For a set of data, the number of bit-planes is determined by the absolute maximum value in the data set.

6.1.2 Significance of Bits

The *significance of bits* refers to whether a bit ('1' or '0') involved in a coefficient is significant or not. The significance status of a coefficient changes from *insignificant* to *significant* at the bit-plane where the first '1' bit of the coefficient is found. This '1' bit just found is recorded as significant. The remaining '1's appearing in the binary representation of the coefficients are regarded as insignificant. In the example shown in Table 6-1, the significant '1' bit for coefficient 21 is at bit-plane 5, this bit is recorded as significant; the second and the fourth '1' bits appear at bit-plane 3 and bit-plane 1, respectively; they will not be recorded as significant because they are not the significant bits of 21. Similarly, for coefficient 10, the '1' bit showing up at the bit-plane 4 is recorded as significant while the '1' in bit-plane 2 is insignificant.

6.1.3 Code-Block

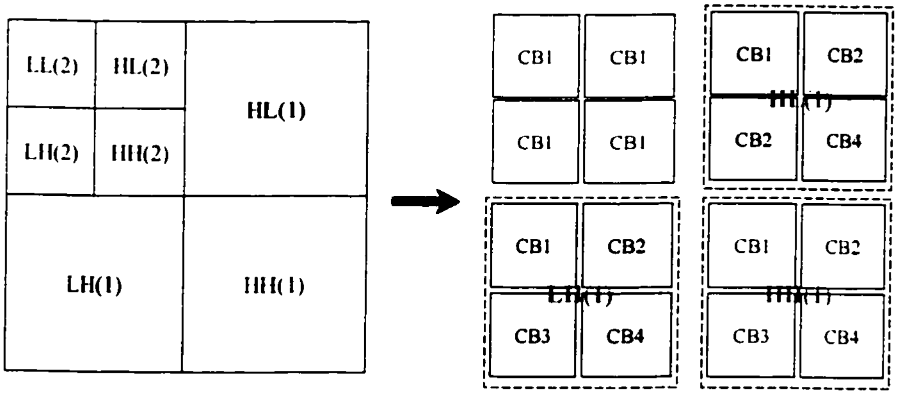


Figure 6-2 Illustration of sub-bands and code-blocks (CB)

Each sub-band in the wavelet domain is partitioned into non-overlapping rectangular blocks, *code-blocks*. The code-block is the smallest independent unit in the JPEG2000 standard. Partitioning the wavelet coefficients into code-blocks facilitates coefficient modeling and coding, and makes the processing more efficient and flexible in organizing the output compressed bitstream. That is, DWT coefficients are sliced into bit-planes in units of code-blocks, and the bit-planes are coded to generate the output bitstreams with respect to the required order. It is noted that the bit-plane coding order descends from MSB to LSB. Note that the size of each code-block is the same for all sub-bands at the same resolution.

6.1.4 Packet and Packet Header

As mentioned above, after dividing the wavelet coefficients into code-blocks and slicing the code-blocks into bit-planes, the arithmetic entropy encoding is applied to all bit-planes in the required order. The output of arithmetic encoder and the necessary information constitute the final output bitstream, in which consists of main header and data body, *i.e.*, multiple so-called *packets*.

A packet, according to the JPEG2000 standard, comprises a packet header and a part of the bitstream from an entropy-coded binary image. The packet header is a portion of a packet that provides auxiliary information about the binary data that follows, and the coded binary data of each packet comes from the bit-planes sliced from at least one complete code-block. Hence, it is expected that the necessary and concise information about a code-block can be extracted from a packet header directly without decompressed the coded binary data.

One important information contained in a packet header is the number of the bit-planes for each code-block involved in the corresponding packet.

6.1.5 The Number of Bit-Planes

Typically, as shown in Figure 6-3, the number of the bit-planes for a given data set is determined by the number with maximum value in the data set. In the JPEG2000 standard, such a data set refers, in particular, to the wavelet coefficients either from a subband or a code-block. The number of bit-planes available for the representation of coefficients in any sub-band b is given by M_b, and in any code-block by P_0. As mentioned before (see "code-block" part above), any sub-band may include one or more code-blocks. Hence, obviously, for those code-blocks from a subband, the value of any P_0 for the code-blocks will not exceed the M_b for that subband. This M_b with respect to a subband can be calculated from the main header of the coded image data, while P_0s for code-blocks within the subband can be obtained from packet headers. In general, the number of actual bit-planes P_0 may vary from code-block to code-block in the sub-band. It has to be noted that, with a known M_b of a subband, $(M_b - P_0)$ instead of P_0 for each code-block within the subband is preserved in the corresponding packet header. This $(M_b - P_0)$, annotated as P in the standard, is actually the number of MSB bit-planes that are all zeros.

coefficients	bp6	bp5	bp4	bp3	bp2	bp1	# of significant bit-planes
21	0	1	0	1	0	1	5
1	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0
10	0	0	1	0	1	0	4

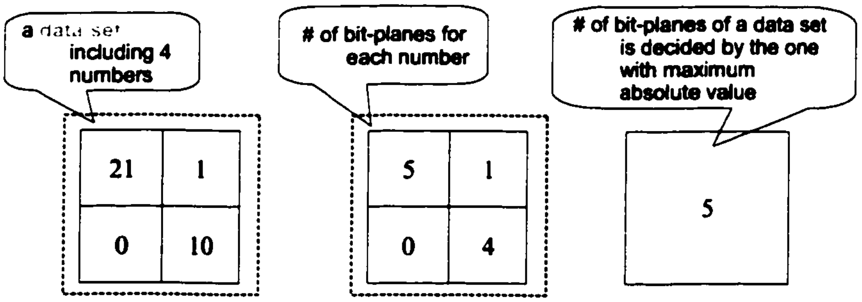

a data set including 4 numbers
21
1
0
10
# of bit-planes for each number
5
1
0
4
# of bit-planes of a data set is decided by the one with maximum absolute value
5

Figure 6-3 Illustration of the number of bit-planes and significance of bit '1'. The bit '1' in red is the most significant over the other '1's.

6.1.6 Layer

The term *layer* used in this work refers to a collection of contiguous bit-planes. A layer may contain a fraction of a bit-plane, or may contain multiple bit-planes. However, in this thesis, we assume for simplicity that a layer contains just one bit-plane. The number of layers is determined by the maximum value of wavelet coefficients, which equals to M_b. For layer $l = 1$, it refers to the MSB bit-plane M_b, layer $l = 2$, refers to the second most significant bit-plane, *i.e.*, $M_b - 1$, and so on. For example, layer 1 is bit-plane 5 in Table 6-1.

6.1.7 Stage

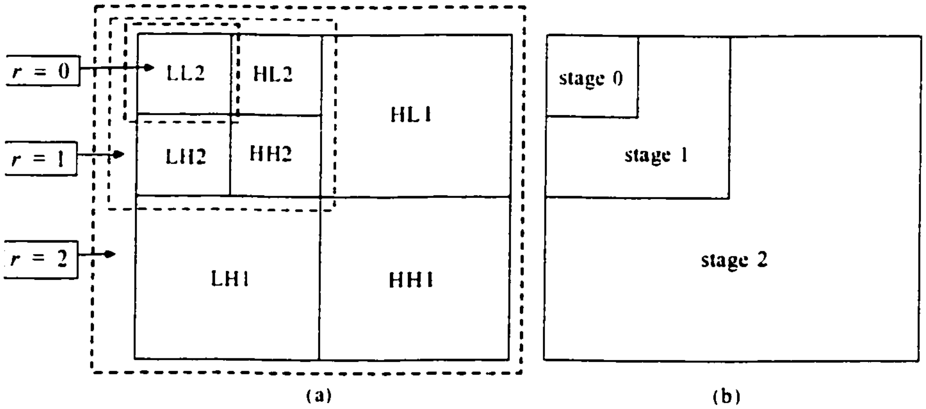


Figure 6-4 Illustration of resolutions and stages. 2-level decomposition is applied ($DL = 2$). (a) $r = 0$: LL2; $r = 1$: LL2 + LH2 + HL2 + HH2; $r = 2$: LL2 + LH2 + HL2 + HH2 + LH1 + HL1 + HH1; (b) $s = 0$: LL2; $s = 1$: LH2 + HL2 + HH2; $s = 2$: LH1 + HL1 + HH1.

In the JPEG2000 standard, if an image is wavelet transformed with n decomposition levels, the image will have $n+1$ distinct resolutions, denoted by $r = 0, 1, \cdots, n$, with a total of $3n+1$ sub-bands. As shown in Figure 6-4-(a), the lowest resolution ($r = 0$) is represented by the LL band. For the rest, resolution r is obtained by discarding sub-bands HHj HLj, LHj from $j = 1$ to $n - r$, and reconstructing the image from the remaining sub-bands. The dimensions of different resolutions will differ each other by powers of two.

Here, a *stage* is defined as another representation of resolution. That is, there are total $n+1$ distinct stages, denoted as $s = 0, 1, \cdots, n$. The lowest stage, which is denoted by

$s = 0$, is represented by the LL band. Any other stage s is equal to the resolution s without the subbands in resolution $s-1$. In other words, stage s is the composition of the LH*s*, HL*s*, and HH*s* sub-bands, as shown in Figure 6-4-(b).

6.2 Significant Bit Map

In the JPEG2000 standard, wavelet coefficients are processed in the basis of bit-planes, and the generated bitstream is also organized on this basis with significance descending to achieve SNR scalability. In other words, the bit-planes, *i.e.*, layers, in the packets are organized in descending order from the MSB to the LSB. This suggests that the layers can be used to construct a new index, significant bit map (referred to as Index-SBM).

6.2.1 Index-SBM Generation

Index-SBM is composed of a set of significant-bit-maps (SBM) derived from the LL subband at specified layers. Recall that Index-LLBM (refers to Section 4.3) is a binarization map of LL subband coefficients with respect to a given threshold. It is found that Index-SBM is also in the style of a binarization map which is similar to Index-LLBM. However, unlike Index-LLBM, Index-SBM is obtained by slicing coefficients into bit-planes instead of applying a threshold, and therefore it includes multiple binarization maps due to the multiple bit-planes. Hence, Index-SBM is actually a 1-D array of such binarization maps. Each component of the array is an SBM derived from a bit-plane at layer l. The dimension of the array is L, where L is the number of layers used, which starts from layer 1 to layer L, for the purpose of retrieval. However, the SBM is not bit-plane itself. A bit-plane consists of '1's and '0's, where the '1's are either significant '1's or insignificant '1's (refers to explanation of bit significance in Subsection 6.1.2). The insignificant '1's are just used to refine the coefficients that their significance status have already been significant, instead of recognizing their significance, their existence may interfere with the extraction of the main feature for the image. Therefore, the insignificant '1's in a layer will be regarded as '0's and only the significant '1's left in SBMs.

Table 6-2 An example of Index-SBM derived from the R color component of Lena color image in size of 256×256, which is 5-level wavelet decomposed. Only the red '1' contributes to SBMs.

