" المستخلص "

في هذه الأطروحة تم اقتراح طريقة لإخفاء العلامة المائية في الصور الملونة. حيث تتطلب وجود الصورة الأصلية عند استخلاص العلامة المائية. وقد استثمرت هذه الطريقة مُميزات التحويلات المويجية من خلال استثمار المعاملات الكبيرة في الحزم الثانوية للترددات العالية كمضيف لإخفاء العلامة المائية الغير مرئية.

وتم استثمار تقنية التقسيم الجزئي في توليد العلامة المائية بطريقة التوليد الذاتي من خلال شرط أولي باستخدام تقنية إزاحة النقطة الوسطى والاستفادة من خواص هذه التقنية (باستخدام ابعاد صورة التقسيم الجزئي المحصورة بين البعد الواحد و البعدين) من اجل زيادة امن الصورة وقوتها عند التعرض للمهاجمة المتعمدة والغير متعمدة.

ممكن استخدام هذه الطريقة لأغراض تعريف المالك واثبات الملكية وتحديد شرعية المستخدم و أيضا قادر على الصمود ضد درجة عالية من الضغط الناتج من البرنامج القياسي JPEG2000.

وهذه الطريقة ارتكزت على استخدام خطوات تحليلية لإيجاد التأثيرات الحاصلة في قيم المعاملات عند تعرض الصورة المضيفة إلى الضغط وقبل إجراءات إخفاء العلامة المائية وهذا يفيدنا في حساب العامل التحليلي لإخفاء العلامة المائية وفي نفس الوقت يجب أن تحافظ على الصمود إلى مستوى معين من الضغط. حيث تصل نسبة ضغط الصورة إلى (١:٣) الصورة الأصلية. بالرغم من هذا النقصان الحاصل في الحجم إلا أن عدد الإعداد الثنائية جدا قليل أو معدوم والعلامة المائية قابلة للاسترجاع.

عملياً، طبق النظام على مدى واسع من الصور واخضع على مقاييس عالمية، حيث أظهرت إن عدد الأعداد الثنائية المشوهة تزداد بزيادة درجة الضغط بالتالي نوعية الصورة يقل. باستخدام طريقة إزاحة النقطة الوسطى نستطيع توليد صور ذات أحجام متعددة ومواصفات عالمية قياسية.
تم استخدام النموذج اللوني المناسب لتطبيق عملية إخفاء العلامات المائية في الصور وهو (RGB).

ذو الحجة – ١٤٢٦ هـ

Abstract

In this thesis, a method for color image watermarking is suggested. It requires the original image for watermark extraction. This method exploit the discrete wavelet transform features, where the large coefficients in high frequency sub bands are used as host place to embed the invisible fractal watermark image using the Haar transform to embed the huge data.

The proposed system exploit the fractal image technique using the midpoint displacement method fractal technique (By using the H-dimension as entered real number) to more secure and preserve the watermark from intentional or unintentional attacker.

The system can be used for owner identification, proof of ownership, and transactional watermarks (fingerprinting) also it is capable to survive against the high degree of compression produced by JPEG2000. By using Haar transform we can embed the fractal watermark image with modulation factor value to satisfy robustness against compression.

The fractal watermarking image was fully extracted under degree of compression that makes the size of image decrease into (1:3) of the original size. In spite of this decreasing in the size of image the distortion bits is very little or underprivileged and the image is capable to survive the extraction.

To be practical, the system was tested by using various images evaluated by adopting many fidelity measurement, the test results indicated that the distortion bits is increased when the compression size decreases, and the quality of image also decreases. The system was incomplete and robust watermark, then it considered as a private watermark.

By using midpoint displacement method can generate image of any size with mean equal to zero and standard deviation equal to one. The proposed system uses RGB color space.

General Introduction

1.1 Introduction

In recent years, the distribution of works of art, including pictures, music, video and textural documents, has become easier. With the widespread and increasing use of the internet, digital forms of these media (still images, audio, video, text) are easily accessible. This is clearly advantageous in that it is easier to market and sell one's work of art, at same time it threatens copyright protection [Ali04].

Digital documents are easy to copy and distribute, allowing for pirating. Another problem with digital document and video is that undetectable modifications can be made with very simple and widely available equipment, which put the digital material for evidential purposes under question [Jam02]. Therefore, there are a number of methods for protecting ownership. One of these is known as digital watermarking.

Digital watermarking is the process of inserting a digital signal or pattern (indication of the owner of the content) into digital content. The signal, known as a watermark, can be used later to identify the owner of the work, to authenticate the content and to trace illegal copies of the work [Cox02].

While the cryptographic techniques provide secrecy for the communications by scrambling a message that cannot be understood, a cryptographic message can be intercepted by an eavesdropper because the encrypted message brings suspicion especially in military communications. So, there is need for embedding data in suggested way that make the input number as a fractal image by Midpoint Displacement Method as a watermark to embed in the cover image and should be invisible to a human observer and doesn’t make any suspicion [Jam02].

1.2 Problem Statement and Goal of Watermarking

The desire for the availability of information and quick distribution has been a major factor in the development of new technology in the last decade. The increased use of multimedia across the internet, multimedia distribution has become an important way to deliver services to people around the world. It is commonly applied in Internet marketing campaigns and electronic commerce web sites [Cha00].

Due to the growing usage of multimedia content on the Internet, serious issues have emerged. Counterfeiting, forgery, fraud, and pirating of this content are rising. Virtually anyone with a sound card, scanner, video frame grabbers, or multimedia authoring systems allow them to incorporate copyrighted material into presentations, web designs, and Internet marketing campaigns. Consequently, copyright abuse is extensive among multimedia users who are rarely caught [Isa04].

This copyright abuse is the motivating factor in developing new encryption technologies. One such technology is digital watermarking. A digital watermarking is a digital signal or pattern inserted into digital content (The digital content could be a still image, an audio clip, a video clip, a text document, or some form of digital data that the creator or owner would like to protect). The main purpose of the watermark is to identify who is the owner of the digital data, but it can also identify the intended recipient [Ali04].

1.3 Watermarking History

Paper watermarks appeared in the art of handmade papermaking nearly 700 years ago. The oldest watermarked paper found in archives dates back to 1292 and has its origin in the town of Fabriano in Italy, which has played a major role in the evolution of the papermaking industry. At that time paper mills produced raw paper with very coarse surfaces not yet suitable for writing. The introduction of watermarks was the perfect method to eliminate any possibility of confusion. Watermarks quickly spread in Italy and then over Europe and although initially used to indicate the paper brand or paper mill, they later served as indication for paper format, quality, and strength, and were also used as the basis for dating and authenticating paper [Ali04]. With the invent of the digital computers and the widespread and increasing use of the internet a new life was given to the watermarking system known as the digital watermarking system.

1.4 Watermarking Terminology

Over the years, researches have coined numerous terms to describe and classify watermarking techniques. We clarify these terms in this section.

The image into which we are hiding the information is called the *host* or *cover data*, and the hidden information is referred to as the *payload* [Cha00].

Most image watermarking systems involve making *imperceptible* alterations to the host image to convey the hidden information, but there also exist *visible* watermarks, which are visible patterns (like company logos) overlaid on top of an image [Cox02]. However, in this thesis, we will concentrate on invisible watermarks and it should be *imperceptible* to refer to *invisible* watermarks.

If the original, *unwatermarked* image is required in order to retrieve the watermark, the system is known as *non-blind* or *non-oblivious*, otherwise it is known as *blind* or *oblivious*.

Watermarking system typically require the use of a key (like that used in the cryptographic sense) for retrieving the embedded *watermark*. If the *same* key as in the watermark embedder must be used for retrieving the watermark, the scheme is known as *private*, because only the person who has the key can read the watermark. If a different key is needed to read the watermark, the scheme is known as *public*. Public watermarking is sometimes also known as *asymmetric* watermarking [Cox02].

Watermarking system can be *robust* or *fragile*. Robust watermarks are required to resist any modifications that do not decrease the commercial value of the cover image. On the contrary, fragile watermarks are designed to *fail* when the cover image is modified.

Fingerprinting and *labeling* refer to specific applications of watermarking. Labeling is similar to the use of serial numbers to identify an individual copy of a product. Fingerprinting can be used to trace the origin of a piece of data if unauthorized copies of it are found [Cox02].

1.5 Relationship of Watermarking with Compression

There exist a duality between watermarking (information hiding in general) and data compression. While compression aims to identify the perceptually insignificant parts of the data and remove them, watermarking technique try to insert information into them. Moreover, compression is one of the most common operations on images; therefore one must take into account the effects of compression when designing a watermarking system.

The most common compression standard during last few years is JPEG, which is based on discrete cosine transform (DCT). JPEG2000, which operates in the wavelet domain, was proposed in 2000 as a future standard of compression, and the experiments showed that it has superior performance compared with JPEG [Ali04, Lur03].

1.6 Digital Watermarking with Images

The solution to the problem of protecting ownership and authenticity of digital images is explored through digital watermarking. This process attempts to add some small digital structure to a host image that cannot be perceived or removed unless by the owner. Therefore a large body in research has focused on techniques that embed an imperceptible symbol to the original media that will “mark” ownership of it and authenticate it.

Then we should know that watermarking is not similar to encryption (see section 2.2), therefore the asserted digital watermark must be:

1. Visually imperceptible within the host image so that there is no visual interference.
2. Discreet or statistically invisible to prevent unauthorized removal.
3. Easily extracted and unambiguously identify the owner.
4. Robust to incidental and intention distortions of the watermarked image including:
    a. Signal processing manipulations such as compression.
    b. Geometric distortions that include rotation, translation, cropping, and scaling.
    c. Subterfuge or attacks to change or destroy watermark for collusion or forgery [Ali04, Jam02].

1.7 <u>Literature Survey</u>

Different techniques were developed and appeared in literature that attempt to meet all possible criteria both successfully and optimally. Various technique are presented for digital watermarking, primarily focusing on still images. This survey allows gaining an understanding of basic watermarking ideas from published research, applications and needs for watermarking in real work.

1- Kundur, [Kun97], had utilized a multi-level wavelet decomposition combined with a Human Visual System (HVS). The authors assert that this process merges the watermark and image more robustly and less perceptibly. The use of wavelet decomposition has the property of space-frequency localization, which allows greater resilience from intentional distortions, i.e. the watermark can still be recovered. The overall process is shown in figure (1.1).

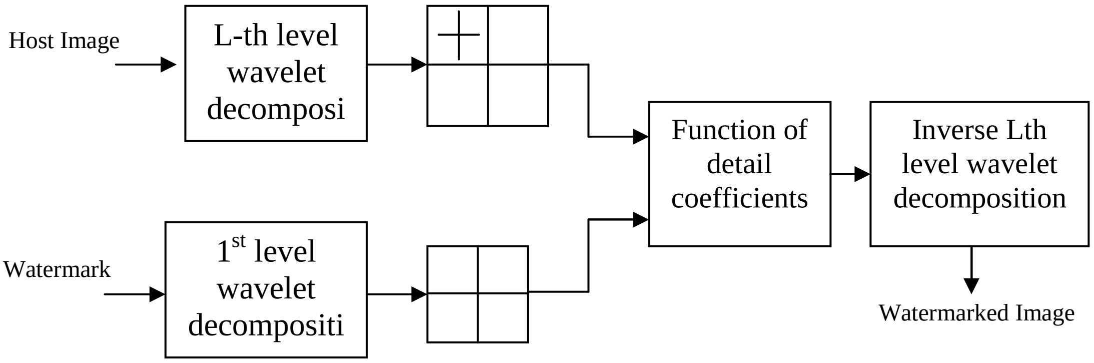

Host Image
L-th level wavelet decomposi
Watermark
1st level wavelet decompositi
Function of detail coefficients
Inverse Lth level wavelet decomposition
Watermarked Image

Figure (1.1): Watermarking method proposed in Kundur, 1997.

The process uses an L-th level discrete wavelet transform on the image, and a first level DWT on the watermark. The merging process begins by computing the salience, or a numerical measure of visual importance of an image component, for each of the DWT detail coefficients.

2- Kundur, [Kun98], had used a similar approach in [Kun97] but does not use an HVS approach. Instead the authors consider a more practical approach, which doesn’t require the original image to extract the watermark. The general overview of this process is shown in Figure (1.2).

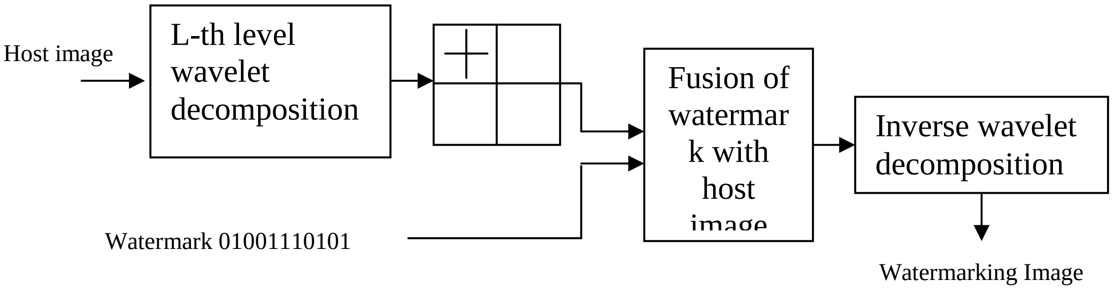

Host image
L-th level wavelet decomposition
Fusion of watermar k with host image
Inverse wavelet decomposition
Watermark 01001110101
Watermarking Image

Figure (1.2): Watermarking method proposed in Kundur, 1998.

The method again begins with an L-th level DWT, however this time the watermark is kept simply as binary values, and a random key governs the embedding process of the watermark. If, for a particular coefficient location in the DWT domain, the key has a bit value one, then the three DWT detail coefficients at that location are ordered in ascending order. Then, the middle-valued detail coefficient is quantized to a particular bin depending on whether the watermark bit value is high or low, the inverse DWT is taken and the image is watermarked.

3- Issar, [Issar04], he toke into account the fact that the Human Visual System (HVS) is not sensitive to small changes in high frequencies of the image. Therefore, the mark is embedded into the wavelet domain, thus making it imperceptible to the human eye. That places the mark into the HH, LH and HL subbands, selecting only part of these coefficients, leaving the LL subband unmodified, as shown in figure (1.3):

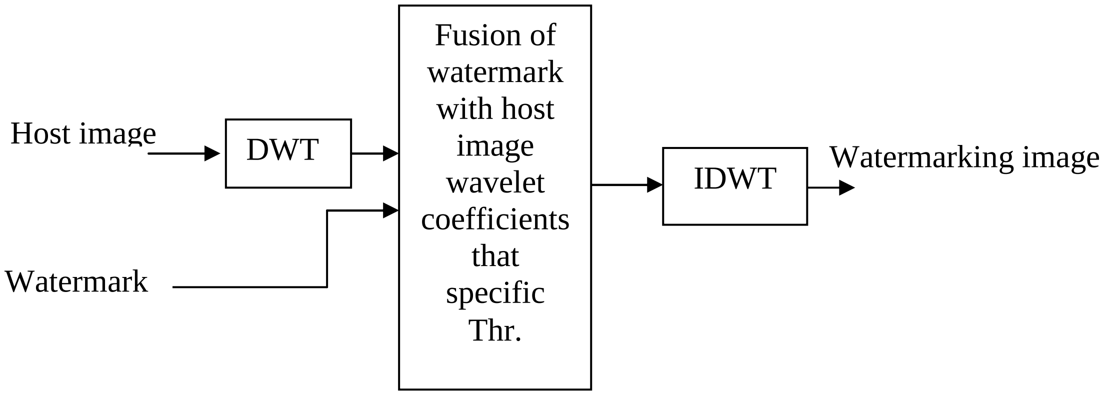

Host image
DWT
Watermark
Fusion of watermark with host image wavelet coefficients that specific Thr.
IDWT
Watermarking image

Figure (1.3): Watermarking method proposed in Issar,2004.

4- Ali Kadhim, [Ali04], he suggested three methods that don't require the original image for watermark extraction, these method exploit the discrete wavelet transform features. The first suggested method can be used for owner identification, proof of ownership and transaction watermarks (fingerprinting). The second suggested method was oriented to embed the watermark such that it is capable to survive against the highest degree of compression produced by JPEG2000. The watermark was fully extracted under high degree of compression ratio (1:19). The third suggest method is the covert communication method (a method of sending a secret messages), a general method is developed for embedding and extracting hidden message without need to know the key, with absence of the original image. Figure (1.4) illustrates the general layout of this method.

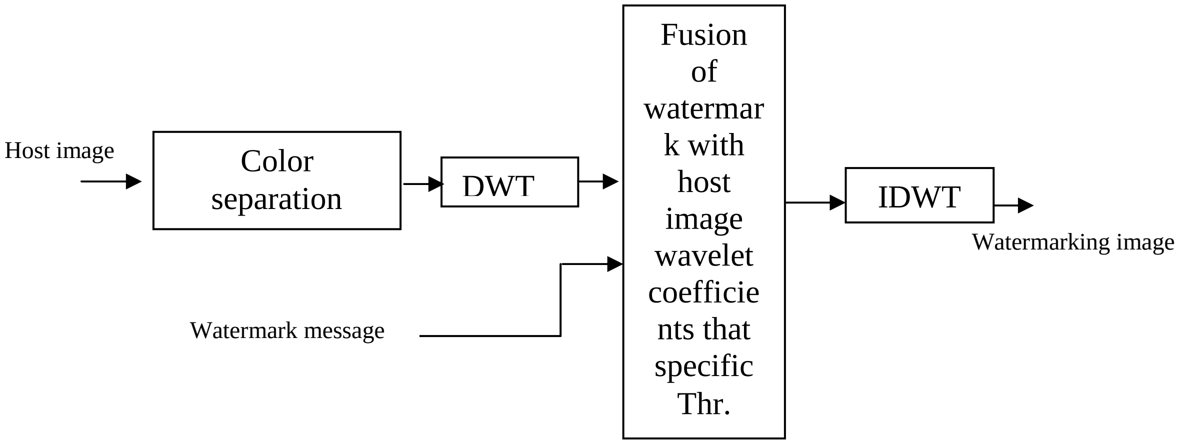

Host image
Color separation
DWT
Fusion of watermark with host image wavelet coefficients that specific Thr.
IDWT
Watermarking image
Watermark message

Figure (1.4): Watermarking method proposed in Ali Kadhim, 2004.

1.8 <u>Aim of the Work</u>

The objectives of this work is to build and apply efficient algorithm for watermarking color still images by a fractal image generated through Midpoint displacement method, which require the original still image through watermark extraction. This algorithm has to be robust against JPEG2000 compression distortions, and also has to be secure and imperceptible.

1.9 Thesis Layout

Chapter 1 and 2 of this thesis provide the introductory material.

Chapter 2**:** describe the conceptual review of the watermark, relationship of information hiding with watermarking, watermarking system, purpose of digital watermarking, watermarking application, watermarking requirements, evaluation and benchmarking of watermarks, the aim of digital watermarks, the classification of watermarking system, resistance to attacks.

Chapter 3: is dedicated to present an overview to the wavelet transform, fractal and JPEG2000 that provide the Introduction to Wavelet Transform, Wavelet Analysis, Fourier Analysis, Haar Wavelet Transform, properties of the Wavelet Transform, Fractals - a definition, characteristics of fractal, different Fractals, Euclidean and Fractal Geometry, Fractal Dimension, The use of JPEG2000, the Application of JPEG2000, Technical Description.

Chapter 4: provide the proposed digital watermarking algorithm, random midpoint displacement method, embedded module, extraction module, practical investigation.

Chapter 5: provide the conclusion and future work

Wavelet, Fractal and JPEG2000 - Conceptual Review -

3.1 Introduction

The choice of embedding locations in the host image should be concerned with the human eyes that less sensitive to noise in regions of edge and textures than in smooth areas; Cox et al. [Cox97], argues that one should embed watermarks in perceptually significant parts of an image, which survive to compression. Numerous approaches have been proposed which choose embedding locations based on this principle.

In addition, one can also select the locations using a key, usually a random number seed, to choose the coefficients to be marked [Isa04].

The best way to define and describe why the selection of a fractal is through its attributes: a fractal is “rugged”, which means that it is nowhere smooth (i.e. the Digital fractal watermarking ambiguous to the attacker what it mean), it is “self-similar”, which means that parts look like the whole, it is “developed through iterations”, which means that a transformation is repeatedly applied and it is “dependent on the starting conditions”. Another characteristic is that a fractal is “complex”, but nevertheless it can be described by simple algorithms - that also means that beneath most natural rugged objects there is some order [Wol04].

3.2 Introduction to Wavelet Transform

Everywhere around us are signals that can be analyzed. For example, there are seismic tremors, human speech, engine vibrations, medical images, financial data, music, and many other types of signals. Wavelets have scale aspects and time aspects; consequently every application has scale and time aspects. To clarify them we try to untangle the aspects somewhat arbitrarily.

For scale aspects, we present one idea around the notion of local regularity. For time aspects, we present a list of domains. A wavelet is a waveform of effectively limited duration that has an average value of zero [Mat00].

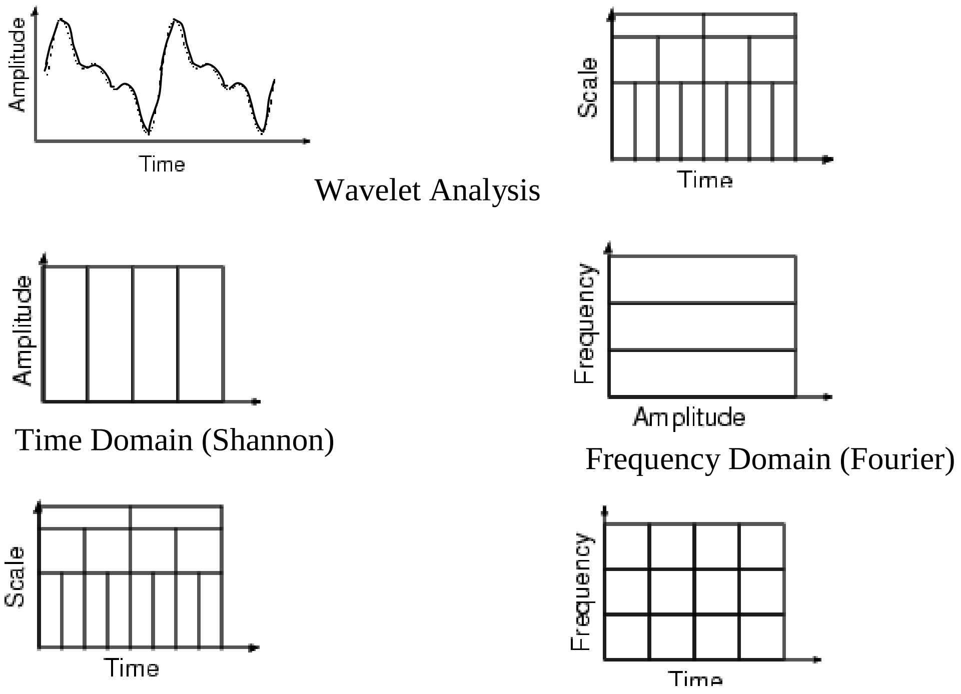
Wavelet has advantage over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Interchanges between these fields during the last ten years have led to many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction [Gra95].

