AN IMAGE SEARCHING TOOL USING WAVELET WATERMARKING

by

Eugen Toaxen
M.Eng., "Politehnica" University , 1998
B.Sc., "Politehnica" University, 1997

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

We accept this thesis as conforming
to the required standard

[redacted]

Dr. J. C. Muzio, Co-supervisor (Dept. of Computer Science)

[redacted]

Dr. K. Cattell, Co-supervisor (Dept. of Computer Science)

[redacted]

Dr. R. N. S. Horspool, Departmental Member (Dept. of Computer Science)

[redacted]

Dr. V. K. Bhargava, Outside Member (Dept. of Electrical and Computer Engineering)

[redacted]

Dr. W. Lu, External Examiner (Dept. of Electrical and Computer Engineering)

University of Victoria

Supervisor: Dr. Jon Muzio

ABSTRACT

New images and videos are added everyday and others are replaced or removed entirely. In order to search the visual information, a highly efficient automated system is needed that regularly traverses the Web detects the information embedded in images and processes it in such a way to allow for efficient and effective search and retrieval.

In this thesis we are trying a new approach: searching for embedded watermarks in the images. The watermark may contain a variety of information, and could serve different goals: for example, some image processing software may have the option to save an invisible watermark together with the image. The web crawler searches for the existence of the watermark, and the vendor of the image composer can get a measure of the usage of his product on the Web.

The watermark insertion is based on a wavelet transform and is realized in the transform domain. This thesis presents an improved technique of image watermarking adapted for Internet image searching. Results of the experiments of this watermarking scheme are also presented.

Also, we propose a framework named VkMark that could be used for generating watermarks, watermarking the images and searching for the images on the web. Instead of searching for image properties, contents, or similarity measures, VkMark searches for the watermark code.

Examiners:

[redacted]

Dr. J. C. Muzio, Co-supervisor (Dept. of Computer Science)

[redacted]

Dr. K. Cattell, Co-supervisor (Dept. of Computer Science)

[redacted]

Dr. R. N. S. Horspool, Departmental Member (Dept. of Computer Science)

[redacted]

Dr. V. K. Bhargava, Outside Member (Dept. of Electrical and Computer Engineering)

[redacted]

Dr. W. Lu, External Examiner (Dept. of Electrical and Computer Engineering)

TABLE OF FIGURES

1 INTRODUCTION

Digital images, graphics, animation and video are being created and disseminated on the World Wide Web at an incredible rate. The ease of creating and capturing digital imagery has enabled its proliferation, making our interaction with today's on-line information sources largely a "visual" one. Various Web-based applications, such as multimedia publishing, conferencing, information retrieval, content production, demand advancement in various areas of image technologies.

Two important concepts have been introduced to balance the development of the digital media:

- Search and Authoring Engines for Images and Videos
- Authentication, and Watermarking,

The first topic is concerned with Web-based systems and tools that allow general users to search for images from distributed sources, and create new visual content on-line. The target user group includes general individual users who are using various types of computing and communication platforms. The second topic is concerned with techniques which facilitate a trustable environment in which the rights of both consumers and providers are well protected. This thesis focuses on creating a searching tool based on wavelet watermarking to help searching the web for images. A *watermark* is an invisible mark placed on an image that later can be detected and used for a variety of purposes.

Many search engines index the plethora of documents on the World Wide Web. For example, recent systems by Yahoo, Alta Vista, and Excite index Web documents by their textual content. These systems periodically scan the Web, analyze the documents, and create compact and searchable indices. Users enter query terms and/or select subjects to more easily find the desired Web documents. A large component of the content on the Web consists of visual information such as images, graphics, animations, and videos. The visual information on the Web proves highly volatile. Web site developers are constantly adding, removing, and replacing images and videos. Cataloging them requires a highly efficient automated system that regularly traverses the Web, detects visual information, and processes it and indexes it in such a way as to allow efficient and effective search and retrieval. The detection of the useful visual information could be achieved by searching the previously inserted watermarks in images.

A prototype framework that can be used for image watermarking and searching has been developed to fulfill this need. The system searches for images from the Web and catalogs them depending on the embedded information found. The complete system possesses several powerful functionalities. It automates collection of visual information from the Web and automates watermarked image detection.

Much research has been conducted in the topics of image databases and digital watermarking. As the size of image databases grows, traditional methods of image searching break down. It is much harder to locate an image among several thousand images from a database or from the Internet. For this reason, various techniques **[NeJo99]** have been developed to assist in querying images like:

- annotation (index the image database with keywords),
- image properties (colors, textures, and general size, that can be processed without human intervention or interpretation),
- image contents (colors can be used to provide spatial segmentation, scene breaks, and color grouping within an image),
- semantics (semantics is the understanding of the image and information about the image). Categories under this heading may be perceptual similarity and image or picture metadata.

Recently, several World Wide Web image search engines have been reported. An example of an implementation of "semantic" search technique would be "The Webseer" system from the University of Chicago . It detects faces within the images and lets users search by the number of faces. The Interpix **[Sc95]** image search engine does content-based image searching using color histograms. The Interpix system has been integrated with Yahoo's search engine for retrieving images related to some of Yahoo's Web categories. In WebSeek **[SmCh97]** , autonomous Web agents or "spiders" collect the images and videos. The spiders traverse the Web by following hyperlinks between documents. They detect images and videos, retrieve and process them, and add the new information to the catalog.

In this thesis we are trying a new approach: searching for embedded watermarks in the images. The watermark may contain a variety of information, and could serve different goals: for example, some image composers may have the option to save an invisible watermark together with the image. The web crawler searches for the existence of the watermark (a boolean response is enough) and the vendor of the image composer can get a measure of the usability of his product on the Web.

To embed the watermark we used the wavelet transforms. Chapter 2 gives an introduction to the theory of the wavelets. Although the wavelet theory is a relatively new and evolving discipline, in recent years there has been growing interest in wavelet transforms in image processing and time-frequency signal analysis. As compared to the traditional Fourier analysis, the wavelet transform is less rigid, because the wavelet functions have a better space-frequency localization and thus they are suited for analyzing images where most of the informative content is represented by components localized in space such as edges and borders. The main property of wavelet functions is to process data at different scales or resolutions, highlighting both large and small features. Wavelet functions have advantages over traditional Fourier methods in analyzing signals containing many discontinuities or sharp changes and they have been used in several fields: image compression, signal denoising, image smoothing and texture analysis. The main advantages of inserting watermarks in the wavelet transform domain instead of other frequency domains, such as Fourier, are:

- The capability to better localize the features (edges, textures) to whose changes the human eye is less sensitive.
- The capability to localize information in space and frequency. This localization makes a watermark scheme based on wavelets more robust against geometric attacks, such as cropping and scaling.

- The wavelet transform requires a lower computational cost O(n) than the Fourier or the Cosine transform O(n×log(n)), where *n* is in the length of the signal to be transformed.

The image is decomposed into several bands with a pyramid structure using one of the wavelet families, and the watermark is inserted in the most perceptually significant region of the image, spreading the signature over the middle-low frequency coefficients.

A review of the image compression and, especially, wavelet image compression techniques is presented in Chapter 3. Recently, several embedded wavelet coders, e.g. the embedded zerotree wavelet (EZW), the layer zero coder (LZC), the modified LZC, MTWC (Multi-Threshold Wavelet Codec) and the method of set partitioning in hierarchical tree (SPIHT), have been proposed, where the global wavelet transform instead of block DCT (Discrete Cosine Transform) is adopted. The first advantage of embedded wavelet coders is that they give a much better rate-distortion performance than JPEG and the blocking artifact is reduced significantly. The process of watermark embedding and some of the steps from wavelet image compression are sometimes overlapping so we may obtain a speed up of the watermarking process if we use them together.

Chapter 4 defines the watermark, presents the properties of the watermark and gives some examples of watermarking techniques. In general, there are two types of digital watermarking. One of these is embedding the watermark in the spatial domain as against in the frequency domain. Most of the current research focuses on developing various watermarking methods in the frequency domain since most currently popular information transmission and compression techniques are in the frequency domain. A spatial domain method analyzes the original data in the spatial domain and manipulates the LSB (Least Significant Bit) to embed watermark data in which the change should be a perceptibly minimal. Even though this technology easily embeds a watermark, it is not robust to some basic data manipulations.

Some blind watermarking techniques are presented in Chapter 5. Blind watermarking means that the verification of the existence of the watermark is done without requiring prior knowledge of the initial image.

In Chapter 6 a framework prototype (VkMark) for watermarking and searching for images on the World Wide Web is described. VkMark uses a wavelet blind watermarking scheme as the underlying mechanism. Chapter 6 also presents an improved watermarking scheme adapted to Internet image searching (VkWatermark). VkWatermark is not intended to protect against malicious attacks, but just to enable more effective image searching. Results of this watermarking scheme are presented.

This thesis focuses on invisible blind watermarking techniques that are designed to exploit perceptual information in the watermarking process. Digital watermarking adds other information to the image that aids the searching and cataloging of that image.

2 TIME-FREQUENCY TRANSFORMS

The goal of this chapter is to define the underlying mathematical foundations of the time-frequency transforms including the multiresolution approach of wavelet theory.

2.1 Fourier Transform

The conventional signal processing methods use the Fourier transform, whose basis functions are sines and cosines, to analyse signals:

$$F(w) = < f(x), e^{jwx} > \qquad (2\text{-}1)$$

<*> denotes the scalar product, defined by

$$< f(x)\psi(x) > = \int f(x)\psi^{*}(x)dx \qquad (2\text{-}2)$$

where $\psi^{*}(x)$ is the complex conjugate of $\psi(x)$.

Unfortunately, the sine and cosine functions, which are well adapted to analyze stationary signals, are not appropriate to describe nonstationary signals. Therefore, other methods have been developed in order to take into account the variation in time of nonstationary signal features. Among those methods the two best known linear ones are presented in the next sections, that is the Short Time Fourier Transform (STFT) and the Wavelet Transform (WT).

2.2 The Short Time Fourier Transform (STFT)

There is only a minor difference between STFT and Fourier Transform (FT). In STFT, the signal is divided into small enough segments, where these segments (portions) of the signal can be assumed to be stationary. For this purpose, a window function ψ is chosen. The width of this window must corespond to the segment of the signal where its stationarity is valid.

This window function is first located at the very beginning of the signal. That is, the window function is located at $t=0$. Let's suppose that the width of the window is T seconds. At this time instant ($t=0$), the window function will overlap with the first $T/2$ seconds (assuming that all time units are in seconds). The window function and the signal are then multiplied. By doing this, only the first T/2 seconds of the signal is being chosen, with the appropriate weighting of the window (if the window is a rectangle, with amplitude "1", then the product will be equal to the signal). Then this product is assumed to be just another signal, whose FT is to be taken. In other words, FT of this product is taken, just as taking the FT of any signal.

The result of this transformation is the FT of the first *T/2* seconds of the signal. If this portion of the signal is stationary, as it is assumed, then there will be no problem and the obtained result will be a true frequency representation of the first *T/2* seconds of the signal.

The next step, would be shifting this window (for some T_1 seconds) to a new location, multiplying with the signal, and taking the FT of the product. This procedure is followed, until the end of the signal is reached by shifting the window with T_1 seconds intervals.

The following definition of the STFT summarizes all the above explanations in one line:

$$STFT(t,w) = <f(x)\,,\ \psi(x-t)e^{jwx}>\ , \tag{2-3}$$

where $\psi(x-t)$ is the window function centred on t.

If we use a window of infinite length, we get the FT, which gives perfect frequency resolution, but no time information. Furthermore, in order to obtain the stationarity, we have to have a short enough window, in which the signal is stationary. The narrower we make the window, the better the time resolution, and better the assumption of stationarity, but poorer the frequency resolution:

A narrow window results in a good time resolution but a poor frequency resolution, and a wide window results a good frequency resolution but a poor time resolution. This is a consequence of the Heisenberg inequality:

$$\Delta x * \Delta w \geq 1/2$$

2.3 Wavelet Transform

2.3.1 The Continous Wavelet Transform (CWT)

The continuous wavelet transform was developed as an alternative approach to the short time Fourier transform to overcome the resolution problem. The wavelet analysis is done in a similar way to the STFT analysis, in the sense that the signal is multiplied with a function, *the wavelet*, similar to the window function in the STFT, and the transform is computed separately for different segments of the time-domain signal. However, the main differences between the STFT and the CWT is that the width of the window is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform.

The continuous wavelet transform is defined as follows:

$$W(s,\tau) = \int f(t)\psi^{*}_{s,\tau}(t)dt \qquad (2\text{-}4)$$

where τ and s are respectively the translation and the scale ($s>1$ corresponds to a dilation and $s<1$ to a contraction of $\psi(t)$, and * denotes complex conjugation. This equation shows how a function $f(t)$ is decomposed into a set of basis functions $\Psi_{s,\tau}(t)$, called the wavelets.

The inverse wavelet transform is:

$$f(t) = \int\int W(s,\tau)\Psi_{s,\tau}(t)d\tau ds \qquad (2\text{-}5)$$

The wavelets are generated from a single basic wavelet $\psi(t)$ (*the mother wavelet*), by scaling and translation:

$$\psi_{s,\tau} = \frac{1}{\sqrt{s}}\psi(-\frac{t-\tau}{s}) \qquad (2\text{-}6)$$

The mother wavelet $\psi(x)$ must satisfy specific properties in order to be an "admissible" wavelet and to produce an inversible transform.

2.3.2 The Discrete Wavelet Transform

In the continuous wavelet transform, let's consider the family:

$$\psi_{s,\tau} = \frac{1}{\sqrt{s}}\psi(-\frac{t-\tau}{s}) \tag{2-7}$$

where $b \in R, a \in R_+$ with $a \neq 0$, and ψ is admissible.

First we restrict τ,s to discrete values only. The discretization of the dilation parameter is choosen: $s = s_0^m, where$ m∈Z and $s_0 > 1$. For m=0, we discretize τ by taking only the integer (positive and negative) multiples of one fixed τ_0, where τ_0 is appropriately chosen so that the $\psi(t - n\tau_0)$ "cover" the whole line. For different values of m, the width of $s_0^{-m/2}\psi(s_0^{-m}t)$ is s_0^m times in width of $\psi(t)$ (as measured, e.g., by width

$$(f) = \left[\int t^2 |f(t)|^2 dx\right]^{1/2},$$

where we assume that

$$(f) = \left[\int t |f(t)| dx\right]^2 = 0),$$

so that the choice $\tau = n\tau_0 a_0^m$, where m,n range over Z, and $s_0>1$, $\tau_0>0$ are fixed. The appropriate choices for s_0, τ_0 depend, of course, on the wavelet ψ. This corresponds to:

$$\psi_{m,n}(t) = s_0^{-m/2}\psi\left(\frac{t - n\tau_0 s_0^m}{s_0^m}\right) = s_0^{-m/2}\psi\left(s_0^{-m}t - n\tau_0\right) \tag{2-8}$$

2.3.3 A Multiresolution Formulation of Wavelet Systems

Multiresolution analysis (MRA) was formulated based on the study of wavelet bases. Wavelet theory and their applications are rapidly developing fields in applied mathematics and signal analysis. Both theoretically and practically, wavelet basis representations of certain signals show advantages over the traditional Fourier basis representation. The MRA concept was initiated by Mallat **[Ma89]** and provides a natural framework for the understanding of wavelet bases. The central idea of the MRA is to decompose a function into a single low-resolution part and a sequence of detail information components at decreasing orders.

2.3.3.1 Definitions

A *function space* is a linear vector space (finite or infinite dimensional) where the vectors are functions and the scalars are real numbers (sometime complex numbers).

$L^2(\mathbf{R})$ (the space of square integrable functions), is defined as the space of Lebesque measurable functions for which:

$$\|f\|^2 = \int_{-\infty}^{+\infty} |f(x)|^2 < \infty$$

||f|| defines a *norm* or "length" of a vector.

Function g(t) is denoted as a member of that space by writing: $g \in L^2(\mathbf{R})$ or simply $g \in L^2$. If we have a signal vector space S, then if any $f(t) \in S$ can be expressed as $f(t) = \sum_k a_k \varphi_k(t)$, the set of functions $\varphi_k(t)$ is called an *expansion set* for the space S. If the representation is unique, the set is a *basis*. Alternatively, one could start with the expansion set or basis set and define the space S as the set of all functions that can be expressed by $f(t) = \sum_k a_k \varphi_k(t)$. This is called the *span* of the basis set.