Bit plane	11	10	9	8	bp7	bp6	bp5	bp4	bp3	bp2	bp1	bp0
layer	1	2	3	4	5	6	7	8	9	10	11	12
1698	0	1	1	0	1	0	1	0	0	0	1	0
474	0	0	0	1	1	1	0	1	1	0	1	0
594	0	0	1	0	0	1	0	1	0	0	1	0
926	0	0		1	1	0	0	1	1	1	1	0
784	0	0	1	1	0	0	0	1	0	0	0	0
689	0	0	1	0	1	0	1	1	0	0	0	1
-2137	1	0	0	0	0	1	0	1	1	0	0	1
...	...											...
...	...											...
1719	0	1	1	0	1	0	1	1	0	1	1	1
894	0	0	1	1	0	1	1	1	1	1	1	0
1560	0	1	1	0	0	0	0	1	1	0	0	0
100	0	0	0	0	0	1	1	0	0	1	0	0
394	0	0	0		1	0	0	0	1	0	1	0

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0

(a) bp11

1	0	0	0	0	0	1	0
1	0	0	0	0	0	1	0
0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	1	1	1	0	0	1
1	0	1	0	0	0	0	0
0	0	0	1	0	1	0	0

(b) bp10

0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	1
0	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
0	0	1	0	0	1	0	1
1	0	0	0	0	0	1	0
0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	0

(c) bp9

0	1	0	0	0	0	0	1
0	1	1	0	0	1	0	0
0	1	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	1	0	1	0
0	1	0	0	0	0	0	1

(d) bp8

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(e) bp7

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0

(f) bp6

Figure 6-5 The first 6 layers of Index-BPM derived from Table 6-2.

Here we still use Γ_n to denote the binarization map under the assumption that an image is in size of 384×256 and 3-level decomposed, and $\Gamma_3(l)$ the SBM component in the array at $L=l$, $\Gamma_3(l)$ is expressed as:

$$\Gamma_n(l)=\begin{bmatrix} b(0,0) & b(0,1) & \cdots & \cdots & b(0,31) \\ b(1,0) & \ddots & & & \vdots \\ \vdots & & b(x,y) & & \vdots \\ \vdots & & & \ddots & \vdots \\ b(47,0) & \cdots & \cdots & \cdots & b(47,31) \end{bmatrix} \tag{6.1}$$

where $0\le x\le47$, $0\le y\le31$, and $b(x,y)$ is the binary value in the coordinate (x,y) of the map.

Now, the Index-SBM is array of $\Gamma_n(l)$ at $DL=n$:

$$Index_{sbm}^{(n)}(x)=\left[\Gamma_n(1),\Gamma_n(2),\cdots,\Gamma_n(L)\right] \tag{6.2}$$

where $1\le L\le M_b$.

As shown in Eq. (6.2), Index-SBM can be generated using different values of L. The maximum value of L can reach the total number of bit-planes M_b.

Table 6-2 shows an example of Index-SBM derived from the R color component of the Lena color image of size of 256×256. Here, the Lena image is 5-level wavelet decomposed to generate a 8×8 LL subband. The maximum absolute value is $|-2137|$, which is represented by 12 bits with non-zero MSB, so that the number of bit-planes is 12 and so is that of layers. To illustrate more clearly, bit-plane 11 through bit-plane 6 (layer 1-6) are drawn in Figure 6-5. From the figure, it is found that the '1's in each layer do not overwhelm the '0's in number statistically, which overcomes the drawback of Index-LLBM when a lower threshold is applied. Therefore it can be expected that the retrieval performance using LSB layers will be better than that of Index-LLBM for a lower threshold-level. It is also found that most of significant bits are typically found in the first several layers, leaving fewer significant bits in the remaining layers. Therefore, we can surmise that the layers corresponding to LSB bit-planes may not influence the retrieval results.

6.2.2 Computational Complexity

For Index-SBM, differences of SBMs between the query image Q and the candidate image C for various layers can be employed to determine the similarities. Since any SBM corresponding to a layer is a binarization map (bit-map), similar to Index-LLBM, a bitwise exclusive OR operation between the two indices corresponding to each layer is employed to obtain the distance for each layer as shown in Eq. (3.4). The total distance for multi-layers is the summation of distance for each layer.

The magnitude of computational complexity of Index-SBM is the same as that of Index-LLBM because both depend on the image dimension and decomposition level. As introduced in Subsection 6.2.1, the Index-SBM includes multiple SBMs, each of them is equivalent to Index-LLBM in size. For a given L, the computational complexity of Index-SBM is about as L times as that of Index-LLBM. The operations include additions, multiplications and XORs. At given L, the complexity can be expressed as follows with n-level decomposition and image size in $X_{siz} \times Y_{siz}$:

$$\begin{cases} O_{sbm}(\pm) = 3L \times \frac{X_{siz}}{2^n} \times \frac{Y_{siz}}{2^n} + 18L \\ O_{sbm}(\oplus) = 3L \times \frac{X_{siz}}{2^n} \times \frac{Y_{siz}}{2^n} \\ O_{sbm}(\times) = 3 \times L \end{cases} \tag{6.3}$$

From the Eq. (6.3), the complexity increases linearly with L, *i.e.*, the smaller the decomposition levels and image dimension, the lower the computational complexity.

At $DL = 4$, $L = 4$ and $T = 2\%$, the average running time for a query image with 60-image retrieved in a 3000-image database is about 3474 ms.

6.2.3 Performance

The retrieval performance of Index-SBM versus the number of layers used is shown in Figure 6-6 for different tolerances T ($T = 1\%$, 2% and 3% at $DL = 5$) and Figure 6-7 for different wavelet decomposition levels DL ($DL = 3$, 4 and 5 at $T = 2\%$).

In Figure 6-6, wavelet coefficients contains total of 12 layers, and so Index-SBM can be generated using $1 \sim 12$ layers. The retrieval efficiency η has a same trend with increasing L at a fixed decomposition level. When T is fixed and L is increasing, η has a step increase at first for small L (about 13% increase for $L = 2$ over $L = 1$), then the increasing tendency slows down and saturates at $L = 5$. Since then, η keeps at a steady level until $L = 12$ with a tiny decrease. The reason for the trend of the η curve is that, when $L = 1$, *i.e.*, when only layer 1 is used as the index, the number of '1's in layer 1 is small and it cannot provide enough details for retrieval. When L is increased by 1 ($L = 2$), *i.e.*, both layer 1 and layer 2 are jointly used to generate Index-SBM, more significant coefficients are found. More details are provided by layer 2 to refine the retrieval. With further increase's L, more layers are added on and provide more details. However, as we know, the significance of layers descends from layer 1 to layer 12; in other words, layer 1 is the most significant and then layer 2, layer 3, ..., until layer 12. For the last layers, their significance becomes more and more insignificant with respect to that of the first layers. When L is increasing, the number of less significant layers is increasing as well. For example, for $L = 10$, we have 5 layers after layer 5. When finding the difference between any two indices, these layers with less significance will influence the resulted distance, and therefore leads to the degrading in retrieval performance. However, Index-SBM is only constructed from the LL subband, and the coefficients of LL subbands are generally large for natural images. It is possible that the significance status of the all coefficients has been changed from insignificant to significant in the first layers, and therefore, the last layers may consist of almost full '0's (Figure 6-5 shows a typical example) such that they may not impact on the distance calculation too much. This explains why η has a neglectable degrading since $L > 5$.

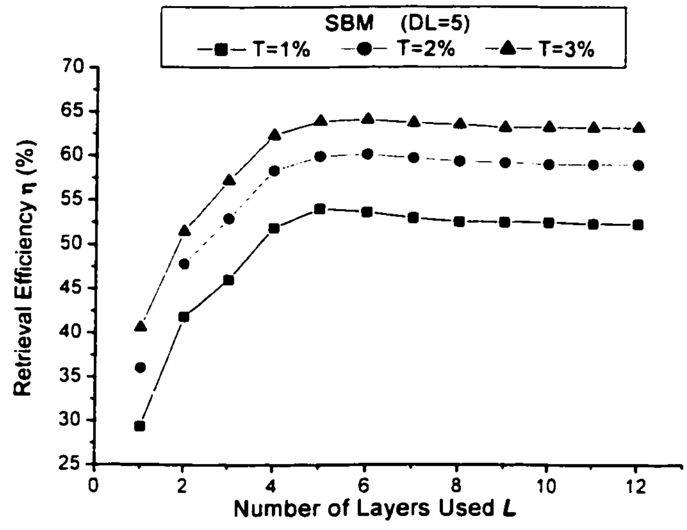


Figure 6-6 Retrieval performance of Index-SBM when DL **is fixed (** η **vs.** L **). Here,** $DL = 5$, $T = 1\% \sim 3\%$, **and** $L = 1 \sim 12$.

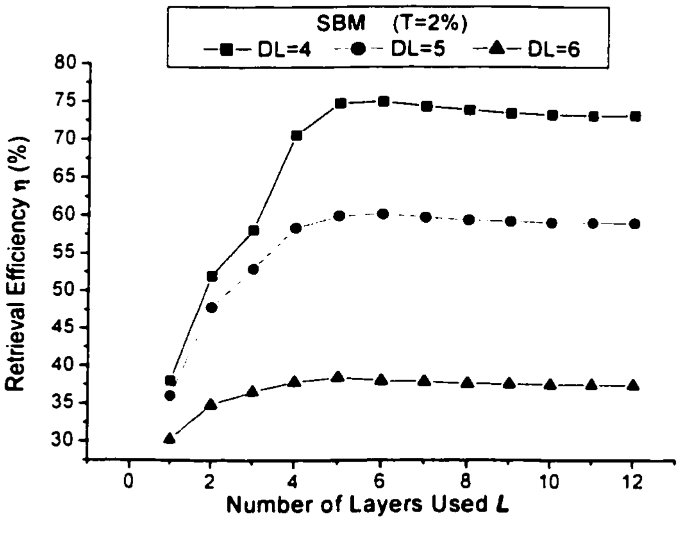


Figure 6-7 Retrieval performance of Index-SBM when T **is fixed (** η **vs.** L **). Here,** $T = 2\%$, $DL = 3 \sim 5$, **and** $L = 1 \sim 12$.

In a word, for a fixed decomposition level and a fixed tolerance T, the best performance is achieved at $L = 5$. Considering the computational complexity, the analysis in Subsection 6.2.2 exhibits a conclusion in opposition to that for retrieval performance, that is, the index in larger L provides good retrieval performance, while complexity computational is increased as well.

On the other hand, when L is fixed, η has an average 5% improvement with every 1% increase in tolerance T (Figure 6-6), which is obvious, whereas when T is fixed, η degrades with decomposition level DL (Figure 6-7). However, the results shown in the figures, in which the best performance is 60% reached at $T = 2\%$, $DL = 5$ and $L = 4$, tell that Index-SBM is not an efficient technique.

6.3 Histogram of the Number of Bits in Bit-Plane

In JPEG2000 standard, the packets involved in the final output bitstream are not only organized in order of layers, but also in order of stages (refers to Section 6.1) descending from stage 0 to stage n if n-level wavelet decomposition is applied, to achieve resolution scalability. That is, when receiving JPEG2000-compressed image, the first received bits are from lower stages, which provide more significant information but less detail than the others. With more bits coming, details with less significance are reinforced to refine the quality of the reconstructed image. This implies us that retrieval can be done progressively in a similar manner if using different number of stages to construct an new index, histogram of the number of bits, which is referred to as Index-NBH [80].