3.3 Wavelet Domain

Wavelet is becoming a key technique in the ongoing source compression standard JPEG2000. The positive arguments closely resemble those for advocating DCT for JPEG (i.e. preventing watermark removal by JPEG2000 lossy compression, reusing previous studies on possibility of embedding in the compressed domain). In addition to those criteria, the multiresolution aspect of wavelets is helpful in managing a good distribution of the watermark in the cover in terms of robustness versus visibility [Kat00].

3.4 Wavelet Analysis

Wavelet analysis represents the next logical step: a windowing technique with variable-sized regions. Wavelet analysis allows the use of long time intervals where we want more precise low-frequency information, and shorter regions where we want high-frequency information; we can compare the analysis of Shannon, Fourier, Gabor and Wavelet as shown in figure (3.1).



Wavelet Analysis

Time Domain (Shannon)

Frequency Domain (Fourier)

Wavelet Analysis

STFT (Gabor)

***Figure (3.1): Compare the description of Shannon, Fourier, Gabor and Wavelet analysis**.*

You may have noticed that wavelet analysis does not use a time-frequency region, but rather a time-scale region. Compare wavelets with sine waves, which are the basis of Fourier analysis, as shown in figure (3.2).

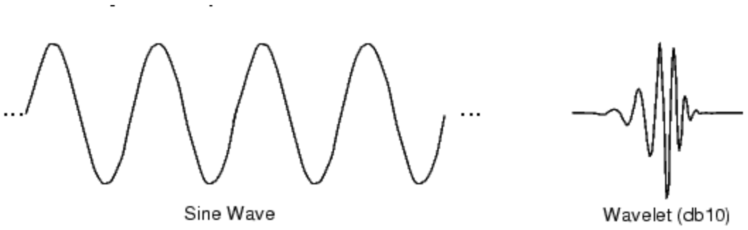


Figure (3.2): Sine and Wavelet wave.

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet.

3.5 Fourier Analysis

Fourier breaks down a signal into constituent sinusoids of different frequencies. Another way to think of Fourier analysis is as a mathematical technique for transforming our view of the signal from time-based to frequency-based.

For many signals, Fourier analysis is extremely useful because the signal's frequency content is of great importance. But it has a serious drawback. In transforming to the frequency domain, time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell when a particular event took place. However, most interesting signals contain numerous no stationary or transitory characteristics: drift, trends, abrupt changes, and beginnings and ends of events. These characteristics are often the most important part of the signal, and Fourier analysis is not suited to detecting them. Gabor's adaptation, called the Short-Time Fourier Transform (STFT) maps a signal into a two-dimensional function of time and frequency. The STFT represents a sort of compromise between the time- and frequency-based views of a signal. It provides some information about both when and at what frequencies a signal event occurs. However, you can only obtain this information with limited precision, and that precision is determined by the size of the window [Mat00]. The embedding capacity in the wavelet domain is greater than in DC domain [Ali04].

3.6 Wavelet Transform: Continuous and Discrete

The continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function ψ :

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\psi(scale, position, t)dt \qquad \text{..... (3.1)}$$

The results of the CWT are many wavelet coefficients C, which are a function of scale and position. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelet of the original signal.

The basic idea of the Discrete Wavelet Transform (DWT) for a one-dimension signal is the following. A signal is split into two parts, usually high frequencies and low frequencies. The edge components of the signal are largely confined in the high frequency part. The low frequency part is split again into two parts of high and low frequency. This process is continued until the signal has been entirely decomposed or stopped before by the application at hand. For compression and watermarking application, generally no more than five decomposition steps are computed. Furthermore, from the DWT coefficients, the original signal can be reconstructed. The reconstructed process is called the Inverse Discrete Wavelet Transform (IDWT).

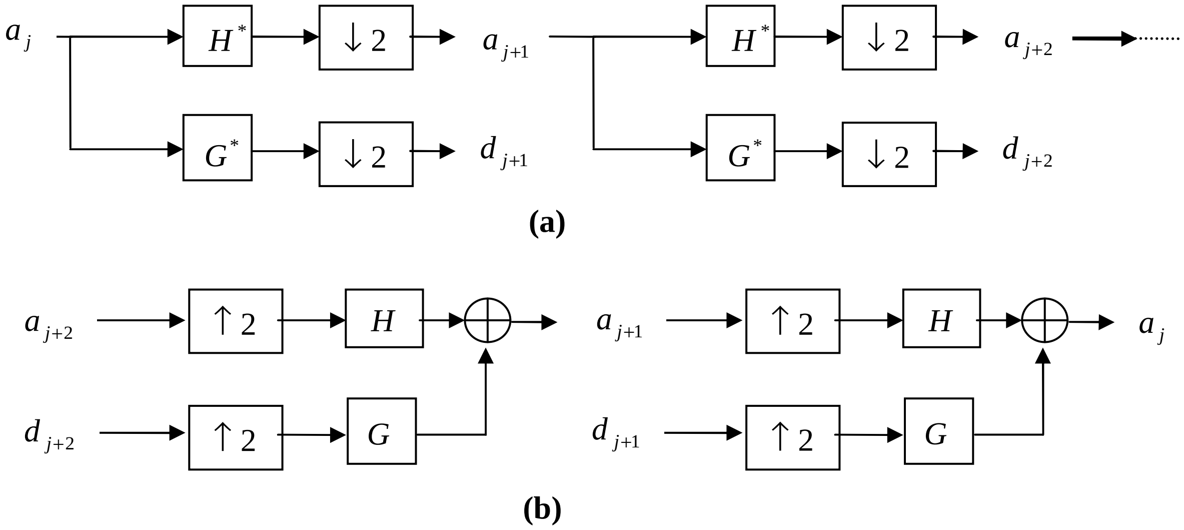


Figure (3.3): The (a)DWT and the (b) IDWT of 1D signal

The DWT for two-dimensional signals, like images, is similar to the DWT for one-dimensional signals. The difference is that one has to implement separately for each dimension the DWT and IDWT respectively. The image will be decomposed for each resolution level into a high-high (HH), high-low (HL), and low-high (LH) sub band, and a low-low (LL) sub band for the coarsest resolution level. The LL band is also known as the approximation sub image because it contains most of the information from the image. The HL, LH, HH sub bands are the detail sub images containing the horizontal, vertical and diagonal details (see figure 4 and 5).

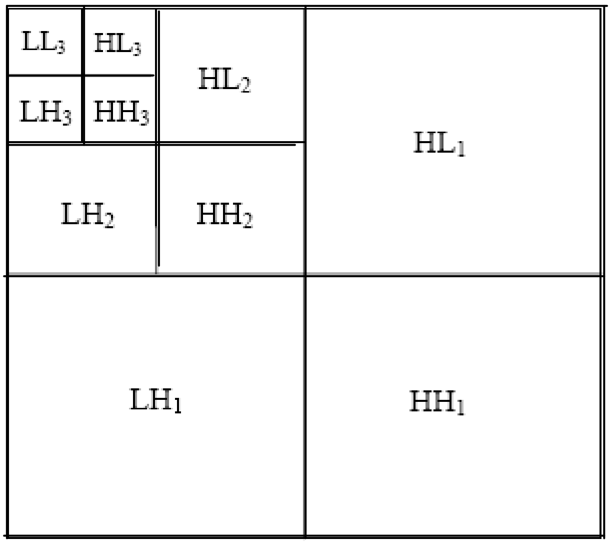


Figure (3.4): DWT pyramid decomposition of an image for three resolution levels.

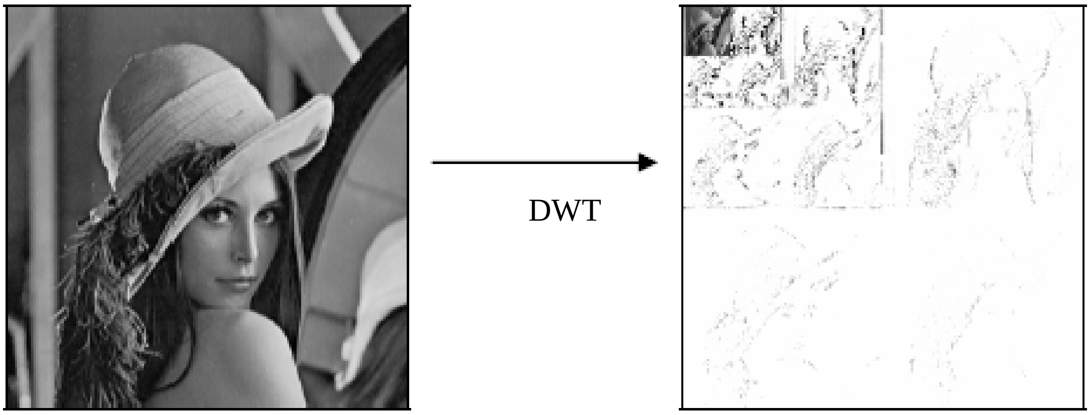


Figure (3.5): Example of a multiresolution decomposition for images.

3.7 The Haar Transform

The Haar transform uses a scale function $\phi(t)$ and a wavelet $\psi(t)$, to represent a large number of functions $f(t)$. The representation is the infinite sum [Sal00]

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \phi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{j,k} \psi(2^j t - k) \quad \ldots\ldots (3.10)$$

Where c_k and $d_{j,k}$ are the coefficients to be calculated.

The basic scale function $\phi(t)$ is time unit pulse

$$\phi(t) = \begin{cases} 1 & 0 \le t < 1 \\ 0 & otherwise \end{cases} \quad \text{….. (3.11)}$$

The function $\phi(t-k)$ is a copy of $\phi(t)$, shifted k units to the right. Similarity $\phi(2t-k)$ is a copy of $\phi(t-k)$ scaled to half the width of $\phi(t-k)$. The shifted copies are used to approximate $f(t)$ at different times t. The scaled copies are used to approximate $f(t)$ at higher resolutions and j represent the interval time basic Haar wavelet is the step function

$$\psi(t) = \begin{cases} 1 & 0 \le t < 0.5 \\ -1 & otherwise \end{cases} \quad \text{….. (3.12)}$$

From this we can see that the general Haar wavelet $\psi(2^j t-k)$ is a copy of $\psi(t)$ shifted k unites to the right and scaled such that its total width is $1/2^j$.

There are two main types of transforms, *orthogonal and subband*. An orthogonal linear transform is done by computing the *inner product* of the data (pixel values) with a set of *basis functions*. The result is a set of transform coefficients. There are several examples of important orthogonal transforms, such as the DCT.

The other main type of transforms is the *subband transform*. The Haar transform is *subband transform*. It is done by computing a convolution of the data with a set of *bandpass filters*. Each resulting subband encodes a particular portion of the frequency content of the data.

The principle of the Haar transform is to calculate averages and differences. It partitions the image into regions such as that one region contains large numbers (averages), and the other regions contain small numbers (differences).

However, these regions, which are called subbands, are more than just sets of large and small numbers. They reflect different geometrical artifacts of the image. Figure 3.6 (Left Part) shows an 8 x 8 image whose pixel values are 12, except for a vertical line with pixel values of 14 and a horizontal line with pixel values of 16.

Figure 3.6 (Right Part) shows the results of applying the Haar transform. The upper-right subband now contains traces of the horizontal line, whereas the lower-left subband shows traces of the vertical line. These subbands are denoted by HL and LH, respectively. The lower-right subband denoted by HH reflects diagonal image artifacts. The most interesting is the upper-left subband, denoted by LL that consists entirely of averages. Here, we have implemented one stage of the forward and inverse discrete wavelet transform using Mallat's fast wavelet transform algorithm. The 2 x 2 convolution kernels used are:

$$h_0(x)h_0(y) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad h_1(x)h_0(y) = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$h_0(x)h_1(y) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \qquad h_1(x)h_1(y) = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

This subband is a one-quarter version of the entire image, containing traces of both the horizontal and the vertical lines.

12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
16	16	16	16	14	16	16	16
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12

24	24	26	24	0	0	2	0
24	24	26	24	0	0	2	0
28	28	28	28	0	0	0	0
24	24	26	24	0	0	2	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-4	-4	-2	-4	0	0	2	0
0	0	0	0	0	0	0	0

Figure (3.6): An 8 x 8 image and its subband decomposition

The method of averaging and differencing can also be expressed as *filtering* the data. Averaging corresponds to a low pass filtering. It removes high frequencies of the data. Since details (sharp changes in the data) correspond to high frequencies, the averaging procedure tends to smooth the data. The low pass filter can be expressed as $\frac{1}{\sqrt{2}}(1,1)$ in the Haar case and when we average the data; we move this filter along our input data.

The differencing corresponds to high pass filtering. It removes low frequencies and responds to details of an image, since details correspond to high frequencies. The high pass filter can be expressed as $\frac{1}{\sqrt{2}}(1,-1)$ in the Haar case, in which we take the difference of the data; we simply move this filter along our input data. The low pass and high pass filters make up what in signal processing language is referred to as a *filter bank*. The method of averaging and differencing is referred to as analysis. The reverse procedure (going in opposite way) is called *synthesis*.

A standard decomposition of a two dimensional image is easily done by first performing a one-dimensional transformation on each row followed by a one-dimensional transformation on each column.

3.8 Properties of the Wavelet Transform

The wavelet transform has a number of advantages over other transforms: [Cox97]

1. The wavelet transform is a multi-resolution description of an image, the decoding can be processed sequentially from a low resolution to a higher resolutions.
2. The wavelet transform is closer to the human visual system than the DCT. Hence, the artifacts introduced by wavelet domain coding with high compression ratio are less annoying than those introduced at the same bit rate by the DCT. Additionally- in the JPEG case- block shaped artifacts are clearly visible, since image coding based on the DCT usually operates on independent (8 x 8) blocks.
3. The wavelet transform generates a data structure known as scale-space representation. In this image representation, the high frequency signals are precisely located in the pixel domain, while low-frequency signals are precisely located in the frequency domain.

   The spatial resolution of the wavelet transforms increases with frequency. Therefore sharp edges, which are localized spatially and have a significant high-frequency content, can be seen in the detail subbands and form the contours of the image's objects. While the frequency resolution is independent of the frequency in the DCT domain, it is inversely proportional to frequency in the wavelet domain.

Barni [Bar99], Dugard [Dug98] and other authors identified several advantages which can be exploited by watermarking schemes operating in the wavelet transform domain:

1. The hierarchical image representation, due to the multi-resolution characteristics of the transform, is especially suitable for applications where the image is transmitted progressively, where large amounts of data have to be processed, such as in video application, or for real-time systems. Watermarking algorithms that embed a hierarchical or nested watermark can save a lot of computational effort when the mark can be detected early in a progressive transmission. They have to reset to the higher resolution subband only when the watermark could not be detected or extracted from the subbands analyzed previously.
2. The Wavelet domain allows superior modeling of the Human Visual System (HVS). It is closer to the hypothetical Cortex transform than the DCT, since it splits the signal into individual bands that can be processed independently. Moreover, the visibility of Wavelet quantization noise and the possibilities of visual masking in the Wavelet domain have been extensively studied.
3. While the high resolution subbands allow locating image features such as edges or textured area easily in the transform domain. Watermarking schemes often put more watermark energy into large DWT coefficients, thus affecting regions with high contrast, like edges and texture, to which the HVS is not sensitive. This is just one

example of implicit masking that can be easily exploited in the Wavelet domain.

4. The Wavelet transform is computational efficient and can be implemented in a variety of ways, e.g. by means of filter convolution.

3.9 Fractals - A Definition

The word fractal comes from the Latin adjective fractus which means broken. A more scientific approach of the term fractal is that it is a fragmented mathematical geometric shape that can be subdivided into smaller pieces that represent a reduced-size copy of the whole that is in contrast to fractal objects such as mountains and coastline [Batty94].

"Fractals are objects of any kind whose spatial form is nowhere smooth, hence termed "irregular", and whose irregularity repeats itself geometrically across many scales", and the fractal as the B. Mandelbrot definition is *a rough or fragmented geometric shape that can be subdivided in parts, each of which is (at least approximately) a reduced/size copy of the whole.*

Mathematical, *the fractal is a set of point whose fractal dimension exceeds its topological dimension.* It is fairly common in nature to notice objects that do not have well-defined geometric shapes, but appear to be constructed according to some simple mathematical rules, example are found in mountains, coastline, volcanic, seashells, plant and clouds [Wol04].

3.10 Characteristics of fractal

The world is chaotic, discontinuous, irregular in its superficial physical form but… beneath this first impression lies an order which is regular, unyielding and of infinite complexity [Bat94].

1. A Fractal is Rugged:

a. Coastline

Benoit Mandelbrot introduced fractal geometry by the question of how long the coastline of Britain is.

b. Border

There are again limits as with the coastlines above - like the definition of the border, that is defining its way and the limit by the smallest and largest scale.

c. Richardson

As early as in the year 1961, Lewis Fry Richardson examined the growth rate of the length for different curves such as coastlines and borderlines.

2. A Fractal is Self-Similar:

Many objects and patterns in the real world possess the property of self-similarity:" *no matter what scale is used to view the pattern, the magnified portion of the shape looks like original pattern*”. Such objects include mountains and coastlines, as well as several classes of pattern derived purely from mathematics. In general terms, an object is spatially self-similar if it appears, exactly or statistically, same under different levels of magnification. Any structure is self-similar if it has under-gone a transformation in which the proportions of the structure have all been modified by the same scaling factor. The new shape may be smaller, larger, rotated, and/or translated, but its shape remains similar, which means that the relative proportions of the shapes’ sides and the internal angles remain the same [Bov96] - these transformations can be produced by a reduction-copy machine as shown in figure (3.7). Fractals produced by self-similar transformations are “true” fractals; the underlying algorithm is the same from scale to scale - zooming into such a fractal shows an object, which is the same as the whole. These "perfect" fractals may produce objects which look similar to nature, but there is mostly something missing - the factor of random. Nevertheless they can be used as a first approach to nature instead of Euclidean objects- remembers: “*... trees are no spheres*”.

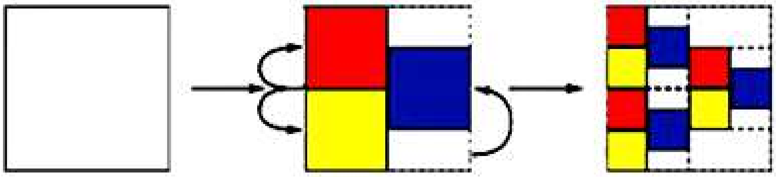


Figure (3.7): The reduction …

The reduction of a paper in a copy machine is a similarity transformation.

If a transformation reduces an object unequally in one or the other way, then the transformation is referred to as a self-affine transformation. In a self - affine transformation the internal angles of the shape and/or the relative proportions of the shape's sides might not remain the same - these curves are not exactly self-similar [Man97].

One example for an early attempt of using self-similarity in architecture is the floor plan of the Tadsch Mahal in Agra/India. The Hindu Temple “Rajarani” in Bhuvanesvar also shows some characteristics of fractals as shown in figure (3.8).

Figure(3.8): The “Rajarani”

In the picture of the Pokrov cathedral in Moscow the prominent element turns out to be the bulb-shaped dome. This element is transformed in size and position from one stage to the next, which is rendered prominent through the colors yellow, orange and red. But in addition to that there are also other forms that are repeated on different scales, as shown in figure (3.9).

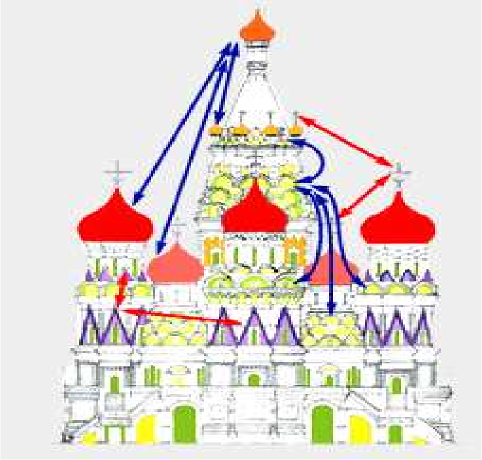


Figure (3.9): The "Pokrov cathedral"

3. A Fractal is Infinitely Complex:

Fractals are highly complex, that means zooming in will bring up more and more details of the object, a characteristic that continues until infinity. It was only in the 70ies that Mandelbrot could show the results of the formula as a picture, which needed the high capacity of computers. The Mandelbrot set is similar from scale to scale, which means zooming closer to the details there will always come up new parts looking similar to each other and sometimes to the whole - see figure (3.10). The only limits are limits of capacity and, rounding mistakes by the computer [Man97].

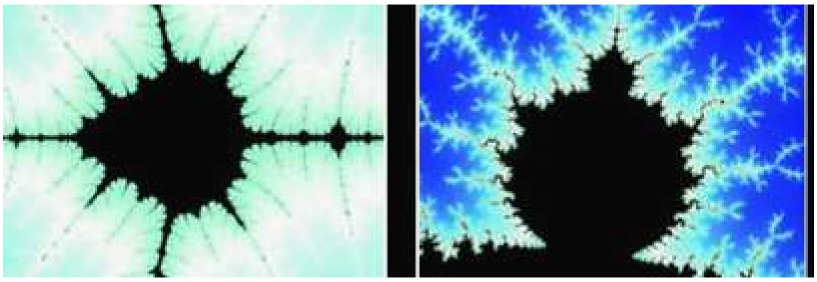


Figure (3.10): The Mandelbrot set

4. A Fractal is developed through Iterations:

Self-similarity, as described before in this chapter, can be produced by iterations, which means that certain kinds of formulas or geometric principles are repeated on the previous result of the calculation or drawing respectively. Examples for geometric rules make up the fern and the Koch curve; those for fractals based on a mathematical equation produce the Mandelbrot set [Mic01].

5. A Fractal Depends on Starting Conditions:

Little differences with regard to the starting conditions may cause great differences in the results. The reason for that lies in the circumstance that for fractal structures always the same rules are repeatedly applied. The behavior of a system can be analyzed by repeating the experiment with the same starting values, under the same conditions so that the same results may be found. This leads us to the principle of causality. If the same causes have the same effects this is called a *"weak"* causality. But mostly it is only possible to get similar starting conditions and not the same ones, which leads to a *"strong"* causality [Mic01].

6. A Fractal is Common in Nature:

Many objects show fractal structures and can be reproduced through fractal geometry such as the cluster of galaxies, the roots of trees, the crater landscape of the room and plants [Mic01].

3.11 Types of Fractal

"Every natural thing around us is a fractal structure in principle, because smooth lines and planes only exist in the ideal world of mathematics. Beside that theoretically any system, which can be visualized or analyzed geometrically, can be a fractal."

This section gives an introduction to some different kinds of fractals like the so-called “true” mathematical fractals, to which the Cantor set belongs, and the “chaotic” fractals, with the Mandelbrot set being an example.