2.3.3.2 The Scaling Function

In order to use the idea of multiresolution, we start by defining the scaling function and define the wavelet in terms of it. We define a set of scaling functions in terms of integer translations of the basic scaling function by

$$\varphi_k(t) = \varphi_k(t-k),\ k \in Z \quad \varphi \in L^2$$

The subspace of $L^2(\mathbf{R})$ spanned by these functions is defined as

$$\upsilon_0 = \overline{\underset{k}{Span}\{\varphi_k(t)\}}$$

for all integers k from minus infinity to infinity. The over-bar denotes closure. This means that

$$f(t) = \sum_k a_k \varphi_k(t) \qquad \text{for any } f(t) \in v_0.$$

One can generally increase the size of the subspace spanned by changing the time scale of the scaling functions. A two-dimensional family of functions is generated from the basic scaling function by scaling and translation as follows

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k)$$

whose span over k is

$$\upsilon_j = \overline{\underset{k}{Span}\{\varphi_k(2^j t)\}} = \overline{\underset{k}{Span}\{\varphi_{j,k}(t)\}}$$

for all integers k $\in$ **Z**. This means that if f(t) $\in$ v_j, then it can be expressed as

$$f(t) = \sum_k a_k \varphi(2^j t + k)$$

For j>0, the span can be larger since $\varphi_{j,k}(t)$ is narrower and is translated in smaller steps. It, therefore, can represent finer detail. For j<0, $\varphi_{j,k}(t)$ is wider and is translated in larger steps. So these wider scaling functions can represent only coarse information, and the space they span is smaller.

2.3.3.3 Multiresolution Analysis

More formally, an MRA of L^2 is an increasing sequence of closed subspaces V_j of L^2 with the following properties:

1) $\ldots \subset v_j \subset v_{j+1} \subset \ldots \subset L^2$ for all $j \in Z$ **(2-9)**

2) $\bigcap_{j=-\infty}^{\infty} v_j = \{0\}$

The space that contains high-resolution signals contains those of lower resolution also. Because of the definition of v_j, the spaces have to satisfy a natural scaling condition

$$f(t) \in \nu_j \qquad \Leftrightarrow \qquad f(2t) \in \nu_j$$

which ensures elements in a space are simply scaled versions of the elements in the next space.

The nesting of the spans of $\varphi(2^j t - k)$, denoted by ν_j is achieved by requiring that $\varphi(t) \in \nu_1$, which means that if $\varphi(t)$ is in ν_0, it is also in ν_1, the space spanned by $\varphi(2t)$. This means $\varphi(t)$ can be expressed in terms of a weighted sum of shifted $\varphi(2t)$ as

$$\varphi(t) = \sum_n h(n)\sqrt{2}\varphi(2t - n), \qquad n \in \mathbf{Z}.$$

where the coefficients h(n) are a sequence of real or, perhaps, complex numbers called the scaling function coefficients (or the scaling filter) and the $\sqrt{2}$ maintains the norm of the scaling function with the scale of two.

2.3.3.4 The Wavelet Functions

The important features of a signal can better be described or parameterized, not by using $\varphi_{j,k}(t)$ and increasing j to increase the size of the subspace spanned by the scaling functions, but by defining a slightly different set of functions $\psi_{j,k}(t)$ that span the differences between the spaces spanned by the various scales of the scaling function.

There are several advantages to requiring that the scaling functions and wavelets be orthogonal. Orthogonal basis functions allow simple calculation of expansion coefficients. The orthogonal complement of ν_j in ν_{j+1} is defined as W_j. This means that all members of ν_j are orthogonal to all members of W_j. We require

$$<\varphi_{j,k}(t), \psi_{j,l}(t)> = \int \varphi_{j,k}(t)\psi_{j,l}(t)dt = 0 \qquad \textbf{(2-10)}$$

for all appropriate j,k,l $\in \mathbf{Z}$.

The concept of the MRA allows detail information to be kept in the complement space W_j of ν_j, such that:

$$\nu_{j+1} = \nu_j \oplus W_j$$

Since these wavelets reside in the space spanned by the next narrower scaling function, $W_0 \subset v_1$, they can be represented by a weighted sum of shifted forms of the scaling function φ(2t) by

$$\psi(t) = \sum_n h_1(n)\sqrt{2}\varphi(2t-n), \qquad n \in Z. \qquad (2\text{-}11)$$

for some set of coefficients $h_1(n)$. From the requirement that the wavelets span the "difference" or orthogonal complement spaces, and from the orthogonality of integer translates of the wavelet (or scaling function), it is shown that the wavelet coefficients are required to be related to the scaling function coefficients by:

$$h_1(n) = (-1)^n h(1-n) \qquad (2\text{-}12)$$

The function generated by (2-11) gives the prototype or mother wavelet $\psi(t)$ for a class of expansion functions of the form:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \qquad (2\text{-}13)$$

where 2^j is the scaling of t (j is the $\log_2$ of the scale), $2^{-j}k$ is the translation in t, and $2^{j/2}$ maintains the L^2 norm of the wavelet at different scales.

The introduction of the hierarchical set of basis functions allows a hierarchical representation of the function f(t):

$$f(t) = \sum_{k=-\infty}^{\infty} c(k)\varphi_k(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \qquad (2\text{-}14)$$

as a series expansion in terms of the scaling function and wavelets.

In this expansion, the first summation in (2-14) gives a function that is a coarse approximation to f(t). For each increasing index j in the second summation, a higher or finer resolution function is added, which adds increasing detail.

In general, this hierarchical representation allows an economical representation of functions. If a function has little or no detail in some region, then the coefficients that code the detail information will be zero or negligibly small in this region.

The coefficients in this wavelet expansion are called the *discrete wavelet transform* (DWT) of the signal f(t).

2.3.4 Haar Basis

The Haar scaling function is the simple unit-width, unit-height pulse function φ(t), and it is obvious that φ(2t) can be used to construct φ(t) by:

$$\varphi(t) = \varphi(2t) + \varphi(2t-1) \qquad \text{(2-15)}$$

which means (2-11) is satisfied for coefficients $h(0) = 1/\sqrt{2}, h(1) = 1/\sqrt{2}$.

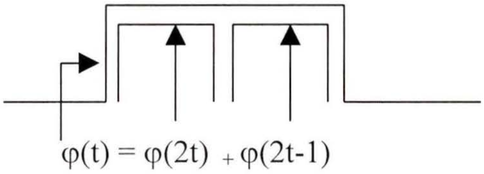


Figure 2-1 Haar scaling function

The Haar triangle wavelets that are associated with the scaling function in Figure 2-1 are shown in Figure 2-2:

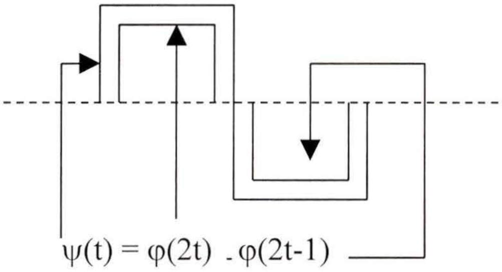


Figure 2-2 Haar wavelet.

2.3.5 Filter Conditions

Ingrid Daubechies **[Da92]** formulated some general conditions suitable families of basis functions to satisfy for exact reconstruction.

$$\sum_k h_k = \sqrt{2}$$

$$g_j = (-1)^j h_{1-j}$$

$$\sum_k g_k = 0$$

$$\forall m : \sum_k h_k h_{k+2m} = \delta_{0,m} \tag{2-16}$$

This class of wavelet functions is constrained, by definition, to be zero outside of a small interval. This is what makes the wavelet transform able to operate on a finite set of data, a property that is formally called *"compact support"*.

Another condition derived by Daubechies that forces the filter to be finite is:

$$\sum_k (-1)^k k^j g_k = 0 \qquad j=0,1,\ldots n-1 \tag{2-17}$$

Most wavelet functions, when plotted, appear to be extremely irregular. The functions, which are normally used for performing transforms, consist of a few sets of well chosen coefficients resulting in a function that has a discernible shape. If in **(2-17)** we choose n= 1, we get the coefficients for the Haar Transform.

2.3.5.1 The Pyramid Algorithm

The pyramid algorithm operates on a finite set of N input data, where N is a power of two; this value will be referred to as the input block size. These data are passed through two convolution functions, each of which creates an output stream that is half the length of the original input. These convolution functions are filters; one half of the output is produced by the "low-pass" filter function:

$$a_i = \frac{1}{2} \sum_{j=1}^{N} c_{2i-j} f_j, \qquad i=1, \ldots, N/2 \tag{2-18}$$

and the other half is produced by the "high-pass" filter function, related to equation (2-18):

$$b_i = \frac{1}{2}\sum_{j=1}^{N}(-1)^{j+1} c_{j+1-2i} f_j \qquad \text{i=1, ..., N/2} \tag{2-19}$$

where N is the input block size, c are the coefficients, f is the input function, and a and b are the output functions. (the low- and high-pass outputs are sometimes referred to as the odd and even outputs, respectively.) In many situations, the odd, or low-pass output contains most of the "information content" of the original input signal. The even, or high-pass output contains the difference between the true input and the value of the reconstructed input if it were to be reconstructed from only the information given in the odd output. In general, higher-order wavelets (i.e., those with more non-zero coefficients) tend to put more information into the odd output, and less into the even output. If the average amplitude of the even output is low enough, then the even half of the signal may be discarded without greatly affecting the quality of the reconstructed signal. An important step in wavelet-based data compression is finding wavelet functions which cause the even terms to be nearly zero.

If the signal is reconstructed by an inverse low-pass filter of the form:

$$f_j^L = \sum_{i=1}^{\frac{N}{2}} c_{2i-j+1} a_i, \qquad \text{j=1, …, N}$$

then the result is a duplication of each entry from the low-pass filter output.

$$f_j^H = \sum_{i=1}^{\frac{N}{2}} (-1)^{j+1} c_{j-2i} b_i, \qquad \text{j=1, …, N}$$

The perfectly reconstructed signal is

$$F = f^L + f^H$$

2.3.6 Properties of Wavelets

We list some important properties of wavelets.

Orthogonality: Orthogonality is convenient to have in many situations , e.g. it directly links the L2 norm of a function to the norm of its wavelet coefficients by

$||f|| = \text{sqrt}(\sum c_{ij})$.

Compact support: If the scaling function and wavelet are compactly supported, the filters h and g are finite impulse response filters, so that the summations in the fast wavelet transform are finite. This obviously is of use in implementations. If they are not compactly supported, a fast decay is desirable so that the filters can be approximated reasonably by finite impulse response filters.

Rational coefficients: For computer implementations it is of use if the filter coefficients h_k and g_k are rationals or, even better, dyadic (power of two) rationals. Multiplication by a power of two on a computer corresponds to shifting bits, which is a very fast operation.

Symmetry: If the scaling function and wavelet are symmetric, then the filters have generalized linear phase. The absence of this property can lead to phase distortion. This is important in signal processing applications.

Smoothness: The smoothness of wavelets plays an important role in compression applications. Compression is usually achieved by setting small coefficients to zero.

If the original function represents an image and the wavelet is not smooth, the error can easily be detected visually.

Vanishing moments: The advantageous effect of vanishing moments is their property to concentrate the signal information in a relatively small number of coefficients. This feature is useful in particular for compression. For example, we have two vanishing moments for Haar wavelet and three vanishing moments for Daubechie 4 wavelet.

Analytic expressions: An analytic expression for a scaling function or wavelet does not always exists but in some cases it is available and nice to have.

It is **not** possible to construct wavelets that have all the properties enumerated in 2.3.6. We have a classic trade-off situation.

2.4 Conclusion

In this chapter we introduce the topic of wavelets by studying the theory of time-frequency transforms. We discuss the decomposition of $L^2(R)$ using a multiresolution theory. Section 2.3.5 concerns itself with the construction of wavelets. Daubechies' filter conditions are enumerated and the simplest orthogonal wavelets (Haar functions) are presented.

Because of their properties, wavelet techniques have had a particularly significant impact on data compression. In Chapter 3 some of the compression schemes that use wavelets are presented.

3 WAVELET IMAGE COMPRESSION

3.1 Introduction To Image Compression

There are two general approaches to image compression. In the first approach, the sample values of the original image are used directly to generate the compressed bitstream. Such an approach is often referred to as a spatial-domain approach. In the second method, a transformation is applied to the samples of the original image and then the transform data are used to produce the compressed bitstream.

The compression of images with wavelets is an example of transform coding (others are for example the DCT-based JPEG, or pyramid decomposition schemes). The set of pixels making up the image is transformed into a set of transform coefficients. The goal is to find a representation of the image which eases the compression task, as when for example the coefficients have better statistical properties than the original pixels.

Usually the transform itself is lossless (it allows perfect reconstruction), and thus it could be used for lossless compression. But the common real-valued wavelets are not well suited for lossless applications, since a naïve implementation inflates the amount of data needed to describe a picture. Quantization is the fundamental step of lossy compression. It means that only some of the coefficients are actually used in reconstructing the image (the others are set to zero). And the retained coefficients are kept in reduced precision. Obviously the reconstructed image will contain errors (distortion), but this can be traded off with the compression ratio, by varying the number of coefficients and their precision. In an actual image compression scheme, the quantization stage will be followed by entropy coding, which is just some “classic" lossless method, such as arithmetic coding.

3.2 Wavelet Still Image Compression

A wide variety of wavelet-based image compression schemes have been reported in the literature, ranging from simple entropy coding to more complex techniques such as vector quantization, adaptive transforms, tree encodings, and edge-based coding.

Compression is accomplished by applying a wavelet transform to decorrelate the image data, quantizing the resulting transform coefficients, and coding the quantized values. Image reconstruction is accomplished by inverting the compression operations.

3.2.1 Choice of Wavelet Basis

One of the most commonly used approaches for analyzing a signal f(x) is to represent it as a weighted sum of simple building blocks, called *basis functions*:

$$f(x) = \sum c_i \varphi_i(x),$$

where the $\varphi_i(x)$ are basis functions and the c_i are the coefficients, or weights. Since the basis functions φ are fixed, it is the coefficients which contain the information about the signal.

The simplest such representation uses translates of the impulse function as its only bases, yielding a representation that reveals information only about the time domain behavior of the signal. Choosing sinusoids as the basis functions yields a Fourier representation that reveals information only about the signal's frequency domain behavior. For the purposes of signal compression, neither of the above representations is ideal. What we would like to have is a representation that contains information about both the time and frequency behavior of the signal. More specifically, we want to know the frequency content of the signal at a particular instant in time. However, resolution in time (Δx) and resolution in frequency ($\Delta\omega$) cannot both be made arbitrarily small at the same time because their product is lower bounded by the Heisenberg inequality:

$$\Delta x * \Delta\omega >= 1/2$$

This inequality means that we must trade off time resolution for frequency resolution, or vice versa. Thus, it is possible to get very good resolution in time if you are willing to settle for low resolution in frequency, and you can get very good resolution in frequency if you are willing to settle for low resolution in time.

Deciding on the optimal wavelet basis to use for image coding is a difficult problem. A number of design criteria, including smoothness, accuracy of approximation, size of support, and filter frequency selectivity are known to be important.

3.2.2 Boundaries

Careful handling of image boundaries when performing the wavelet transform is essential for effective compression algorithms. For symmetrical wavelets an effective strategy for handling boundaries is to extend the image via reflection. Such an extension preserves continuity at the boundaries and usually leads to much smaller wavelet coefficients than if discontinuities were present at the boundaries.

3.2.3 Basic Compression Scheme

A very basic image compression system would contain the following modules:

- the wavelet transform module,
- the quantization module,
- the entropy coding;

3.2.3.1 Wavelet Transform Module

Each step of the transform for such bases involves two frequency splits instead of one. Suppose we have an N × N image. First each of the N rows in the image is split into a low-pass half and a high-pass half. The result is an N × N / 2 sub-image and an N × N / 2 high-pass sub-image. Next each column of the sub-images is split into a low-pass and a high-pass half. The result (Figure 3-1) is a four-way partition of the image into horizontal low-pass/vertical low-pass, horizontal high-pass/vertical low-pass, horizontal low-pass/vertical high-pass, and horizontal high-pass/vertical high-pass sub-images.

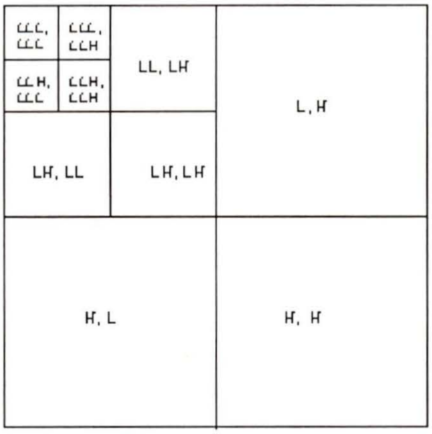


Figure 3-1 Three level wavelet transform

3.2.3.2 Wavelet Coefficient Quantization

Mallat **[Ma89]** has found experimentally that the histograms of the wavelet coefficients (in the transformed domain) can be modeled with the following family of histograms:

$$hist(u) = Ke^{-(|u|/\alpha)\beta} \tag{3-1}$$

where β modifies the decreasing rate of the peak and α models the variance. K is adjusted so that

$$\int_{-\infty}^{+\infty} hist(u)du = N \tag{3-2}$$

where N is the total number of pixels in the detail image.