6.3.1 *Index-NBH Generation*

Although Index-NBH is designed based on stages, it cannot be constructed without layers because the JPEG2000-compressed bitstream is progressive on both layers and stages. Hence, Index-NBH is actually a 2-D histogram of the number of '1's in the bit-plane. The value of each bin is equal to the number of '1' bits derived from a bit-plane for a specified layer l and a specified stage s. Let $Index_{nbh}$ denote the Index-NBH of image

x, and $p_{l,s}$, the number of '1's of the bit-plane in current layer and stage, is the bin value of the histogram. Index-NBH is defined as follows:

$$Index_{nbh}(x)=\begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & \cdots & p_{0,S-1} \\ p_{1,0} & \ddots & & & \vdots \\ \vdots & & p_{l,s} & & \vdots \\ \vdots & & & \ddots & \vdots \\ p_{L-1,0} & \cdots & \cdots & \cdots & p_{L-1,S-1} \end{bmatrix} \tag{6.4}$$

where $1 \le L \le M_b$ and $1 \le S \le n+1$.

Table 6-3 Illustration of Index-NBH. The index is extracted from component R of the Lena color image in size of 256×256. $DL = 5$. sb: subband; bp: bit-plane

	bp11	bp10	bp9	bp8	bp7	bp6	bp5	bp4	bp3	bp2	bp1	bp0
sb0	3	16	30	34	34	26	30	29	27	31	39	34
sb1	0	6	17	25	23	27	42	34	36	32	30	29
sb2	0	0	4	11	18	32	33	22	36	30	36	32
sb3	0	0	2	12	16	26	36	30	38	28	33	31
sb4	0	0	7	55	67	103	116	120	114	130	119	131
sb5	0	0	0	5	25	52	86	99	117	127	119	136
sb6	0	0	1	6	45	59	72	93	128	121	124	133
sb7	0	0	0	12	100	199	273	342	405	438	504	494
sb8	0	0	0	1	19	69	144	226	325	411	444	496
sb9	0	0	0	1	17	78	153	223	297	357	455	488
sb10	0	0	0	0	23	209	465	744	1030	1405	1725	1865
sb11	0	0	0	0	1	53	183	373	671	1097	1465	1775
sb12	0	0	0	0	0	35	174	389	608	956	1413	1744
sb13	0	0	0	0	0	8	224	837	1670	3033	5335	6966
sb14	0	0	0	0	0	3	69	367	927	2068	4164	6183
sb15	0	0	0	0	0	0	2	134	615	1548	3308	5721

Eq. (6.4) shows that the value of L and S for Index-NBH may be different. The maximum value of L and S can reach the number of total bit-planes M_b in subband b and the number of total stages $n+1$, respectively. On the other hand, M_b may vary from subband to subband due to the asymmetric energy distribution in different subbands. Typically, M_b is larger for subbands in lower resolutions and vice verse. Namely, the first layer for different subband may locate in different bit-plane. For instance, in Table 6-3, $M_b = 12$ for subband 0 (LL subband), 11 for subband 1, and 10 for subband 2, 3, and 4, therefore, layer 1 means bit-plane 11 for subband 0, bit-plane 10 for subband 1, and bit-plane 9 for subband 2, 3, and 4, and so on for the remaining layers. Hence, when more than one stages are used to construct Index-NBH, a same layer for different subbands is not necessary from the same bit-plane. In Table 6-3, the cells covered in gray are layer 1, in pink layer 2.

6.3.2 Distance Metric

For Index-NBH, the L^1 metric is used to derive the difference of histograms between the query image Q and the candidate image C when $DL = n$:

$$D_{nbh}(Q,C) = \sum_{c=1}^{3}\sum_{l=1}^{L}\sum_{s=0}^{S-1}\left|p_{l,s}{}^{Q} - p_{l,s}{}^{C}\right|, \quad (1 \le L \le M_b, 1 \le S \le n+1) \tag{6.5}$$

6.3.3 Computational Complexity

The computational complexity includes $L \times S$ subtractions, $L \times S - 1$ additions and $L \times S$ multiplications for each color component. For the distance calculation with 3 color components, the total number of computational operations is given as follows:

$$\begin{cases} O_{nbh}(\pm) = 6 \times L \times S \\ O_{nbh}(\times) = 3 \times L \times S \end{cases} \tag{6.6}$$

From the Eq. (6.6), the computational complexity increases with both L and S.

At $DL = 3$, $L = 2$, and $S = 4$, the average running time for a query image with 60-images retrieved in a 3000-image database is about 61.33 ms.

6.3.4 Performance

The retrieval performance of Index-NBH is shown in Figure 6-8 and Figure 6-9 for different L and S at $DL = 3$ and $T = 2\%$, Figure 6-10 and Figure 6-11 for different wavelet decomposition level DL at $L = 3$ and $T = 2\%$, and Figure 6-12 and Figure 6-13 for different tolerance T at $DL = 3$..

From Figure 6-9, there are 4 stages and 11 layers with respect to $DL = 3$. The retrieval efficiency η has a similar trend with increases in either S or L at a fixed decomposition level. When T is fixed, if L is fixed as well (Figure 6-8), η increases with S when $S < 2$ or S<3 and then decreases slightly after that, except that η increases linearly at $L = 1$. The curves follow this trend because the wavelet coefficients in first several stages (lower frequencies) are more significant for natural images but small number, while those in the final one or two stages (highest frequencies) are least significant but in much larger number. For example, a n-level decomposed image has $n + 1$ stages, in which the first n stages contributes one-fourth of the overall coefficients, and the remaining three-fourths belong to stage $n + 1$. Hence, in the first several stages, the increase in the number of stages contributes more detail with less number of coefficients involved so as to distinguish indices among images more efficiently. For the last 1 or 2 stages, however, the details (highest frequencies of images) are provided with much larger number of coefficients than that of the first stages, and so the indices constructed by all stages may fail to tell the main difference among images. The curves shown in Figure 6-10 illustrate this observation. When $L = 3$ and $T = 2\%$, the best performance for each DL is always achieved at the last second or third stage, *i.e.*, $S = 3$ for $DL = 3$, $S = 3$ for $DL = 4$, and $S = 4$ for $DL = 5$. In addition, Figure 6-10 and Figure 6-11 indicates that the best performance achieved is almost independent of wavelet decomposition level.

On the other hand, if both T and S are fixed (Figure 6-9), η increases when $L \leq 3$ and saturates after $L > 3$. The reason for this is the same as that for Index-SBM, and so it will not be repeated here.

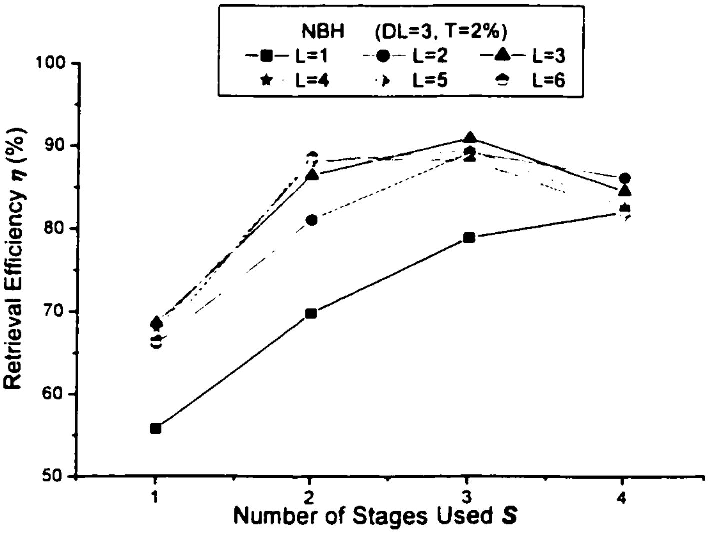


Figure 6-8 Retrieval performance of Index-NBH at a fixed DL **and** T **(**η **vs.** S**). Here,** $DL = 5$, $T = 2\%$, **and** $L = 1 \sim 6$

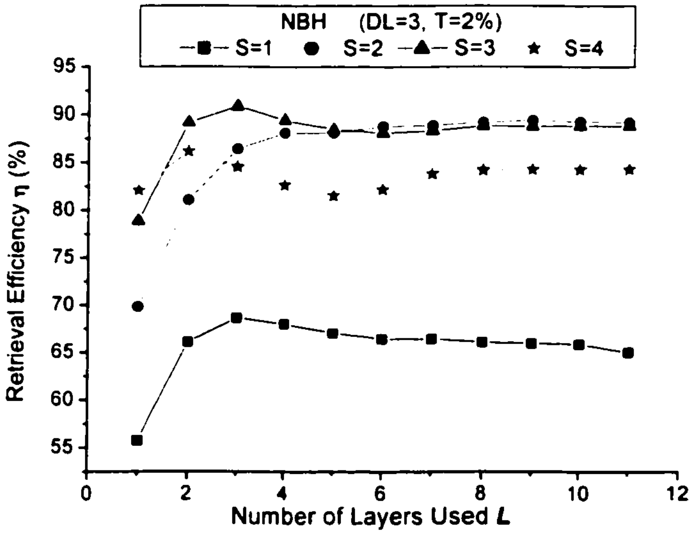


Figure 6-9 Retrieval performance of Index-NBH at a fixed DL **and** T **(**η **vs.** L**). Here,** $DL = 5$, $T = 2\%$, **and** $S = 1 \sim 4$

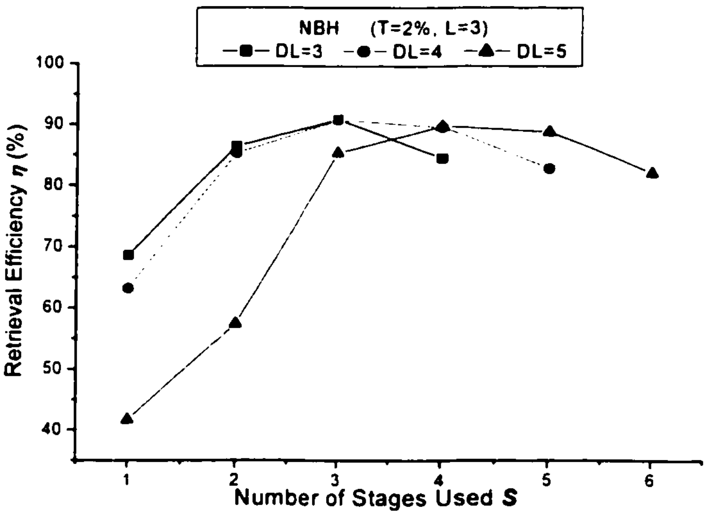


Figure 6-10 Retrieval performance of Index-NBH at a fixed T and L (η vs. S). Here, $L = 3$, $T = 2\%$, $DL = 3 \sim 5$, and $S = 1 \sim 6$