Beside that some other methods of creating fractals such as the iteration function systems, the DLA model, the L-system and the Midpoint displacement method will be introduced. The form of strange attractors as a connection to deterministic chaos also offers fractal characteristics and will be described at the end of this chapter. The one or other type of fractal may help in creativity, analysis, comparison, construction, organization and other questions arising in architecture [Man97].

3.11.1 <u>The "true" Mathematical Fractals:</u>

The development of this kind of fractals consists of simple rules - a starting image, the so-called initiator, is replaced by another image, the so-called generator. But nevertheless they are very complex and always strictly self-similar: it does not matter which part we analyze, it always looks exactly like a scaled down copy of the whole set. The tools to create such fractals are called iteration and feedback: Iteration means that the procedure is repeated based on the result of the previous step.

a. <u>Cantor Set:</u>

For producing the Cantor Set the initiator, a straight line of a certain length, is replaced by a generator consisting of two lines, each of the length of 1/3 of the initiator, in such a way that the new lines are located in each case at the end of the initiator. This geometric rule is repeated again with the two new lines, which leads to four lines and so on [Pau91, Wol04], as shown in figure (3.11).

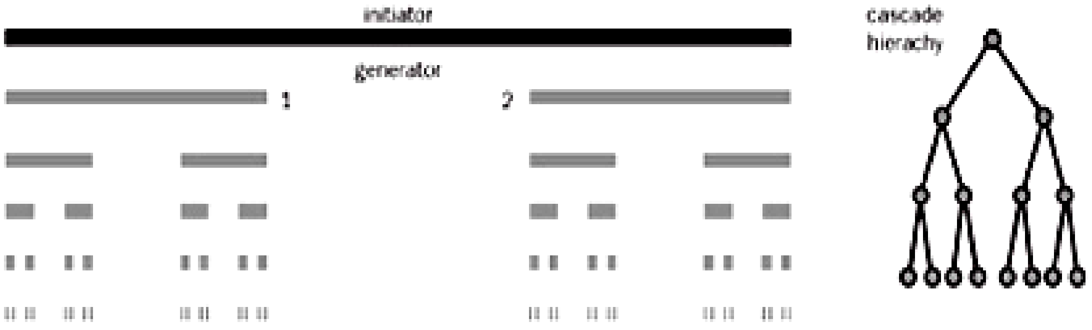


Figure (3.11): Cantor Set

b. Sierpinski Gasket

For producing the Sierpinski Gasket, the initiator, an equilateral triangle, is replaced by a generator consisting of three equilateral triangles, each of the size of half the initiator, in such a way that the new triangles are located in each case at the three corners of the initiator. In other words an equilateral triangle is cut out in the middle. This cut-out triangle is half the size of the initiator and rotated by 180 degrees - the side-points of the triangle are defined by the midpoints of the sides of the original triangle. The same procedure is repeated for each of the three new triangles, and so on. The remaining triangles or the set of points that are left after infinite iterations is called the Sierpinski Gasket [Wol04]. For further details see figure (3.12).

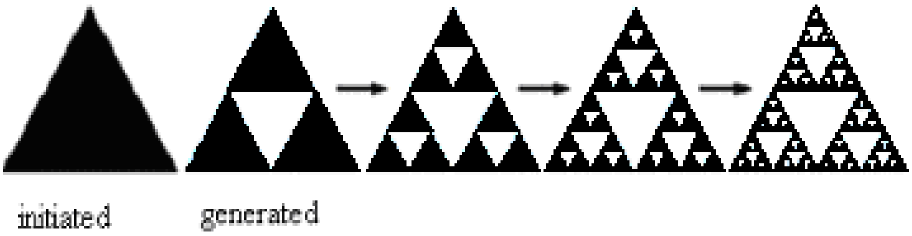


Figure (3.12): Sierpinski Gasket

c. Koch Curve

The initiator of this fractal is again a line, the generator four lines of 1/3 of the initiator. For their creation, the initiator-line is divided into three equal parts, with the middle part being replaced by an equilateral triangle of the side length of 1/3 of the initiator - the lower part of the triangle, however, is taken away. This procedure is then repeated for the four new lines. After infinite steps the construction leads to the Koch curve - the geometric rule for this fractal is shown in figure (3.13) [Wol04].

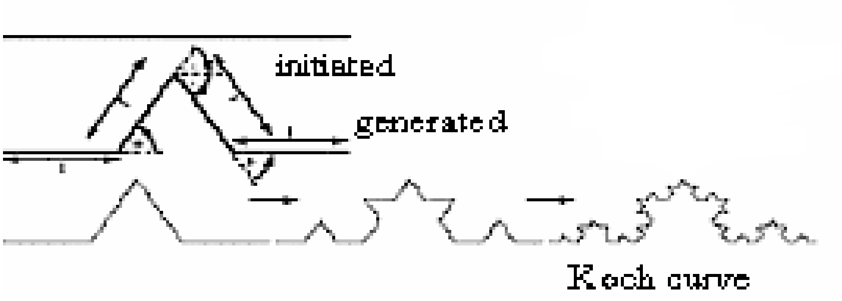


Figure (3.13): Koch curve

d. Minkowski Curve

For constructing the Minkowski curve the initiator, a line of example a unity length equal to one, is replaced by a generator consisting of eight lines. These eight lines, each 1/4 of the original line, are arranged in the following manner: horizontal lines, which are kept in position, build the first fourth and the last fourth of the original line. The second fourth consists of a line turned up 90 degrees, followed by a horizontal line and finally by a line moving down 90 degrees again. The third fourth is constructed by a sequence of lines that is first turned down 90 degrees then moving horizontally and finally turning up 90 degrees again to connect the last fourth. This rule of construction is then repeated for all eight new lines of the first iteration, 64 lines of the second iteration, 512 lines of the third iteration and so on [Wol04], as shown in figure (3.14).

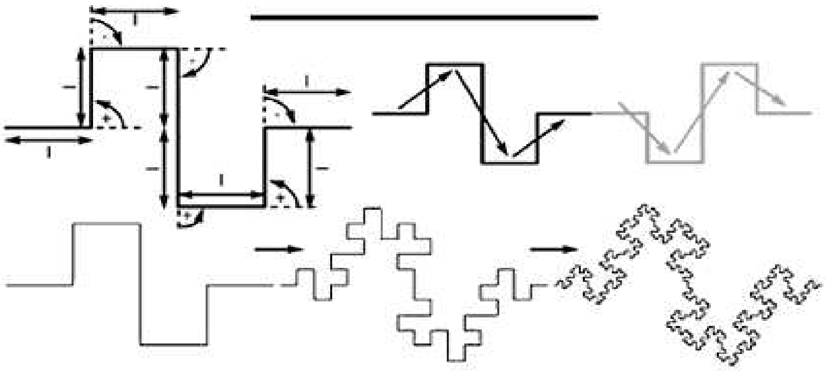


Figure (3.14): Minkowski curve

e. Peano Curve

The initiator of the Peano curve is once more defined through a straight line and the correspondent generator consists of nine lines 1/3 of the initiator. The first line of the generator runs horizontally, the second turns up by 90 degrees, then a horizontal part follows before the curve turns down again by 90 degrees. The fifth line moves back to the end of the first line without touching it. The next part of the curve heads down by 90degrees, followed by a horizontal part before it goes up again. Finally a line located in horizontal position again forms the last section [Wol04] - see figure (3.15).

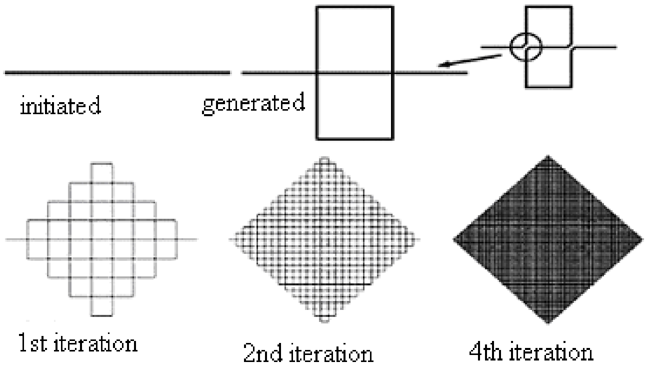


Figure (3.15): Peano curve

3.11.2 <u>Chaotic Fractals:</u>

This fractal type is connected with the theory of chaos, and its elements are obtained by a simple mathematical equation [Bov96]. For visualizing them, each point on the paper or screen is related to a certain number - e.g. in the case of the “Mandelbrot set” this is a complex number. This number is then iterated, that means it is used in a formula and the new number resulting from that is then again used in the same formula, which leads to the next iteration. This sequence of operations is "similar" to the work of the "copy-machine" of linear fractals - with regard to insertion. The insertion is repeated until the numerical values approach infinity, converge or fluctuate between several numbers. Depending on the result, the original point may be colored differently.

a. The Mandelbrot Set

The Mandelbrot set as shown in picture 04 is one of the famous fractals. It was born when Mandelbrot was playing with the simple quadratic equation $Z_{(n+1)} = Z_n^{\ 2} + C$. In this equation, both z and c are complex numbers. In other words, the Mandelbrot set is the set of all complex c such that iterating $Z_{(n+1)} = Z_n{}^2 + C$ dose not diverge [Bat94].

To generate the Mandelbrot set graphically, the computer screen becomes the complex plane. Each point on the plane is tested into the equation $Z_{(n+1)} = Z_n{}^2 + C$. If the iterated z stayed within a given boundary forever, then the c point considered inside the set and the point will be plotted black. If the iteration went out of control, the point will be plotted color. There is no way to sketch the Mandelbrot set or eve describes it without a computer [Abd01].

b. The Julia Sets

The Julia set as shown in figure (3.16) is another very famous fractal, which happens to very closely related to Mandelbrot set. It was named after Gaston Julia. The main difference between the Julia set and Mandelbrot set is in the way in which the function is iterated. The Mandelbrot set iterates $Z_{(n+1)} = Z_n{}^2 + C$ with z always starting at 0 and varying the c value. The Julia set iterates $Z_{(n+1)} = Z_n{}^2 + C$ for a fixed c value and varying z value. In other words, the Mandelbrot set is in the parameter space, or c-plane, while the Julia set is in the dynamical space, or the z-plane [Abd01].

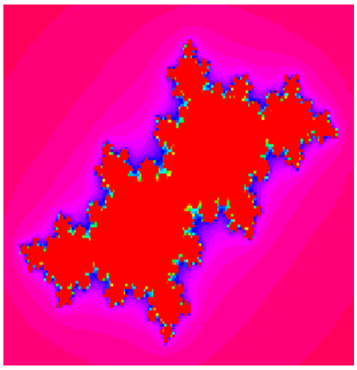


Figure (3.16): Julia set

3.11.3 DLA Model - Diffusion-Limited Aggregation Model:

In physics and chemistry diffusion means a certain behavior of two different gases or liquids, which get in touch with each other. This behavior is characterized by the circumstance that two different gases or liquids are mixed when they are brought together. This mixture happens because of molecular heat emission, but the way of diffusion, its “form”, cannot be forecast, which means that it cannot be calculated by mathematics.

Through simulation: a certain point is marked, for example in the middle of the computer-screen. This point is the starting point of the diffusion. Now another point, anywhere on the screen starts its wanderings. The way it follows is random; the movement is stopped when it touches the stationary point, because at that moment it is also turned into a fixed point. We can also think of a screen full of dead cells - one cell of which is turned to life in the middle part of the screen. Touching this living point means that the moving cell is also turned to life. The random movement is repeated for the next cell or point anywhere on the screen- see figure (3.17).

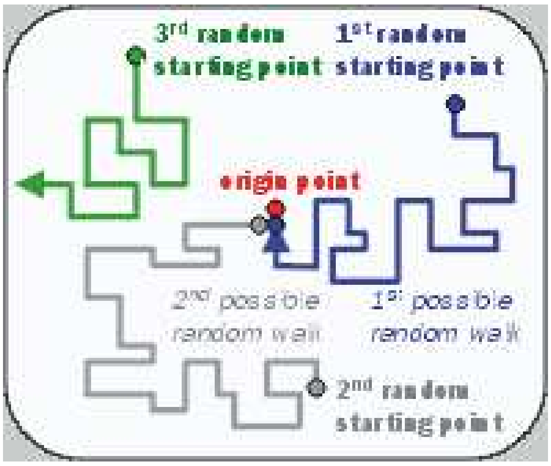

3rd random starting point
1st random starting point
origin point
2nd possible random walk
1st possible random walk
2nd random starting point

Figure (3.17): DLA

Beside that the way of the Brown-movement nearly fills the plane - this would mean that the way of the Brown-movement is topologically a curve of the dimension 1, but because it nearly fills the plane it is a fractal of higher dimension [Wol04].

3.11.4 L-Systems - Lindenmayer-Systems:

Lindenmayer-Systems are like the IFS very close to naturally looking objects. The biologist Aristid Lindenmayer developed this variant to describe plant-forms similarly to the transformation rules of IFS [Wol04], as shown in figure (3.18).

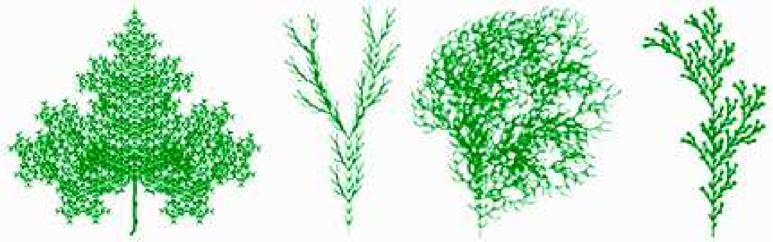


Figure (3.18): produced with L-Systems:

3.11.5 Strange Attractors:

The long-term behavior of a system can be represented in the so-called n-dimensional “phase space”. Attraction areas, to which these trajectories aim, are called attractors. Put into other words an attractor is a preferred position for the system to which it evolves no matter what the starting position is. Once such a position is reached it will then stay on the attractor in the absence of other factors. One classical strange attractor is the “Lorenz attractor” that is used for the weather forecast. The weather forecast depends on many parameters such as season, vegetation, temperature or direction of the wind. Edward Lorenz tried to describe meteorological processes with the support of differential equations.

3.11.6 IFS - Iteration Function Systems:

That concerned previous in section (3.10) the characteristic of Fractal in the self-similarity description.

3.11.7 Random Midpoint Displacement Method

Random midpoint displacement method introduced by Fouriner *et al* [Cox97]. The mathematical, non-linear and linear fractals presented above are deterministic, which means that repeating the transformations under the same starting conditions will always result in the same figure. The midpoint displacement method, however, belongs to the category of random fractals, such as the fractals generated by the DLA-method, which in general produce more nature-like "random" objects.

We draw a triangle on the screen and mark the centers of the three borderlines. Then we move these points perpendicularly to the lines up or down by a random factor. The resulting object consists of four smaller triangles; one of them is the combination of the three newly constructed points, as shown in figure (3.19). The same is applied to the new triangles, and so on. For the decision whether the center point is moved up or down we use a coin [Ivo02].

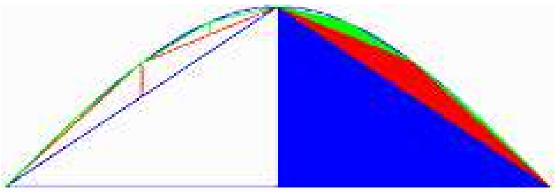


Figure (3.19): Midpoint Displacement

The midpoint displacement method shows very clearly that the ideas around fractal geometry have already been known for a long period of time. Archimedes (287-212 B.C.) used the midpoint displacement for measuring the area under a parabola. For this purpose a vertical line is drawn from the center of the base line of the parabola until it touches the curve. This upper point is then connected with the base-points of the parabola. In the next step vertical lines are drawn from the centers of the two new outer lines until they again reach the curve. Once more they are closed to form triangles whose area can be given easily. Each step produces twice as many triangles as the step before. After some "iterations" the calculated area does not increase very much, that means that the remaining area between the triangles and the parabola becomes smaller and smaller, as shown in figure (3.20). The height of a certain step is related to the heights of the steps before by the formula: height(n+1) = (1/4)*height(n).

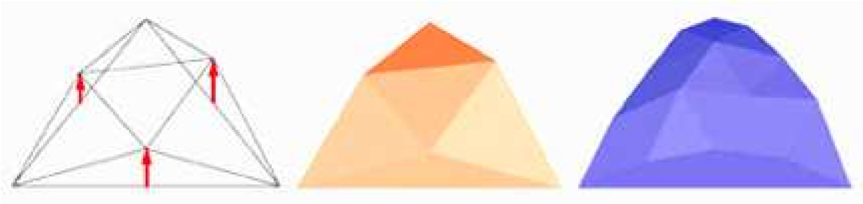


Figure (3.20): Iteration of mountain

Moving up or down the midpoints of the sides of a triangle by a certain value can produce natural looking landscapes. The value of such a displacement can be calculated through different distribution rules - in this way rough or smooth looking mountain ranges can be generated. The construction of the final mountain-like surface requires an additional graphic procedure which connects all the generated points in the way that the space can be visualized, example modeling by a rectangular network, as shown in figure (3.21).

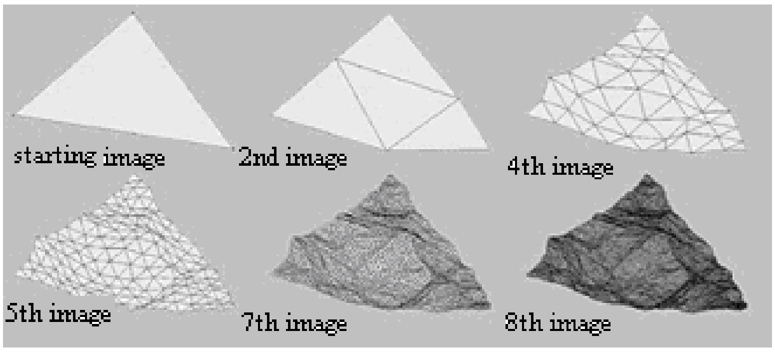


Figure (3.21): The construction of the final mountain

Then the principle of work of this project represented as follows:

An initial square is subdivided into four smaller squares, as shown in figure (3.22).

Let us have four points $[x_0, y_0, f(x_{0,} y_0)], [x_1, y_0, f(x_1, y_0)],$ $[x_0, y_1, f(x_0, y_1)], [x_1, y_1, f(x_1, y_1)]$. In the first step we add one vertex into the middle. The vertex is denoted by $\lfloor x_{1/2}, y_{1/2}, f(x_{1/2}, y_{1/2}) \rfloor$, where

$$x_{1/2} = \frac{1}{2}(x_0 + x_1)$$

$$y_{1/2} = \frac{1}{2}(y_0 + y_1)$$

$$f(x_{1/2}, y_{1/2}) = \frac{1}{4}(f(x_0, y_0) + f(x_1, y_0) + f(x_0, y_1) + f(x_1, y_1)) \qquad \ldots(3.13)$$

This procedure is recursively repeated for each subsquare, then for every their descendants, and so on.

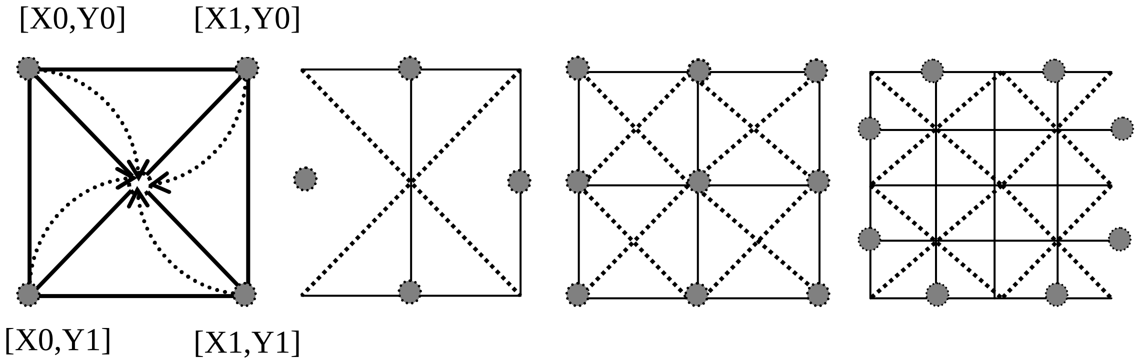


Figure (3.22): First four steps in random midpoint displacement method

The random number must be generated with Gaussian distribution [μ=0, σ=1] and in the i-th iteration step the variation σi have to be modified according to

$$\sigma^2{}_i = \frac{1}{2^{2H(i+1)}}\sigma^2 \qquad \ldots..\ (3.13)$$

Where H denotes Hurst exponent $(1 \le H \le 2)$. From equation (3.13) we can see, that the first iteration has the biggest influence to the resulting shape of the surface and influence of the others decreases.

In the second step we calculate the points on the edges of Initial Square. We virtually rotate square by 45^0 and calculate the values as in the previous step.

In the next step we virtually rotate the square back by 45^0 and we recursively apply the first two steps on the four new squares as is mentioned above. This recursive process ends after given number of iteration [Ivo02].

Fractal dimension D of surface is obtained by

$$D=3-H$$

An example of fractal terrain obtained with random midpoint displacement algorithm is in figure (3.23).

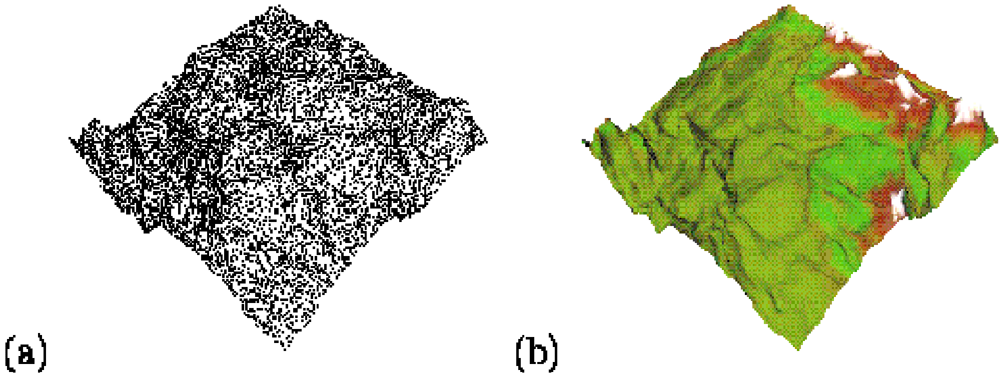


Figure (3.23): Example of fractal terrain (a) Wire frame model (b) the same model textured

3.12 Euclidean and Fractal Geometry

Up to these days we have been used to think and talk in the words of traditional Euclidean geometry [Bov96]. But many complex objects described and composed by single Euclidean sections [Bat94] do not really reflect the characteristic of the whole real-world object; clouds and mountains respectively do not correspond to simple geometric rules. Fractal curves consist of infinite elements which are infinitely small and which are, because of that, not tangible. These infinite elements are the reason why the length increases to infinity at an infinitely small scale and by that makes it impossible to define a point of a fractal curve by co-ordinates or describe its position on the curve exactly after all. Some of the major differences between fractal and Euclidean geometry are outlined in table (3.1).