The histograms of the detail subimages present only one very large peak value at the origin. This means that many wavelet coefficients are zero and, further, that most of the wavelet coefficients are nearly zero. This observation allow us to derive some interesting rules to quantize the wavelet coefficients, which should take into account the following facts:

- at a given resolution, in a given position on a detail subimage corresponding to that resolution, the pixel “value" represents the amplitude of the wavelet function scaled at that resolution and shifted in that position;
- the wavelets are designed to perform local signal description, and are therefore characterized by a very fast decay .

From the above remarks, it follows that the influence domain of a wavelet coefficient is very limited. A small error on a wavelet coefficient will have a negligible influence not only on the pixel itself where it is evaluated, but also on the surrounding pixels thanks to the limited support and to the fast decay of the wavelet function.

The forward wavelet transform decorrelates the pixel values of the original image and concentrates the image information into a relatively small number of coefficients.

We can also take advantage of the energy invariance property of the wavelet transform to achieve high-quality lossy compression. The energy invariance property says that the total amount of energy in an image does not change when the wavelet transform is applied. This property can also be viewed in a slightly different way: any changes made to the values of the wavelet coefficients will result in proportional changes in the pixel values of the reconstructed image. In other words, we can eliminate (set to zero) those coefficients with small magnitudes without creating significant distortion in the reconstructed image. In practice, it is possible to eliminate all but a few percent of the wavelet coefficients and still get a reconstructed image of reasonable quality. The elimination of small valued coefficients can be accomplished by applying a thresholding function

$$T(\tau, x) = \begin{cases} 0, if|x| < \tau \\ x, otherwise \end{cases} \qquad (3\text{-}3)$$

to the coefficient matrix. The amount of compression obtained can now be controlled by varying the threshold parameter τ.

Higher compression ratios can be obtained by quantizing the non-zero wavelet coefficients before they are encoded.

A quantizer is a many-to-one function Q(x) that maps many input values into a (usually much) smaller set of output values. Quantizers are staircase functions characterized by a set of numbers $\{d_i; i = 0, .., N\}$ called *decision points* and a set of numbers $\{r_i; i = 0, ... ,N-1\}$ called reconstruction levels. An input value x is mapped to a reconstruction level r_i if x lies in the interval $(d_i, d_{i+1}]$.

To achieve the best results, a separate quantizer should be designed for each scale, taking into account both the properties of the Human Visual System and the statistical properties of the scale's coefficients. The characteristics of the Human Visual System guide the allocation of bits among the different scales, and the coefficient statistics guide the quantizer design for each scale.

The design of scalar quantizers also depends on the type of encoder to be used.

3.2.3.3 Entropy Coding

a) Arithmetic coding provides a near-optimal entropy coding for the quantized coefficient values. The coder requires an estimate of the distribution of quantized coefficients

b) The non-zero coefficients can be coded using Huffman codes. The coefficients contain a large number of zero elements and are therefore easily encoded using run length coding combined with Huffman codes.

3.3 Examples of Wavelet Transform-Based Image Compression Systems

Recently, several embedded wavelet coders, e.g. the embedded zerotree wavelet (EZW), the layer zero coder (LZC), the modified LZC, MTWC and the method of set partitioning in hierarchical tree (SPIHT), have been proposed, where the global wavelet transform instead of block DCT is adopted.

The first advantage of embedded wavelet coders is that they give a much better rate-distortion performance than standard JPEG and the blocking artifact is reduced significantly.

The second advantage is their progressive coding property due to the use of successive approximation quantization (SAQ) and bit layer coding.

The successive approximation quantization (SAQ) scheme adopted by all embedded wavelet coders is crucial to the design of embedded coders.

The entropy coding stage follows the successive approximation quantization stage. Since there are correlations between insignificant bits, several methods are developed to predict or to group these insignificant bits for effective lossless entropy coding. EZW uses the parent-children zerotree relationship to predict insignificant bits in the descendant while the parent bit is insignificant. More tree structures of bits are identified in SPIHT, for effective coding. Instead of identifying special classes of bit structures, modified LZC uses a context adaptive QM coder to estimate the conditional probability for a given pixel based on the bit pattern of its neighborhood and parent position and encode the content of bit streams.

Since LZC and modified LZC do not require the determination of the zerotree relationship between the parent and children, their computational speed is faster. Moreover, LZC and modified LZC have a better performance than EZW.

3.3.1 EZW Coding Scheme

The idea of embedded coding was first popularized by Shapiro in his well-known paper on EZW coding **[Shap93]** .

EZW technique is based on three concepts:

(1) partial ordering of the transformed image elements by magnitude, with transmission of order by a subset partitioning algorithm that is duplicated at the decoder,

(2) ordered bit plane transmission of refinement bits, and

(3) exploitation of the self similarity of the image wavelet transform across different scales.

EZW introduced a data structure called a zerotree, built on the parent-child relationship. The zerotree structure takes the advantage of the principle that if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then all wavelet coefficients of the same orientation at the same spatial location at finer wavelet scales are also likely to be insignificant with respect to that T. **Figure 3-3** shows a wavelet tree descending from a coefficient in the subband LL3 . The EZW scheme encodes all wavelet coefficients in order of importance with respect to a number of decreasing thresholds. Scanning order of the coefficients is shown in Figure 3-2 (**[Shap93]**). The initial threshold is usually set as half of the largest wavelet coefficients of a picture, and the threshold is halved every time the coding procedure progresses to the next threshold. For every threshold, two passes are performed: a dominant pass and a subordinate pass.

There are two types of passes implemented: a *dominant pass* and a *subordinate pass*. The dominant pass finds pixel values above a certain threshold and creates a binary decision tree called a *significance map* and the subordinate pass quantizes all significant pixel values found in this and all previous dominant passes previous.

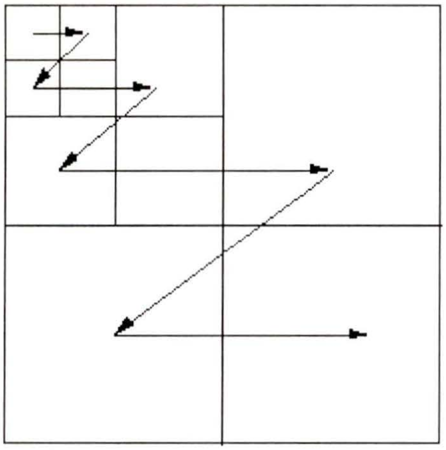


Figure 3-2 Scanning order of subbands for encoding

The first step in the EZW algorithm is to apply a discrete wavelet transform to the original image. This provides a pyramid representation of the image.

A *dominant pass* checks all trees for significant pixel values with respect to a certain threshold. The initial threshold is chosen to be one-half of the maximum magnitude of all pixel values. Subsequent dominant pass thresholds are always one-half the previous pass threshold. When an insignificant pixel value is found, and a check of all its children reveals that they too are insignificant, then it is possible to encode that pixel and all its children with one symbol, a zero tree root (ZTR), in place of a symbol for that pixel and a symbol for each of that pixel's children, thus achieving compression. Pixel values found to be significant in the dominant pass are encoded with the symbol positive (POS), for a value greater than zero, or negative (NEG) for a value less than zero, then those pixel values are added to a subordinate list for quantization, and the pixel value in the subband is then set to zero for the next dominant pass. Pixel values found to be insignificant in the dominant pass but with significant children are coded as isolated zeros(IZ).

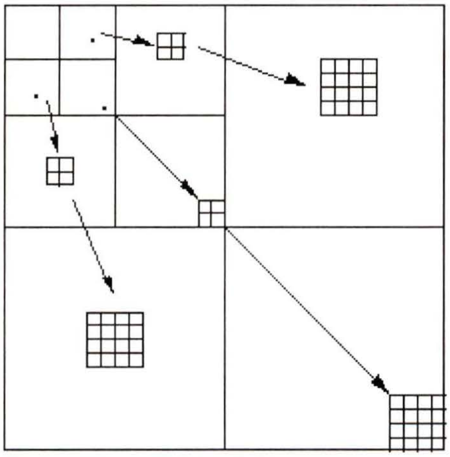


Figure 3-3 Zerotree structure

A *subordinate pass* is then performed on the subordinate list which contains all pixel values previously found to be significant. The subordinate pass performs pixel value quantization which achieves compression by telling the decoder with a symbol roughly what the pixel value is instead of its exact magnitude. Since the initial threshold is one-half the maximum magnitude of all pixel values for the first dominant pass, then in the first subordinate pass only two ranges are specified in which a significant pixel value could lie: the upper half of the range between the maximum pixel value and the initial threshold, or the lower half of the same range. A pixel value in the upper half of the range gets coded with the symbol upper (for upper part of the range), while a pixel value in the lower half gets coded with the symbol lower. A pixel value found to be in a particular range is quantized, from the decoder's viewpoint, to the midpoint of that range. So, the subordinate passes quantize pixel values to a two symbol alphabet which then get encoded by using an adaptive arithmetic coder.

Since its inception in 1993, many enhancements have been made to make the EZW algorithm more robust and efficient. One very popular and improved variation of the EZW is the SPIHT algorithm.

3.3.2 Set Partitioning in Hierarchical Trees (SPIHT) Algorithm

Said and Pearlman **[PeSa96]** , offered an alternative explanation of the principles of operation of the EZW algorithm to better understand the reasons for its excellent performance. According to them, the three key concepts in EZW are partial ordering by magnitude of the transformed coefficients with a set partitioning sorting algorithm, ordered bitplane transmission of refinement bits, and exploitation of self-similarity of the image wavelet transform across different scales of an image. In addition, they offer a new and more effective implementation of the modified EZW algorithm based on set partitioning in hierarchical trees, and call it the SPIHT algorithm. They also present a scheme for progressive transmission of the coefficient values that incorporates the concepts of ordering the coefficients by magnitude and transmitting the most significant bits first. They use a uniform scalar quantizer and claim that the ordering information made this simple quantization method more efficient than expected. An efficient way to code the ordering information is also proposed. According to them, results from the SPIHT coding algorithm in most cases surpass those obtained from EZW algorithm.

3.3.3 MTWC (Multi-Threshold Wavelet Codec)

Another still image compression algorithm based on the multi-threshold wavelet coding (MTWC) technique is proposed in **[HoWa98]** . It is an embedded wavelet coder in the sense that its compression ratio can be controlled depending on the bandwidth requirement of image transmission. At low bit rates, MTWC can avoid the blocking artifacts of JPEG, resulting in a better reconstructed image quality. A subband decision scheme is developed based on rate-distortion theory to enhance the image fidelity. Moreover, a new quantization sequence order (SAQ-successive approximation quantization) is introduced based on analysis of error energy reduction in significant and refinement maps.

3.3.3.1 Coder Description

MTWC adopts different initial threshold values for different subbands. The initial threshold for each subband is equal to the largest absolute value of all wavelet coefficients in that subband. Instead of encoding all wavelet coefficients of all subbands at the same time, the algorithm searches for the subband with the largest threshold value and encodes the bit layer obtained via SAQ at the subband. After the bit layer coding, the threshold is halved.

The maximum magnitude of coefficients in each subband is used as the initial quantization threshold for coefficients in that subband. Once coefficients in the subband are quantized, the corresponding threshold value is halved. The subband decision scheme always chooses the subband with the maximum threshold value for quantization.

Bits are classified as significant bits and refinement bits. The transmission of k^{th} refinement map of subband *i* is followed immediately by the transmission of (k+1)th significant map of subband *i*.

Any efficient binary entropy coder can be applied, e.g. JBIG, binary Arithmetic coder and even the JBIG-2.

3.4 Conclusion

Wavelet techniques have had a particularly significant impact on data compression. In this chapter several image compression algorithms have been presented (embedded zerotree wavelet (EZW), the layer zero coder (LZC), the modified LZC, MTWC and the method of set partitioning in hierarchical trees (SPIHT)).

Some of the watermarking schemes are based in part on the compression algorithms seen in this chapter. For example the wavelet watermarking scheme to be covered in Section 4.7 is based on Zerotrees. Also some of the steps seen in image compression are common with steps in image watermarking and because the final goal is to track watermarks in both compressed and uncompressed images .

In the next chapter, we introduce the image watermarking concept and also present some of the watermarking algorithms.

4 WATERMARKS FOR DIGITAL IMAGES

4.1 Introduction

Multimedia applications that operate on audio, images and video data will enable new applications for computers. The recent growth of networked multimedia has caused problems relative to the protection of intellectual property rights. This is particularly true for image and video data.

The types of possible protection systems involve the use of both encryption (used to scramble images so that an adversary can not obtain the original image without knowing the secret key) and watermarking (used for owner identification). In this chapter, we explore watermarks, their desirable properties and the general process of watermarking. It is also important for us to consider the possibility of attacks upon the watermarks, as well as any possible degradation of the image that they may cause. In section 7, we look at several watermarking schemes based on wavelets.

4.2 Definition of a Watermark

A ***watermark*** is an invisible mark placed on an image that later can be detected and used for purposes such as:

- owner identification,
- identifying the content of the image,
- authentication,
- determining if data has been altered

Thus, the goal is for the watermark to **always** remain present in the data. Embedded signaling or watermarking could be used for a variety of other purposes other than copyright control.

4.3 Desired Properties of watermarks

There are a number of desirable characteristics that a watermark should exhibit depending on the use. These include properties such as being difficult to notice, being robust to common distortions of the signal, having resistance to malicious attempts to remove the watermark, supporting a sufficient rate commensurate with the application, allowing multiple watermarks to be added and the scalability of the decoder. These characteristics are discussed in more detail below.

Difficult to notice The watermark should not be noticeable to the viewer nor should the watermark degrade the quality of the content. In earlier work, the term "imperceptible" was used, and this is certainly the ideal. Current state-of-the-art compression algorithms still leave room for an imperceptible signal to be inserted.

Robustness Music, images and video signals may undergo many types of distortions. Lossy compression has already been mentioned, but many other signal transformations are also common. For example, an image might be contrast enhanced and colors might be altered somewhat, or an audio signal might have its bass frequencies amplified. In general, a watermark must be robust to transformations that include common signal distortions as well as digital-to-analog and analog-to-digital conversion and lossy compression. It has been argued that robustness can only be attained if the watermark is placed in perceptually significant regions of an image. This is because image fidelity is only preserved if the perceptually significant regions of the image remain intact. Conversely, perceptually insignificant regions can be removed without affecting the image quality. Consequently, watermarks that are placed in perceptually insignificant regions will not be robust and can be easily removed.

Tamper-resistance. As well as requiring the watermark to be robust to legitimate signal distortions, a watermark may also be subject to signal processing that is solely intended to remove the watermark. This may be the case when watermarking is used for copyright.

Bit rate. The bit rate of a watermark refers to the amount of information a watermark can encode in a signal.

4.4 The General Process

The process of watermarking an image can be represented by the addition of a noise term that is a function of the watermark signal, w, and possibly of the original image, I. The watermarked image, I' is then given by:

$$I' = I + f(I,w)$$

The watermarked image may then be subject to any number of distortions due to tampering or common use:

$$I'' = I' + n = I + f(I,w) + n(I)$$

At the decoder, we wish to extract the watermark signal. It should be noted that the magnitude of I needs to be very much larger than the inserted watermark, $f(I, w)$, and the distortions, n, otherwise the image fidelity would not be preserved.

4.5 Attacks on Watermarking

4.5.1 Counterfeit Attacks

Ramkumar **[MaRa99]** classified watermark detectors into two categories. Some detectors need the original image I to check for the presence of the signature S in I'(watermarked image). Such schemes, are called cover ***image escrow schemes***.

On the other hand, schemes that do not require the original image for detection of the signature are called ***oblivious detection schemes***.

The watermark detection functions could be written as following:

$W = f(I', S, I)$ in the case of image escrow schemes

$W = f(I', S)$ in the case of oblivious detection schemes

The detection statistic is an indication of the degree of certainty with which the signature S is detected in the image I'.

Ramakumar imagined some scenarios of counterfeit attacks on watermarking. The main conclusion was that watermarks are not yet mature enough to be used in court as a proof of ownership.

4.5.2 Direct Attacks On Watermarks

Kutter **[Ku99]** proposes a list of attacks against which watermarking system could be judged. They do not make a difference between intentional and unintentional processing.

- JPEG compression - JPEG is currently one of the most widely used compression algorithms for images and any marking system should be resilient to some degree of compression with JPEG.
- Geometric transformations
  - Horizontal flip, rotation, cropping and scaling. Many images can be flipped, rotated, cropped and/or scaled, very few watermarking systems could survive all these geometric transformations.
- Enhancement techniques
  - Low pass filtering (this includes linear and non-linear filters like median, Gaussian, and standard average filters), sharpening, gamma correction (a very frequently used operation to enhance images or adapt images for display, for example after scanning) and color quantization.