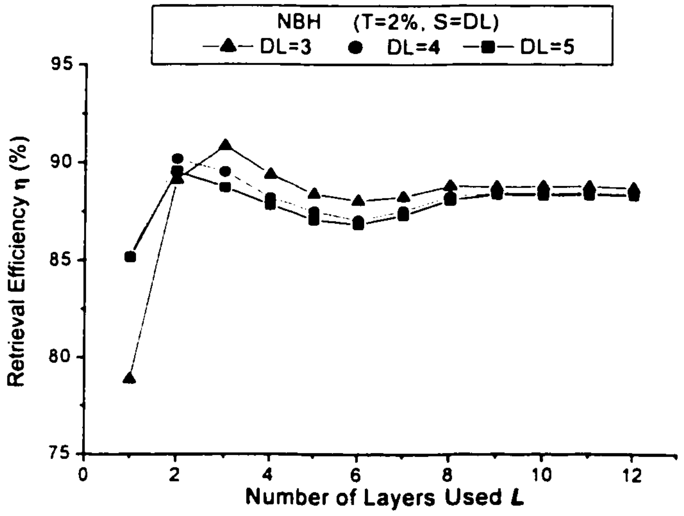


Figure 6-11 Retrieval performance of Index-NBH at a fixed T and S (η vs. L). $T = 2\%$, $DL = 3 \sim 5$, $S = DL$, and $L = 1 \sim 12$

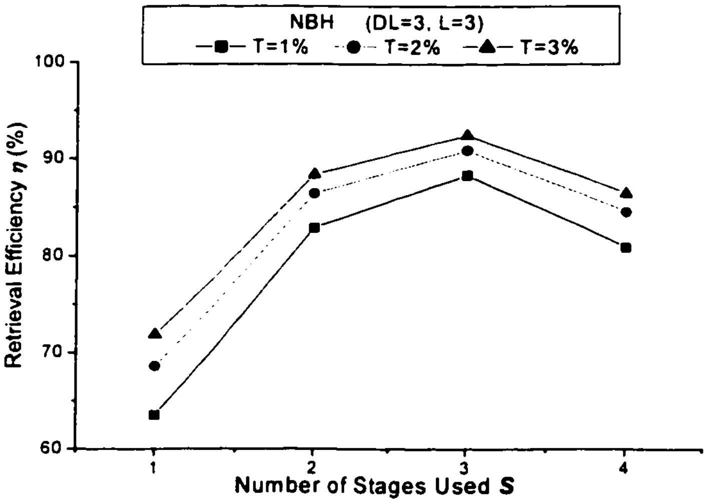


Figure 6-12 Retrieval performance of Index-NBH at a fixed DL and L (η vs. S). Here, $DL = 3$, $L = 3$, $T = 1\% \sim 3\%$, and $S = 1 \sim 6$

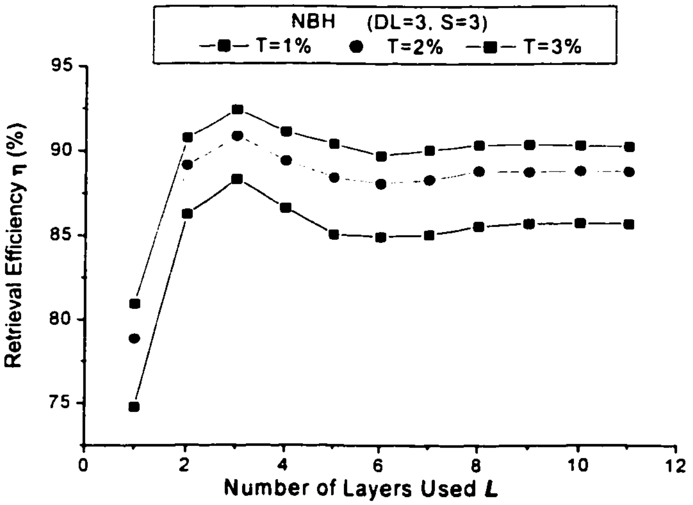


Figure 6-13 Retrieval performance of Index-NBH at a fixed DL and S (η vs. L). Here, $DL = 3$, $S = 3$, $T = 1\% \sim 3\%$, and $L = 1 \sim 11$

At last, the curves in both Figure 6-12 and Figure 6-13 show that η will increase with tolerance T. It is found that the best performance is about 91% achieved at $S = 3$ and $L = 3$.

6.4 Joint Indexing of SBM and NBH

In this section, we discuss the combination of Index-SBM and Index-NBH, which is referred to as Index-CSN.

6.4.1 Computational complexity

The distance calculation for Index-CSN is the summation of the distance for Index-SBM and Index-NBH. Likewise, its computational complexity is also the summation of that for the later two, which is expressed as follows:

$$\begin{cases} O_{csn}(\pm) \approx 3L \times \frac{X_{sz}}{2^n} \times \frac{Y_{sz}}{2^n} + 6L \times (S+3) \\ O_{csn}(\oplus) = 3L \times \frac{X_{sz}}{2^n} \times \frac{Y_{sz}}{2^n} \\ O_{csn}(\times) = 3 \times L \times (S+1) \end{cases} \qquad (6.7)$$

Eq. (6.7) shows that the computational complexity of Index-CSN is determined by four factors: size of the image, wavelet decomposition level, number of layers used, and number of stages used.

6.4.2 Performance

The retrieval performance of Index-CSN will be analyzed from three aspects, *i.e.*, influence of 1) the number of layers or stages (Figure 6-14 and Figure 6-15); 2) tolerance (Figure 6-16 and Figure 6-17); 3) decomposition level (Figure 6-18 and Figure 6-19). In each case, the other parameters are fixed.

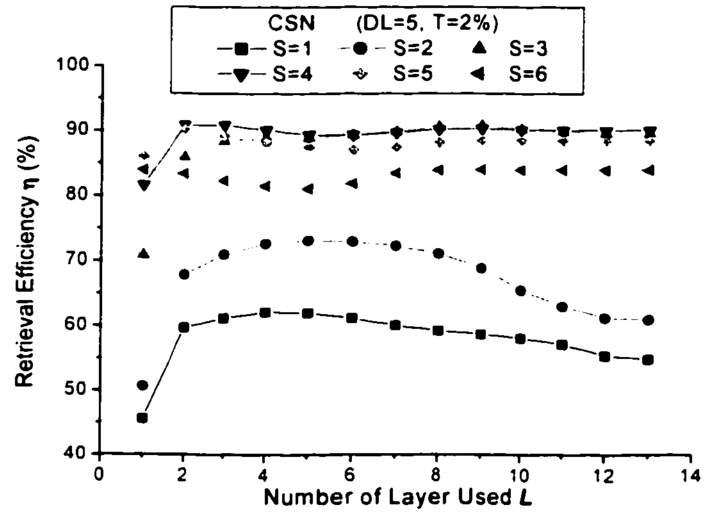


Figure 6-14 Retrieval performance of Index-CSN at a fixed *DL* **and** *T* **(**η **vs.** *L* **). Here,** $DL = 5$, $T = 2\%$, $S = 1 \sim 6$, **and** $L = 1 \sim 13$

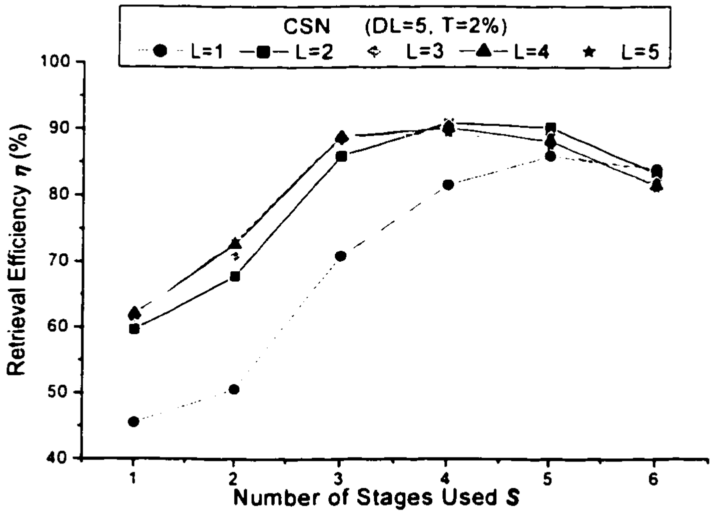


Figure 6-15 Retrieval performance of Index-CSN at a fixed *DL* **and** *T* **(**η **vs.** *S* **). Here,** $DL = 5$, $T = 2\%$, $L = 1 \sim 5$, **and** $S = 1 \sim 6$

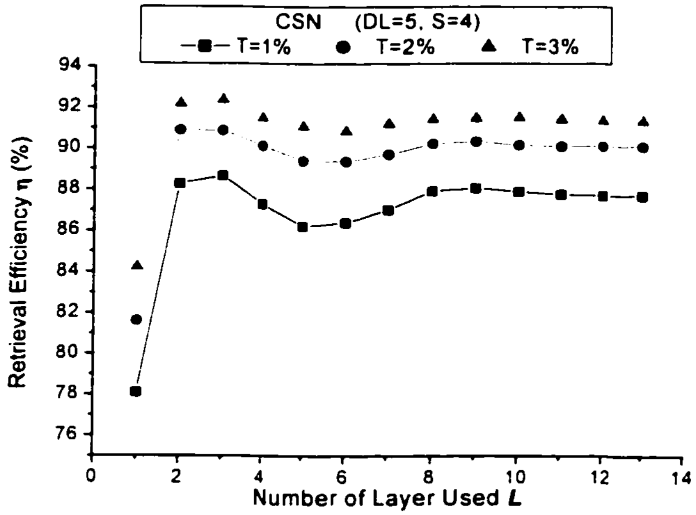


Figure 6-16 Retrieval performance of Index-CSN at a fixed DL **and** S **(**η **vs.** L **). Here,** $DL = 5$, $S = 4$, $T = 1\% \sim 3\%$, **and** $L = 1 \sim 13$

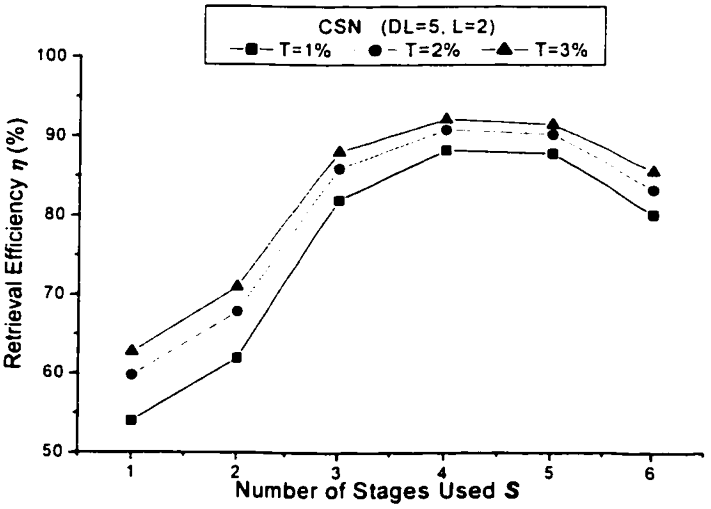


Figure 6-17 Retrieval performance of Index-CSN at a fixed DL **and** L **(**η **vs.** S **). Here,** $DL = 5$, $L = 2$, $T = 1\% \sim 3\%$, **and** $S = 1 \sim 6$.