Table (3.1): Differences between fractal and Euclidean geometry

Fractal	Euclidean
Modern invention.	Traditional.
No specific size or scale.	Based on a characteristic size or scale.
Appropriate for geometry in nature.	Suits description of man made objects.
Described by an algorithm.	Described by a usually simple formula.

Firstly the recognition of fractal is very modern, they have only formally been studied in the last ten years compared to Euclidean geometry which goes back over 2000 years. Secondly whereas Euclidean shapes normally have a few characteristic sizes or length scales (e.g.: the radius of a circle or the length of a side of a cube) fractals have so characteristic sizes. Fractal shapes are self-similar and independent of size or scaling. Third, Euclidean geometry provides a good description of man-made objects whereas fractals are required for a representation of naturally occurring geometries. It is likely that this limitation of our traditional language of shape is responsible for the striking difference between mass produced objects and natural shapes. Finally, Euclidean geometries are defined by algebraic formulae.

3.13 Geometrical Dimensions

The concept of dimension used in school mostly deals with Euclidean geometry. In short, in an E-dimensional system of co-ordinates at least E-co-ordinates are needed for defining the position of a point. Consequently a point corresponds to a zero-dimensional system of co-ordinates, points on a line to a one-dimensional, a line on a plane to a two-dimensional and finally a plane on a cube to a three dimensional system of co-ordinates, as shown in figure (3.24).

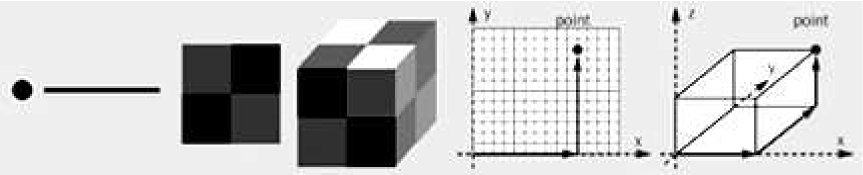


Figure (3.24): Euclidean Geometry

The point has no width, no height, no length and therefore no dimension. As well as a line in the Euclidean sense cannot be drawn exactly, because it has no thickness and is characterized by infinity to both sides, something similar is true for fractal curves. First they also have no thickness and second they are unrestricted, which means that they are of infinite length bound between two ends. The thickness in general is no problem, but also infinity does no harm, because the character and attributes of fractals can be shown after only a few iterations, for example only a few iterations can produce fern-like, mountain-like or cloud-like fractals [Pau91,BaL94].

Each point on a two-dimensional surface can be described by exactly two numbers, e.g. a grid can be put on the surface and the distances from the borders can be given. A certain width and a certain length define the plane, but it has no height.

A point in a three-dimensional space, with the three dimensions being the length, the width and the height, is described by three numbers, for example by the three ordinates [Wol04].

3.14 Fractal Dimension

Visually the fractal dimension is the expression of the degree of roughness, which means how much texture an object, has. It also shows how fast the length of a fractal increases from one iteration to the next. Fractal dimension is not an integer in contrast to the dimension in Euclidean geometry. The complex forms of clouds, blood vessels, coastlines or mountains seem to have an unrestricted complexity, but they nevertheless have a geometric regularity, their scale-independence. That means, if we analyze the structure on different scales, we will always find the same basic elements. Fractal dimension also expresses the connection between these different scales [Bov96].

The Hausdorff Dimension

Take an object residing in Euclidean dimension D and reduce its linear size by $1/r$ in each spatial direction, its measure (length, area or volume) would increase to $N = r^D$ times the original [Rob95, Man97, Cyn04], as shown in figure (3.25).

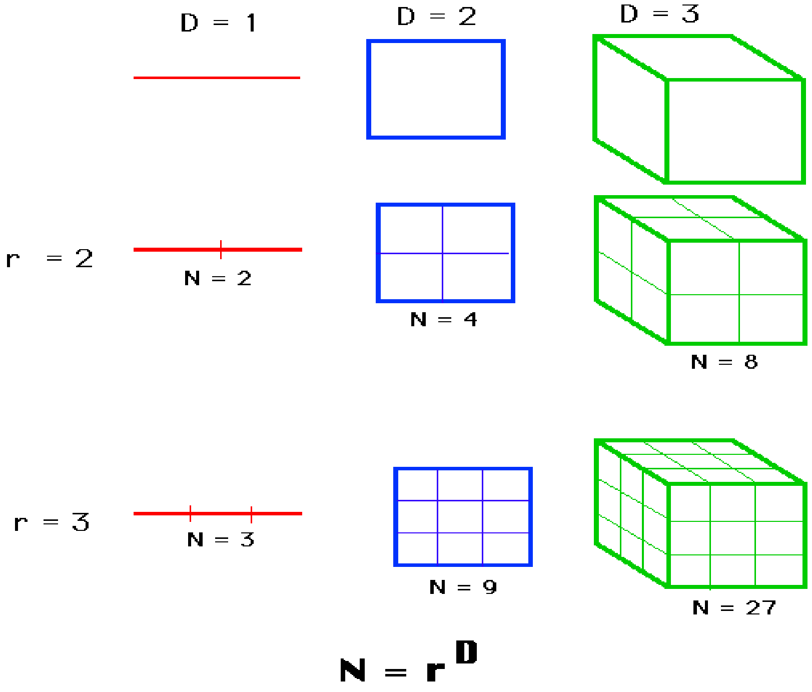


Figure (3.25): The Hausdorff Dimension

Take the log of both sides, and get $\log(N) = D\log(r)$. If we solve for D, $D = \log(N)/\log(r)$. D need not be an integer, as it is in Euclidean geometry. It could be a fraction, as it is in fractal geometry.

For **Cantor set,** we calculate the dimension by, as shown in figure (3.26):

$$D = \log(N)/\log(r)$$

$D = \log(2)/\log(3) = 0.63$, We have an object with dimensionality less than one, between a point and a line, as shown in figure (3.26).

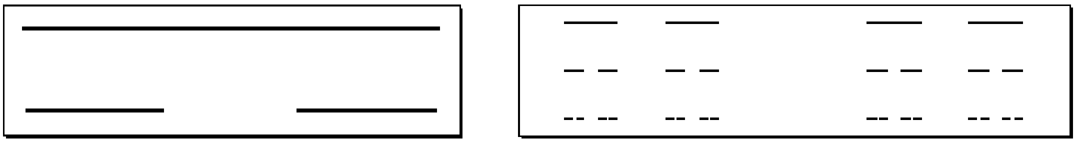


Figure (3.26): Cantor set

For **Sierpinski Triangle,** Calculating the dimension by:

$D = \log(N)/\log(r) = \log(3)/\log(2) = 1.585$. This time we get a value between 1 and 2, as shown in figure (3.27).

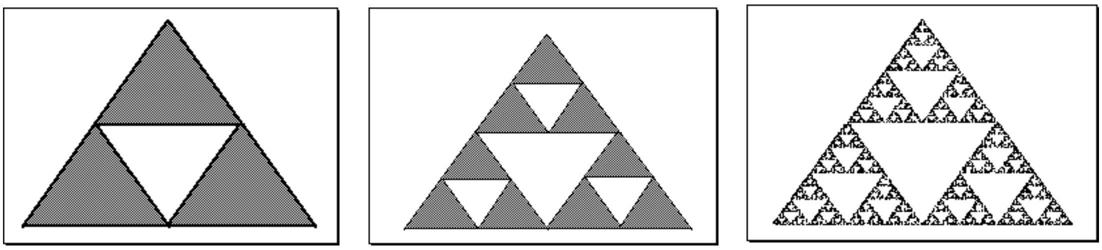


Figure (3.27): Sierpinski Triangle

For **Koch curve,** $D = \log(\text{ N})/\log(\text{r}) = \log(4)/\log(3) = 1.26$,as depicted in figure (3.28).

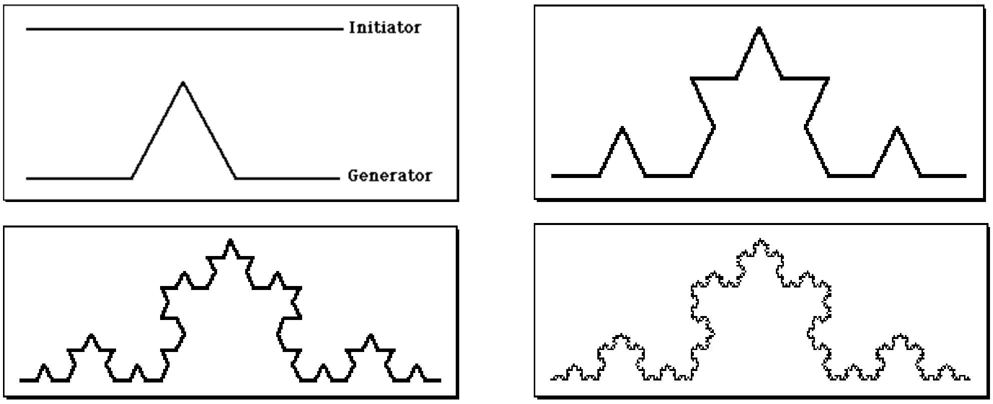


Figure (3.28): Koch curve

3. 15 JPEG2000

JPEG committee has decided, as early as 1995, to develop a new, wavelet based standard for the compression of still images, to be know as JPEG2000 (or JPEG Y2K). The JPEG2000 architecture advances a number of different applications in the digital imaging market, everything from digital cameras, pre-press, medical imaging and other key sectors. As digital imagery expands in quality, size, and application there is a greater need for image compression with flexibility and efficient interchangeability. JPEG2000, file extension .JP2, is a more dynamic, more powerful file format for today and tomorrow's applications delivering better compression efficiency, as well as features not available in previous standards. **Wavelet-based compression** technology is the core strength of JPEG2000, which is designed to meet the growing application needs not addressed by the current JPEG standard. While offering state-of-the-art compression, JPEG2000 also offers unprecedented access into the image while still in compressed form. Thus, images can be accessed, manipulated, edited, transmitted, and stored in a minimal information form. JPEG2000 supports a wide set of features, achieving in a single file format what the original baseline JPEG offers in 44 largely incompatible modes. JP2 is a feature-rich, flexible format that is striving to be an open standard by the end of year 2000 [Chr00].

3.15.1 The Use of JPEG2000

With the increasing use of multimedia technologies, image compression requires higher performance and new features. The JPEG2000 standard is intended to advance standardized image coding systems to serve applications into the next millennium. JPEG2000 will provide superior rate-distortion and subjective image quality performance than existing standards; its features will define JPEG2000. It will provide functionality vital to many high-end and emerging imaging applications by taking advantage of new modern technologies. Specifically, this new standard will address areas where current standards fail to produce the best quality or performance and will provide capabilities to markets that currently do not use compression [Iso00].

3.15.2 The Application of JPEG2000

JPEG2000's desired capabilities shall serve markets and applications such as: Internet, Facsimile, Printing, Scanning, Digital Photography, Remote Sensing, Mobile/ Wireless, Medical, Digital Library and E-Commerce [Iso00].

3.15.3 Technical Description

JPEG2000 is a new wavelet-based image coding system for different types of still images (bi-level, gray-level, color, multi-component) with different characteristics (natural images, scientific, medical, remote sensing imagery, text, rendered graphics, etc.) allowing different imaging models (client/server, real-time transmission, image library archival, limited buffer and bandwidth resources, etc.) preferably within a unified system.

This coding system is intended to provide low bit-rate operation with rate distortion and subjective image quality performance superior to existing standards, without sacrificing performance at other points in the rate-distortion spectrum.

This standard will serve still image compression needs that are currently not served by the JPEG standards. For example, very low bit-rate, progression for the WWW, medical imagery, pre-press, etc. It is intended to complement, not to replace, the current JPEG standards. Indeed, this standard is expected to include an architectural context that will allow the previous standards to be used as desired on different tiles and/or components within a single image [Cru00].

Following is a list of features that this new standard is expected to improve on existing methods:

1. High compression efficiency. Bitrates of less than 0.25 bpp for highly detailed grayscale images.
2. The ability to handle large images. Up to $2^{32} \times 2^{32}$ pixels (the original JPEG can handle images of up to $2^{16} \times 2^{16}$).
3. Progressive image transmission. It can decompress an image progressively by resolution, color component, or region of interest.
4. Easy, fast access to various points in the compressed stream.
5. The decoders can plan/zoom the image while decompressing only parts of it.
6. The decoder can rotate and crop the image while decompressing it.
7. Error resilience. Error-correcting codes can be included in the compressed stream, to improve transmission reliability in noisy environments.

JPEG2000 is based on the Embedded Block Coding with Optimized Truncation (EBCOT) scheme proposed by Dr. Tauhman. Basically, EBCOT can be viewed as a block-based bit-plane coder (i.e. that the basic coding unit is a block instead of the whole image as used in coding schemes and each block is then encoded independently by using the same algorithm. That for each block a separate bit-stream is generated without resorting to any information from other blocks).

The implementation of EBCOT is a pipeline structure as shown in figure (3.29), which attempts to minimize the internal memory size [Chr00, Cru00].

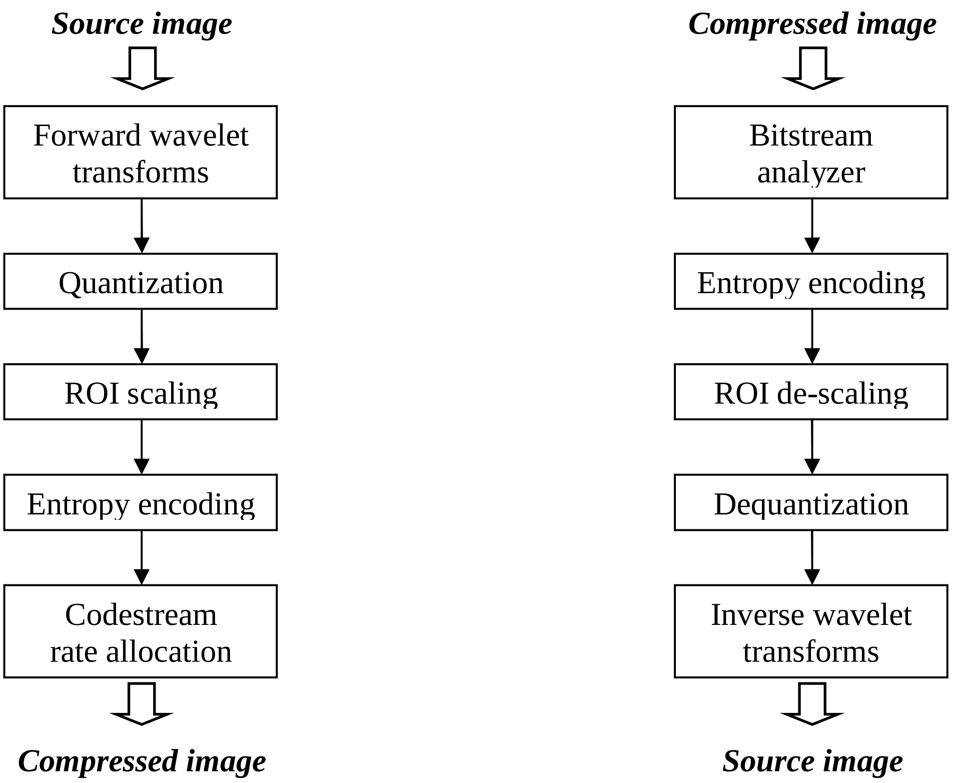


Figure (3.29): The JPEG2000 coding pipeline

Digital Watermarking

2.1 Introduction

Digital watermarking can be defined as the practice of imperceptibly altering the software product (Digital Document) to embed a watermark (can be message, image or number) about that product, while Steganography represents the art of concealed communication that keeps the very existence of a message secret [ISO00].

Steganography methods are usually not robust against modifications of the data, or have only limited robustness to product the embedded information against technical modifications that may occur during transmission and storage, like format conversion, and compression [Jam02].

On the other hand, watermarking has the additional notion of resilience against attempts to remove the hidden data. Thus, watermarking, rather than Steganography principles are used whenever the cover data is available to parties who know the existence of the hidden data and may have an interest in removing it. Although watermarking methods have been robust, in general, different levels of required robustness can be identified depending on the specific application-driven requirements [Cox02].

2.2 Relationship of Information Hiding with Watermarking

Watermarking is closely related to the fields of Information hiding and Steganography. These three fields have a great deal of overlap and share many technical approaches. However, there are fundamental philosophical differences that effect the requirements, and thus the design, of a technical solution.

Information hiding (or data hiding) is a general term encompassing a wide range of problems beyond that of embedding message in content. The term hiding here can refer to either making the information imperceptible or keeping the existence of the information secret [Cox02].

Steganography is a term derived from Greek words steganos, which means "covered", and graphia, which means "writing". It is the art of concealed communication. The very existence of a message is secret. An example of Steganography is a story from Herodotus [Cox02].

Cryptography is the art and science of keeping messages secures [Cox02].

Therefore the difference between the Steganography and watermarking that the information hidden by watermarking system is always associated to the digital object to be protected or to its owner while steganographic systems just hide any information. The "robustness" criteria are also different, since steganograhy is mainly concerned with detection of the hidden message while watermarking concerns potential removal by a pirate. Finally, steganography communications are usually point-to-point (between the sender and receiver) while watermarking techniques are usually one-to-many [Ali04].This can be seen in table (2.1) [Fri98]:

Table (2.1): Comparisons between Steganography and Watermarking.

Viewpoint	**Steganography**	**Watermarking**
Object	• Unobservable, Confidential communication.	• Protecting authorship.
Attack	• No modification of stego-object. • Identifying communication. • Extracting embedded message.	• Massive modification of the cover-object. • Destroying/changing embedded copyright data.
Properties	• Embedding as much data as possible. • No precautions against destruction of cover-object. • Not verifiable without stego-key.	• Few data to embed. • Data has to be embedded robust. • Resistant to compression, linear filtering, additive noise, quantization, etc.

		• Redundant embedding of copyright data. • Embedded data is imperceptible. • According to algorithm copyright data is not verifiable without key.

And, the analogy between the cryptography and watermarking is the watermark embedding and detection can some times be considered analogous to encryption and decryption.

In symmetric key cryptography, we have an encryption function, $E_K(.)$ that takes a key, K and some cleartext, m and produces a ciphertext, m_c:

$m_c = E_K(m)$ ... (2.1)

And, the decrypted the ciphertext by:

$m = D_K(m_c)$ ... (2.2)

In watermarking, we have embedding function $\mathrm{E}(.)$ that takes a message, m, and an original Work, $c_{\circ}$, and outputs a watermarked Work, c_W. Similarly we have a detection function, $D(.)$, that takes a watermarked Work and outputs a message. The mapping between watermarked Works and messages is controlled by a watermark key, K ,for this we can characterize watermarking system by the equations:

$c_W = \mathrm{E}(c_{\circ}, m)$ ... (2.3)

And

$m = D_K(c_W)$ ... (2.4)

Which are similar to the previous equations [Cox02].

2.3 Watermarking System

Digital watermarking is the process of embedding a signal, called the watermark, into another signal, called the host or cover, robustly and at the same time imperceptibly.

The host signal can either be an image, audio, video or a text document (for example, program source code).

The earliest known Work describing digital watermarking in the 1954 patent obtained by Emil Hembrooke of the Muzak Corporation in which a method for embedding an identification code into music for proving ownership was described.[Jam02]

Watermarking process consists of an embedded and a detector, as illustrated in Figure (2.1). The embedder takes two inputs. One is the message we want to encode as a watermark, and the other is the cover work in which we want to embed the mark. The output of the watermark embedded is typically transmitted or recorded. Later, that work is presented as input to the watermark detector. Most detectors try to determine whether a watermark is present, and if so, output the message encoded by it [Cox02].

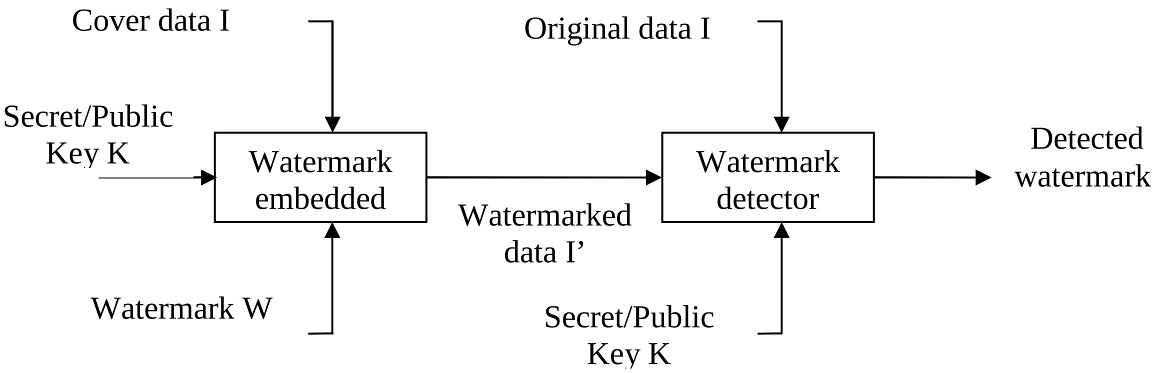


Figure (2.1): The Watermarking System (embedding and detection).

2.4 Purpose of Digital Watermarking

Watermarks added to digital content serve a variety of purposes. The following list details some of purposes of digital watermarking.

- **Ownership Assertion –** to establish ownership of the content (i.e. image).
- **Fingerprinting –** to avoid unauthorized duplication and distribution of publicly available multimedia content.
- **Authentication and integrity verification –** the authenticator is inseparably bound to the content whereby the author has a unique key associated with the content and can verify integrity of that content by extracting the watermark.
- **Content labeling –** bits embedded into the data that gives further information about the content such as a graphic image with time and place information.
- **Usage control –** added to limit the number of copies created whereas the watermarks are modified by the hardware and at some point would not create any more copies (i.e. DVD).
- **Content protection –** content stamped with a visible watermark that is very difficult to remove so that it can be publicly and freely distributed.

Unfortunately, there is no a universal watermarking technique to satisfy all of these purposes [Cox02].