- Noise addition
  - Printing-scanning (this process introduces geometrical as well as noise-like distortions),
  - Statistical averaging and collusion (given two or more copies of the same image but with different watermarks, it should not be possible to remove the marks by averaging these images or by taking small parts of all images and reassembling them).

4.6 Visual Quality Metrics

The watermark robustness depends directly on the embedding strength, which in turn influences the visual degradation of the image. For fair benchmarking and performance evaluation, visual degradation due to the embedding is an important, and unfortunately often neglected, issue.

4.6.1 Pixel Based Metrics

Signal-to-noise (SNR) measures are estimates of the quality of a reconstructed image compared with an original image. The basic idea is to compute a single number that reflects the quality of the reconstructed image. Reconstructed images with higher metrics are judged better. However, traditional SNR measures do not equate well with human subjective perception.

The actual metric we will compute is the peak signal-to-reconstructed image measure which is called PSNR. Assume we are given a source image f(i,j) that contains N by N pixels and a reconstructed image F(i,j) where F is reconstructed by decoding the encoded version of f(i,j). Error metrics are computed on the luminance signal only, so the pixel values f(i,j) range between black (0) and white (255).

First, the mean squared error (MSE) of the reconstructed image is computed as follows

$$MSE = \frac{\sum_{i}\sum_{j}[f(i,j) - F(i,j)]^2}{N^2} \tag{4-1}$$

The summation is over all pixels. The root mean squared error (RMSE) is the square root of MSE.

PSNR in decibels (dB) is computed by using

$$PSNR = 20\log_{10}(\frac{255}{RMSE}) \tag{4-2}$$

Typical PSNR values range between 20 and 40. They are usually reported to two decimal places (e.g. 21.34). The actual value is not meaningful, but the comparison between two values for different reconstructed images gives a relative measure of quality. The MPEG committee used an informal threshold of 0.5 dB PSNR to decide whether to incorporate a coding optimization because they believed that an improvement of that magnitude would be visible.

Some definitions of PSNR use 255^2/MSE rather than 255/RMSE. Either formulation will work because we are interested in the relative comparison, not the absolute values.

The other important technique for displaying errors is to construct an error image which shows the pixel-by-pixel errors. The simplest computation of this image is to create an image by taking the difference between the reconstructed and original pixels. These images are hard to see because zero difference is black and most errors are small numbers which are shades of black. The typical construction of the error image multiplies the difference by a constant to increase the visible difference and translates the entire image to a gray level. The computation is

$$E(i,j) = 2[f(i,j) - F(i,j)] + 128 \tag{4-3}$$

You can adjust the constant (2) or the translation (128) to change the image. Some people use white (255) to signify no error and difference from white as an error which means that darker pixels are bigger errors.

4.6.2 Perceptual Quality Metrics

The weakness of the pixel-based distortion metrics has been known for a long time. In recent years more and more research has been concentrated on distortion metrics adapted to the human visual system by taking various effect into account. In **[Ku99]**, the perceptual quality measure exploits the contrast sensitivity and masking phenomena of the HVS (Human Visual System) and is based on a multi-channel model of the human spatial vision.

Computing the metric involves the following steps: coarse image segmentation, decomposition of the coding error and the original image into perceptual components using filter banks, computing the detection threshold for each pixel using the original image as masker, dividing the filtered error by the decision threshold, pooling over all color channels. The units for the metric is given in *units above threshold* also referred to as Just Noticeable Difference (JND). The overall metric, Masked Peak Signal to Noise Ratio (MPSNR) is then given by:

$$MPSNR = 10 \log_{10} 255^2 / E^2$$

where E is the computed distortion. Since this quality metric does not have exactly the same meaning as the usual dB's, it is referred to as visual decibels (vdB). A normalised quality rating is often more useful. Kutter uses the ITU-R Rec. 500 quality rating Q. The rating is computed as:

$$Q = 5 / (1 + N \times E)$$

where E is the measured distortion and N a normalisation constant. N is usually chosen such that a known reference distortion maps to the corresponding quality rating.

4.7 Wavelet-Based Watermarking Schemes

4.7.1 Scheme 1 Based on Zerotrees

4.7.1.1 Zerotrees of Wavelet Coefficients

In **[InMi98]** a scheme based on zerotrees is presented. Given an amplitude threshold T, if a wavelet coefficient x satisfies $|x| < T$, then the x is said to be insignificant with respect to a given threshold T.

If a coefficient and all of its descendants (i.e. coefficients corresponding to the same spatial location but at finer scales of similar orientation) are insignificant with respect to T, then we call the set, of these wavelet coefficients zerotree for the threshold T.

A parent-child relationship can be defined between wavelet coefficients at different scales corresponding to the same location. With the exception of the highest frequency subbands, every coefficient at a given scale can be related to a set of coefficients at the next finer scale of similar orientation. The coefficient at the coarse scale is called the parent, and all coefficients corresponding to the same spatial location at the next finer scale of similar orientation are called children.

For the wavelet pyramid subband decomposition, the parent-child dependencies are also shown in Figure 3-3. With the exception of the lowest frequency subband, all parents have four children. Given a threshold T to determine whether or not a coefficient is significant, an element of a zerotree for the threshold T is said to be a zerotree root if it is not the descendant of a previously found zerotree root for the threshold T.

4.7.1.2 Classification of Wavelet Coefficients

Using the idea of zerotrees of wavelet coefficients mentioned above, insignificant and significant coefficients may be defined.

Definition 1: If a wavelet, coefficient x at the coarsest scale and all of its descendants u,v, …, satisfy $(|x| <T, |u| <T, |v|) <T,\ldots,)$ for a given threshold T, then they are called *insignificant coefficients*.

Definition 2: If a wavelet coefficient E at the coarsest scale satisfies $|x| >T$ for a given threshold T, then it is called a *significant coefficient*.

4.7.1.3 Watermark Method Using Significant Coefficients

The watermark is embedded by thresholding and modifying significant coefficients at the coarsest scale except for the lowest frequency subband LL3. However, it is well known that modification of these components can lead to perceptual degradation of the signal. To avoid this, in the proposed algorithm, the regions in which the watermark is embedded are edges in the coarsest scale component.

The algorithm is described as follows:

> *step1* Two thresholds T_1 and T_2 are defined such that $T_2>T_1> \alpha C_{max}$, α and C_{max} being calculated "a priori". One of the subbands LH, HL, HH in the coarsest scale is selected and the significant coefficients C_k $(k = 1,2, \ldots, N)$, satisfying $T_1< |C_k| <T_2$ are calculated.
>
> *step2* Information data is encoded to binary digits $W(k)$, $k= 1,2,..,N$.
>
> *step3* For $k = 1,2, \ldots, N$, the watermark are embedded by modifying C_k as follows:
>
> If $W(k) = 1$ and $C_{k,} > 0$, then $C_k =T2$,
>
> if $W(k) = 0$ and $C_k > 0$, then $C_k =T1$,
>
> if $W(k) = 1$ and $Ck < 0$, then $C_k = -T2$, and

```
        if $W(k) = 0$ and $Ck < 0$, then $C_k = -T_1$.
    step4 Save the embedded position, subband label, and two thresholds.
```

The watermark is detected using the embedded position and the threshold values after the wavelet decomposition of the watermarked image, as follows.

```
For $k = 1,2, \ldots, N$,
    step 1 Using the subband label and the embedded position, the corresponding wavelet coefficients $C'_k$ are read,
    step 2 Check $C'_k$.
        If $|C'_k| < (T1+T2)/2$, the embedded data is "0".
        If $|C'_k| >= (T1+T2)/2$, the embedded data is "1".
End for
```

4.7.2 Watermarking in the DWT Domain (Scheme 2)

Another watermarking scheme is presented in **[XiBo97]**. The basic idea using the DWT for a one dimensional signal is the following. A signal is split into two parts: the high frequencies and the low frequencies. The part with the high frequencies mostly contains the edge components of the signal. The part with the low frequencies is split again into two parts for the high and low frequencies. This process is continued an arbitrary number of times, which is usually determined by the application at hand. From these DWT coefficients, the original signal can be reconstructed. This reconstruction process is called the inverse DWT (IDWT). The DWT and IDWT for two dimensional images x[m, n] can be similarly defined by implementing the one dimensional DWT and IDWT for each dimension m and n separately: $DWT_n[DWT_m[x[m, n]]]$.

Watermarking in the DWT domain includes two parts: encoding and decoding. In the encoding part, they **[XiBo97]** first decompose an image into several bands with a pyramid structure as shown in **Figure 3-1** and then add pseudo-random sequence (Gaussian noise) to the large coefficients which are not located in the lowest resolution, i.e., the corner at the left and top, as follows. Let y[m, n] denote the DWT coefficients, which are not located at the lowest frequency band, of an image x[n, m]. A Gaussian noise N[m, n] is added with mean 0 and variance 1 to y[m, n]:

$$y'[m,n] = y[m,n] + \alpha(y[m,n])^2 N[m,n], \tag{4-4}$$

where α is a parameter to control the level of the watermark, the square 2 indicates the amplification of the large DWT coefficients. The DWT coefficients at the lowest resolution are not changed. Then the two dimensional IDWT of the modified DWT coefficients j and the unchanged DWT coefficients at the lowest resolution are selected. Let x'[m, n] denote the IDWT coefficients. For the resultant image to fit within the 0 to 255 integer values, as for typical image data, it is modified as:

$$x'[m, n] = \lceil (255\, x'[m, n] - \min_{m,n}(x'[m, n])) / (\max_{m,n}(x'[m, n] - \min_{m,n}(x'[m, n]))) \rceil \tag{4-5}$$

The resultant image x'[m, n] is the watermarked image of x[m, n].

The decoding method they propose is hierarchical and described as follows. First the received image is decomposed and the original image (it is assumed that the original image is known) is decomposed with DWT into four bands, i.e., low-low (LL_1) band, low-high (LH_1) band, high-low (HL_1) band, and high-high (HH_l) band, respectively. They, then, compare the signature added in the HH_1 band and the difference of the DWT coefficients in the HH_l bands of the received and the original images by calculating their cross cross-correlations.

If there is a peak in the cross correlations, the signature is deemed to be detected. Otherwise, the original and the received signals are decomposed in the LL_l band into four additional subbands LL_2, LH_2, HL_2 and HH_2 and so on until a peak appears in the cross correlations. Otherwise, the signature can not be detected.

4.7.3 WaveMark

This watermarking scheme is called WaveMark **[WaWi98]**, and is a wavelet-based multiresolution digital watermarking system for color images. The algorithm in WaveMark uses discrete wavelet transforms and error-correcting coding schemes to provide robust watermarking of digital images. The algorithm uses Daubechies' advanced wavelets and extended Hamming codes to deal with problems associated with JPEG compression and random additive noise. The use of Daubechies' advanced wavelets makes the watermarked images more perceptively faithful than the images watermarked with the Haar wavelet transform. The watermark is adaptively applied to different frequency bands and different areas of the image, based on the smoothness of the areas, to increase robustness within the limits of perception.

4.7.3.1 Encoding Algorithm

The encoding algorithm of the image watermarking system uses Daubechies' wavelet transforms and error correcting coding. The system is designed to place a 32-bit watermark in the image with high redundancy.

The image is first converted and stored in a component color space with intensity and perceived contrasts. For each color component, a 4-level transform using the Daubechies-4 wavelet is performed. For very large images, a wavelet transform with a higher number of levels can be applied.

Also a smoothness analysis is realized: a rough smoothness region overlay for each image is extracted. For watermarking purposes, the watermark is applied with lower strength to regions classified as highly smooth (such as sky) because watermark coding with higher strength in these regions introduces noticeable distortion. They apply watermark coding with higher strength to regions with lower levels of smoothness.

Error Correcting Coding

A Hamming code is used to add redundancy to the bits, so that errors can be detected or corrected to a certain extent. A Hamming code is a linear block code. The main advantage of linear block codes is their simplicity in implementation and low computational complexity. A linear block code is usually composed of two parts. The first part contains the information bits, which are the original bits to be transmitted. The second part contains the parity checking bits, which are obtained by summing over a subset of the information bits. A linear block code with length n and k information bits is denoted as a (n, k) code.

4.7.3.2 Inserting the Watermark Code

The entire wavelet transform of each color component is partitioned into 10x10 non-overlapping blocks. The lower bits of the block borders are altered to code '0' in order to assist the decoding process. With 8x8 matrix entries within each block, a 64-bit watermark code could be hidden. The Hamming encoded watermark has 64 bits. Each entry in the transformation matrix is encoded with one bit of the watermark code. The lower bits of the coefficients are altered according to the frequency represented in the band, the smoothness region overlay, and the encoded watermark code. For the lower frequency bands of coefficients, a watermark coding with higher strength is applied. That is, the higher order bits in these bands are altered. Similarly, higher order bits in regions with higher variations are altered.

4.8 Conclusion

In this Chapter, the idea of a watermark is defined and some of the desired properties of the watermarks in general are enumerated. In section 4.4 the general process of watermarking is presented.

The image containing the watermarks can suffer some intentional or unintentional processing. The possible attacks are presented in section 4.5. The insertion of the watermark may introduce a visual degradation of the image. The visual degradation can be measured using Visual Quality Metrics presented in section 4.6.

Section 4.7 presents several watermarking schemes based on wavelets. These watermarking schemes need the original image for the detection of the watermarks so they are well suited for copyright purposes. The next Chapter introduces blind watermarking schemes that do not need the original image to detect the watermark. These types of techniques are more suitable for image searching tools.

5 WAVELET-BASED BLIND WATERMARK DETECTION TECHNIQUES

5.1 Introduction

The blind watermark retrieval technique is important for applications in very large image databases. To give an example, for archived movie films, art libraries and Internet image distributors, it may not be convenient to search the original image from a huge database for watermark detection.

Current techniques described in the literature for the watermarking of images can be grouped into two classes: transform domain methods which embed the data by modulating the transform domain coefficients and spatial domain techniques which embed the data by directly modifying the pixel values of the original image.

It is today widely accepted that robust image watermarking techniques should largely exploit the characteristics of the Human Visual System (HVS) for more effectively hiding watermarks. From this point of view the Digital Wavelet Transform (DWT) is a very attractive tool, given its intrinsic similarity to the theoretical models of the HVS perception.

5.2 Threshold Dependent Watermarking

In **[AhRa98]** is presented a method that can add the watermark to the significant coefficients in the DWT domain but which does not require the original image in the detection process. Since the watermark is added to significant coefficients in the DWT domain, the method is much more resistant to common image manipulations. The time-frequency localization properties lead to implicit visual masking; this improves the correlation coefficient in the detection process considerably.

5.2.1 Method Description

Figure 5-1 shows a block diagram of the proposed method. A three level DWT with a Daubechies 8-tap filter [Da92] is used. The low pass sub-band is not used (only coefficients in the other sub-bands which are above a given threshold (T_1) are used).

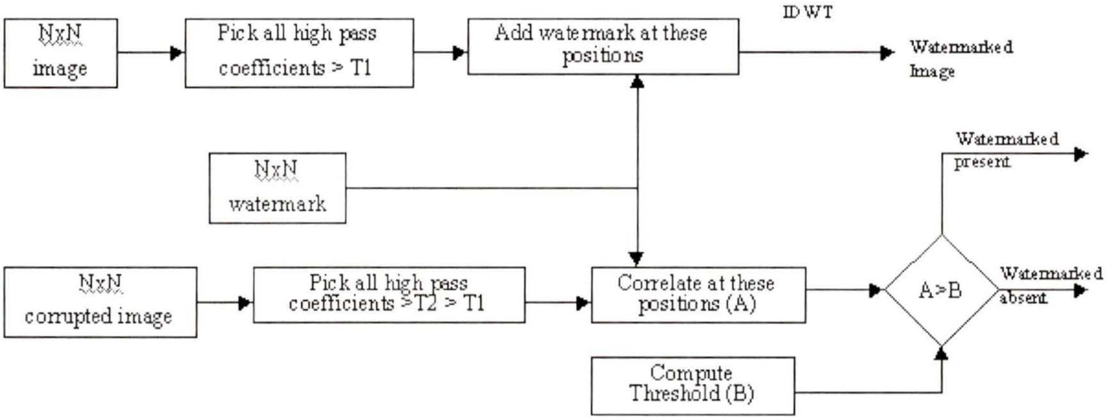


Figure 5-1 Watermark casting and detection [AhRa98]

A small image is used as a watermark and added only to the significant coefficients. The watermark is added to all coefficients (barring the lowpass component) above a threshold T_1. Small coefficients in the DWT domain are more susceptible to being corrupted by compression and other image manipulations like denoising, compared to the large coefficients.