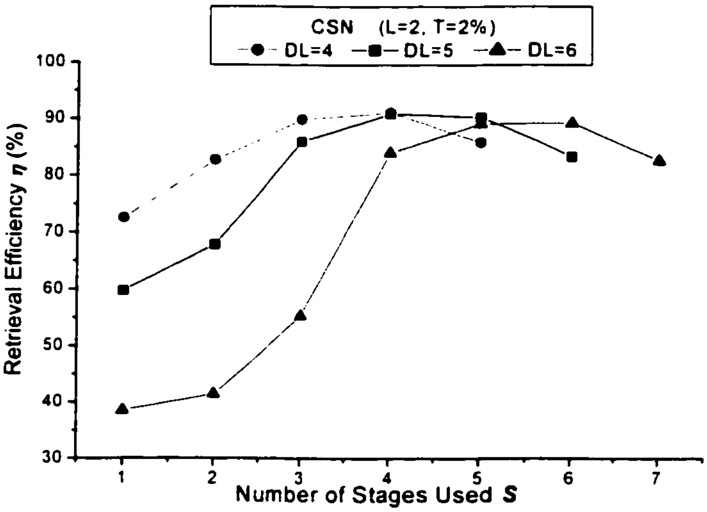


Figure 6-18 Retrieval performance of Index-CSN at a fixed L **and** T **(**η **vs.** S**). Here,** $L = 2$**,** $T = 2\%$**,** $DL = 4 \sim 6$**, and** $S = 1 \sim 7$**.**

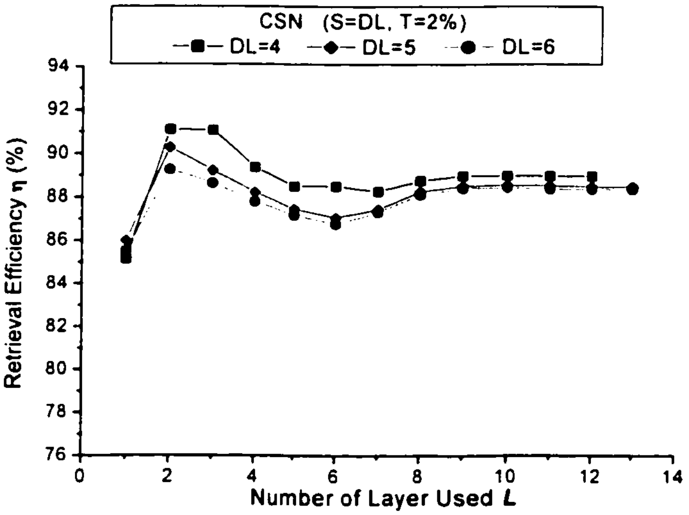


Figure 6-19 Retrieval performance of Index-CSN at a fixed S **and** T **(**η **vs.** L**). Here,** $T = 2\%$**,** $DL = 4 \sim 6$**,** $S = DL$**, and** $L = 1 \sim 13$**.**

When decomposition level and tolerance are fixed, with increasing L, η has an average 10~15% jump from $L=1$ to $L=2$, and is relatively independent of L for $L>2$. While, with the increase of S, η improves rapidly at average 15% increase for each higher S over the previous S when $S \leq 3$, and reaches a maximum at the last second or third S for a given DL. When DL, and L or S are fixed, η has an average 3~5% improvement with every one percent increase in tolerance. Meanwhile, when tolerance is fixed, η decreases with the increase of DL at first several S (typically, when $S < DL$, refers to Figure 6-18), reaches the maximum value at $S = DL$, and degrades slightly at $S = DL+1$. Interestingly, the maximum η achieved at $S = DL$ for different DL are very close. This can be observed more clearly in Figure 6-19 that η curves are almost overlapped each other for different DL when S is chosen to be the value of DL.

The observations above and their reasons are similar to Index-NBH and they will not be repeated. When the JPEG2000-based bitstream is organized in layers, Index-CSN can obtain good retrieval performance by using the first several layers (typically the first 2 or 3 layers) because η is almost saturated when $L>3$ or $L>4$. Moreover, the performance is independent of decomposition level.

From Figure 6-14, when $T=2\%$, $DL=5$, $S=5$, and $L=2$, the retrieval efficiency can achieve as high as 91%.

6.5 Moment of Packet Headers

So far, we have introduced two JEGP2000-based indexing techniques by extracting image features from wavelet coefficients after entropy decoding. As mentioned in Section 6.1, the packet header of a packet provides complete and concise information about the code-blocks included in the packet, and can be extracted without decompressing the bitstream. A useful information concerns the actual number P_0 of bit-planes of each code-block. These P_0s represent the significance of the corresponding code-block in subbands. This inspired us to consider taking advantage of the information inside of packet headers to construct an index for the purpose of retrieval, which is a vector of moments of packet headers (referred as to Index-PHMV) [81].

6.5.1 Index-PHMV Generation

It is known from Section 6.1 that packet headers do not store P_0 for each code-block. Instead, with previously known M_b for each subband, $M_b - P_0$, *i.e.*, P, for each code-block in the corresponding sub-band is stored. Hence, M_b for each subband and P for each code-block have to be extracted first in order to obtain P_0.

In the following, n-level wavelet decomposition is assumed for all images, and the number of sub-bands will be $(3n+1)$. The index generation includes two steps, P_0 collection and index calculation. The details are as follows.

P_0 Collection

1) Collect all packet headers from a coded image bitstream.
2) Extract the values of M_b for each subband from main header.
3) Extract the P values for code-blocks in all sub-bands.
4) Calculate P_0 from $M_b - P$.

Index Calculation

1) Calculate mean and variance of P_0 for each subband.

If the image is n-level wavelet decomposed, there will be a total of $(3n+1)$ pairs of mean and variance. Assume that sub-band b is divided into t_b code-blocks, let $\varepsilon_{phmv,b}$ and $\sigma_{phmv,b}$ denote the mean and variance, respectively:

$$\varepsilon_{phmv,b} = \sum_{j=0}^{t_b-1} \frac{P_{0,j}}{t_b} \tag{6.8}$$

$$\sigma_{phmv,b}^{2} = \sum_{l=0}^{t_b-1} \frac{\left(P_{0,l} - \varepsilon_{phmv,b}\right)^2}{t_b} \tag{6.9}$$

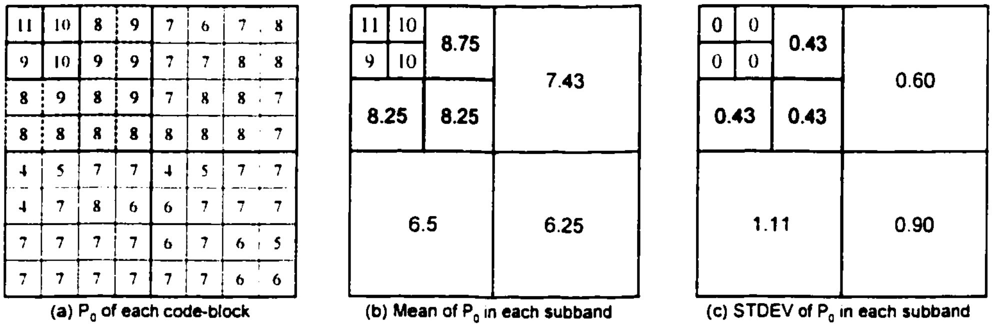
2) Vector of means and vector of standard deviation are stored as an index

Because the wavelet coefficients have the inherent hierarchical property, the distribution of the numbers of bit-planes in all subbands will also appear hierarchically in the form of resolution. To take advantage of this point, the vectors can be constructed progressively. Hence, the dimension of each vector is determined by the wavelet decomposition level of images and the number of levels to be used. We have assumed an n-level decomposition whereas S-stages will be used for retrieval ($1 \le S \le n+1$, $S = 1$ equivalent to the lowest resolution in DWT). Therefore, the vector size will be $\left(3(S-1)+1\right)$. Let $v_{phmv,\varepsilon}^{(S)}$ and $v_{phmv,\sigma}^{(S)}$ denote vector of means and vector of standard deviations, respectively:

$$v_{hJmv,\varepsilon}^{(S)} = \left(\varepsilon_{phmv,0}, \varepsilon_{phmv,1}, \cdots, \varepsilon_{phmv,b}, \cdots, \varepsilon_{phmv,3(S-1)}\right) \tag{6.10}$$

$$v_{phmv,\sigma}^{(S)} = \left(\sigma_{phmv,0}, \sigma_{phmv,1}, \cdots, \sigma_{phmv,b}, \cdots, \sigma_{phmv,3(S-1)}\right) \tag{6.11}$$

where b denotes a sub-band ($b = 0$ refers to the lowest resolution sub-band).



11	10	8	9	7	6	7	8
9	10	9	9	7	7	8	8
8	9	8	9	7	8	8	7
8	8	8	8	8	8	8	7
4	5	7	7	4	5	7	7
4	7	8	6	6	7	7	7
7	7	7	7	6	7	6	5
7	7	7	7	7	7	6	6

(a) P_0 of each code-block

(b) Mean of P_0 in each subband

(c) STDEV of P_0 in each subband

Figure 6-20 Illustration of Index-PHMV. The index was extracted from component R of the Lena color image. $DL = 3$**, i.e.,** $3 \times 3 + 1 = 10$ **subbands.**

And, when S-stages are used, Index-PHMV is the combination of them:

$$Index_{phmv}^{(S)} = \left\{ v_{phm,\varepsilon}^{(S)}, v_{phm,\sigma}^{(S)} \right\} \tag{6.12}$$

Figure 6-20 shows an example of Index-PHMV. The index is extracted from component R of the Lena color image. Here, wavelet decomposition level is 3, *i.e.*, there are $3 \times 3 + 1 = 10$ subbands, so as to result in 10 pairs of means and variances in the index.

6.5.2 Computational Complexity

Assume that S-stages be used to generate an index, the distance of indices between the query image Q and the candidate image C is the summation of distance for vector mean and vector variation, each of them is determined using the L^1 metric as shown in Eq. (3.3).

The computational complexity is only decided by the number of stages used, which is the same as Index-WMV in this aspect. When $DL = n$, if S stages are used to construct the index, where $1 \le S \le n+1$, the computational complexity includes $2(3S-2)$ subtractions, $6S-5$ additions and $2(3S-2)$ multiplications for each color component. For distance calculation with 3 color components, the complexity is 3 times of that for one component, which is given as follows:

$$\begin{cases} O_{phmv}(\pm) = 35S + 11 \\ O_{phmv}(\times) = 11S + 6 \end{cases} \tag{6.13}$$

From the Eq. (6.13), the computational complexity increases with only S and independent of the number of wavelet coefficients.

For $DL = 3$, $S = 4$, the average running time for a query image with 60-image retrieved in a 3000-image database is about 152 ms.