2.5 Watermarking application

Watermarking is distinguished from other techniques in three important ways. First, watermarks are imperceptible. Unlike bar codes, they do not diminish from the aesthetics of an image. Second, watermarks are inseparable from the works in which they are embedded. Finally, watermarks undergo the same transformations as the works. This means that it is sometimes possible to learn something about those transformations by looking at the resulting watermarks[Cox02]. It is these three attributes that make watermarking invaluable for certain application, then we examine some of proposed or actual watermarking applications[Fer98]:

1) **Broadcast monitoring:** Identifying when and where works are broadcast by recognizing watermarks embedded in them, which mean watermarking each video or sound clip prior to broadcast. Automated monitoring stations can then receive broadcasts and look for these watermarks, identifying when and where each clip appears[Fri98, Katz02].
2) **Owner identification:** Embedding the identity of a Work's copyright holder as a watermark. The form of the copyright notice is usually "© date, owner". On books and photographs, the copyright is placed in plane sight. A digital watermark can be used to provide complementary copyright marking functionality because it becomes an integral part of the content. Their watermark embedder and detector are bundled with Adobe's popular image processing program (Photoshop) [Fri98, Katz02].
3) **Proof of ownership:** Using watermarks to provide evidence in ownership disputes[Cox02].
4) **Transaction tracking:** Using watermarks to identify people who obtain content legally but illegally redistribute it. That Monitoring and Owner identification application place same watermark in all copies of the same content. However, electronic distribution of content allows a unique watermark to be embedded in each individual copy. Transactional watermarks, also called **Fingerprints,** allow a content owner or content distributor to identify the source of an illegal copy[Fri98,Katz02].
5) **Content authentication:** Embedding signature information in content that can be later checked to verify it has not been tampered with. Verification watermarks are required to be fragile, so that any

modification to the image will destroy the mark. Unlike cryptographic message digests which can only validate identical copies, watermarking for image authentication should tolerate some well-defined image distortion(e.g. file format conversion, recompression, resampling) [Fri98, Katz02].

6) **Copy control:** Transactional watermarks as well as watermarks for monitoring, identification, and proof of ownership do not prevent illegal copying. Therefore, using watermarks to tell recording equipment what content may not be recorded. Then all manufactured recorders must include watermark detection circuitry. This system is currently being developed for DVD video and for digital music distribution [Cox02].
7) **Device control:** Using watermarks to make devices, such as toys, react to displayed content [Cox02].

2.6 Watermarking Requirements

The performance of a given watermarking system can be evaluated on the basis of a small set of properties. For example, *robustness*, describes how well watermarks survive common signal processing operations, *fidelity* describes how imperceptible the watermarks are, and so forth. The relative importance of these properties depends on the application for which the system is designed. For example in applications where we have to detect the watermark in a copy of a Work that has been broadcast over an analog channel, the watermark must be robust against the degradation caused by that channel. In this section, we can characterize by a highlight ten of defining properties. The relative importance of each property is dependent on the requirements of the application, even the interpretation of a watermark property can vary with the application [Cox02, Ali04].

1) **Embedding effectiveness:** we define a watermarked work as a work that when input to a detector results in a positive detection, the

effectiveness is the probability of detection immediately after embedding, the definition implies that a watermarking system might have an effectiveness of less than 100%.

2) **Fidelity:** refers to the perceptual similarity between the original and watermarked versions of the cover work when they are presented to a consumer. A watermark is said to have high fidelity if the degradation it causes is very difficult for a viewer to perceive. However, it only needs to be imperceptible at the time that the media is viewed [Isa04].
3) **Data payload:** refer to the number of bits a watermark encodes within a unit of time or within a work, the data payload would refer to the number of bits encoded within the image.
4) **Blind or informed detection:** this often substantially improved detector performance in the original can be subtracted from the watermarked copy to obtain the watermark pattern alone. We refer to a detector that requires access to the original, unwatermarked work as an informed detector.
5) **False positive rate:** is the detection of a watermark in a work that doesn't actually contain one [Cox02].
6) **Robustness:** refer to the ability to detect the watermark after common signal processing operations. A watermark is said to be robust if it survives against common signal processing operations (such as lossy compression and digital-to-analog-to-digital conversions). There has been an increased concern that video and still image watermarks also be robust to geometric transformations. Robustness is often thought of as a single-dimensional value, but this is incorrect. A watermark that is robust against one process may be very fragile against another. In many applications, robustness to all possible processing is excessive and unnecessary. Usually, a watermark must survive against common signal processing only during the time interval between the time of embedding and the time

of detection. For example, in television and radio broadcast monitoring, the watermark need only survive during the transmission process. For television, the signal processing processes may include lossy compression, analog transmission and some small amount of horizontal and vertical translation. It need not survive against rotation, scaling, high-pass filtering, or any of a wide variety of distortions that do not occur during broadcast. In some cases, robustness may be completely irrelevant, or even undesirable. Watermarks used for covert communication need not be robust at all, if the cover media will be transmitted digitally without compression [Ali04].

A watermark for simple authentication, which just indicates whether the media has been altered, should be fragile. On the other hand, when the signal processing between embedding and detection is unpredictable, the watermark may need to be robust to every conceivable distortion. This is the case for owner identification, proof of ownership, fingerprinting, copy control and device control. It is also true for any application in which hackers might want to remove the watermark [Isa04].

7) **Security:** refer to the ability to resist hostile attacks; a hostile attack is any process specifically intended to thwart the watermark's purpose. The types of attacks we might be concerned in the resistance to attacks [Cox02]. The embedding algorithm is said to be secure if the embedded information cannot be removed beyond reliable detection by targeted attacks based on a full knowledge of the embedding and the detection algorithms (except the secret key). In general, watermarking systems use one or more cryptographically secure keys to ensure security against manipulation and erasure of the watermark. As soon as someone can read a watermark, the same person may easily destroy it, not only because of the embedding strategy, but also

because the locations of the watermark are known in this case [Jam02].

8) **Cipher and watermark keys:** the security is provided by the use of secret keys. These can be thought of as more or less arbitrary sequences of bits that determine how messages are encrypted.
9) **Modification and multiple watermarks:** the possibility of changing embedded watermarks or embedding several watermarks in one work.
10) **Cost:** the computational cost of the embedder and detector [Ali04].

2.7 Evaluation and Benchmarking of Watermarks

Most people who deal with watermarking system need some way of evaluating and comparing them. The interested in applying watermarking to an application need to identify the systems that are most appropriate. Those interested in developing new watermarking systems need measures by which to verify algorithmic improvements. These measures lead to ways of optimizing various properties [Cox02, Fer98]: -

1) **The notion of "Best":** we need to have some idea of what makes one system better than another, or what level of performance would be the best. If we are interested in using a watermark for some specific application, our evaluation criteria must depend on that application [Cox02].
2) **Benchmarking**: once the appropriate tests have been determined, we can turn our attention to developing a test. This benchmark could be used by a researcher to assign a single, scalar score to a proposed watermarking system. The score could then be used to compare it against other systems similarly tested. Example of these tests, Stirmark, It is used as a benchmark against attacks. Stirmark applies attacks to a watermarked image in nine different categories [Vol01]. This program applies distortions that have little effect on the

perceptual quality of images, but are known to render most watermarks undetectable. The system's performance in these tests is combined into a single score [Fer98].

3) **Scope of test:** watermarking systems should be tested on a large number of Works drawn from a distribution similar to that expected in the application [Fer98].

2.8 The Aim of Digital Watermarks

Watermarks are a way of dealing with many problems by its properties providing a number of services [Chri00]:

1. The aim to mark digital data permanently and unalterably, so that the source as well as the intended recipient of the digital work is known. Copyright owners can incorporate identifying information into their work. Watermarks are used in the protection of ownership. The presence of a watermark in a work suspected of having been copied can prove that it has been copied.
2. By indicating the owner of the work, they demonstrate the quality and assure the authenticity of the work.
3. With a tracking service, owners are able to find illegal copies of their work on the Internet, to become a unique watermark embedded in her/his copy, any of unauthorized copies that she/he has distributed can be traced back to him/her.
4. Watermarks can be used to identify any changes that have been made to the watermarked data.
5. Some more recent techniques are able to correct the alteration as well.

2.9 The Classification of Watermarking System

There are several types of watermarking systems. They are defined by their inputs and outputs [Cox02]:

1. **Private watermarking** systems require at least the original image. Type 1 systems, extract the watermark W from the possibly distorted image $I^{'}$ and use the original image as a hint to find where the watermark could be in $I^{'}(I^{'} \times I \times K \rightarrow W)$. Type 2 system also require a copy of the embedded watermark for extraction and just yield a 'yes' or 'no' answer to the question: does $I^{'}$ contain the watermark W? $(I^{'} \times I \times K \times W \rightarrow \{0,1\})$. It is expected that this kind of scheme will be more robust than the other since it conveys very little information and requires access to secret material.
2. **Semi-private watermarking** does not use the original image for detection $(I^{'} \times K \times W \rightarrow \{0,1\})$ but answers the same question. The only use of private and semi-private watermarking seems to be evidence in court to prove ownership and copy-control in applications such as DVD where the reader needs to know whether it is allowed to play the content or not. A large number of the currently proposed schemes fall in this category.
3. **Public watermarking** (also referred to as blind watermarking) remains the most challenging problem since it requires neither the secret original I nor the embedded watermark W. Indeed such systems really extract n bits of information (the watermark) from the watermarked image: $(I^{'} \times K \rightarrow W)$. Public watermarks have much more applications than the others and we will focus our benchmark on these systems. Actually the embedding algorithms used in public systems can always be used into a private one improving robustness at the same time.

4. There is also **asymmetric watermarking** (or public key watermarking) which has the property that any user can read the watermark, without being able to remove it.

Also, the watermark can be attributed as follows[Joh01,Katz02]:

Visible watermark	**Invisible watermark**
The intention is for the presence of the watermark to be very obvious but equally to make it impossible to remove without destroying the image.	Can be used for any application also resist any detection and decoding.

Complete watermark	**Incomplete watermark**
Doesn’t need the original copy for the hidden massage decoding.	This scheme needs the original copy for message decoding which means that it is strongly resistant to detection and decoding.

Robust watermark	**Fragile watermarks**
These are designed to withstand accidental and malicious attack.	Have just the opposite characteristics and are used to detect tampering.

After grouping the different systems, we can now identify important parameters and variable [Cox02, Cru00]:

- **Amount of embedded information -** This is an important parameter since it directly influences the watermark robustness. The more information one wants to embed, the lower is the watermark robustness. The information to be hidden depends on the application.
- **Watermark embedding strength –** There is tradeoff between the watermark embedding strength and quality. Increased robustness

requires a stronger embedding, which in turn increases the visual degradation of the images.

- **Size and nature of the picture -** Although very small pictures do not have much commercial value, watermarking software needs to be able to recover a watermark from them. Furthermore the nature of the image has also an important impact on the watermark robustness. Very often methods featuring a high robustness for scanned natural images have a surprisingly reduced robustness for synthetic images. A fair benchmark should use a wide range of picture sizes, from few hundred to several thousands pixels, and different kind of images.
- **Secret information -** Although the amount of secret information has no direct impact on the visual fidelity of the image or the robustness of the watermark, it plays an important role in the security of the system. The key space, that is the range of all possible value of the secret information, must be larger enough to make exhaustive search attacks impossible. The reader should also keep in mind that many security systems fail to resist to very simple attacks because of bad software engineering.

2.10 Resistance to Attacks

Attack can be defined as any processing that may impair detection of the watermark or communication of the information conveyed by the watermark. The processed watermarked data is then called attacked data.

There exist four classes of the attacks [Fer98]:

1. **Removal attacks:** Removal attacks aim at the complete removal of the watermark information from the watermarked data without cracking the security of the watermarking algorithm (e.g. without the key used for watermark embedding). No processing can recover the watermark information from the attacked data. This category includes denoising, quantization (e.g. for compression), remodulation, and collusion attacks [Fer98].
2. **Geometric attacks:** Intend to distort the watermark detector synchronization with the embedded information. The detector could recover the embedded watermark information when perfect synchronization is regained. Robustness to geometric distortion often relies on the use of either a transform invariant domain or an additional template, or specially designed periodic watermarks whose Auto Covariance Function (ACF) allows estimation of the geometric distortions [Cox02].
3. **Cryptographic attacks:** Cryptographic attacks aim at cracking the security methods in watermarking schemes and thus finding a way to remove the embedded watermark information or to embed misleading watermarks. One such technique is brute-force search for the embedded secret information. Another attack in this category is the so-called Oracle attack, which can be used to create a non-watermarked signal when a watermark detector device is available. Application of these attacks is restricted due to their high computational complexity [Fer98].

4. **Protocol attacks:** Protocol attacks aim at attacking the entire concept of the watermarking application. One type of protocol attack is based on the concept of invertible watermarks the idea behind inversion is that the attacker subtracts his own watermark from the watermarked data and claims to be the owner of the watermarked data. Another protocol attack is the copy attack. The goal is to estimate a watermark from watermarked data and copy it to some other data called target data [Fer98].

An important aspect of any watermarking scheme is its robustness against attacks. The notion of robustness is intuitively clear: A watermark is robust if it cannot be impaired without also rendering the attacked data useless. Watermark impairment can be measured by criteria such as miss probability, probability of bit error, or channel capacity. For multimedia, the usefulness of the attacked data can be gauged by considering its perceptual quality or distortion. Hence, robustness can be evaluated by simultaneously considering watermark impairment and the distortion of the attacked data. An attack succeeds in defeating a watermarking scheme if it impairs the watermark beyond acceptable limits while maintaining the perceptual quality of the attacked data. The developers of watermarking algorithms refer to the results of experimental testing performed in the scope of some benchmark. The benchmark combines the possible attacks into a common framework and weights the resulted performances depending on the possible application of the watermarking technology [Fri98].

Conclusions and Suggestions

5.1 Conclusions

From our previous discussion we have reported some results that conducted with the behavior and performance resulted from the proposed Wavelet Fractal Image Watermarked System. Therefore, some of the following conducted conclusions are derived:

1. The results of the test confirm the idea that wavelet transform is suitable for watermark application, and the watermark was generated from the Midpoint Displacement Method using H-dimension.
2. The proposed Fractal watermarked image scheme can establish watermarks capable to survive against JPEG2000 up to compression ratio (1:3) at quality 88.0.
3. By using the Haar transform we can embed the fractal watermark with modulation factor value to satisfy the degree of robustness against compression.
4. While the compression ratio increases, the number of survived embedded fractal watermark bits is decreases.
5. The distortion bits are increased when quality and compression size decreases.
6. The best result of the proposed system occur when quality =88.0, compression size =64.0 Kb when the original size =192.1 Kb, compression ratio =1/3, then the distorted bits is very little or underprivileged.

5.2 Suggestions

1. Can use this project to determine the modified area of the cover image from the Tamper detection.
2. Can use the proposed system to make watermarking robust against other types of attacks.
3. Can use other types of transform that used to images.
4. Can suggest algorithm to survive other type of the cover image not only (.Bmp) images and the watermark possible to be a message or number.

Proposed Wavelet Fractal Image Watermarking System (WFIWS)

4.1 Introduction

The steps of the proposed watermarking algorithm consist of two Modules:

a. **Embedding Module.**
b. **Extraction Module.**

The embedding module is used for embedding fractal true color image (watermark) inside still cover true color image, then the result of this process is watermarking image. The overall of operations of embedding and implemented algorithms are described in section (4.3).

The extraction module is used for extraction fractal image from watermarking image. The whole processes are described in section (4.4).

Some times watermarking image might be attacked and effected in a digital method such as: compression JPEG, filters or add noise …etc. So the proposed method has the ability to resist the attack and extract the fractal image from the watermarking image.

4.2 The Proposed Digital Watermarking Algorithm

The proposed watermarking algorithm consists of many stages represented in scheme as shown in figure (4.1) and the stages of algorithm:

Stage1: generate the fractal image by a Midpoint Displacement Method at size (64x64).

Stage2: convert the generated fractal image into a sequence of bits (0 & 1).

Stage3: transform the sequence of bits into 2-level (±Factor).

Stage4: select a cover image of type BMP.

Stage5: apply Haar wavelet transform on to the cover image.

Stage6: add the 2-level of fractal image into the coefficients of cover image.

Stage7: apply the inverse Haar wavelet transform.

Stage8: display watermarking image.

Stage9: attack by JPEG2000 compression watermarking image, then the fractal image should be not be affected.

Stage10: extract the watermark image.

Stage11: display the fractal image.

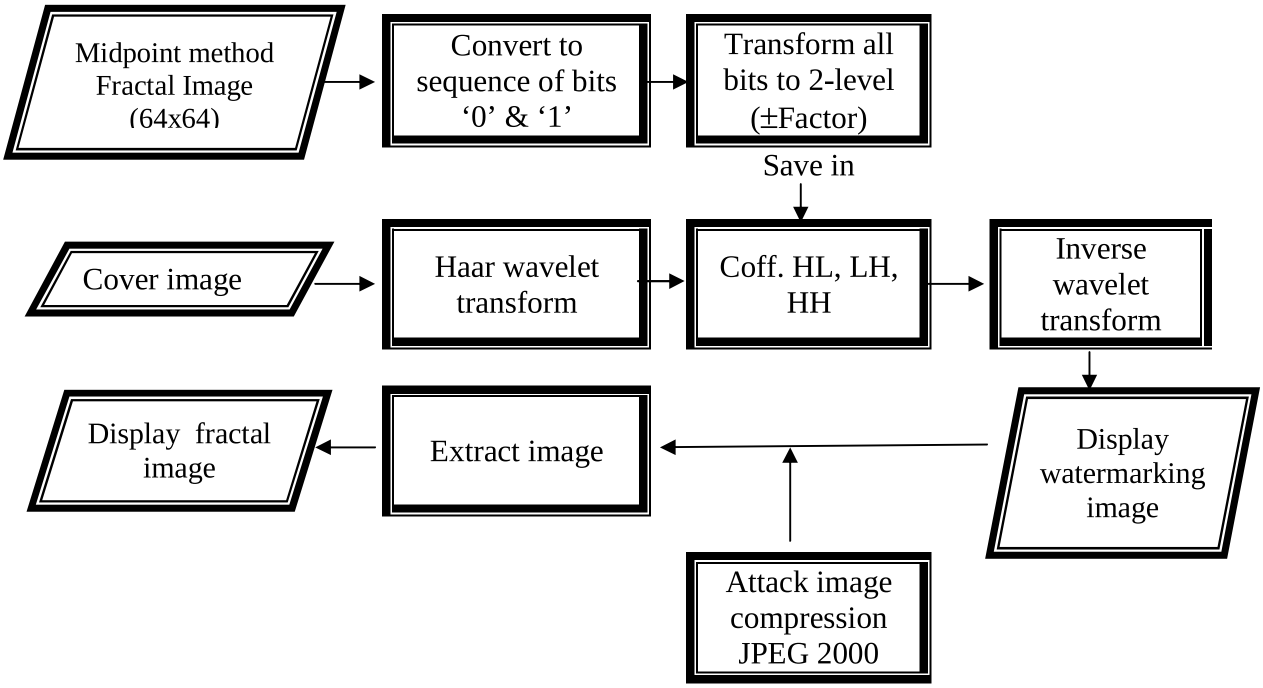

Midpoint method
Fractal Image
(64x64)
Convert to sequence of bits '0' & '1'
Transform all bits to 2-level (±Factor)
Save in
Cover image
Haar wavelet transform
Coff. HL, LH, HH
Inverse wavelet transform
Display fractal image
Extract image
Display watermarking image
Attack image compression JPEG 2000

Figure (4.1): General block diagram of (embedding and extracting) modules.

4.3 Random Midpoint Displacement Method

This method used to generate the fractal image used as a watermark, the principles of this method as follows:

Step 1: Apply Gaussian Random Generation algorithm to find values of all points in fractal image below:

```
Function  GetRandomGeneraion
    Sum=0;  for I=0 to 11  Sum = Sum +Rnd;  Next I;
    GetRandomGeneraion = (Sum / 12  -0.5) * 12.00014555 * Val;
End Function
```

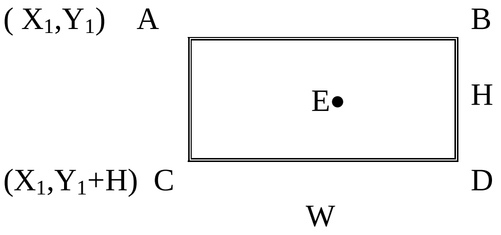
Step 2: An initial square of image

F(x1, y1)   = GetRandomGeneraion(1)

F(x1+w, y1)   = GetRandomGeneraion(1)

F(x1, y1+H)   = GetRandomGeneraion(1)

F(x1+w, y1+H) = GetRandomGeneraion(1)



Figure(4.2): Represent the middle Point algorithm.

Step 3: We add one vertex in to middle. Its vertex is denoted by

$$X_C = X_1 + W / 2$$

$$Y_C = Y_1 + H / 2$$

$$F(X_C, Y_C) = (A + B + C + D) / 4 + \text{GetRandomGenerator}$$

Step 4: We repeat step 3 for each sub square with applied algorithm as below in:

```
While Sx > 0 And Sy > 0
      Sx = Wt / 2: Sy = Ht / 2: Ys = 0: Ye = Wt: Yc = Sy
        While Ye <= Hm
            Xs = 0: Xe = Ht: Xc = Sx
            While Xe <= Wm
```

```
                A = F(Xs, Ys); B = F(Xe, Ys)
                C = F(Xs, Ye); D = F(Xe, Ye)
                E = (A + B + C + D) / 4 + GetRandomGeneraion (Std)
                F(Xc, Yc) = E
                F(Xc, Ys) = (A + B + E) / 3 + GetRandomGeneraion (Std)
                F(Xc, Ye) = (C + D + E) / 3 + GetRandomGeneraion (Std)
                F(Xs, Yc) = (A + C + E) / 3 + GetRandomGeneraion (Std)
                F(Xe, Yc) = (B + D + E) / 3 + GetRandomGeneraion (Std)
                Xs = Xe; Xe = Xe + Wt; Xc = Xc + Wt
                Wend
          Ys = Ye; Ye = Ye + Ht; Yc = Yc + Ht;
          Wend
        Wt = Sx; Ht = Sy; Std = Std / Sqr(2 ^ (H * (I + 1))); I = I + 1;
Do
```

Step 5: Combine (Red, Green and Blue) components with the header of the image to construct fractal image.

Step 6: We display fractal image which results in step (5). This image includes:

a- Size equal (64*64).

b- Bitmap (BMP) image format.

c- Type of 24 bit.

4.4 Embedded Module

The embedding module is used for hiding fractal image which constructed from applied Random Midpoint Displacement Algorithm and it is demonstrated by:

a. 24 bit color or true color.

b. Size of fractal image (64*64) pixels.

All bits=64*64 *24= (98304) bits.

c. We enter the High_Dimensional factor's value which determines the characterizations image.

The embedded module uses cover image demonstrated by:

a. 24 bit color or true color.

b. Size of Cover image greater than (218*218) pixels.

Generally, cover and fractal image are BMP pictures, which utilize RGB color model. They are divided into three components (Red, Green and Blue).

4.4.1 Algorithmic Steps to construct watermarking image

Steps 1: Apply "Random Midpoint displacement method" to construct fractal image (64*64), it is described in section (4.3).