Visual masking is implicit due to the time-frequency localization properties of the DWT. High pass bands, where the watermark is added, typically contain edge related information of the image. Furthermore, each coefficient in the high frequency bands affects only a spatially limited portion of the image. Thus, adding the watermark to significant coefficients in the high frequency bands is equivalent to adding the watermark to only the edge areas of the image, which tends to make the watermark invisible to the human visual system.

During watermark detection, all the high pass coefficients above T_2 are picked and then correlated with the original copy of the watermark. They use $T_2 = 50$ and $T_1 = 40$ (T_1 is the threshold used for watermark casting). $T_2 >= T_1$ is necessary because correlation should be computed only over watermarked coefficients. T_2 is chosen to be strictly larger than T_1 for robustness since some coefficients, which were originally below T_1, may become greater than T_1 due to image manipulations.

The equations used for watermark casting and detection are:

$$V_i' = V_i + \alpha\, |V_i|\, x_i \tag{5-1}$$

where i runs over all DWT coefficients > T_1 (barring the lowpass component). V_i denotes the corresponding DWT coefficient of the original image and V_i' denotes the DWT coefficient of the watermarked image. x_i is the watermark value at the position of V_i. x_i is generated from a uniform distribution of zero mean and unit variance. α is taken as 0.2.

For watermark detection the same procedure as above is followed but now only the coefficients (again barring lowpass component) > T2 > T1 are used as explained above. The correlation z between the DWT coefficients V' of the corrupted watermarked image and a possibly different watermark y is computed as

$$z = 1/M \sum V_i'\, y_i \tag{5-2}$$

where i runs over all coefficients > T2 > T1 and M is the number of such coefficients.

The threshold S is defined as

$$S = (\alpha/2M) \sum |V_i'| \tag{5-3}$$

If the S is greater than threshold z then the watermark is detected.

5.3 Key Watermarking

In **[KuHa98]** they present a technique for the digital watermarking of still images based on the concept of multiresolution wavelet fusion. The original unmarked image is not required for watermark extraction.

5.3.1 Watermark Technique Description

In a multiresolution decomposition, the image is separated into bands of approximately equal bandwidth on a logarithmic scale. The binary watermark is of length N_w and consists of elements from the set {-1,1}. **[KuHa98]** embeds the watermark into the detail wavelet coefficients of the host image with the use of a key. This key is randomly generated and is used to select the exact locations in the wavelet domain in which to embed the watermark. For each coefficient within the wavelet domain, the key has a corresponding value of one or zero to indicate if the coefficient is to be marked or not, respectively. The number of ones in the key must be greater or equal to the size of the watermark. The watermark values are repeatedly embedded in different coefficients selected by the key if the length of the watermark is less than the number of ones in the key.

The technique is comprised of the three steps described below:

Step I: Compute the *Lth*-level discrete wavelet decomposition of the host image to produce a sequence of 3L detail images, corresponding to the horizontal, vertical and diagonal details at each of the L resolution levels, and a gross approximation of the image at the coarsest resolution level. $f_{k,l}(m, n)$ denotes the *kth* detail image component at the *lth* resolution level of the host image, where $k = \{h, v, d\}$ (which stands for "horizontal", "vertical" and "diagonal", detail coefficients, respectively) and l = 1, …, L.

Step II: Consider each resolution level l, and coefficient location (m, n). If the associated value of the key y is one then proceed as follows, otherwise do not embed a mark. Sort the detail coefficients in ascending order so that $f_{k1,l}(m, n)$, $f_{k2,l}(m, n)$, and $f_{k3,l}(m, n)$ are coefficients such that

$$f_{k1,l}(m, n) \leq f_{k2,l}(m, n) \leq f_{k3,l}(m, n), \tag{5-4}$$

where $k_1, k_2, k_3 \in \{h,v,d\}$ and k_1, k_2, k_3 are distinct.

To embed the watermark, $f_{k2,l}(m, n)$ is quantized as shown in Figure 5-2

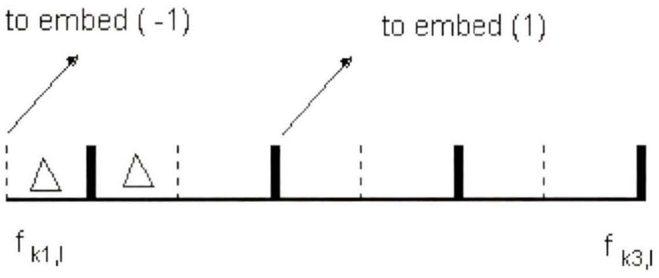


Figure 5-2 Quantization process used in Key Watermarking Scheme

The range of values between $f_{k1,l}(m, n)$ and $f_{k3,l}(m, n)$ is divided into bins of width

$$\Delta = (f_{k3,l}(m, n) - f_{k1,l}(m, n)) / 2Q - 1 \tag{5-5}$$

where Q is a user-defined variable. To embed a watermark bit of value one, $f_{k2,l}(m, n)$ is quantized to the nearest value shown with bold verticals bars in Figure 5-2. Alternatively, to embed a negative one, $f_{k2,l}(m, n)$ is quantized to the nearest value shown by dashed vertical lines.

Step III: The corresponding Lth level inverse wavelet transform of the image is computed to form the watermarked image.

5.3.2 Watermark Extraction and Detection

The detection process requires knowledge of the watermark $\omega(m,n)$ and the key $y(m, n)$. $r(m, n)$ represents the image to which the extraction process is applied. The key y is used to find the locations in which the watermark was embedded for each resolution level l. The watermark bit value is computed from the relative position of $r_{k2,l}(m, n)$. Using the same constant Q as for embedding, a particular watermark bit is determined by finding the closest quantized value (shown in Figure 5-2) to $r_{k2,l}(m, n)$ and determining if this quantized value was used to embed a one or a negative one.

A given watermark is detected if the correlation of the extracted watermark with the given watermark is above a pre-specified threshold. More precisely, the watermark detection condition is given by

$$\rho(\omega,\tilde{\omega})=\frac{\sum\omega(n)\tilde{\omega}(n)}{\sqrt{\sum\omega^2(n)}\sqrt{\sum\tilde{\omega}^2(n)}}\geq T_1 \qquad (5\text{-}6)$$

where ω is the given watermark, ω' is the extracted one, and T is a pre-specified threshold. The quantity $p(\omega,\omega')$ is known as the correlation coefficient between the given and extracted watermarks.

5.4 HVS Sensitive Blind Watermarking

In **[PiCa99]** a watermarking algorithm that does not rely on the original image for recovering the watermark embedded in the DWT domain, is presented. The watermark is a binary {+1, -1} pseudo-random sequence that is added to the DWT coefficients of the three largest detail (i.e. LH, HL, HH) subbands of the image. Each binary value is multiplied, before adding it, by a weighting parameter, which is obtained by the noise sensitivity function used in the framework of a compression system, for adaptively tuning the quantization steps. In this way, the maximum visibly tolerable level of disturbance (i.e. watermark coefficient) is added to each DWT coefficient. The construction of the sensitivity function is mainly based on the analysis of the degree of image activity in the neighborhood of the pixel to be modified.

For watermark detection, the correlation between the watermark to be tested for presence and the marked coefficients is computed. The value of the correlation is compared to a threshold to decide if the watermark is present or not. An optimum threshold is theoretically set to minimize the probability of false positive detection. The value of this threshold depends on the variance of the DWT coefficients of the watermarked image, and can thus be computed a-posteriori.

5.4.1 Watermarking in the DWT domain

The image to be watermarked is first decomposed through DWT in four levels: let us call I_j the sub-band at resolution level j = 0, 1, 2, 3 and with orientation $\theta \in \{LL, LH, HL, HH\}$. In **[PiCa99]** , the Daubechies-6 filtering kernel has been used for computing the DWT. The watermark, consisting of a pseudorandom binary sequence, is then inserted by modifying the wavelet coefficients belonging to the three detail bands at level 0, i.e. I_0^{LH}, I_0^{HL}, and I_0^{HH}. This choice has been done based on experimental tests as offering the best compromise between robustness and invisibility. Indeed inserting the watermark into these bands makes it more sensitive to attacks (e.g. compression or low-pass filtering) but, given the low visibility of disturbs added to these frequencies, a higher level of watermark strength is allowed, thus compensating for the outlined bigger fragility.

In more detail, given a pseudorandom binary sequence $x_i \in \{+1, -1\}$, with i=0,…3MN – 1, where 2M x 2N is the dimension of the original image, the three highest level detail sub-bands are modified according to the rule:

$$\tilde{I}_0^{LH}(i,j) = I_0^{LH}(i,j) + \alpha\,\omega^{LH}(i,j)x_{iN+j}$$

$$\tilde{I}_0^{HL}(i,j) = I_0^{HL}(i,j) + \alpha\omega^{HL}(i,j)x_{MN+iN+j} \qquad (5\text{-}7)$$

$$\tilde{I}_0^{HH}(i,j) = I_0^{HH}(i,j) + \alpha\omega^{HH}(i,j)x_{2MN+iN+j}$$

where α is a parameter accounting for watermark strength and $\omega(i, j)$ is a weighting function taking into account the local sensitivity of the image to noise. This weighting function allows us to exploit the masking characteristics of the HVS.

Watermark detection is accomplished without referring to the original image. The correlation between the marked DWT coefficients and the watermarking sequence to be tested for presence is computed :

$$\rho = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \tilde{I}_0^{LH}(i,j)x_{iN+j} + \tilde{I}_0^{HL}(i,j)x_{MN+iN+j} + \tilde{I}_0^{HH}(i,j)x_{2MN+iN+j} \tag{5-8}$$

and compared to a threshold T, chosen a posteriori, based on estimates of the statistical parameters of the watermarked image, in such a way to minimize the probability of false positive detection.

The quantization step takes into consideration HVS (the eye is less sensitive to noise in those areas of the image where brightness is high, to noise in high resolution bands and to noise in the highly textured areas of the image).

For watermark detection, the correlation is computed between the watermarked image DWT coefficients and the watermarking code to be tested for presence; the value of the correlation is then compared to a threshold T_p chosen in such a way as to minimize the probability of a false positive detection.

$$T_\rho = 3.97\sqrt{2\sigma_{\rho B}^2} \tag{5-9}$$

The threshold can be computed a posteriori on the watermarked image: it is possible, for example, to use an unbiased estimate:

$$\sigma_{\rho\beta} \approx \frac{1}{(3MN)^2}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(\tilde{I}_0^{LH}(i,j)\right)^2 + \left(\tilde{I}_0^{HL}(i,j)\right)^2 + \left(\tilde{I}_0^{HH}(i,j)\right)^2 \tag{5-10}$$

5.5 Adaptive Blind Watermarking

A wavelet-based watermark casting scheme and a blind watermark retrieval technique are investigated in [**HoKu98**]. An adaptive watermark hiding method is first developed to determine significant wavelet subbands and to select significant wavelet coefficients in these subbands automatically. Then, a blind watermark retrieval technique that can detect the embedded watermark in the wavelet domain without the help from the original image is proposed.

5.5.1 Significant Subband Search

It is assumed that wavelet coefficients in each subband follow the Gaussian distribution. The purpose of perceptually significant coefficient search is to find significant wavelet coefficients which are robust with respect to attack. If their fidelity is lost, the reconstructed image should be very different from the original image. Furthermore, due to the limitation on the size of the watermark, the watermark is embedded in the first N_w significant coefficients, where N_w is the length of the watermark sequence. In this work, the upper bound of N_w is chosen to be 50% of the original image size.

With bit-plane coding, coefficient $C_s(x, y)$ in subband s is represented by

$$C_s(x,y) = sign \times \left(\alpha_0 \frac{T}{2^0} + \alpha_1 \frac{T}{2^1} + \ldots\ldots + \alpha_b \frac{T}{2^b} + \ldots\right) \tag{5-11}$$

where “sign’" is the sign value (e.g. +1 for positive sign or –1 for negative sign) of coefficient $Cs(x, y)$, α_b is the binary bit at bth bit plane and T is the initial threshold of subband calculated by:

$$T = C_{max,s} / 2$$

and where $C_{max,s}$ is the maximum absolute coefficient value in subband s. Parameter b is called the layer number of the bit plane, and b = 0 indicates the most significant bit (MSB) layer.

5.5.2 Significant Coefficient Skipping

A significant coefficient skipping scheme is used to achieve non-invertibility. A random sequence of integers ranging from 0 to M is generated by another seed to indicate the skipping number.

5.5.3 Summary of Watermarking Algorithm

The algorithm can be summarized as follows:

1. Initialization: Set the initial threshold T_s of each subband to one half of its maximum absolute value of coefficients inside the subband. Set all coefficients un-selected.
2. Select the subband (except the DC term) with the maximum value of (β_s x T_s), where β_s is the weighting factor of subband s. For the selected subband, all un-selected coefficients $C_s(x; y)$ are examined, and coefficients which are greater than the current threshold T as significant coefficients are selected.
3. The watermark is cast in the selected significant coefficients obtained in Step 2.
4. Update the new threshold in subband s via $T_s^{new} = Ts / 2$.
5. Repeat Step 2 to Step 4 until all watermark symbols are cast.

The parameter β_s is used to control the subband selection order.

5.5.4 Blind Watermark Detection

The most difficult problem associated with blind watermark detection in the frequency domain is to identify coefficients with the embedded watermark and the embedded watermark values. The blind watermark detection algorithm is done by truncating selected significant coefficients to some specified value.

5.6 Conclusion

Several wavelet-based blind watermarking techniques are presented. These techniques do not require the original image for recovering the watermark.

Some of the blind watermarking schemes presented are very computationally intensive and are not suitable for fast image searching. Also, most of the algorithms presented in this chapter were intended to be used in the copyright field and are not appropriate for image searching. The use of a key for determining the coefficients to be watermarked (Section 5.3) is not indicated for a large image database because the image structures could be very different.

In the next chapter we present a modified blind watermarking technique more suitable for Internet image searching. Also a framework prototype based on blind watermarking techniques is presented.

6 FRAMEWORK IMPLEMENTATION

6.1 Searching the Web for Images

In this chapter, a framework prototype (VkMark) for watermarking and searching for images on the World Wide Web is described. New visual information in the form of images, graphics, animations, and videos is being published on the Web at an incredible rate. However, searching all this information exceeds the capabilities of current text-based Web search engines.

No efficient tools are currently available for searching for images and videos. This absence is particularly notable given the highly visual and graphical nature of the Web. Visual information is published both embedded in Web documents and as stand-alone objects. The visual information takes the form of images, graphics, bitmaps, animations and videos. As with Web documents in general, the publication of visual information is highly volatile. New images and videos are added every day and others are replaced or removed entirely. In order to search the visual information, a highly efficient automated system is needed that regularly traverses the Web, detects watermark information embedded in images, and processes it in a way that allows for efficient and effective search and retrieval.

6.1.1 Interactive Searching of Image Databases

Our goal is to develop an integrated framework that is capable of searching digital image databases for user-defined watermarks. Databases of interest would include digital photograph albums and such databases as are found on the World Wide Web. These databases can often be too large to allow every image to be inspected visually, and automatic techniques are required to assist with searching and interpretation. A user could define a watermark ID using a graphical interface, and the images that contain the watermark are then found automatically.

In this thesis we are proposing a new approach: searching for embedded watermarks in the images. The watermark may contain different information, and could serve different goals: for example some image composers may have the option of saving an invisible watermark together with the image. The web crawler searches for the existence of the watermark (a boolean response is enough) and the vendor of the image composer can get a measure of the usability of his product on the Web.

6.2 Interface Implementation Considerations

The framework user interface uses MFC's (Microsoft Foundation Class) document/view architecture that makes it easy to support multiple views, multiple document types, splitter windows, and other valuable user-interface features. At the heart of the document/view approach are four key classes:

- CDocument (or COleDocument), an object used to store or control the program's data.
- CView (or one of its many derived classes), an object used to display a document's data and manage user interaction with the data.
- CFrameWnd (or one of its variations), an object that provides the frame around one or more views of a document.
- CmultiDocTemplate, an object that coordinates one or more existing documents of a given type and manages creation of the correct document, view, and frame window objects for that type.

These classes have been used to display the image and the watermark in different perspectives (original, transformed, quantized, inverse quantized, watermarked).
This application needs to support two major document types: watermark, and image. Each document type is represented by its own document class and by its own view class as well. When the user chooses the File New command, the framework displays a dialog box that lists the supported document types. Then it creates a document of the type that the user chooses. Each document type is managed by its own document template object.

The image document is shown in a splitter window containing four scrollable panes (Figure 6-1). A splitter control (or "split box") in the window frame next to the scroll bars allows the user to adjust the relative sizes of the panes. Each view has its own class, each with a different purpose.