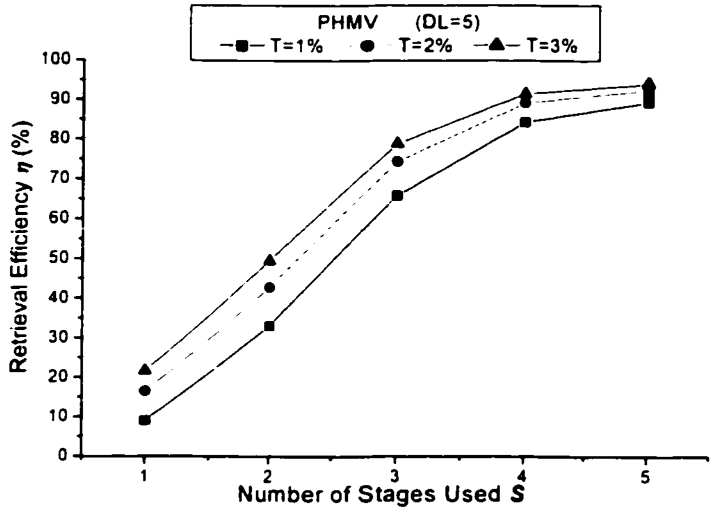


Figure 6-21 Retrieval performance of Index-PHMV. η **vs.** S **at** $DL = 5$**, and** $T = 1\% \sim 3\%$.

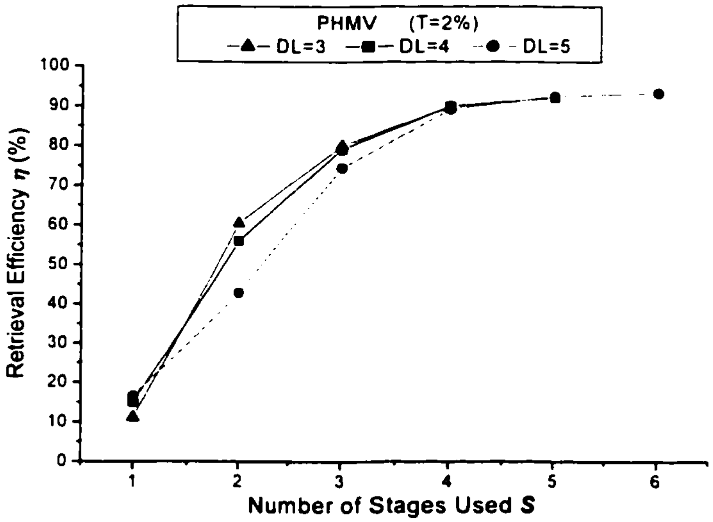


Figure 6-22 Retrieval performance of Index-PHMV. η **vs.** S **at** $T = 2\%$**, and** $DL = 3 \sim 5$.

6.5.3 Performance

The retrieval efficiency of Index-PHMV versus S is shown in Figure 6-21 for different T at $DL = 5$, and Figure 6-22 for different wavelet decomposition level DL at $T = 2\%$.

From Figure 6-21, the retrieval efficiency η follows the same trend with increasing S at a fixed decomposition level. When T is fixed, η increases with S in a big slope, the improvement even reaches 30% from $S = 2$ to $S = 3$. This is because, when $S \leq 2$, the number of code-blocks in each subband is very small (even down to 1 for subband 0~3 if the size of code-block is right the size of LL subband), resulting in the same small number of P_0. The mean and variance using a data set in such small size may generate large bias from the actually statistical characteristics of the corresponding code-block. When $S > 2$, the number of code-blocks increases exponentially so as that the mean and variance calculated from P_0 can approach the actually statistical characteristics, and the retrieval performance is improved as well.

On the other hand, for different T and fixed DL (Figure 6-21), η is improved with the increase of$T$, which is obvious. While for different DL and fixed T (Figure 6-22), the curve for each DL makes little difference from the others so that η can similarly regarded as independent of DL.

6.6 Comparison of JPEG2000-based Techniques

So far, three JPEG2000-based and one derived indexing techniques have been introduced. In this section, both the computational complexity and retrieval performance of the 4 techniques and related previous work WMV technique will be compared in the following. To clearly state the difference of the techniques, they will be divided into two groups, *i.e.*, group one, SBM, NBH and CSN, and group two, NBH vs. PHMV vs. WMV. For simplicity, the weights for all indices are set to unity so that the computational complexity due to the multiplication operations does not have to be considered.

6.6.1 SBM vs. NBH vs. CSN

In this group, both SBM and NBH indexing techniques extract indices from the decoded bit-planes of wavelet coefficients, and CSN technique was derived from the both. It is useful to compare these bit-plane-based indexing techniques.

As introduced in the earlier sections, the computational complexity of Index-SBM and Index-CSN is dependent on not only the number of layers used, but also image-size and decomposition level. Therefore, if the size of retrieved images is too large, *e.g.*, larger than 256×256, or the decomposition level is too low, *e.g.*, $DL = 2$, the running time for retrieving a set of images with respect to a query image will increase in the order of the power of 4. Hence, it is unpractical to apply SBM and CSN techniques in such case. On the other hand, the complexity for Index-NBH is just dependent on the number of layers and stages used, and independent on the size of image and decomposition level. As analyzed before, the best performance is typically achieved at the first several layers but in higher stages, however, the number of stages is very small compared to the size of the image, consequently, Index-NBH is suitable for variable size of images and decomposition level from the viewpoint of the computational complexity. Table 6-4 shows an example of the complexity for the 3 techniques at $DL = 5$, $S = 4$, and image size in 256×256. Even for DL as large as 5 in the example, the complexity for Index-SBM is still considerably high compared to Index-NBH when L is large.

Table 6-4 Computational complexity of Index-SBM, Index-NBH and Index-CSN at $DL = 5$, $T = 1\%$, and $S = 4$ if necessary. (Image size: 256×256)

L		1	2	3	4	5	6	7	8
$O_{sbm}^{(5)}(\ )$	±	206	416	626	836	1046	1256	1466	1676
	⊕	192	384	576	768	960	1152	1344	1536
$O_{nbh}^{(5)}(\ )$	±	23	47	71	95	119	143	167	191
$O_{csn}^{(5)}(\ )$	±	230	464	698	932	1166	1400	1634	1868
	⊕	192	384	576	768	960	1152	1344	1536

Concerning the retrieval performance, we first compare Index-NBH and Index-CSN. Index-NBH is a subset of Index-CSN. It is expected that the latter could provide improvement over the former. However, from Figure 6-23, in which the decomposition level and the number of layers used are fixed, it is found that Index-CSN only has improvement over Index-NBH at lower S ($S \leq 2$), and the two curves are almost overlapped since $S \geq 3$. The reason for this is that Index-CSN is a combination of Index-SBM and Index-NBH. Because Index-SBM is from just LL subband, when $S \leq 2$, the contribution from Index-SBM helps Index-CSN to achieve higher retrieval efficiency over Index-NBH, for $S \geq 3$, however, it is no longer to provide more details on the higher stages, and consequently, η of Index-CSN tends to go together with that of Index-NBH.

Figure 6-24 and Figure 6-25 show the performance comparison among Index-SBM, Index-NBH and Index-CSN when the decomposition level and the number of stages used are given. As analyzed before, SBM technique is relatively low efficient whereas NBH is high, η of Index-CSN always provides improvement over Index-SBM, but only improves over Index-NBH at lower S, which is in consistent with the observation from Figure 6-23.

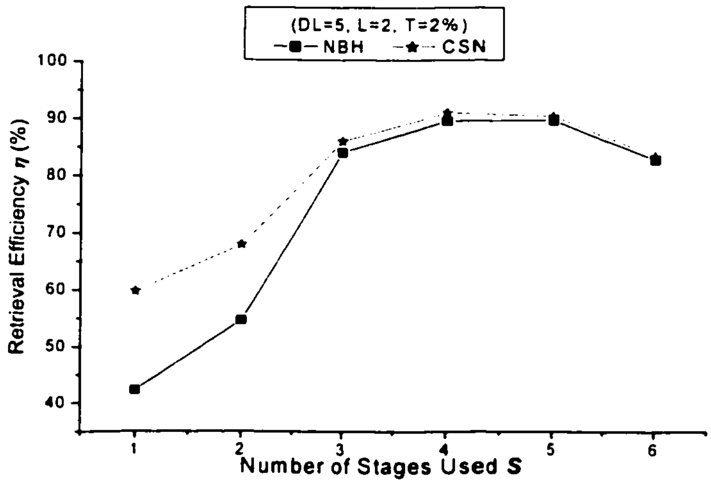


Figure 6-23 Retrieval performance comparison between Index-NBH and Index-CSN. Here, $DL = 5$, $L = 2$, and $T = 2\%$

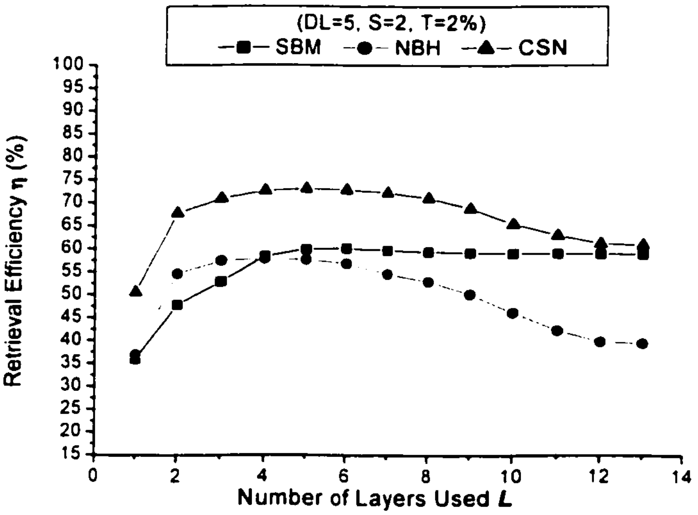


Figure 6-24 Retrieval performance comparison among Index-SBM, Index-NBH and Index-CSN. Here, $DL = 5$, $S = 2$, **and** $T = 2\%$

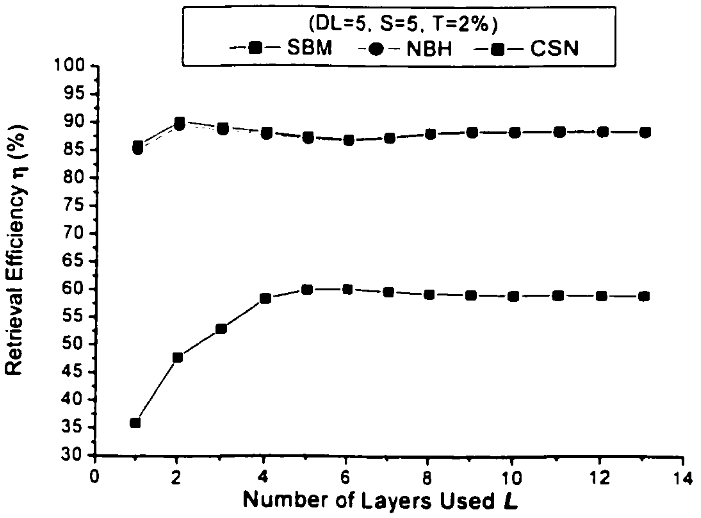


Figure 6-25 Retrieval performance comparison among Index-SBM, Index-NBH and Index-CSN. Here, $DL = 5$, $S = 5$, **and** $T = 2\%$

Compared to Index-NBH, Index-CSN provides the better retrieval performance at lower S and is not superior when S is large, whereas, its complexity is considerably higher especially for large image sizes and low decomposition level. Hence, Index-CSN is only suitable to apply when the information needed to construct an index can be only extracted from the first one or two stages, while Index-NBH can be chosen if the information can be obtained from more stages. As for Index-SBM, it is better not to be used independently but jointly with Index-NBH due to its relatively low retrieval efficiency.