Step 2: Choose cover image and check it, it must satisfy the following:

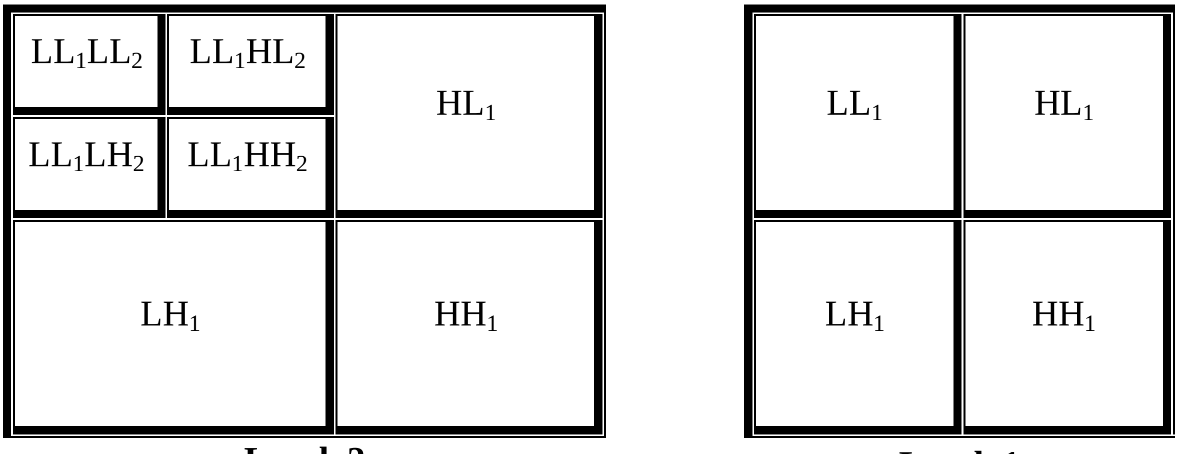
a- Bitmap image format (BMP).

b- 24-bit or true color type.

c- Size of image greater than (218*218).

Step 3: Input and check

a- Number of Levels (1-3).



Level 2 **Level 1**

Figure (4.3): Represent the coefficients for 1 and 2 level.

b- Watermark image Name and it must be bitmap (BMP) format.

c- Magnitude factor's range is (± (1-50)), it will be minus value when the embedded bit equal '0' and plus value when the embedded bit equal '1'.

Step 4: Convert all components data (Red, green, blue) fractal image to sequence of bits ('0' and '1').

Number bits = width × height × 24

= 64 × 64 × 24

= 98304 bits.

Step 5: We take Red component in first stage from cover image and execute Haar Wavelet Transform which contains "Low Pass Filter LPF and High Pass Filer HPF".

The results are four sub bands as follows:

a- Approximation image band Low Low (LL) component.

b- Horizontal image band High Low (HL) component.

c- Vertical image band Low High (LH) component.

d- Diagonal image band High High (HH) component.

The principle of the Haar wavelet transform is to calculate average and difference for values of neighboring data, we execute the equations and algorithms below to find all coefficients (LL, HL, LH and HH),as shown in figure (4.4):

```
Int k, D, X, Y;
K=0;
For(Y=0; Y<BMP.Height; Y+=2)
        D=0;
        For(X=0; X<BMP.Width; X+=2)
        D=0;
        L1=BUFD[Y][X] + BUFD[Y][X+1];
        L2=BUFD[Y+1][X]+BUFD[Y+1][X+1];
        LL[k][d]=(L1+L2) / 4;
        LH[k][d]=(L1-L2) / 4;
        H1=BUFD[Y][X] - BUFD[Y][X+1];
        H2=BUFD[Y+1][X]-BUFD[Y+1][X+1];
        HL[k][d]=(H1+H2) / 4;
        HH[k][d]=(H1-H2) / 4 ;
        D++;
k++;
```

LL **Approximate**	**HL** **Horizontal**
LH **Vertical**	**HH** **Diagonal**

Figure (4.4): Refers to all coefficients and concepts.

<u>Step 6</u>: Save bits of one component which resulted from step (4) in coefficients (HL, LH, HH) for cover image as follow:

a- All bits mapped to values of magnitude factor, where as the bit equal ‘0’ the sign of magnitude of factor is minus, and when it equal ‘1’ sign is plus.

b- Choose the same location in (LH, HL ,HH) coefficients for check and embedded 3 bits and this means adding (± magnitude factor) to any above coefficients.

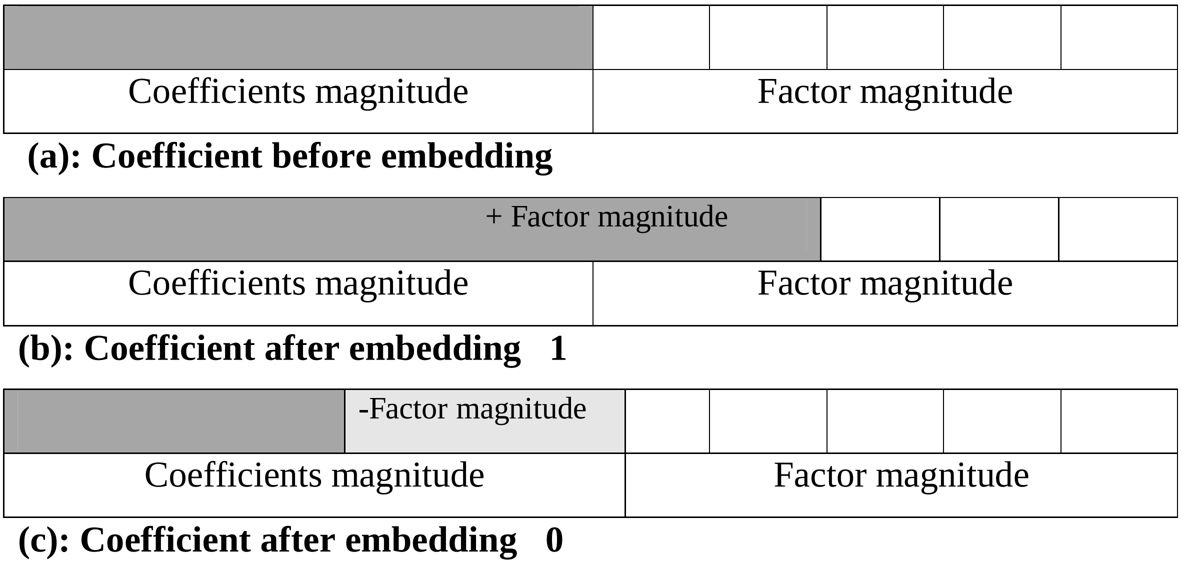
c- As shown in figure(4.5), the addition or subtraction magnitude of factor to coefficients according to:

If embedded bit equal ‘1’:

New coefficient = original coefficient + magnitude factor

If embedded bit equal ‘0’:

New coefficient = original coefficient - magnitude factor



(a): Coefficient before embedding

(b): Coefficient after embedding 1

(c): Coefficient after embedding 0

Figure (4.5): The wavelet coefficient value before and after adding magnitude factor ‘0’ or ‘1’.

d- Repeat operation in steps (b and c) above until for all bits of Red component of fractal image,

I.e. embedded number bits equal (Width × Height × Byte) = (64 × 64 × 8)

<u>Step 7</u>: Rearrange coefficients (LL, HL, LH and HH) to construct level image in form Gray scale for only Red component.

<u>Step 8</u>: Execute Inverse Haar Wavelet Transform to replace Red component of watermarking image after embedding all bits from the essential Red component of fractal image as follow:

a- distribute the coefficients (LL, HL, LH and HH) according to figure (4.6):

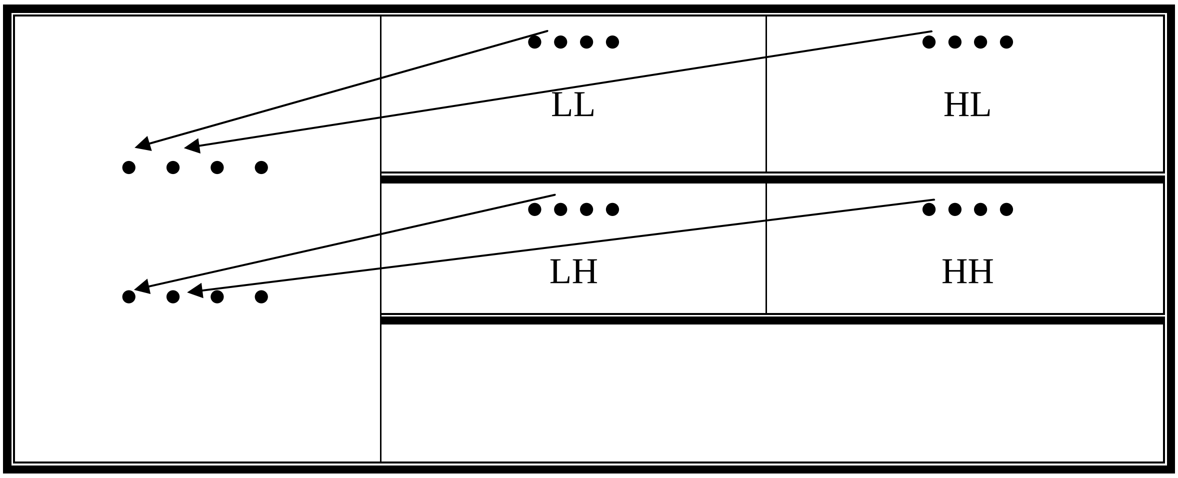


Figure (4.6): The position of the set of Permutation the coefficients (LL, HL, LH and HH) in Inverse Haar Wavelet transform.

Get the value of the coefficient (LL) in location (X, Y), (HL) coefficients in location (X+1, Y), (LH) in location (X, Y+1), and (HH) in location (X+1, Y+1) for all data of image.

b- Apply the code below to calculate coefficients (ILL, IHL, ILH and IHH).

```
Int k, D, X, Y;
K=0;
For(Y=0; Y<BMP.Height; Y+=2)
   D=0;
        For(X=0; X<BMP.Width; X+=2)
        D=0;
           L1=BUFD[Y][X] + BUFD[Y][X+1];
           L2=BUFD[Y+1][X]+BUFD[Y+1][X+1];
           LL[k][d]=(L1+L2) ;
           LH[k][d]=(L1-L2) ;
```

```
                H1=BUFD[Y][X] - BUFD[Y][X+1];
                H2=BUFD[Y+1][X]-BUFD[Y+1][X+1];
                HL[k][d]=(H1+H2) ;
                HH[k][d]=(H1-H2) ;
          D++;
    k++;
```

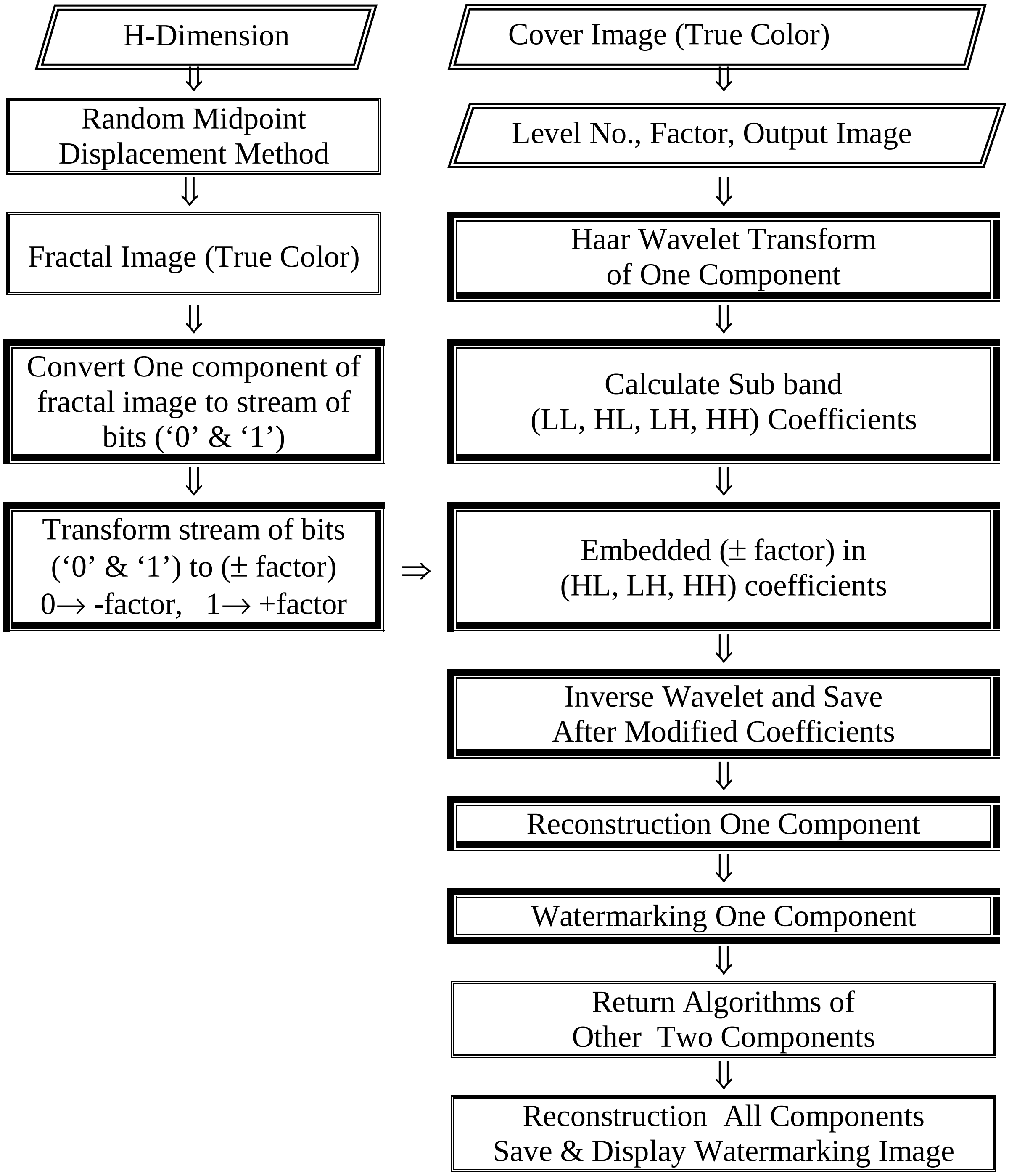
c- Repeat step (8-a) to construct and save watermarked image.

Step 9: Repeat steps (6, 7 and 8) for components (Green and Blue) of cover and fractal image and construct Gray image of (Green and Blue) components.

Step 10: After embedding fractal image in coefficients of cover image, combine (Red Green and Blue) components with the header of the image to construct watermarked image.

Step 11: Display all images (cover, fractal, level, watermarking) in output.

4.4.2 Flow Diagram of the embedding watermark method


H-Dimension
Random Midpoint Displacement Method
Fractal Image (True Color)
Convert One component of fractal image to stream of bits (‘0’ & ‘1’)
Transform stream of bits (‘0’ & ‘1’) to (± factor)
0→ -factor, 1→ +factor
Cover Image (True Color)
Level No., Factor, Output Image
Haar Wavelet Transform of One Component
Calculate Sub band (LL, HL, LH, HH) Coefficients
Embedded (± factor) in (HL, LH, HH) coefficients
Inverse Wavelet and Save After Modified Coefficients
Reconstruction One Component
Watermarking One Component
Return Algorithms of Other Two Components
Reconstruction All Components Save & Display Watermarking Image

4.5 Extraction Module

This module is used for extraction all data of fractal image from the coefficients of watermarking image.

The process of retrieving the fractal image depends on the calculating the coefficients of the watermarked and cover image, then find the difference between same coefficient (HL, LH, HH) in the same location and suggest if ‘0’ or ‘1’. The steps of the watermark image extraction module are shown in figure (4.6) and the stages are explained in the following steps:

4.5.1 Algorithm Steps Extraction Module

Step 1: Choose and check the watermark and cover image, magnitude factor demonstrated by:

a- Bitmap image format (BMP).

b- 24-bit or true color type.

c- Size of image greater than (218*218).

Step 2: Take the Red component of watermarked image and execute Haar wavelet on the red data, then calculate the coefficients (LLW, HLW, LHW and HHW), it is explained in section (4.4.1, step 5).

Step 3: Take the Red component of cover image and execute Haar wavelet on the image, then calculate the coefficients (LLC, HLC, LHC and HHC), it is explained in section (4.4.1, step 5).

Step 4: Find the difference between two coefficients of the same location to these bands (IHLW with IHLC, ILHW with ILHC and IHHW with IHHC), when the difference greater than zero the output bit is ‘1’ else the output bit is ‘0’; the following code represents the process.

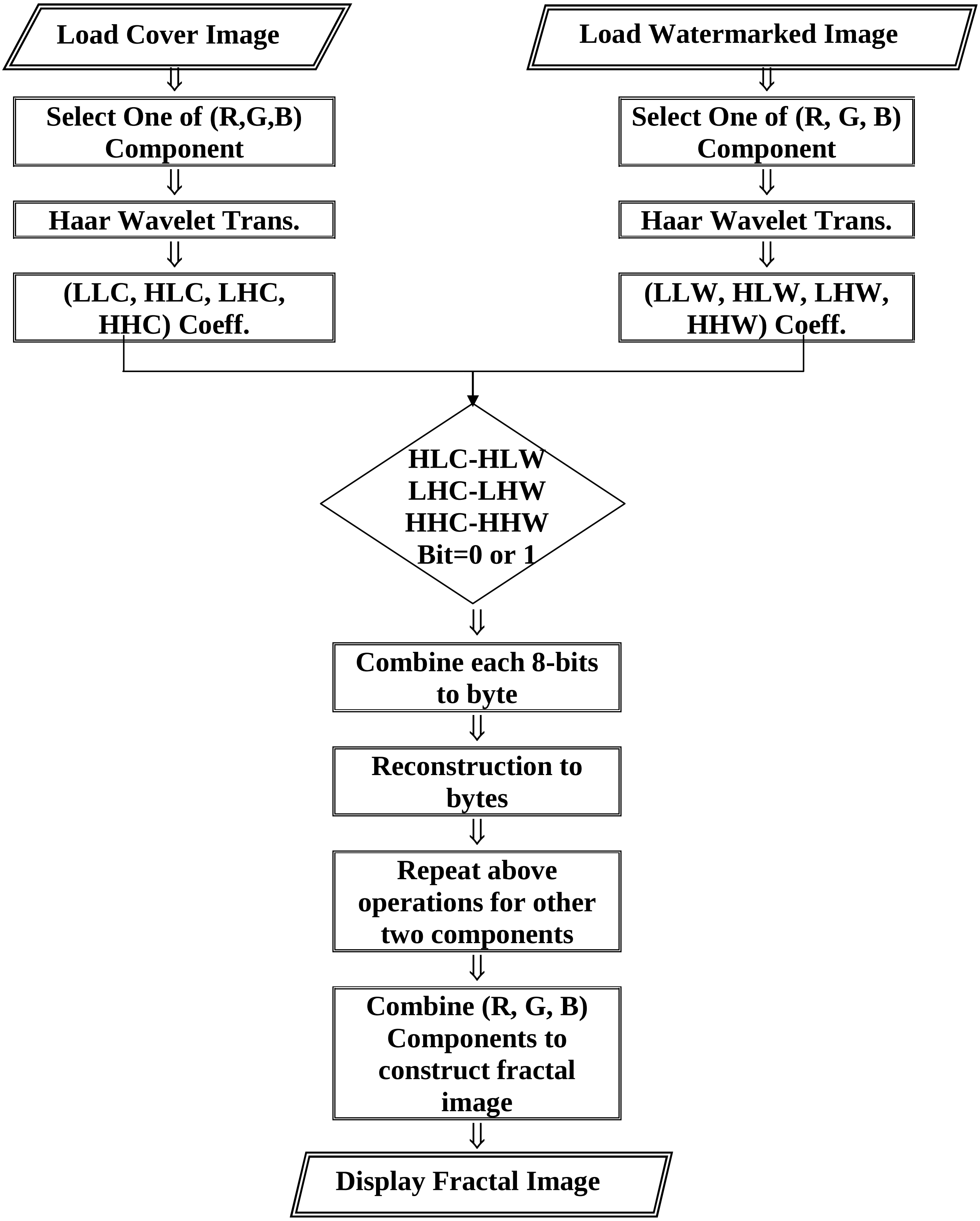
```
EndLoc=0;EE=0;
for(i=0;i<NDD;i++)
    for(E=0;E<NWW;E++)
        Diff=ILHC[i][E]-ILHW[i][E];
        if(Diff>0) Out[ii][EE]=1; else Out[ii][EE]=0;
        EE++;if(EE>=8){EE=0;ii++;}
        EndLoc++;
        Diff=IHLC[i][E]-IHLW[i][E];
        if(Diff>0)Out[ii][EE]=1; else Out[ii][EE]=0;
        EE++;if(EE>=8) EE=0;ii++;
        EndLoc++;
        Diff=IHHC[i][E]-IHHW[i][E];
        if(Diff>0)Out[ii][EE]=1; else Out[ii][EE]=0;
        EE++;if(EE>=8) EE=0;ii++;
        EndLoc++;
        if (EndLoc>=32768){E=NWW;i=NDD;EndLoc=0;
```

Step 5: The sequence of bits resulted from step 4 are transformed to bytes, which is the constructed Red component of output image.

Step 6: Repeat steps (2, 3, 4, and 5) in order to find the (Green and Blue) components of the output image.

Step 7: Combine the three components Red, Green and Blue with header image to construct output bitmap and true color image (64×64) and display.

4.5.2 Flow Diagram represents Extraction Watermarking Image


Load Cover Image
Load Watermarked Image
Select One of (R,G,B) Component
Select One of (R, G, B) Component
Haar Wavelet Trans.
Haar Wavelet Trans.
(LLC, HLC, LHC, HHC) Coeff.
(LLW, HLW, LHW, HHW) Coeff.
HLC-HLW
LHC-LHW
HHC-HHW
Bit=0 or 1
Combine each 8-bits to byte
Reconstruction to bytes
Repeat above operations for other two components
Combine (R, G, B) Components to construct fractal image
Display Fractal Image

4.6 Practical investigation

The system should be tested by using many different measurements to investigate the capability of the system and its sufficiency to use.

4.6.1 Criteria for Evaluation

Since we are looking for suitable decomposition scheme and transform type that can satisfy the following:

a. Large number of bits can be embedded.

b. Higher ratio of compression can be applied.

c. Less shift factor.

To investigate the proposed system, we should tested by using standard measurements such as Mean Square Error (MSE), Signal to Noise Ratio (SNR), Peak Signal to Noise Ratio (PSNR) and Mean Absolute Error (MAE) also measure the number of different bits of fractal image after extraction.