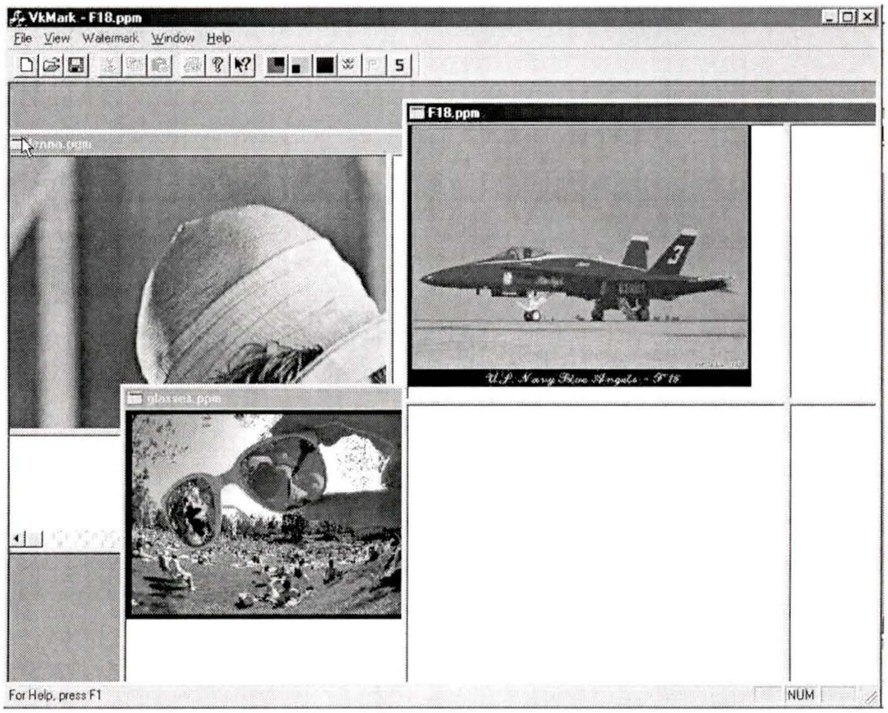


Figure 6-1 VkMark framework, overall multidocument display

There are four panels used to display the image in different stages of the transformation. The top left one is the original image. In the top right panel is displayed the transformed image using one of the wavelet transforms (Daubechies4, Daubechies6, Daubechies8, Daubechies10, Daubechies12, Battle Lemarie, Coiflet2, Coiflet4, Coiflet6, and Haar). In the bottom left panel is displayed the watermarked image after the inverse transform and the bottom-right panel corresponds to the difference between the watermarked image and the original watermark.

6.3 VkMark Framework Structure

The VkMark framework is composed from 5 subsystems:

-backbone system

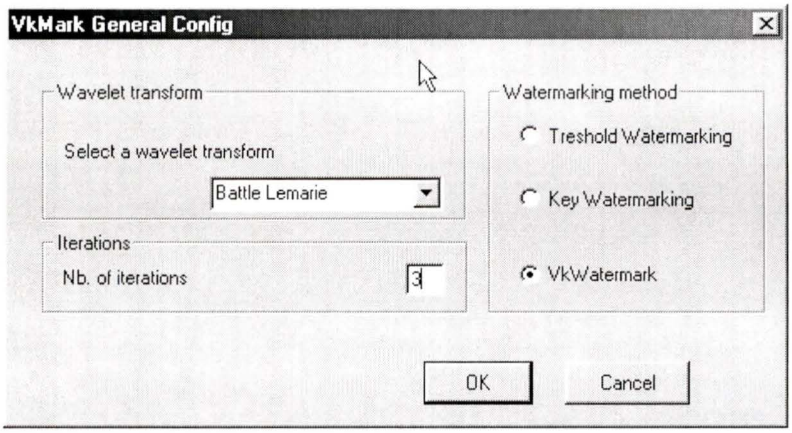
-watermark encoding system
-watermark detecting system
-watermark benchmark
-search engine

6.3.1 Backbone system

All the five systems are using the same backbone and are "living" in the same framework. The images are stored in ppm (pgm) format. This format handles both the gray and color scale images. After reading from a pgm (ppm) file the image is stored in an internal format. In this way the system can be very easily extended to read from other desired formats.

The design of the class structure of the backbone actually follows the flow of the image compression (not including the entropy encoding). Each step has associated a class, and could be very easily modified by inheriting from the desired class.

The wavelet transform is chosen interactively from the following implemented transforms: Daubechies4, Daubechies6, Daubechies8, Daubechies10, Daubechies12, BattleLemarie, Coiflet2, Coiflet4, Coiflet6, and Haar. In Figure 6-2 the VkMark Dialog for parameter configuration (wavelet transform, number of iterations, watermarking method) is shown.



VkMark General Config

Wavelet transform
Select a wavelet transform: Battle Lemarie

Iterations
Nb. of iterations: 3

Watermarking method
○ Treshold Watermarking
○ Key Watermarking
● VkWatermark

Figure 6-2 Basic configuration of the VkMark parameters

A signal is split into two parts of high frequencies and low frequencies. The part with the high frequencies is basically the edge components of the signal. The part with the low frequencies is split again into two parts of high and low frequencies. This process is continued an arbitrary number of times, which is set by the user (Number of iterations – in VkMark General Config). Figure 6-3 and Figure 6-4 present two cases of number of iterations parameter (N=2 and respectively N=4) for two different images.

Figure 6-3 Wavelet transform (Daubechies 4) , No of iterations = 2

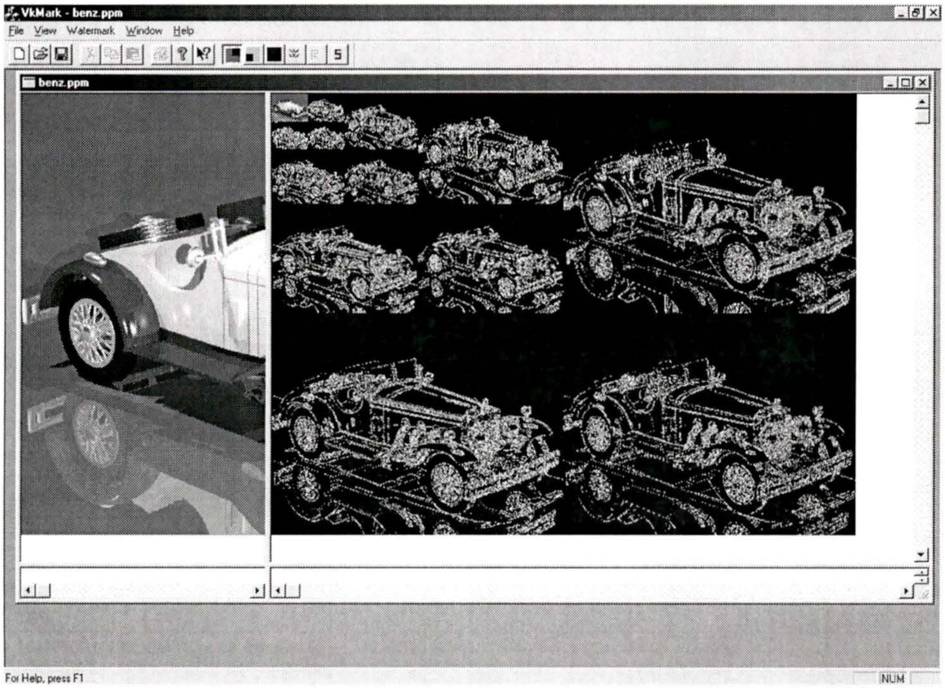


Figure 6-4 Wavelet transform (Daubechies 4) , No of iterations = 4

The user can also set the desired watermarking method. At this time three types of watermarking are implemented:

1. Threshold watermarking as described in Section 5.2
2. Key watermarking as described in Section 5.3
3. VkWaterMark (our method designed for Internet images) to be presented in Section 6.3.2

All three are blind watermarking schemes and they do not need the original image to detect the watermark.

6.3.2 Watermark Encoding System

Watermarking in the DWT domain includes two parts: encoding and decoding. Two types of watermarking have been implemented:

1) key watermarking: the watermark is a digital ID or a small image which could be set by the user. This method uses the spread spectrum technique.

2) random type watermarking: the watermark is a random sequence in {1, -1}. The image is first decomposed into several bands with a pyramid structure and then a pseudo-random sequence is added to the "important" coefficients which are not located in the lowest resolution, i.e., the corner at the left and top, as follows. There are several algorithms that search for these "important" coefficients. In general they use the properties of the HVS (Human Visual System) to find the desired coefficients.

In order for them to be used in searching in large databases these methods must add the watermark to the "important" coefficients in the DWT domain and not require the original image in the detection process. Since the watermark is added to significant coefficients in the DWT domain, the method is much more resistant to common image manipulations.

All the watermark classes are derived from the base class *CWatermarkB* and implement a series of methods declared as virtual in the parent class.

```
class CWatermarkB
{
protected:
        /*the watermark string in binary form (1,-1)*/
        int *wBinaryBuff;
        CInternalStruct             *iStruct;
        CWaveForwardTransform         *wFTrans;
        CWaveInverseTransform       *wITrans;
public:
        virtual void LoadWatermark(const char *fileName);
        virtual void DoWatermark();
        virtual int DetectWatermark();
        virtual CString GetWatermark();
        virtual void GenerateWatermark(int length);
};
```

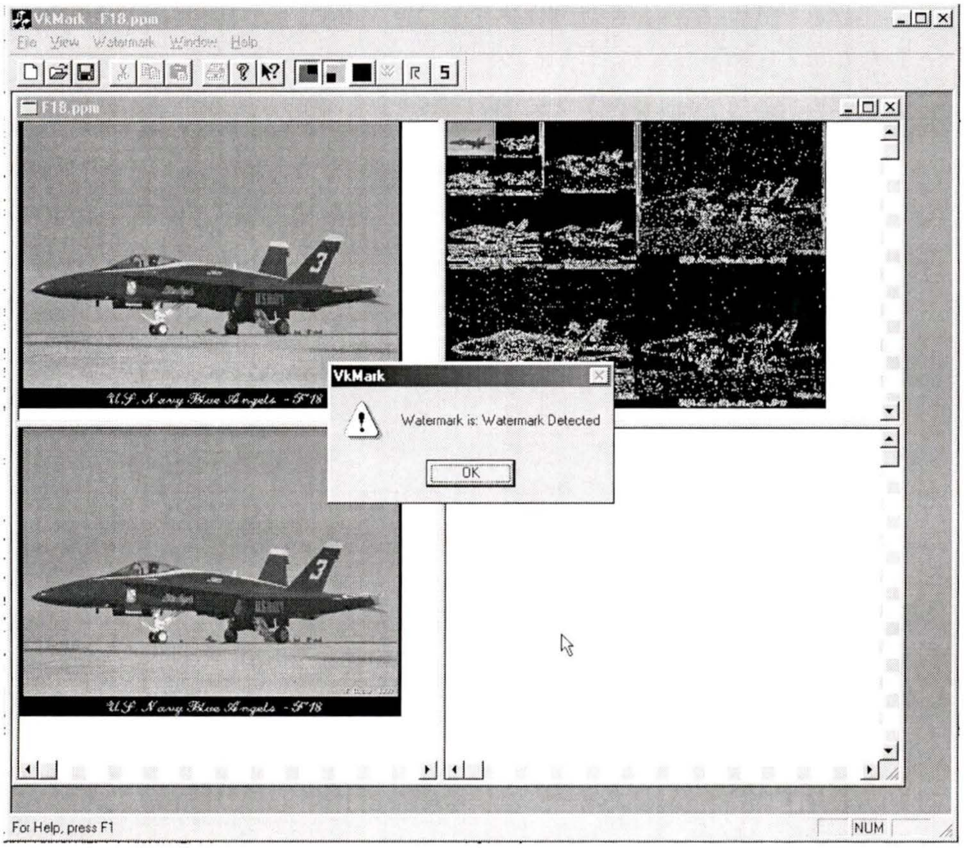
The *GenerateWatermark* method generates a uniform distribution of numbers with values {1} and {–1}. The watermark is saved in a file, and the name of the file together with an ID is entered in a database. When you want to use a watermark, you choose an ID and the system automatically loads the watermark from a file and does the watermarking or searches for that watermark.

6.3.3 Watermark Detection System

Given an image in ppm (pgm) format and depending on the watermark detection method, a watermark is extracted or a boolean response is given indicating the existence of the watermark.

Usually, in the first case the original image is needed. The objective of the watermark extraction process is to reliably obtain an estimate of the original watermark from a possibly distorted version of the watermarked image. For watermark detection, the correlation between the watermark to be tested for presence, and the marked coefficients is computed. The value of the correlation is compared to a threshold to decide if the watermark is present or not.

In Figure 6-5 the "F18.ppm" image is transformed using forward Daubechies6 wavelet transform, watermarked using the threshold watermarking scheme, and then it is transformed again using inverse Daubechies6 transform. The final step is the watermark detection. All the steps can be viewed in the four panels.



Watermark is: Watermark Detected

Figure 6-5 The process of watermarking and detecting the watermark

6.3.4 Benchmark system

The goal of the benchmark is to help in determining if a watermark can survive after common processing steps (usually unintentional):

- JPEG compression - JPEG is currently one of the most widely used compression algorithms for images and any marking system should be resilient to some degree of compression. To transform from ppm (pgm) to jpeg we used the free software from *The Independent JPEG Group's JPEG software* (ftp://ftp.uu.net/graphics/jpeg/ jpegsrc.v6b.tar.gz). This package contains C software to implement JPEG image compression and decompression. JPEG is a standardized compression method for full-color and gray-scale images. JPEG is lossy, meaning that the output image is not exactly identical to the input image.

- Geometric transformations

  -*Scaling* - This happens when an image is scaled (modified the width and the height)

Publishers often use these manipulations to enhance images. Watermarking algorithms should be resistant to these "attacks".

The benchmark uses metrics to measure also the quality of the image. The most popular distortion measures in the field of image and video coding and compression are the Signal to Noise Ratio (SNR), and the Peak Signal to Noise Ratio (PSNR). They are usually measured in decibels

(dB): $\mathrm{SNR(dB)} = 10 \log_{10} (\mathrm{SNR})$.

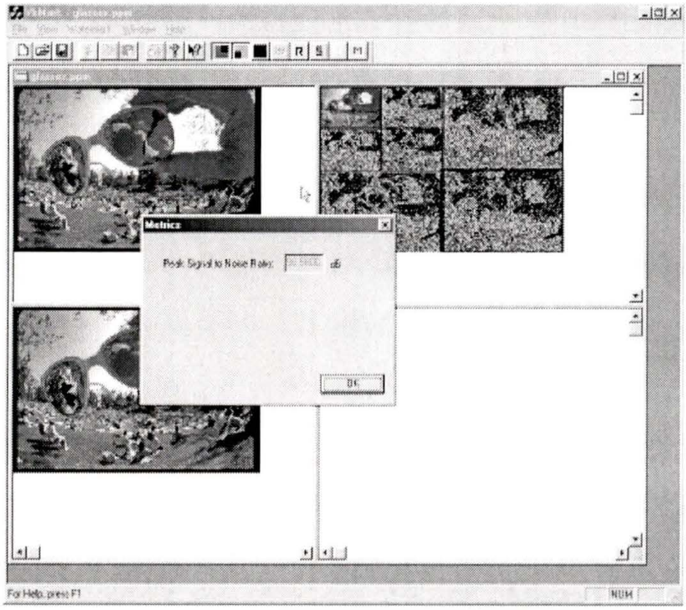


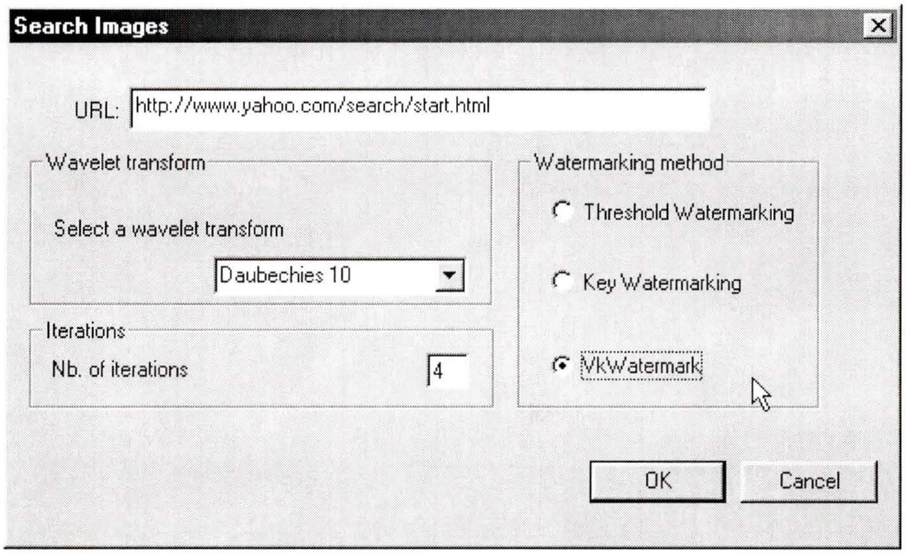
Figure 6-6 PSNR display after watermarking and inverse transformation steps

Figure 6-6 displays a capture of the dialog box that shows the calculation of PSNR after the image was watermarked.