6.6.2 NBH vs. PHMV vs. WMV

In this group, both Index-PHMV and Index-WMV consist of mean and standard deviation pairs with respect to each subband, namely, their structure is exactly the same except the process of generating the mean and standard deviation pair. For Index-WMV, the moment pairs corresponding to each subband are calculated directly from the wavelet coefficients inside the subband, while, for Index-PHMV, they are derived from P_0s (the numbers of bit-planes of code-blocks within the subband). It is worth comparing the two techniques. Meanwhile, we still want to compare bit-plane-based techniques and packet-header-based technique. Because Index-NBH is a good one among the previous bit-plane-based techniques, it is selected to compare with Index-PHMV as well.

The computational complexity of each technique is shown in Table 6-5 at $DL = 5$ and $L = 2$. It is found that all indices have comparable low complexities. Therefore, they all can be employed for retrieving in large size database.

Table 6-5 Computational complexity of Index-NBH, Index-PHMV and Index-WMV at $DL = 5$, $T = 2\%$, and $L = 2$ if necessary. (Image size: 256×256)

S		1	2	3	4	5
$O_{nbh}^{(5)}(\ )$	±	11	23	35	47	59
$O_{phmv}^{(5)}(\ )$	±	47	83	119	155	191
$O_{wmv}^{(n)}(\ )$	±	47	83	119	155	191

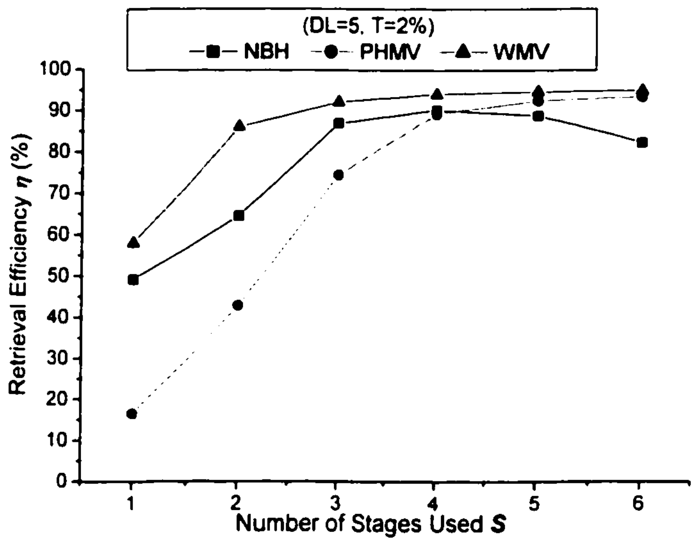


Figure 6-26 Retrieval performance comparison among Index-NBH, Index-PHMV and Index-WMV, at $DL = 5$, $T = 2\%$

Figure 6-26 shows the retrieval performance for the 3 techniques. All the techniques provide better performance at higher S. Index-WMV has the best performance at all Ss, which is obvious because the moment pairs of the index are calculated using fully lossless wavelet coefficients inside each subband. Interestingly, the performance of Index-NBH when $S \leq 4$ and of Index-PHMV when $S \geq 4$ is close to Index-WMV. This is because when $S \leq 4$, Index-NBH can provide enough details to calculate index distance, but for Index-PHMV, the moment pairs are calculated from a small number of P_0s, which expresses the distribution corresponding to the subbands far from statistically and resulting in large bias in statistical calculation, while, the situation is inverse when $S \geq 4$ and leads to the above observation. That is, from the view point of performance, Index-NBH may be a better choice unless all stages can be used to generate Index-PHMV. However, Index-NBH has to be extracted after decoding JPEG2000 bitstream while Index-PHMV does not because it uses packet header directly. Typically, the decoding of JPEG2000 bitstream is time-consuming, and this implies that the generation of bit-plane-based indices including Index-NBH be time-consuming as well even though the computational complexity for distance may be quite small. On the other hand, according to JPEG2000 standard, the packet headers are allowed to be included in the main header such that the information needed for all stages can be extracted without decoding the bitstream. At this point, Index-PHMV is much better than Index-NBH.

6.7 Summary

In this chapter, four indexing techniques in the JPEG2000 framework, *i.e.*, Index-SBM, Index-NBH, and their combination Index-CSN, and Index-PHMV, were proposed and analyzed. From the view point of feature extraction, Index-SBM, Index-NBH and Index-CSN belong to bit-plane based techniques, while Index-PHMV is based on the packet header. Index-SBM presents the precise spatial information but lacks robustness in translation and rotation. Instead of using it independently, it is suitable to retrieve images jointly with other techniques to provide auxiliary location information. Except Index-SBM, all the other three can achieve retrieval efficiency of over 90%.

With respect to the computational complexity, Index-CSN has the largest complexity. Index-NBH and Index-PHMV have comparable complexity for index matching lower than Index-CSN. However, for index generation, Index-PHMV has lowest complexity. So, when the index has to be calculated in real-time, Index-PHMV will be a better choice among JPEG2000-based indexing techniques.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

In this thesis, two classes of wavelet-based image indexing techniques are proposed and analyzed. One class is developed in the embedded zerotree wavelet framework while the other class is in the JPEG2000 framework.

In the EZW framework, two basic techniques, Index-MNSCH and Index-DNSSCH, and their combinations, *i.e.*, Index-CMD, Index-CML, Index-CDL and Index-CMDL, are discussed, analyzed and evaluated. These techniques are based on the histogram of significant wavelet coefficients with respect to different thresholds. Experimental results show that these techniques can achieve good retrieval performance. Among them, Index-CMDL provides the best retrieval performance, but it has a large computational complexity. Index-CMD provides the 2nd-best retrieval performance, *e.g.* over 90%; however, its complexity is independent of image size and relatively small. Considering both the retrieval-efficiency and the complexity, Index-CMD might be a better choice than Index-CMDL in the EZW framework.

In the JPEG2000 framework, four indexing techniques, *i.e.*, Index-SBM, Index-NBH, Index-CSN and Index-PHMV, have been proposed and analyzed. Among them, the first three techniques are bit-plane based, and the last technique is packet header based. Here, "bit-plane based" and "packet header based" refer to how indices are extracted instead of how indices are compared. The three bit-plane-based techniques generate indices from the bit-planes decomposed out of wavelet coefficients which are decoded from JPEG2000 bitstream. In this category, Index-SBM is a 2-D significant-bit-map array, which can provide good local information for an image; Index-NBH is a 2-D histogram of the number of significant bits found in bit-planes; and Index-CSN is a combination of Index-SBM and Index-NBH. On the other hand, the packet header based technique, Index-PHMV, extracts indices from, as the name suggested, the packet-header, which can be obtained without decoding the JPEG2000 bitstream. Experimental results show that three of the four techniques, Index-NBH, Index-CSN, and Index-PHMV can provide good retrieval performance with respect to different conditions: each of them can achieve over 90% with appropriate parameter settings. Overall, Index-CSN has the best retrieval efficiency but it also has a high computational complexity. Index-NBH and Index-PHMV provide similar performance with respect to both retrieval efficiency and computational complexity. However, the index generation time for Index-NBH is considerably longer than for Index-PHMV because of the need to decode the JPEG2000 bitstream. Hence, for JPEG2000-based retrieval systems, the technique to be selected for indexing depends on the format of input JPEG2000 bitstream. If the detailed packet headers are prefixed to the JPEG2000 bitstream, Index-PHMV is a better choice; otherwise Index-NBH is probably a better alternative.

7.2 Future Work

The work in this thesis can be extended in many different directions. First, as discussed earlier, the wavelet transform is not RSTN (*i.e.*, rotation, scaling, translation and reflection) invariant. Hence, the indexing and retrieval techniques based on wavelet transform are not robust to rotation and translation. It is expected that a hybrid scheme, for example, DWT combined with DFT, which is RSTN invariant might solve this problem.

Second, a detailed investigation needs to be carried out for large-size databases, *e.g.*, natural images, trademarks and hand-drawings. This is because while one CBIR technique works very well in one image database, it might have a poor performance on another database. Similarly, one technique might have high retrieval efficiency in a smaller-size image database, but not in a large-size one. In this thesis, the database used to evaluate the proposed indexing techniques consists of 3000 natural images. It is necessary to further evaluate them using other databases in different categories and sizes.

Meanwhile, a complete image retrieval system requires a user-friendly interface, which was not implemented in this work. Perl, CGI or JAVA might be good choices to realize this.

REFERENCES

[1] J. van den Berg, "Subject Retrieval in Pictorial Information Systems," *Proceedings of the 18th International Congress of Historical Sciences. Round Table 34: Electronic Filing, Registration, and Communication of Visual Historical Data*, Montreal, 1995.

[2] "IS 10918-1 (ITU-T T.81): JPEG Image Coding System."

[3] "ISO/IEC FCD15444-1: JPEG 2000 Image Coding System," 16 March 2000.

[4] R. Polikar, "Wavelet Tutorial," 1999.

[5] "ISO/IEC JTC1/SC29/WG11 N4030: Coding of Moving Pictures and Audio," 2001.

[6] A. Grossmann and J. Morlet, "Decomposition of Hardy Functions into Square Integrable Wavelet of Constant Shape," *SIAM journal on mathematical analysis*, vol. 15, pp. 723-736, 1984.

[7] I. Daubechies, *Ten Lectures on Wavelets*, 2nd ed. Philadelphia: SIAM, 1992.

[8] J. Sunderland, "Image Collections: Librarians, Users and Their Needs," *Art Libraries Journal*, vol. 7, pp. 41-49, 1982.

[9] V. N. Gudivada and V. V. Raghavan, "Modeling and retrieving images by content," *Information Processing & Management*, vol. 33, pp. 427-452, 1997.

[10] P. Aigrain, H. Zhang, and D. Petkovic, "Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review," *Multimedia Tools and Applications*, vol. 3, pp. 179-202, 1996.

[11] M. De Marsicoi, L. Cinque, and S. Levialdi, "Indexing pictorial documents by their content: a survey of current techniques," *Image and Vision Computing*, vol. 15, pp. 119-141, 1997.

[12] M. K. Mandal, F. Idris, and S. Panchanathan, "A critical evaluation of image and video indexing techniques in the compressed domain," *Image and Vision Computing*, vol. 17, pp. 513-529, 1999.

[13] Y. Rui, T. S. Huang, and S.-F. Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues," *Journal of Visual Communication and Image Representation*, vol. 10, pp. 39-62, 1999.

[14] Y. Rui, T. S. Huang, and S.-F. Chang, "Image Retrieval: Past, Present, And Future," *International Symposium on Multimedia Information Processing*, Taipei, Taiwan, 1997.

[15] R. Brunelli and O. Mich, "On the use of histograms for image retrieval," *Proceedings of IEEE Multimedia Systems '99 International Conference on Multimedia Computing and Systems*, Florence, Italy, 1999.

[16] M. J. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, pp. 11-32, 1991.

[17] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 837-842, 1996.

[18] P. Wu, B. S. Manjunath, S. D. Newsam, and H. D. Shin, "A Texture Descriptor for Image Retrieval and Browsing," *Computer Vision and Pattern Recognition Workshop*, 1999.