When the shift factor increase more degradation occurs in image, which will increase the value of the Mean Square Error (MSE) and decreases the value of Peak Signal to Noise Ratio (PSNR), these measures have been were adopted in our study as an objective distortion measures,

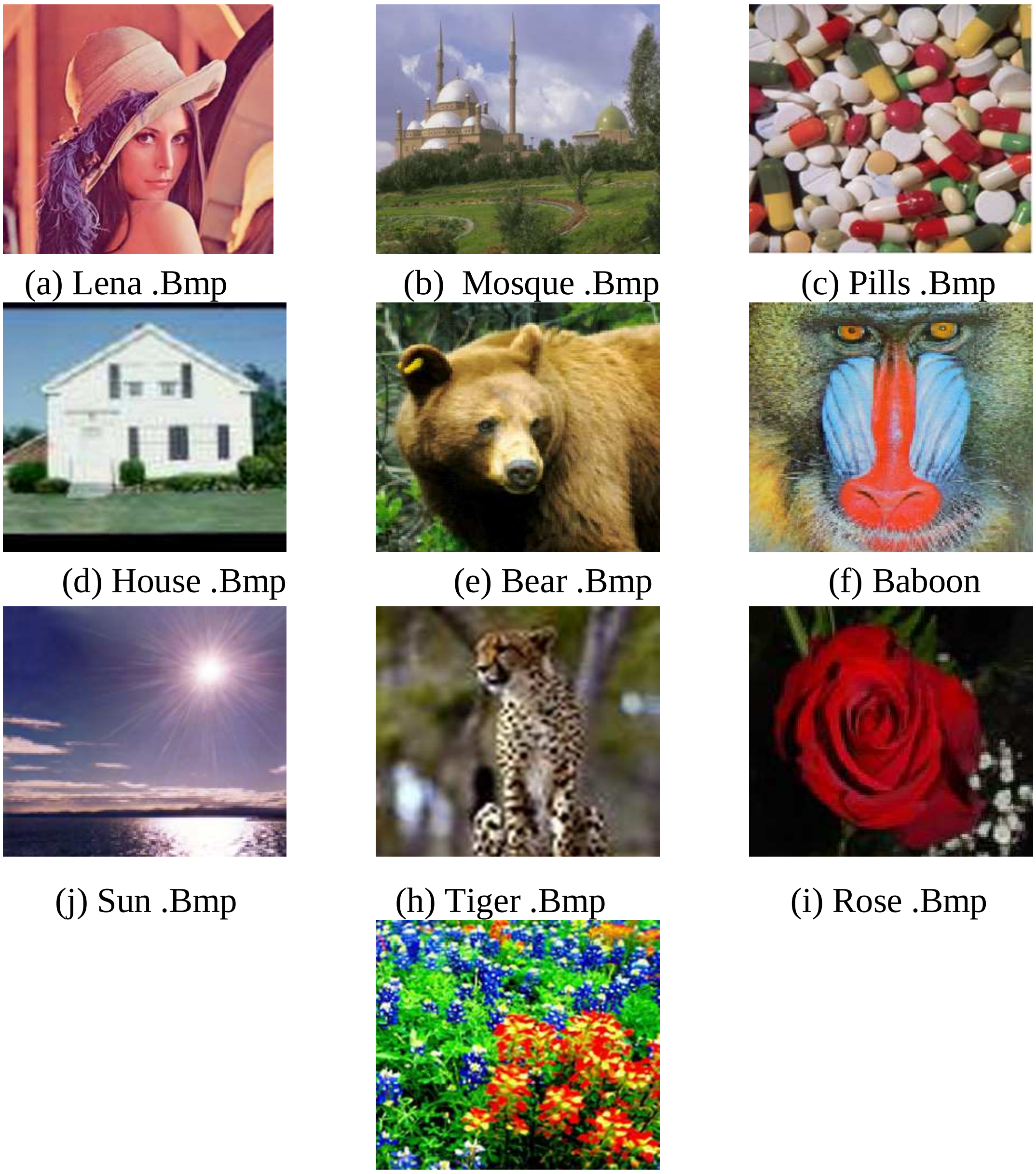
$$MSE = \frac{\sum_{x,y}[O(x,y)-R(x,y)]^2}{W \times H} \qquad .....(4.1)$$

$$PSNR = 10\log_{10}\frac{(255)^2}{MSE} \qquad .....(4.2)$$

Where O(x,y) represents the value of the color component of the original image at pixel (x,y), and R(x,y) represents the corresponding value in the watermarked image. W &H represent the width and height of image respectively. Generally the values of PSNR above (38)db are visually satisfactory, even for the professionals.

4.6.2 Packet Decomposition with Haar Transform

The flow diagrams given in sections (4.3.2) and (4.4.2) show the steps of decomposition and reconstruction, of an image using Haar transform. A set of images were tested by these diagrams, and these tested images are shown in figure (4.7); tables (4.1) to (4.10) shows the results obtained from system implementation.



(a) Lena .Bmp (b) Mosque .Bmp (c) Pills .Bmp

(d) House .Bmp (e) Bear .Bmp (f) Baboon

(j) Sun .Bmp (h) Tiger .Bmp (i) Rose .Bmp

(j) Flowers .Bmp

Figure (4.7): The tested images of type bitmap.

The best results as shown occur when the quality of the compressed watermarked image is greater or equal 88.0, Thr. represent ±magnitude factor, Comp. size represent the size of image after JPEG2000 compression process, Comp. ratio represent the ratio of compression to tested image and the ratio of different bits of fractal image that affected after JPEG2000 compression .

Table (4.1): The test results of image (Lena .Bmp).

Lena .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	95.1	1/2	0.707168	198.034535	49.635578	0.500086	0.000000
	90.0	95.1	1/2	0.707168	198.034535	49.635578	0.500086	0.000000
	88.0	64.0	1/3	1.303892	107.404350	46.978387	0.977020	0.195597
	87.0	34.4	1/5	2.006807	69.784340	45.105749	1.539256	0.382874
5	100.0	111.9	1/1	3.534734	39.619296	42.647236	2.499288	0.000000
	90.0	111.9	1/1	3.534734	39.619296	42.647236	2.499288	0.000000
	88.0	64.1	1/2	4.031352	34.738640	42.076297	3.223902	0.002228
	87.0	34.8	1/5	4.351434	32.183338	41.744480	3.394023	0.116323
7	100.0	117.6	1/1	4.947778	28.304358	41.186702	3.498093	0.000000
	90.0	117.6	1/1	4.947778	28.304358	41.186702	3.498093	0.000000
	88.0	64.0	1/3	5.430842	25.786733	40.782132	4.281214	0.000671
	87.0	34.7	1/5	5.745572	24.374190	40.537471	4.493556	0.071075
9	100.0	122.2	1/1	6.359765	22.020258	40.096393	4.496053	0.000000
	90.0	122.2	1/1	6.359765	22.020258	40.096393	4.496053	0.000000
	88.0	64.1	1/2	6.887464	20.333126	39.750210	5.400976	0.000224
	87.0	35.0	1/5	7.202928	19.442602	39.555713	5.625534	0.049032
15	100.0	132.2	1/1	10.582756	13.233195	37.884816	7.478175	0.000000
	90.0	132.2	1/1	10.582756	13.233195	37.884816	7.478175	0.000000
	88.0	64.0	1/2	11.084140	12.634599	37.683783	8.560176	0.000102
	87.0	34.9	1/5	11.445459	12.235740	37.544471	8.795965	0.034251

Table (4.2): The test results of image (Mosque .Bmp).

Mosque .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	92.7	1/2	0.707394	190.004971	49.634187	0.500407	0.000000
	90.0	92.7	1/2	0.707394	190.004971	49.634187	0.500407	0.000000
	88.0	63.9	1/3	1.265374	106.220360	47.108615	0.944524	0.181498
	87.0	34.8	1/5	1.852240	72.565378	45.453832	1.403773	0.369639
5	100.0	107.8	1/1	3.535671	38.014984	42.646086	2.500661	0.000000
	90.0	107.8	1/1	3.535671	38.014984	42.646086	2.500661	0.000000
	88.0	64.1	1/2	3.972008	33.838922	42.140703	3.142110	0.001027
	87.0	34.8	1/5	4.288257	31.343381	41.807996	3.304387	0.093658
7	100.0	113.2	1/1	4.949877	27.153899	41.184860	3.500641	0.000000
	90.0	113.2	1/1	4.949877	27.153899	41.184860	3.500641	0.000000
	88.0	63.9	1/3	5.388537	24.943407	40.816095	4.224040	0.000153
	87.0	34.9	1/5	5.776627	23.267637	40.514060	4.435989	0.060791
9	100.0	117.5	1/1	6.364088	21.119830	40.093442	4.500519	0.000000
	90.0	117.5	1/1	6.364088	21.119830	40.093442	4.500519	0.000000
	88.0	63.9	1/3	6.800754	19.763758	39.805233	5.264267	0.000020
	87.0	34.9	1/5	7.195027	18.680744	39.560479	5.517980	0.042582
15	100.0	127.1	1/1	10.606791	12.671925	37.874963	7.500814	0.000000
	90.0	127.1	1/1	10.606791	12.671925	37.874963	7.500814	0.000000
	88.0	64.1	1/2	11.094492	12.114882	37.679730	8.460526	0.000000
	87.0	34.9	1/5	11.531430	11.655837	37.511972	8.740504	0.026601

Table (4.3): The test results of image (Pills .Bmp)

pills .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	104.3	1/1	0.725084	200.894624	49.526921	0.506205	0.008626
	90.0	104.3	1/1	0.725084	200.894624	49.526921	0.506205	0.008626
	88.0	63.9	1/3	1.603343	90.851081	46.080539	1.221069	0.255280
	87.0	34.9	1/5	2.647315	55.023853	43.902748	2.035721	0.406738
5	100.0	118.8	1/1	3.535992	41.195075	42.645691	2.500107	0.000061
	90.0	118.8	1/1	3.535992	41.195075	42.645691	2.500107	0.000061
	88.0	64.0	1/2	4.158977	35.024344	41.940938	3.330755	0.013967
	87.0	34.8	1/5	4.614082	31.569758	41.489950	3.632863	0.157349
7	100.0	123.9	1/1	4.944867	29.457914	41.189258	3.494481	0.000071
	90.0	123.9	1/1	4.944867	29.457914	41.189258	3.494481	0.000071
	88.0	64.1	1/2	5.596387	26.028481	40.651726	4.492411	0.002899
	87.0	34.9	1/5	6.038648	24.122197	40.321406	4.786296	0.097178
9	100.0	128.2	1/1	6.353941	22.925213	40.100372	4.488485	0.000071
	90.0	128.2	1/1	6.353941	22.925213	40.100372	4.488485	0.000071
	88.0	64.1	1/2	6.962554	20.921269	39.703118	5.577494	0.001668
	87.0	34.5	1/5	7.364679	19.778929	39.459265	5.808462	0.076274
15	100.0	137.6	1/1	10.576822	13.772139	37.887252	7.466426	0.000041
	90.0	137.6	1/1	10.576822	13.772139	37.887252	7.466426	0.000041
	88.0	63.9	1/3	11.170519	13.040169	37.650070	8.761724	0.001414
	87.0	34.9	1/5	11.652377	12.500922	37.466658	9.110209	0.045247

Table (4.4): The test results of image (House .Bmp)

House .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	63.2	1/3	0.695276	228.694421	49.709232	0.483409	0.000102
	90.0	63.2	1/3	0.695276	228.694421	49.709232	0.483409	0.000102
	88.0	63.2	1/3	0.695276	228.694421	49.709232	0.483409	0.000102
	87.0	34.9	1/5	1.069388	148.688552	47.839451	0.751765	0.243022
5	100.0	91.4	1/2	3.438789	46.238878	42.766749	2.377777	0.000102
	90.0	91.4	1/2	3.438789	46.238878	42.766749	2.377777	0.000102
	88.0	64.1	1/2	3.733582	42.587980	42.409546	2.796743	0.015818
	87.0	34.9	1/5	4.045634	39.303039	42.060937	3.106435	0.064667
7	100.0	98.4	1/1	4.804552	33.094808	41.314275	3.315450	0.000102
	90.0	98.4	1/1	4.804552	33.094808	41.314275	3.315450	0.000102
	88.0	63.9	1/3	5.128599	31.003736	41.030816	3.842046	0.017517
	87.0	34.9	1/5	5.467362	29.082705	40.753026	4.185298	0.054454
9	100.0	103.5	1/1	6.168740	25.776045	40.228839	4.251099	0.000102
	90.0	103.5	1/1	6.168740	25.776045	40.228839	4.251099	0.000102
	88.0	64.2	1/2	6.559318	24.241199	39.962217	4.897105	0.018982
	87.0	34.9	1/5	6.841267	23.242147	39.779439	5.206390	0.051982
15	100.0	114.2	1/1	10.235167	15.535236	38.029854	7.032501	0.000102
	90.0	114.2	1/1	10.235167	15.535236	38.029854	7.032501	0.000102
	88.0	64.0	1/2	10.631783	14.955697	37.864743	7.864522	0.021942
	87.0	34.9	1/5	10.889084	14.602304	37.760890	8.24091	0.045949

Table (4.5): The test results of image (Bear .Bmp)

Bear .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	114.8	1/1	0.700781	159.516850	49.674981	0.475093	0.014038
	90.0	114.8	1/1	0.700781	159.516850	49.674981	0.475093	0.014038
	88.0	63.7	1/3	1.901409	58.791332	45.340048	1.435109	0.313446
	87.0	34.9	1/5	3.306701	33.806013	42.936855	2.504389	0.421773
5	100.0	123.2	1/1	3.430305	32.587880	42.777476	2.352127	0.005371
	90.0	123.2	1/1	3.430305	32.587880	42.777476	2.352127	0.005371
	88.0	63.9	1/3	4.186671	26.700539	41.912115	3.314275	0.050873
	87.0	34.9	1/5	4.911136	22.761818	41.218984	3.852865	0.194539
7	100.0	127.3	1/1	4.799038	23.293495	41.319262	3.291041	0.005697
	90.0	127.3	1/1	4.799038	23.293495	41.319262	3.291041	0.005697
	88.0	64.1	1/2	5.500826	20.321742	40.726524	4.367813	0.034810
	87.0	34.7	1/5	6.105400	18.309428	40.273663	4.795670	0.144440
9	100.0	131.1	1/1	6.154167	18.164339	40.239111	4.216502	0.004588
	90.0	131.1	1/1	6.154167	18.164339	40.239111	4.216502	0.004588
	88.0	64.1	1/2	6.804904	16.427326	39.802584	5.377131	0.031443
	87.0	34.6	1/5	7.383304	15.140427	39.448296	5.818604	0.112915
15	100.0	140.1	1/1	10.225722	10.931881	38.033864	6.992142	0.003296
	90.0	140.1	1/1	10.225722	10.931881	38.033864	6.992142	0.003296
	88.0	64.0	1/3	10.863675	10.289923	37.771036	8.397741	0.032725
	87.0	34.8	1/5	11.349351	9.849582	37.581093	8.888646	0.080831

Table (4.6): The test results of image (Baboon .Bmp)

baboon .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	126.1	1/1	0.725007	188.354823	49.527383	0.506083	0.008331
	90.0	126.1	1/1	0.725007	188.354823	49.527383	0.506083	0.008331
	88.0	64.0	1/3	2.438895	55.991972	44.258873	1.888250	0.339742
	87.0	34.9	1/5	4.327701	31.554515	41.768231	3.351344	0.446737
5	100.0	133.3	1/1	3.550099	38.466119	42.628399	2.513855	0.000936
	90.0	133.5	1/1	3.550099	38.466119	42.628399	2.513855	0.000936
	88.0	64.1	1/2	4.447118	30.707196	41.650017	3.607646	0.037699
	87.0	34.6	1/5	5.413058	25.227609	40.796377	4.257589	0.263794
7	100.0	137.0	1/1	4.962119	27.520202	41.174132	3.513585	0.000936
	90.0	137.1	1/1	4.962119	27.520202	41.174132	3.513585	0.000936
	88.0	64.0	1/2	5.797819	23.553428	40.498157	4.692128	0.012889
	87.0	34.8	1/5	6.570111	20.784810	39.955076	5.202159	0.192332
9	100.0	140.2	1/1	6.375207	21.420248	40.085861	4.512812	0.000936
	90.0	140.2	1/1	6.375207	21.420248	40.085861	4.512812	0.000936
	88.0	63.9	1/3	7.193615	18.983294	39.561331	5.799413	0.006989
	87.0	34.7	1/5	7.953101	17.170475	39.125439	6.314280	0.147217
15	100.0	148.2	1/1	10.607926	12.873253	37.874499	7.501394	0.000936
	90.0	148.2	1/1	10.607926	12.873253	37.874499	7.501394	0.000936
	88.0	64.1	1/2	11.438070	11.938947	37.547276	9.110357	0.001190
	87.0	34.9	1/5	12.200839	11.192551	37.266907	9.690923	0.072286

Table (4.7): The test results of image (Sun .Bmp)

Sun .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	82.9	1/2	0.707847	191.447723	49.631408	0.501038	0.003286
	90.0	82.9	1/2	0.707847	191.447723	49.631408	0.501038	0.003286
	88.0	64.1	1/2	1.055032	128.447020	47.898146	0.783056	0.113159
	87.0	34.9	1/5	1.572053	86.203068	46.166133	1.190384	0.330963
5	100.0	102.2	1/1	3.513956	38.565013	42.672841	2.477809	0.003276
	90.0	102.2	1/1	3.513956	38.565013	42.672841	2.477809	0.003276
	88.0	63.6	1/3	3.888976	34.846125	42.232451	3.055033	0.005910
	87.0	35.0	1/5	4.181723	32.406674	41.917250	3.274144	0.062236
7	100.0	108.3	1/1	4.911248	27.592936	41.218885	3.458928	0.003276
	90.0	108.3	1/1	4.911248	27.592936	41.218885	3.458928	0.003276
	88.0	64.0	1/3	5.328246	25.433462	40.864961	4.148488	0.005809
	87.0	34.6	1/5	5.606455	24.171379	40.643920	4.361923	0.051158
9	100.0	113.0	1/1	5.606455	24.171379	40.643920	4.361923	0.051158
	90.0	113.0	1/1	5.606455	24.171379	40.643920	4.361923	0.051158
	88.0	64.1	1/2	6.723639	20.155121	39.854760	5.187398	0.006124
	87.0	34.8	1/5	7.129367	19.008104	39.600294	5.513026	0.036438
15	100.0	123.2	1/1	10.483474	12.926607	37.925751	7.359639	0.003276
	90.0	123.2	1/1	10.483474	12.926607	37.925751	7.359639	0.003276
	88.0	64.1	1/2	10.917919	12.412232	37.749405	8.275487	0.006836
	87.0	34.5	1/5	11.370663	11.918017	37.572946	8.71935	0.026031

Table (4.8): The test results of image (Tiger .Bmp)

Tiger .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	65.9	1/2	0.710240	138.988274	49.616755	0.504430	0.002157
	90.0	65.9	1/2	0.710240	138.988274	49.616755	0.504430	0.002157
	88.0	63.8	1/3	0.741773	133.079807	49.428095	0.531184	0.010610
	87.0	34.7	1/5	1.100551	89.695972	47.714703	0.782715	0.252777
5	100.0	94.0	1/2	3.519318	28.049462	42.666218	2.485545	0.002136
	90.0	94.0	1/2	3.519318	28.049462	42.666218	2.485545	0.002136
	88.0	64.0	1/3	3.844558	25.676550	42.282340	2.937190	0.004191
	87.0	34.6	1/5	4.067341	24.270150	42.037698	3.107941	0.052917
7	100.0	100.9	1/1	4.922987	20.051841	41.208517	3.473104	0.002136
	90.0	100.9	1/1	4.922987	20.051841	41.208517	3.473104	0.002136
	88.0	63.8	1/3	5.266738	18.743096	40.915386	4.024846	0.004313
	87.0	34.7	1/5	5.535321	17.833649	40.699375	4.225321	0.040782
9	100.0	106.0	1/1	6.324779	15.607656	40.120350	4.455838	0.002163
	90.0	106.0	1/1	6.324779	15.607656	40.120350	4.455838	0.002163
	88.0	64.1	1/2	6.708541	14.718421	39.864523	5.072759	0.004384
	87.0	34.9	1/5	7.022334	14.057289	39.665989	5.332006	0.031911
15	100.0	116.5	1/1	10.518379	9.384999	37.911315	7.402420	0.002136
	90.0	116.5	1/1	10.518379	9.384999	37.911315	7.402420	0.002136
	88.0	64.1	1/2	10.902169	9.054618	37.755674	8.172557	0.004852
	87.0	34.6	1/5	11.318156	8.721825	37.593047	8.575806	0.022146

Table (4.9): The test results of image (Rose .Bmp)

Rose .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	72.6	1/2	0.690560	90.660553	49.738791	0.476873	0.000000
	90.0	72.6	1/2	0.690560	90.660553	49.738791	0.476873	0.000000
	88.0	64.1	1/2	0.904455	69.220165	48.566934	0.629471	0.089162
	87.0	34.5	1/5	1.339580	46.735942	46.861117	0.960063	0.337016
5	100.0	95.8	1/2	3.355346	18.658740	42.873431	2.294011	0.000000
	90.0	95.8	1/2	3.355346	18.658740	42.873431	2.294011	0.000000
	88.0	64.0	1/3	3.692079	16.956984	42.458093	2.791361	0.021942
	87.0	34.5	1/5	3.936210	15.905282	42.180021	2.979955	0.099192
7	100.0	101.6	1/1	4.614197	13.568240	41489842	3.130941	0.000000
	90.0	101.6	1/1	4.614197	13.568240	41489842	3.130941	0.000000
	88.0	64.0	1/3	5.017107	12.478610	41.126270	3.739136	0.021444
	87.0	34.5	1/5	5.209153	12.018561	40.963132	3.924749	0.088359
9	100.0	106.1	1/1	5.849252	10.703339	40.459800	3.938980	0.000000
	90.0	106.1	1/1	5.849252	10.703339	40.459800	3.938980	0.000000
	88.0	64.1	1/2	6.265168	9.992794	40.161477	4.605891	0.023763
	87.0	34.9	1/5	6.424590	9.744829	40.052350	4.787847	0.078644
15	100.0	115.1	1/1	9.431734	6.637860	38.384888	6.227951	0.000000
	90.0	115.1	1/1	9.431734	6.637860	38.384888	6.227951	0.000000
	88.0	64.1	1/2	9.868049	6.344367	38.188491	7.048884	0.027791
	87.0	34.9	1/5	9.904217	6.321199	38.172602	7.265737	0.080129

Table (4.10): The test results of image (Flowers .Bmp)

Flowers .Bmp								
Thr.	**Quality**	**Comp. Size**	**Comp. Ratio**	**MSE**	**SNR**	**PSNR**	**MAE**	**Ratio of diff. bits**
1	100.0	141.7	1/1	0.704740	171.477576	49.650515	0.489894	0.008321
	90.0	141.7	1/1	0.704740	171.477576	49.650515	0.489894	0.008321
	88.0	64.0	1/2	3.396144	35.583619	42.820943	2.580943	0.412567
	87.0	34.6	1/5	7.739917	15.613486	39.243440	5.873322	0.472850
5	100.0	148.0	1/1	3.394108	35.604964	42.823547	2.319963	0.006571
	90.0	148.0	1/1	3.394108	35.604964	42.823547	2.319963	0.006571
	88.0	64.1	1/2	4.919390	24.565462	41.211691	3.833771	0.161326
	87.0	34.9	1/5	8.196608	14.743549	38.994462	6.256887	0.384928
7	100.0	150.6	1/1	4.718728	25.609804	41.392504	3.214874	0.006714
	90.0	150.6	1/1	4.718728	25.609804	41.392504	3.214874	0.006714
	88.0	63.9	1/3	6.152996	19.640367	40.239938	4.837962	0.103536
	87.0	34.9	1/5	8.648433	13.973293	38.761429	6.636154	0.348236
9	100.0	153.0	1/1	6.026656	20052096	40.330039	4.092804	0.006826
	90.0	153.0	1/1	6.026656	20052096	40.330039	4.092804	0.006826
	88.0	64.1	1/2	7.323010	16.502379	39.483907	5.781021	0.079091
	87.0	34.8	1/5	9.225438	13.099333	38.480934	7.109777	0.316203
15	100.0	158.8	1/1	9.854632	12.262973	38.194399	6.630163	0.006734
	90.0	158.8	1/1	9.854632	12.262973	38.194399	6.630163	0.006734
	88.0	64.2	1/2	10.931976	11.054460	34.743817	8.552689	0.055990
	87.0	34.8	1/5	12.915087	9.357048	37.019830	10.050268	0.210378

4.6.3 Evaluation the results:

Through the investigation of the test results, we implement the objectives of this system to build and apply efficient algorithm for watermarking color images by a fractal image generated through Midpoint displacement method utilizing the H-dimension as input real number. The following remarks can be conducted:

1- When the value of shift factor (Thr.) is appropriate to the modification manipulated by JPEG2000 robustness of watermark become powerful.