6.4 Search Engine

The search engine is included in the same framework. Its main purpose is to search given domains, download the images, and then launch the watermarking detection system to see if there is a watermark present in the image. If a watermark is found, the name of the image and the URL is added to database together with the current watermark method name and the watermark ID.

A URL is entered in a dialog box (Figure 6-7) and then the framework searches for <IMG> html tags. The image is saved in a temporary directory, transformed to ppm format, and then the watermarking detection engine is launched.


Search Images

URL: http://www.yahoo.com/search/start.html

Wavelet transform

Select a wavelet transform: Daubechies 10

Iterations

Nb. of iterations: 4

Watermarking method

- ○ Threshold Watermarking
- ○ Key Watermarking
- ● VkWatermark

Figure 6-7 Dialog box for start searching the images

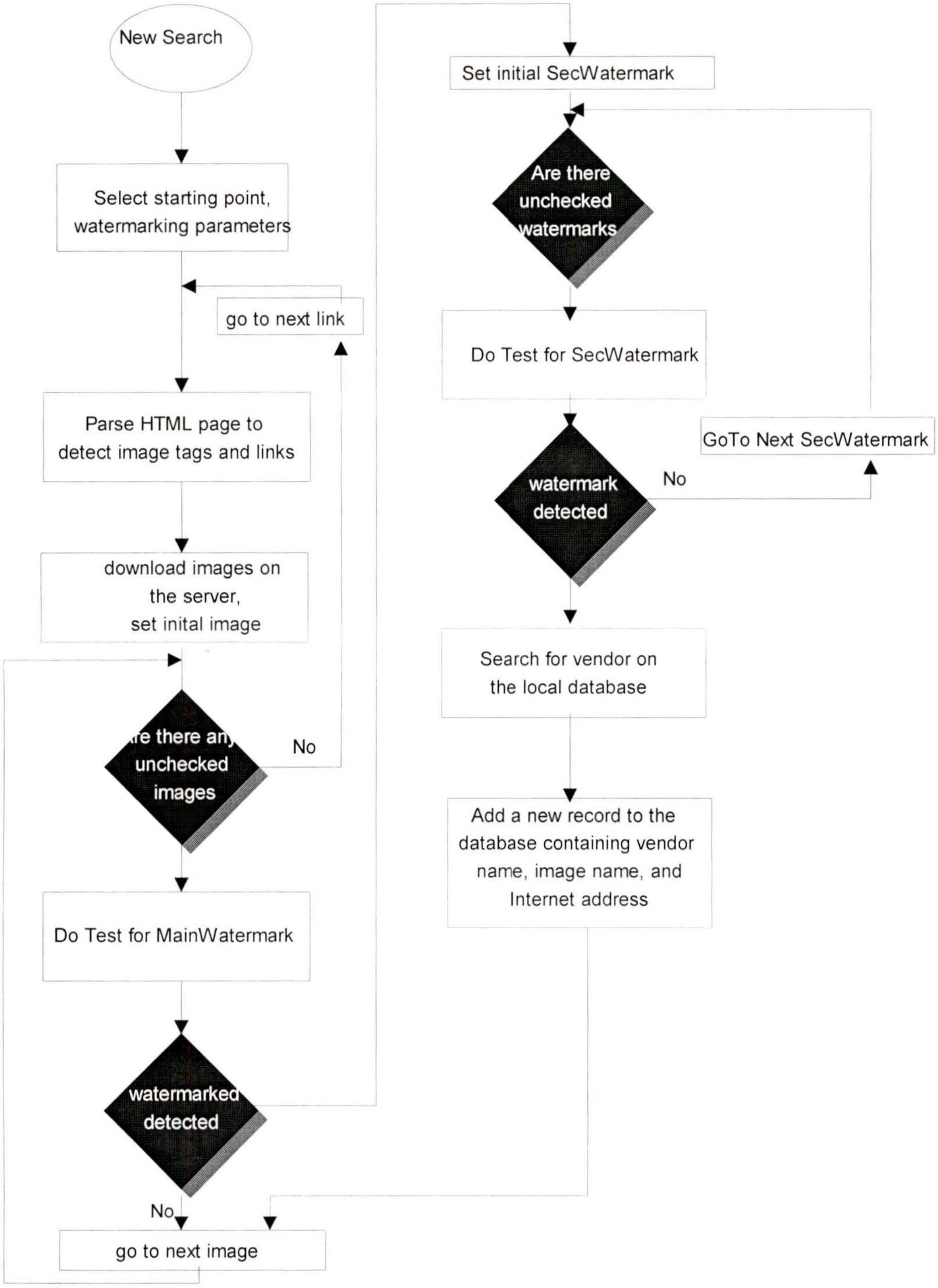


Figure 6-8 **Workflow diagram for Search Engine component**

6.5 Comparison And Results

In this chapter results of watermarking techniques are presented. Besides designing digital watermarking methods, an important and often neglected issue addresses proper evaluation and benchmarking. A fair benchmark should include both evaluation of robustness and evaluation of distortion introduced by the watermark. Examples and results of the tests are summarized below.

A new technique (VkWaterMark) is implemented to adapt the classic watermarking scheme used for copyright detection, to a watermarking scheme used for Internet image searching. A detailed set of experiments is applied to test how the watermarks perform under typical processing operations.

VkWaterMark Scheme Description

The main disadvantage of using classic watermarking methods in image searching is that the search engine has to verify the existence of all watermarks in each image. Taking into consideration the huge number of images on the Internet, this would be unacceptable.

We have improved the method presented in Chapter 5.3 by two aspects: search speed and robustness. The VkWaterMark improves the search speed by splitting the watermark into two smaller watermarks having lengths N1 and N2 respectively. We name these watermarks: *mainWatermark* and *secWatermark*. In this way the search algorithm does not have to search for presence of all watermarks in each image. *mainWatermark* does not change and gives a response if the image is watermarked or not. The "owner" of the image is determined using *secWatermark*.

The initial search is done using *mainWatermark*. If the watermark is detected, the image is tested for the *secWatermark*. All the *secWatermarks* are stored in a local database and the watermark detection is applied until a watermark is found.

The robustness is improved by modifying the watermarking insertion and detection procedure as follows: the watermarking method presented in **[KuHa98]** inserts the watermark in specific points determined by a key. This key is not image dependent so you may get unexpected results on some images, in the case that the key coefficients are insignificant (close to 0).

Our method takes this aspect into consideration and tests if the relevant coefficient is insignificant (smaller than a threshold T_1). If that coefficient is smaller than T_1, the coefficient is not watermarked. Also, another source of false detection in **[KuHa98]** could occur when the interval between the extreme coefficients is very small. The VkWaterMark method tests if the difference between these coefficients is smaller than a threshold T_2, and in this case the coefficient is not watermarked.

In the detection stage, we can have three possible values for the watermark:

- 1: a value of 1 was detected
- -1: a value of –1 was detected
- x: don't care: the value is not determined because of the conditions discussed above

If the detected watermark value is "don't care" the correlation value is not influenced by this watermarking value. Another improvement is that the watermark is inserted as many times as possible, skipping the "don't care" coefficients. A voting procedure is applied in the detection process.

The simplified watermarking insertion algorithm is presented below:

```
watermark = concatenate(mainWatermark, secWatermark)
for each level do
        for each of the coefficients ch, cv, cd
                coeff[0]=ch, coeff[1]=cv, coeff[2]=cd

                SortCoeff(coeff);
```

```
                    If coeff[0]<=T1  AND coeff[2]<=T1 OR (coeff[2]-coeff[0])<=T2
                              *do not embed watermark


                    else
                              *calculate Delta
                              *compute the closest Delta position to  coeff[1]
                               *embed the watermark depending on the watermark value and
                                number of  Delta
                    end if
                    if (watermark position > watermark length)
                              *reinitialize watermark position
          end for
end for
```

The watermarking detection algorithm for *mainWatermark* is presented. A similar algorithm is applied for *secWatermark*:

```
watermark = mainWatermark
lengthMain = length of mainWatermark
lengthSec = length of secWatermarking
k=0
for each level do
          for each of the coefficients ch, cv, cd
                    coeff[0]=ch, coeff[1]=cv, coeff[2]=cd

                    SortCoeff(coeff);

                    If coeff[0]<=T1  AND coeff[2]<=T1 OR (coeff[2]-coeff[0])<=T2
                              * watermark[k] = x (don't care)
```

```
        else
                *calculate Delta
                *compute the closest Delta position to coeff[1]
                *determine the watermark depending on the watermark value and
                 number of Delta
                watermark[k] = watermark[k] + detected value
        end if
        if (k > lengthMain)
                *reinitialize watermark position to 0
                *skip lengthSec coefficients
    end for
end for
```

Both algorithms are used in the overall search engine architecture displayed in Figure 6-8.

6.5.1 Experiments

Different wavelet transforms (Daubechies4, Daubechies6, Daubechies8, Daubechies10, Daubechies12, Battle Lemarie, Coiflet2, Coiflet4, Coiflet6, and Haar) have been tested. For different image sizes we have got different results in a sense that we can not affirm which one is the most suitable for watermarking for the whole range of image sizes. The results show an average of the watermarks used.

False Detection Test

First a false detection test is applied using all three watermarking insertion and detection methods. The images have the original size (uncompressed). This is an important case because many of the images on the Internet are encoded a in lossless format.

If we are using Threshold Watermarking technique presented in Section 5.2, the watermark detection procedure is using only the coefficients (barring lowpass component) that are > T_2 > T_1. The correlation between the DWT coefficients of the corrupted watermarked image and a possibly different watermark Y is computed.

The chart shown in Figure 6-9 presents the variation of correlation for 20 different watermarks when Threshold Watermarking method is used. Daubechies 4 was used as a wavelet transform and the watermark number 10 was the inserted watermark. As we can observe the difference between correlation for detected and undetected watermark is (0.3). This difference is not very big and for a considerable number of images we may get unexpected results.

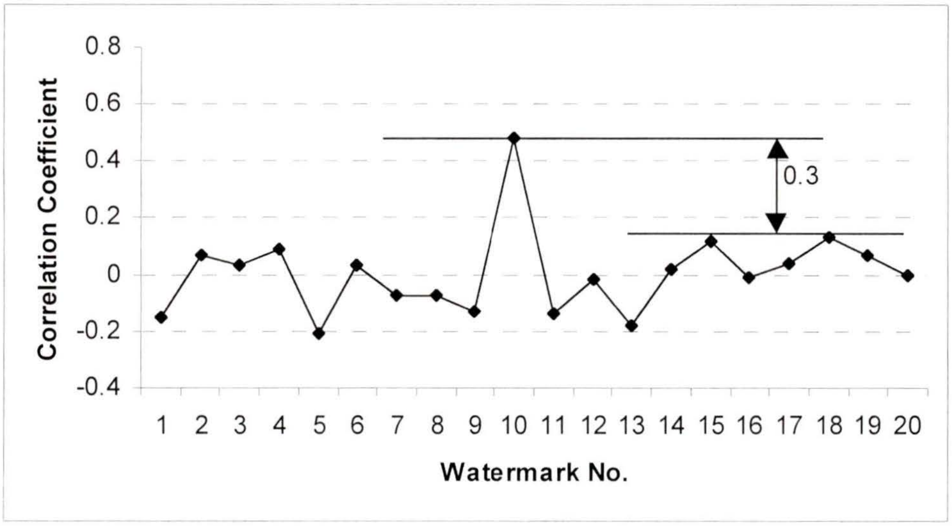


Figure 6-9 False detection test for Threshold Watermarking scheme.

The chart from Figure 6-10 presents the false detection for Key Watermarking scheme, described in Section 5.3, using the same parameters and the same image as in the previous test. We observe a good watermark detection for uncompressed image: Correlation = 1. Also, the difference between the calculated correlation for watermarked and un-watermarked is around 0.8.

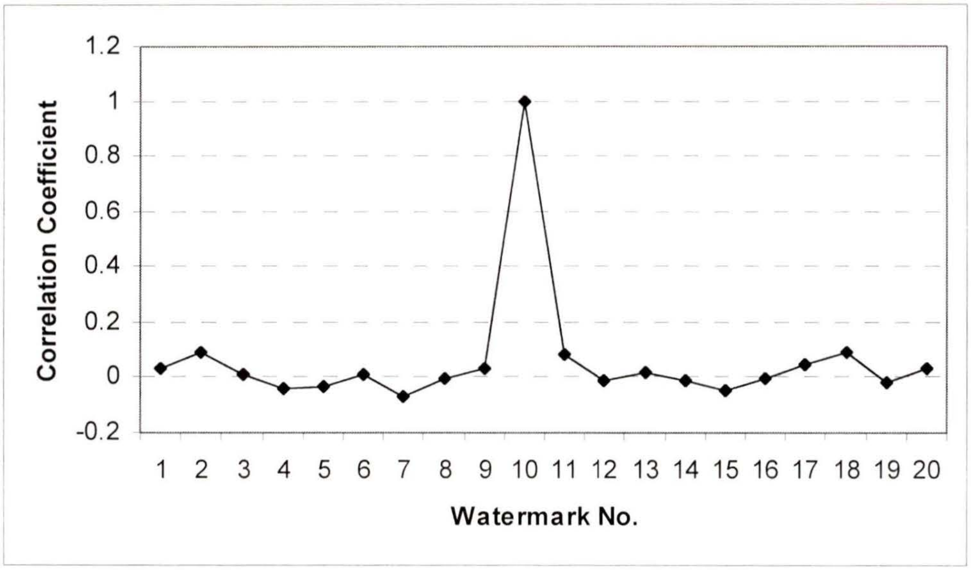


Figure 6-10 **Key Watermarking False Detection test**

The VkWaterMark scheme applied on 20 watermaks gives the results shown in Figure 6-11. Both watermarks (*mainWatermark* and *secWatermark*) are represented in the chart. In the case of uncompressed image we do not see a real improvement over the previous method, because the correlation calculated for the inserted watermark is in the same range as in previous method.

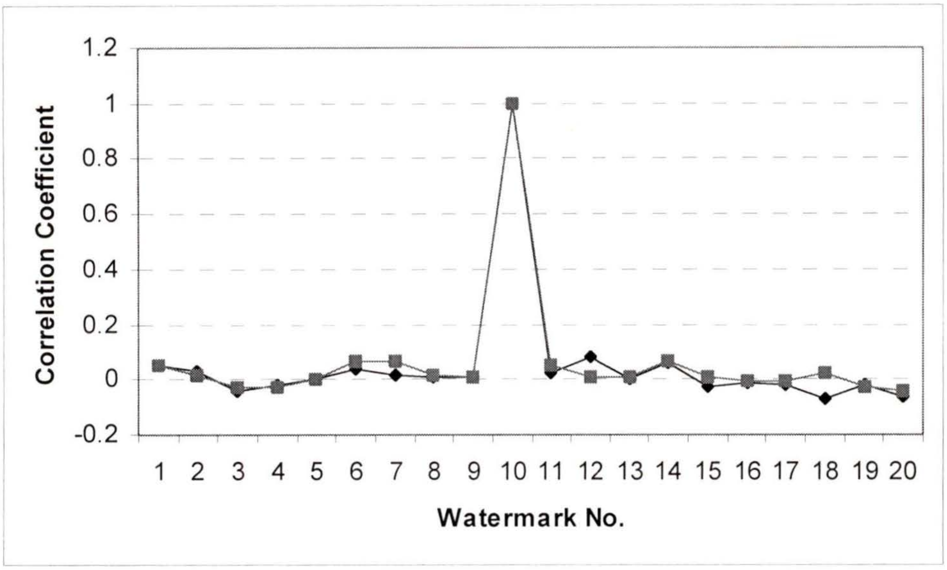


Figure 6-11 VkWaterMark false detection test chart

JPEG compression

In the digital domain, it is very likely that the original source material may be compressed for more efficient storage or transmission. It is therefore important to examine whether the watermarking schemes can survive JPEG compression, since this is the current standard for still image compression.

An experiment was performed to test the preservation strength of watermarks at different compression ratios. The compression ratio (CR) metric is defined as:

CR= (original image size in bits) / (compressed image size in bits)

As it is shown in Figure 6-12, VkWaterMark gives better results than Key Watermarking presented in section 5.3. We have tried with different CR (from 2 to 25) using JPEG compression. As a result, the watermarks after up to CR=25 (JPEG compression) were detected by VkWaterMark detector. Watermark retrieval results are displayed in Figure 6-12. We observe a substantial improvement for low Compression Ratios and some improvement for high Compression Ratios.