[19] T. Gadi, R. Benslimane, M. Daoudi, and S. Matusiak, "Fuzzy Similarity Measure for Shape Retrieval," *Vision Interface '99*, Trois-Rivieres, Canada, 1999.

[20] M. Adoram and M. S. Lew, "IRUS: Image Retrieval Using Shape," *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, 1999.

[21] R. C. Veltkamp and M. Tanase, "Content-Based Image Retrieval Systems: A Survey," Institute of Information & Computing Science, University Utrecht UU-CS-2000-34, 2000.

[22] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, and D. Petkovic, "The QBIC project: Querying images by content using color, texture, and shape. In Storage and Retrieval for Image and Video Databases,," IBM Technical Report 1993.

[23] K. R. Castleman, *Digital Image Processing*. New Jersey: Prentice-Hall, 1996.

[24] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," *Proc. SPIE Storage Retrieval Still Image Video Databases IV*, 1996.

[25] G. Pass and R. Zabih, "Histogram refinement for content based image retrieval," *Proc. IEEE Workshop Applications Computer Vision*, pp. 96-102, 1996.

[26] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1997.

[27] J. R. Parker, *Algorithms for Image Processing and Computer Vision:* John Wiley & Sons, 1997.

[28] J. Posl and H. Niemann, "Wavelet features for statistical object localization without segmentation," *Proceedings of International Conference on Image Processing,* 1997.

[29] M. Tuceryan and A. K. Jain, in *Handbook of Pattern Recognition and Computer Vision,* C. H. Chen, Ed.: World Scientific, 1993, pp. 235-276.

[30] H. Tamura and N. Nokoya, "Image database systems: a survey," *Pattern Recognition,* vol. 17, pp. 29-43, 1984.

[31] H. Zhang and D. Zhong, "A Scheme for Visual Feature Based Image Indexing," *Proc. SPIE: Storage and Retrieval for Image and Video Databases III,* 1995.

[32] F. Liu and R. W. Picard, "Periodicity, Directionality, and Randomness: Wold Features for Image Modeling and Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 18, pp. 722-733, 1996.

[33] J. R. Smith and S.-F. Chang, "Automated binary Texture Feature Sets for Image Retrieval," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Proceeding,* Atlanta, USA, 1996.

[34] S. Loncaric, "A survey of shape analysis technique," *Pattern Recognition,* vol. 31, pp. 983-1001, 1998.

[35] S. Sclaroff and A. Pentland, "Modal Matching for Correspondence and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 17, pp. 545-561, 1995.

[36] H. Nishida, "Shape Retrieval from Image Databases through Structural Feature Indexing," *Vision Interface '99,* Trois-Rivi 鑢 es, Canada, 1999.

[37] P. Enser, "Progress in Documentation: Pictorial Information Retrieval," *Journal of Documentation,* vol. 51, pp. 126-170, 1995.

[38] M. D. Marsicoi, L. Cinque, and S. Levialdi, "Indexing Pictorial Documents by Their Content: a Survey of Current Techniques," *Image and Vision Computing,* vol. 15, pp. 119-141, 1997.

[39] H. S. Stone and C.-S. Li, "Image matching by means of intensity and texture matching in the Fourier domain," *Proc. SPIE Storage and Retrieval for Image and Video Databases IV,* San Jose, CA, USA, 1996.

[40] M. F. Augusteijn, L. E. Clemens, and K. A. Shaw, "Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier," *IEEE Trans. Geoscience and Remote Sensing*, vol. 33, pp. 616-626, 1995.

[41] A. Celentano and E. D. Lecce, "A FFT-based technique for image signature generation," *Proc. SPIE: Storage and Retrieval for Image Databases V*, San Jose, CA, USA, 1997.

[42] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content-base manipulation of image databases," *SPIE Conference on Storage and Retrieval of Image and Video Databases II*, San Jose, CA, USA, 1997.

[43] G. Pass, R. Zabih, and J. Miller, "Comparing Images Using Color Coherence Vectors," *Proc.Fourth ACM Multimedia conference*, 1996.

[44] Y. Deng, B. S. Manjunath, C. Kenney, and M. S. Moore, "An efficient color representation for image retrieval," *IEEE Transaction on Image Processing*, pp. 140-147, 2001.

[45] L. V. Tran and R. Lenz, "PCA-based representation of color distributions for color based image retrieval," *Proc. International Conference Image Processing 2001*, Greece, 2001.

[46] H. Schweitzer, "An eigenspace approach to multiple image registration," *Proceedings of the First Image Registration Workshop*, NASA Goddard Space Flight Center, 1997.

[47] X. Tang and W. K. Stewart, "Texture Classification Using Principle Component Analysis Techniques," *Proc. SPIE*, 1994.

[48] S.-F. Chang, "Compressed-domain techniques for image/video indexing and manipulation," *Proceedings of the 1995 International Conference on Image Processing*, Washington DC, 1995.

[49] B. Furht and P. Saksobhavivat, "A Fast Content-Based Video and Image Retrieval Technique Over Communication Channels," *Proc. of SPIE Symposium on Multimedia Storage and Archiving Systems*, Boston, MA, USA, 1998.

[50] B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," *Proceedings of the SPIE: Storage and Retrieval for Image and Video Database*, 1996.

[51] C. W. Ngo, T. C. Pong, and R. T. Chin, "Exploiting Image Indexing Techniques in DCT domain," *IAPR International Workshop on Multimedia Media Information Analysis and Retrieval*, HongKong, 1998.

[52] M. Shneier and M. Abdel-Mottaleb, "Exploiting the JPEG Compression Scheme for Image Retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 849-853, 1996.

[53] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast Multiresolution Image Querying," *Proceedings of SIGGAPH 95, in Computer Graphics Proceedings, Annual Conference Series*, pp. 277-286, 1995.

[54] J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei, "Content-Based Image Indexing and Searching Using Daubechies' Wavelets," *International Journal of Digital Library*, vol. 1, pp. 311-328, 1997.

[55] J. You and P. Bhattacharya, "A Wavelet-Based Coarse-to-Fine Image Matching Scheme in A Parallel Virtual Machine Environment," *IEEE Trans. on Image Processing*, vol. 9, pp. 1547-1559, 2000.

[56] N. Sebe, Q. Tian, E. Loupias, M. S. Lew, and T. S. Huang, "Color Indexing Using Wavelet-Based Salient Points," *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, Hilton Head Island, USA, 2000.

[57] E. Albuz, E. Kocalar, and A. A. Khokhar, "Scalable color image indexing and retrieval using vector wavelets," *IEEE Transaction on Knowledge and Data Engineering*, vol. 13, pp. 851-861, 2001.

[58] E. Regentova, S. Latifi, and S. Deng, "Wavelet-based Techniques for Image Similarity Estimation," *Proceedings of ITCC 2000*, Las Vegas, USA, 2000.

[59] S. Ardizzoni, I. Bartolini, and M. Patella, "Windsurf: Region-based image retrieval using wavelets.," *IWOSS'99*, Florence, Italy, 1999.

[60] F. Idris and S. Panchanathan, "Image and Video Indexing Using Vector Quantization," *Machine Vision and Applications*, vol. 10, pp. 43-50, 1997.

[61] A. Vellaikal, S. Dao, and C.-C. J. Kuo, "Content-Based Retrieval of Remote Sensed Images Using a Feature-Based Approach," *Proceedings of SPIE Visual Information Processing IV*, Orlando, 1995.

[62] L. Zhu, "Keyblock: An Approach for Content-Based Image Retrieval." New York: University of New York at Buffalo, 2001.

[63] A. E. Jacquin, "Image coding based on a fractal theory of iterated contrative image transformation," *IEEE Trans. on Image Processing*, vol. 1, pp. 18-30, 1991.

[64] D. Saupe, R. Hamzaoui, and H. Hartenstein, "Fractal image compression: an introductory overview," in *Fractal Models for Image Synthesis, Encoding and Analysis, SIGGRAPH '96 Course Notes XX*, J. Hart, Ed. New Orleans, 1996.

[65] M. Vissac, J.-L. Dugelay, and K. Rose, "A Fractals-Inspired Approach to Content-based Image Indexing," *IEEE International Conference on Image Processing,* 1999.

[66] M. Nappi, G. Polese, and G. Tortora, "Content Based Retrieval of Fractal Compressed Images," *Proceedings of Image and Video Content Based Retrieval,* Milano, Italy, 1998.

[67] M. Nappi, G. Polese, and G. Tortora, "FIRST: Fractal indexing and retrieval system for image databases," *Image and Vision Computing,* vol. 16, 1998.

[68] A. Zhang, B. Cheng, and R. Acharya, "A Fractal-Based Clustering Approach in Large Visual Database Systems," *The International Journal on Multimedia Tools and Applications,* vol. 3, pp. 225-244, 1996.

[69] A. Zhang, B. Cheng, R. Acharya, and R. Menon, "Comparison of Wavelet Transforms and Fractal Coding in Texture-based Image Retrieval," *Proceedings of the SPIE Conference on Visual Data Exploration and Analysis III,* San Jose, CA, 1996.

[70] F. Idris and S. Panchanathan, "Storage and Retrieval of Compressed Images Using Wavelet Vector Quantization," *Journal of Visual Languages & Computing,* vol. 8, pp. 289-301, 1997.

[71] C. L. Sabharwal and S. R. Subramanya, "Indexing Image Databases Using Wavelet and Discret Fourier Transform," *Proc. of the 16th ACM SAC2001 Symposium on Applied Computing,* 2001.

[72] K. Liang and C. J. Kuo, "Progressive image indexing and retrieval based on embedded wavelet coding," *IEEE 1997 International Conference on Image Processing,* Santa Barbara, CA, 1997.

[73] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transaction on Signal Processing,* vol. 41, pp. 3455-3462, 1993.

[74] R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," *CVGIP: Graphical Models and Image Processing,* 1992.

[75] M. K. Mandal, T. Aboulnasr, and S. Panchanathan, "Image Indexing Using Moments and Wavelets," *IEEE Transactions on Consumer Electronics,* vol. 42, pp. 557-565, 1996.

[76] C. Liu and M. Mandal, "Image Indexing in the Embedded Zerotree Wavelet Framework," *Proc. of the International Conference on Communications, Control and Signal Processing (CCSP),* Bangalore, India, 2000.

[77] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transaction on Image Processing,* pp. 1158-1170, 2000.

[78] T. Halbach, "Blockwise mixed fixed-length/variable-length coding in JPEG2000," *Proc. NORSIG 1999,* Asker (Norway), 1999.

[79] A. N. Skodras, C. A. Christopoulos, and T. Ebrahimi, "JPEG2000: The Upcoming Still Image Compression Standard," *Proceedings of 11th Portuguese Conference on Pattern Recognition (RECPA00D 20),* Porto, Portugal, 2000.

[80] C. Liu and M. Mandal, "Image Indexing in the JPEG2000 Framework," *Proc. of SPIE: Internet Multimedia Management Systems,* Boston, USA, 2000.

[81] C. Liu and M. K. Mandal, "Fast Image Indexing Based on JPEG2000 Packet Header," *3rd Intl Workshop on Multimedia Information Retrieval (MIR2001),* Ottawa, Canada, 2001.