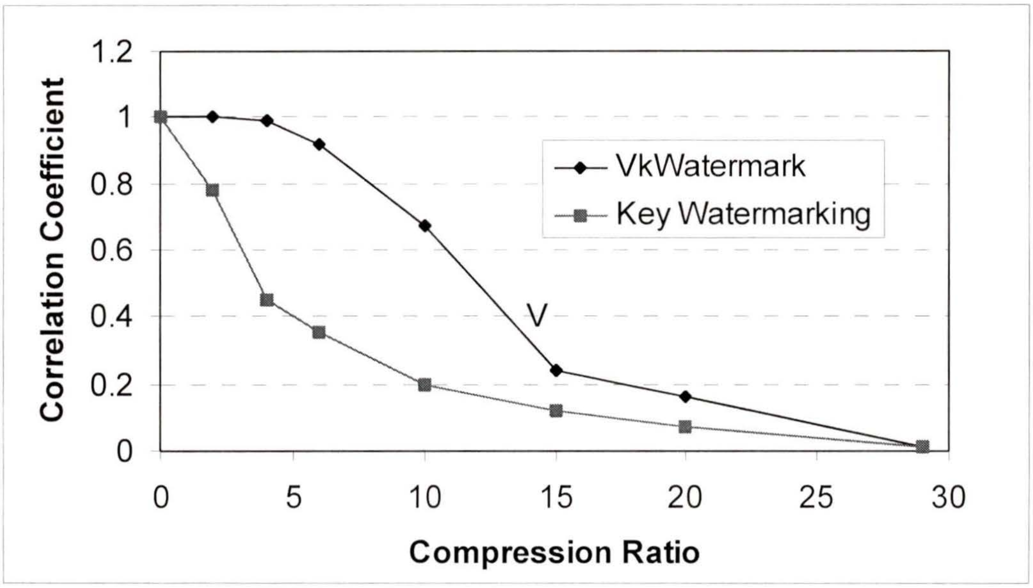


Figure 6-12 **Comparison between Key Watermarking and VkWaterMark schemes**

Figure 6-13 shows the variation of the Correlation with the JPEG Quality. In most of the cases the JPEG quality is around 75%. As we observe in Figure 6-13, the Correlation Coefficient for 75% JPEG Quality is around 0.8 value, and if we set the threshold around 0.3 we have very good watermark detection.

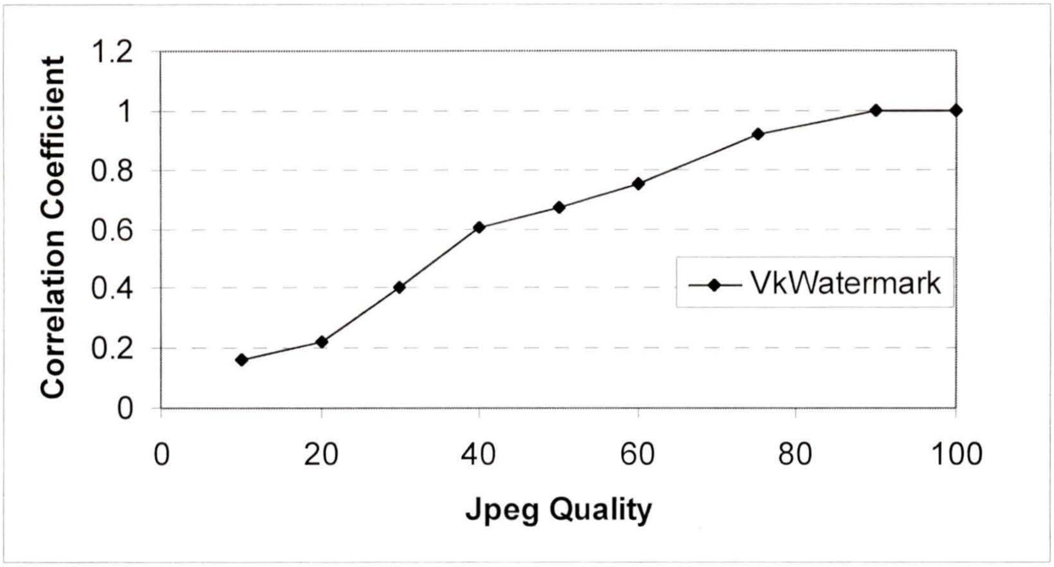


Figure 6-13 VkWatermark detection depending on Jpeg Quality

We have performed watermark insertion on the following standard test images (color and gray scale): f18.ppm, lenna.pgm, benz.ppm, mandrill.ppm

6.6 Conclusion

In this chapter we have addressed the issue of implementation and what are the main components of the framework. The framework is composed from different components that work together to achieve the watermarking utilities. An improved watermarking scheme was proposed to adapt the watermarking used for copyright protection to image searching. The results of the experiments are shown in Section 6.5.1. Two types of experiments were realized: false detection test on uncompressed images and JPEG compression test. The VkWatermark scheme showed an improvement over the Key Watermarking scheme. Also VkWatermark is more suitable for Internet image searching because of the use of two smaller watermarks. The first watermark determines if the image is watermarked or not and the second watermark determines the owner.

7 CONCLUSION

Graphical web sites on the Internet present millions of world wide users with many kinds of design concepts. The major graphic element is the image and as a visual element the image is always what creates the first impression for the users. The image's design function is not just to illustrate information; the image can stand as a visual language itself.

The ever-growing need for retrieving images from large and distributed collections motivates the research for original and sophisticated methods for image retrieval. The results from methods used in watermarking systems, have led us to investigate the feasibility of applying one such technique to image search engines.

In this thesis, we described an integrated WWW image search system named VkMark. Our experimental results show that the integrated system based on watermark retrieval system gives good searching results.

There are several benefits of a watermark based search engine over image database techniques for identifying images. The first is that searches using watermark detection returns results that are more relevant than searches based on color or shape recognition. Many digital watermarks are invariant to scale, changes in color, and image format. A digital watermark is integrated with the image content so it cannot be removed easily without severely degrading the image. Watermarks provide information embedded within the image content that may relate to the owner, license, or tracking of an image. This embedded information may be a code that can be used to identify an image. Another benefit of a watermark search engine is the speed of search and it needs fewer system resources than classic image database search techniques that need to store and process image metadata (color, scale, content, objects, etc.).

In this thesis we propose a framework named VkMark that could be used for generating watermarks, watermarking the images and searching for the images on the web. Instead of searching for image properties, contents, or similarity measures, VkMark searches for the watermark code. The result of finding a matching watermark is the exact image containing that code. If multiple images contain the same watermark (author information), then the set of images containing that code is returned. In image database terms, a query for an image containing an embedded watermark, should yield an exact image match as opposed to "similar" images.

This thesis has presented an improved technique of image watermarking adapted for Internet searching. We have improved the method presented in Chapter 5.3 by two aspects: search speed and robustness. The VkWaterMark improves the search speed by splitting the watermark into two smaller watermarks having lengths N1 and N2 respectively. The first watermark is used as a base watermark to test if the image is watermarked or not. The second watermark is used to determine the "owner" of the image. The watermarks are inserted as many times as possible in the following wavelet transform bands (HL, HH, LH). There is no insertion in low-low (LL) band to conserve the image quality.

The watermark insertion is based on a wavelet transform and is realized in the transform domain. First the image is decomposed with DWT into four bands, i.e., low-low (LL) band, low-high (LH) band, high-low (HL) band, and high-high (HH) band, respectively. The process of decomposition continues a number of times depending on the initial setup. Then the watermark is added using the algorithm presented in section 6.3.2.

This work has also shown that the wavelet transform is useful not only for image compression but also for image watermarking. Chapter 4 presents three examples of watermarking methods based on wavelets. The following discusses several methods of blind watermarking insertion and detection techniques. Because these types of techniques do not need the initial image to detect the watermark, they could be used in image searching techniques.

There are many exciting avenues for extending the work presented in this thesis. We list some of them, including general future research directions, algorithmic enhancements, and extending specific applications:

- add more watermaking schemes

- improve the benchmarking system
- test new wavelet transforms

The successful application of watermarking techniques to image Internet searching will hopefully help it become part of the arsenal of techniques used for Internet image searching.

REFERENCES

[AhRa98] Rakesh Dugad, Krishna Ratakonda and Narendra Ahuja, *"A new wavelet-based scheme for watermarking images"*, ICIP'98

[CoMi97] Ingemar J.Cox, Matt L. Miller, *"A review of watermarking and the importance of perceptual modeling"*, Proc. of Electronic Imaging, February 97

[Cr97] Craver et al. *"Can invisible watermark resolve rightful ownership?"* Proceedings of SPIE, Storage and Retrieval for Image and Video Database, San Jose Feb, 1997.

[Da92] I. Daubechies, *"Ten Lectures on Wavelets."*, CBMS-NSF regional conference series in applied mathematics, no.61, SIAM, 1992.

[DuRa98] Rakesh Dugad, Krishna Ratakonda and Narendra Ahuja *"A New Wavelet-Based Scheme for Watermarking Images"*, ICIP'98.

[FrSw96] C. Frankel, M. Swain, and V. Athitsos, "*Webseer: An Image Search Engine for the World Wide We*b", Tech. Report TR-96-14, Computer Science Dept., Univ. of Chicago, July 1996, http://webseer.cs.uchicago.edu/.

[HiJa94] Michael L. Hilton, D. Jawerth, Ayan Sengupta, *"Compressing still and moving images with wavelets", 1994.*

[HoKu98] Houng-Jyh Wang and C.-C. Jay Kuo, *"Wavelet based blind watermark technique"*, SPIE Symposium on Voice, Video, and Data Communications, Boston, MA, 2-5 November 1998.

[HoWa98] Houng-Jyh Wang, Yi-Liang Bao, C.-C. Jay Kuo and Homer Chen, *"Multi-threshold wavelet codec (MTWC)"*, JPEG 2000 Contribution, ISO/IEC JTC1/SC29/WG1 N819, Convergence Phase Meeting, Geneva Swiss, March 1998.

[InMi98] Hisashi Inoue, Akio Miyazki, Akihiro Yamamotoi, Takashi Katstra, *"A Digital Watermark Based on the Wavelet Transform and its Robustness on Image Compression"*, ICIP'98.

[Ku99] M. Kutter, *"A fair benchmark for image watermarking sytems"*, Proceedings of SPIE: Security and Watermarking of Multimedia, Contents,Volume 3657, pp. 226-239, San Jose, California, January, 1999.

[KuHa97] Deepa Kundur and Dimitrios Hatzinakos *"A Robust Digital Image Watermarking Method using Wavelet-Based Fusion"*, ICIP'97.

[KuHa98] D. Kundur, D. Hatzinakos *"Digital watermarking using multiresolution wavelet decomposition"*, International Conference on Acoustic, Speech and Signal Processing (ICASP), Seattle, Washington, U.S.A., , IEEE, vol. 5, May 1998

[Ma89] Mallat, S. G. *"A Theory for Multiresolution Signal Decomposition: The Wavelet Representation"*, *IEEE Trans. PAMI,* vol. 11, no. 7, July 1989, pp. 674-693.

[MaRa99] Mahalingam Ramkumar, *"Image watermarks counterfeit attacks: some problems and solutions"*, Content Security and Data Hiding in Digital Media, Newark, NJ, May 14 1999.

[NeJo99] Neil F. Johnson, *"In Search of the Right Image: Recognition and Tracking of Images in Image Databases, Collections, and The Internet"*, Technical Report, George Mason University, 1999.

[PeSa96] Amir Said, William Pearlman, *"A New fast and efficient image codec based on Set Partioning in Hierarchical Trees"*, IEEE Transactions on Circuits and Systems for Video Technology, Vol.6, June 1996.

[PiCa99] M. Barni, F. Bartolini, V. Cappellini, A. Lippi, A. Piva, *"A DWT-based technique for spatio-frequency masking of digital signatures"*, in Wong, Delp vol. 3657, San Jose, California, U.S.A., 25–27 The Society for Imaging Science and Technology (IS&T) and the International Society for Optical Engineering (SPIE), January1999.

[Sc95] S. Sclaroff, *"World Wide Web Image Search Engines",* Tech. Report TR 95-016, Computer Science Dept., Boston Univ., Boston, 1995.

[Shap93] J. Shapiro, *"Embedded image coding using zerotrees of wavelet coefficients"*, IEEE Trans. on Signal Processing, Vol.41, No.12, pp.3445--3462, Dec. 1993.

[She96] Sheng, Y. *"Wavelet Transform. The transforms and applications handbook."* Ed. by A. D. Poularikas. P. 747-827.

[SmCh96] J.R. Smith and S.-F. Chang, *"An Image and Video Search Engine for the World Wide Web"*, Proc. SPIE 2670 Storage and Retrieval for Still Image and Video Databases IV, I. Sethi and R. Jain, eds., SPIE, Bellingham, Wash., 1996, pp. 84-95.

[SmCh97] J. R. Smith and S.-F. Chang, *"Searching for Images and Videos on the World-Wide Web,"* IEEE Multimedia Magazine, Summer, 1997. (also Columbia University CU/CTR Technical Report #459-96-25).

[SuSa00] Subhasis Saha, *"Image Compression – from DCT to Wavelets: A Review"*, ACM Crossroads Student Magazine, 2000.

[XiBo97] Xiang-Gen Xia, Charles G. Boncelet, Gonzalo R. Arce *"A Multiresolution Watermark for Digital Images"*, ICIP'97.

[WaWi98] James Ze Wang and Gio Wiederhold, *"WaveMark: Digital Image Watermarking Using Daubechies' Wavelets and Error Correcting Coding"*, Stanford University, 1998.

[WoPo99] Raymond B Wolfgang, Christine I Podilchuk, Edward J Delp, *"Perceptual Watermarks for Digital Images and Video"*, Proceedings of IEEE, July 1999.

GLOSSARY

CWT – Continuous Wavelet Transform

DWT – Discrete Wavelet Transform

MRA – Multiresolution analysis

STFT – Short Time Fourier Transform

lossless compression schemes - compression schemes in which the reconstructed data is identical to the original data

mother function - the transform used to generate the filtered signals from the original data

orthogonality - two vectors u and v in vector space V2 are orthogonal if and only if u.v = 0 (that is, the dot product of the two vectors is 0)

quality - the difference between the reconstructed data and the original

quantization - the process of representing a large, or even infinite set of values with a smaller, finite set of values

rate - the average number of bits required to represent a single sample, where a sample can be a symbol in a text message, a pixel in an image, or a wave sample for audio data

redundancy of a message - the relative redundancy of a message is defined as Rr = 1 - 1/Cr where Cr is the compression ratio

analyzing wavelet - see mother function

basis - two vectors in a vector space from which all other vectors in the vector space can be expressed, using a linear combination of the two vectors e.g. ai + bj where i and j are vectors

compression ratio - the amount of compression achieved; it is defined as Cr = m1 / m2 where m1 is size of the compressed message, or data and m2 is the size of the original message

detail coefficients - values stored at each application of a wavelet transform that allow the original data to be reconstructed

distortion - the difference between the original data and the reconstructed data after compression (only applies to lossy compression techniques)

filtering - isolating certain components of the function

lossy compression schemes - compression schemes in which data may be lost between the encoding and reconstruction steps

EZW - the embedded zerotree wavelet

LZC - the layer zero coder

MTWC –Multi-Threshold Wavelet Codec

SPHIT - method of set partitioning in hierarchical tree

DCT – Discrete Cosine Transform

VITA

Surname: Toaxen	Given Names: Eugen Solomon

Place of Birth: Bucharest, Romania

Educational Institutions Attended:

University of Victoria	1998 to 2001
"Politehnica" University of Timisoara	1992 to 1998

Degrees Awarded:

M.Eng.	"Politehnica" University of Timisoara	1998
B.Sc.	"Politehnica" University of Timisoara	1997

Honours and Awards:

University of Victoria Graduate Teaching Fellowship	1998 to 2000

Publications:

E.Toaxen, V.Cretu, V.Groza, *"A Simulator for Validation of Distributed Real-Time Systems"*,3-rd International Conference on Technical Informatics CONTI '98, Scientific and Technical Bulletin of Technical University of Timisoara, Computers and Automation, Volume 43(57), 1998, pp(204-215), ISSN 1224-600X.

E.Toaxen, *"A Simulator for Validation of Distributed Real-Time Systems", Master Thesis, 1998*

UNIVERSITY OF VICTORIA PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis (or dissertation) to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain by the University of Victoria shall not be allowed without my written permission.

Title of Thesis/Dissertation:

An Image Searching Tool Using Wavelet Watermarking

Author [redacted]
Eugen Solomon Toaxen
April 30, 2001