2- When the shift factor (Thr.) increases, Peak Signal to Noise Ratio (PSNR) decreases, in other words the watermarked image is more degraded, but it is still insignificant subjectively.

3- The ratio of different bits equal to zero when the shift factor (Thr.) equal seven in more of images.

4- When the shift factor (Thr.) increases, the quality of image is decreases.

5- By using Haar transform, we can accomplish more bits embedding with modulation factor value to satisfy the same robustness degree against JPEG2000 compression.

6- Using the Midpoint displacement method, we can generate image of any size with mean equal to zero and standard deviation equal to one.

7- The proposed system is very robust until the size of compressed image decreases into (1:3) of the original image.

8- The distorted bits is occur in any of magnitude factors range at (± (1-50)).

9- The magnitude factor's range should be not greater or smaller than the magnitude of JPEG2000 compression.

10- The proposed system embed a huge data, not message or number, it is a fractal image.

A set of images were tested by the proposed watermark system, given in section (4.3.2) and (4.4.2) the results show that the effect of the hiding system on the images:

Original image

Hide image

Extracted image

Original image

Hide image

Extracted image

Original image

Hide image

Extracted image

Original image

Hide image

Extracted image

Original image

Hide image

Extracted image

Original image

Hide image

Extracted image

Original image

Hide image

Extracted image

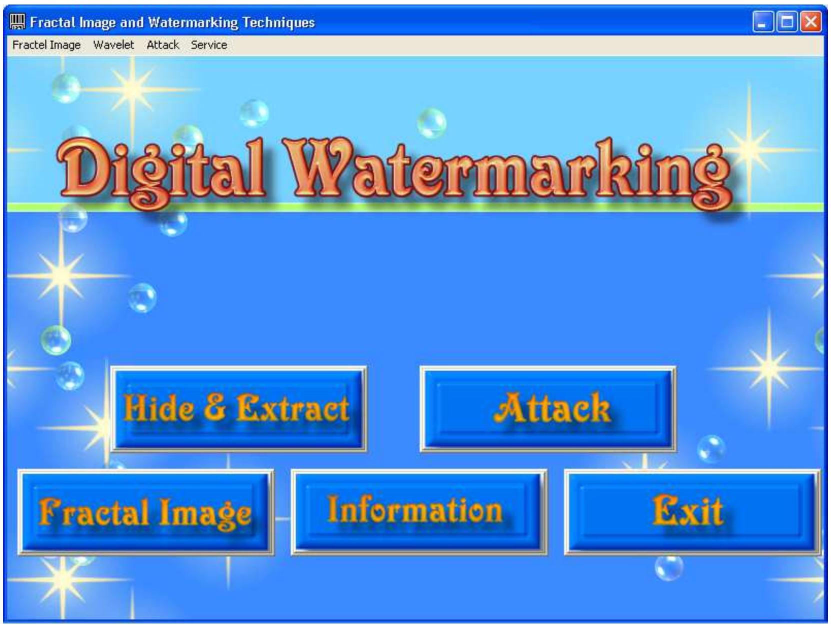
Original image

Hide image

Extracted image

4.7 Main GUI of The Proposed System

The main GUI displays the parts of Fractal Image Watermarking with Wavelet Transform System as shown in figure (4.8):

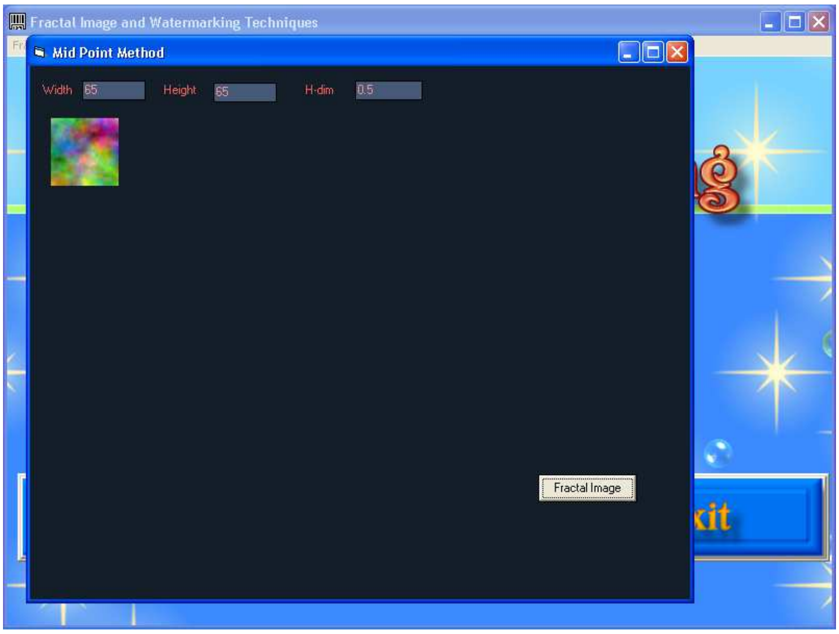


Figure(4.8) Main GUI of The Proposed System

The main window is the first window appears to the administrator when starting the Fractal Image Watermarking with Wavelet Transform System. It consists of the menu items Fractal Image, Hide and Extract, Information and Attack. Each command is explained as follows:

4.7.1 Fractal Image

When the administrator click on the “Fractal Image” command, the fractal image window will be appear as shown in figure (4.9). The fractal image contains H-dimension, Width and Height of the image, if the user needs to make image at any size or any other size he needed to use this procedure to generate it. But in this system the watermark should be at size 64*64 and the H-dim is a real number generate for this image by a Midpoint displacement Method. From this window we make the watermark as a fractal image.

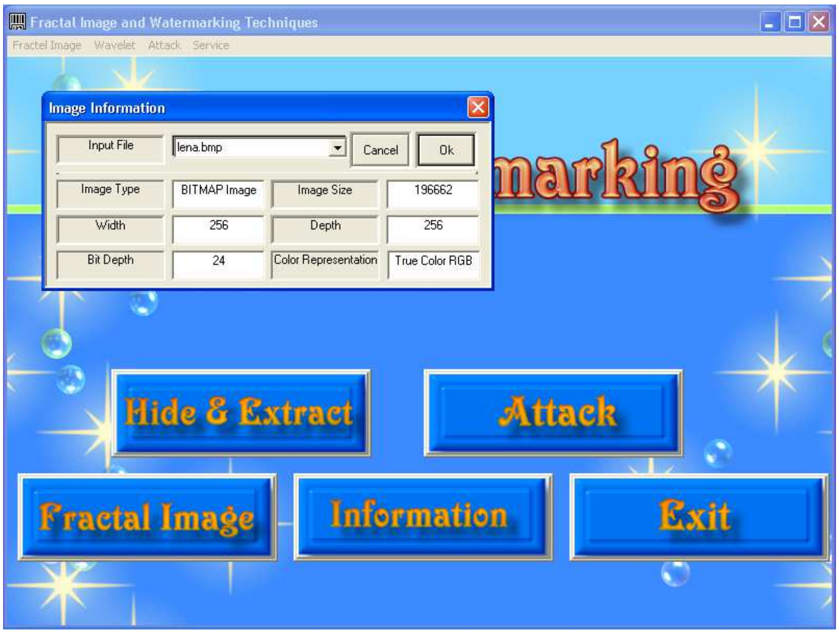


Width 65 Height 65 H-dim 0.5

Figure(4.9): Fractal Image

4.7.2 Information

This window gives information about any image that gives it. It contains the input file, image type, image size, width, depth, and bit depth and color representation as shown in figure (4.10).



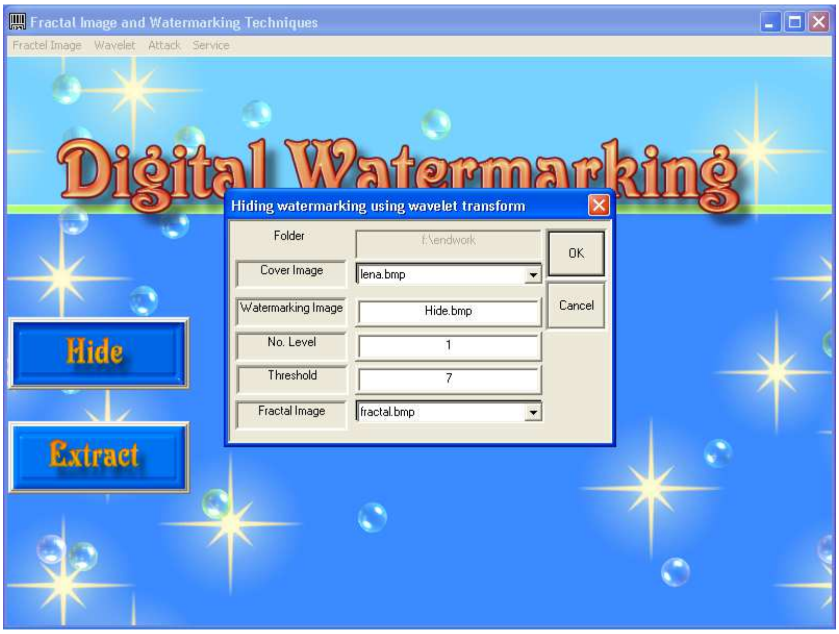
Input File	lena.bmp		
Image Type	BITMAP Image	Image Size	196662
Width	256	Depth	256
Bit Depth	24	Color Representation	True Color RGB

Figure(4.10) Information about the selected image

4.7.3 Hide and Extract

- **Hiding Image**

When the watermark image is produced as a fractal image, then the cover image is selected and the values of the level of wavelet transform, threshold and the fractal image are selected, and the result of watermarking process is the watermark image as shown in figure (4.11).


Fractal Image and Watermarking Techniques
Fractel Image
Wavelet
Attack
Service
Digital Watermarking
Hide
Extract
Hiding watermarking using wavelet transform
OK
Cancel

Folder f:\endwork

Cover Image lena.bmp

Watermarking Image Hide.bmp

No. Level 1

Threshold 7

Fractal Image fractal.bmp

Figure(4.11) Hide Watermark in Cover Image

Figure (4.12) shows as an example the result of Hiding image, and the level of Haar wavelet transform.

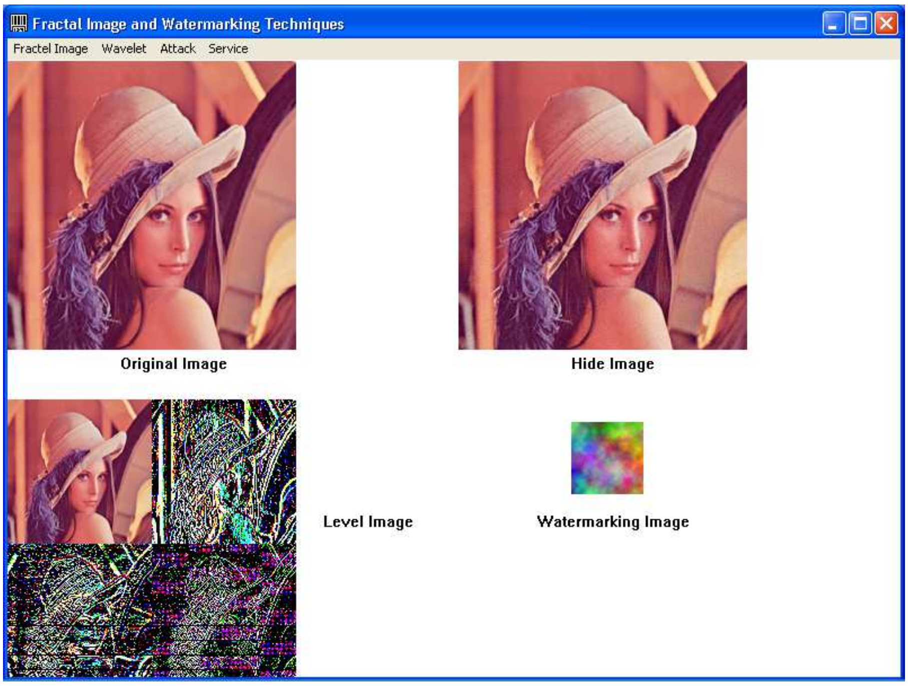
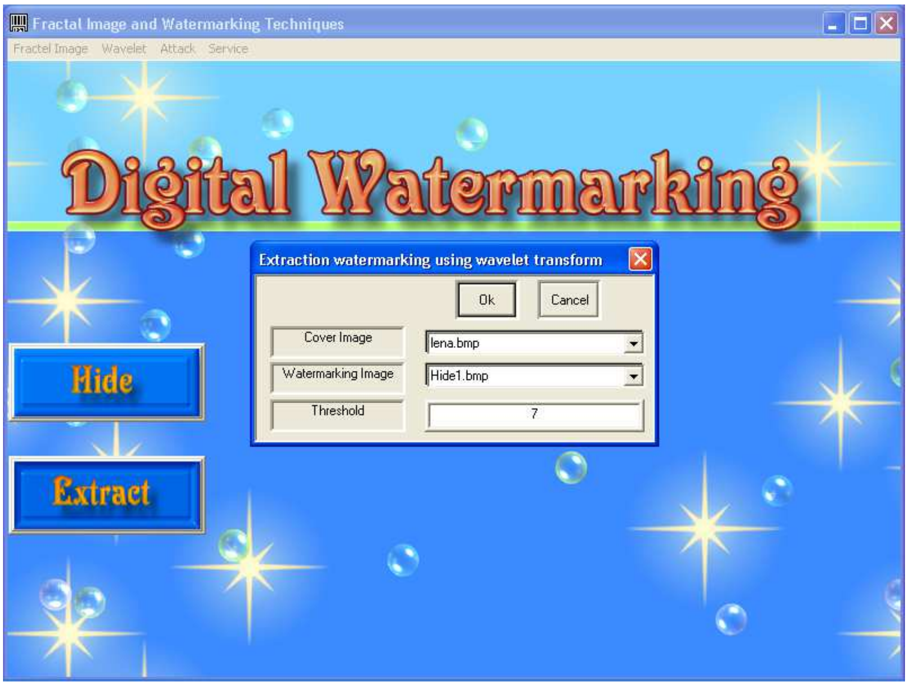


Figure (4.12) Hiding image

- **Extracting Image**

After that the watermark image is attacked by JPEG2000, then the watermark data is retrieved by extract it from the host image, the watermarking image (after hide the fractal watermark image) and the threshold (±Factor), as shown in figure (4.13).


Fractal Image and Watermarking Techniques
Fractel Image Wavelet Attack Service
Digital Watermarking
Hide
Extract
Extraction watermarking using wavelet transform
Ok
Cancel

Cover Image: lena.bmp

Watermarking Image: Hide1.bmp

Threshold: 7

Figure (4.13) Extracting Image

The result of image extraction is shown in figure (4.14).

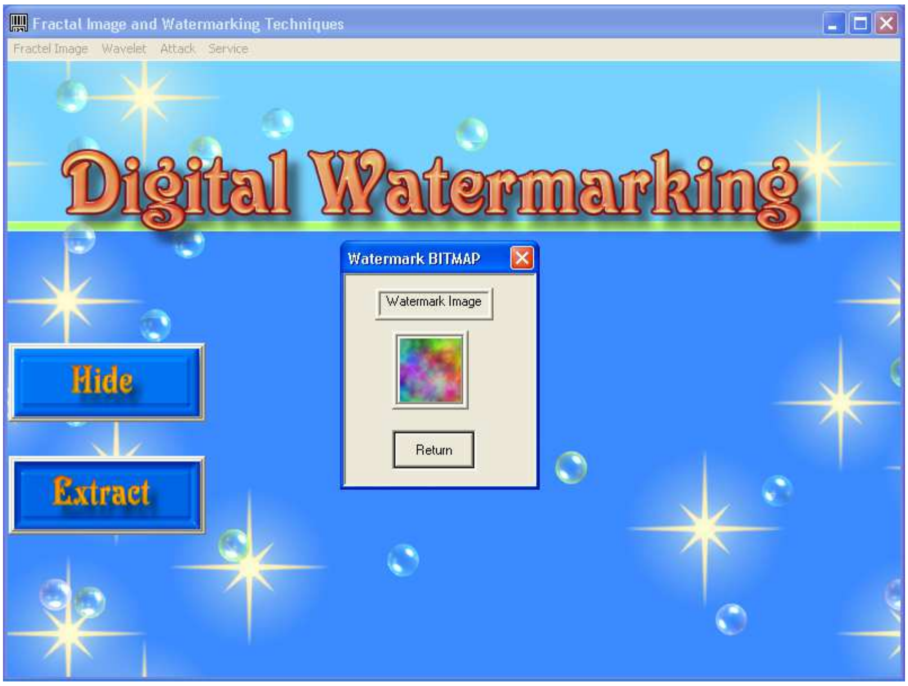
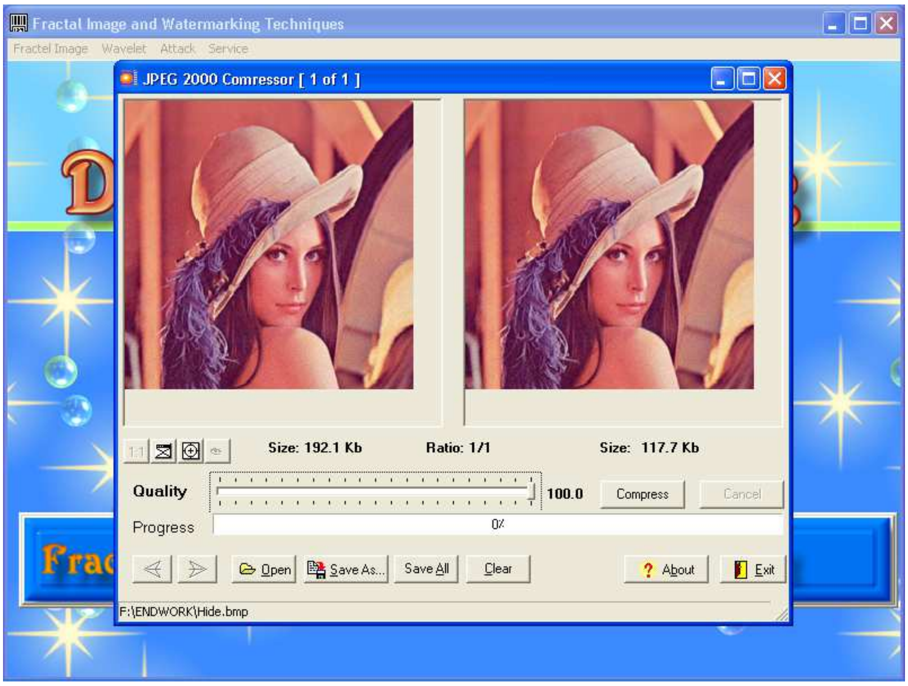

Fractal Image and Watermarking Techniques
Fractel Image Wavelet Attack Service
Digital Watermarking
Hide
Extract
Watermark BITMAP
Watermark Image
Return

Figure (4.14) Fractal Watermark Image

4.7.4 Attack By JPEG2000

- **JPEG2000 Compression**

The watermarked image obtained from the proposed system should not be affected by applying the JPEG2000 compression. Figure (4.15) is used for applying the JPEG2000 compression.



Size: 192.1 Kb Ratio: 1/1 Size: 117.7 Kb

Quality 100.0

Progress 0%

F:\ENDWORK\Hide.bmp

Figure (4.15) JPEG2000 Compression to the Watermarking Image

The output of applying the compression attack by JPEG2000 onto the watermarked image is shown in figure (4.15). The size of the image before and after the compression, the ratio, and the quality, of the compressed image to (.jp2) type is shown.

- **JPEG2000 Converter**

From the compressed version of the watermarked image, the bitmap version of the image is constructed. Therefore we need a conversion procedure that convert the image from (.jp2) type into (.bmp) type should be used, as shown in figure (4.16).

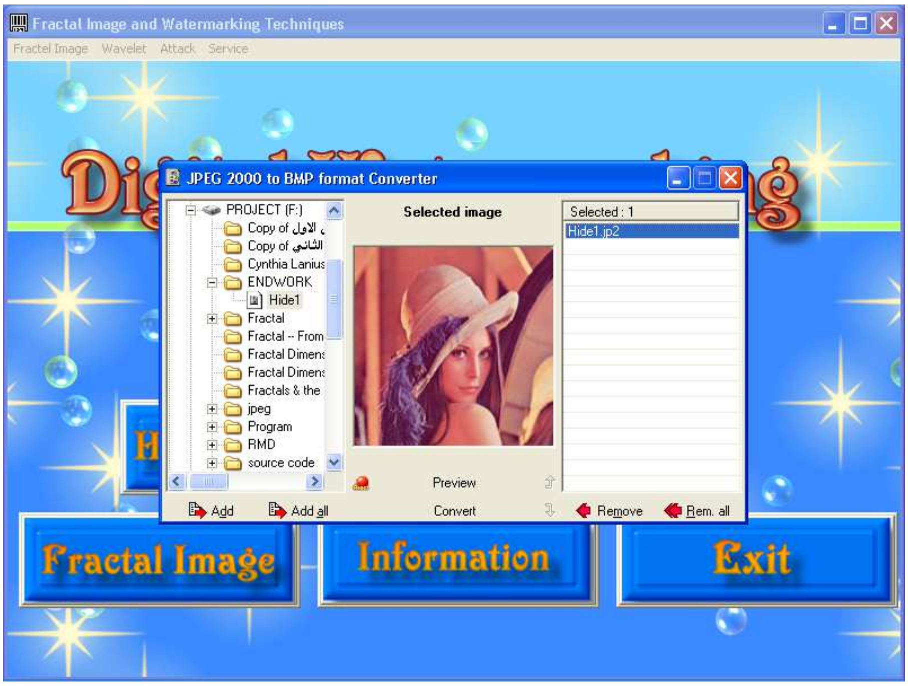
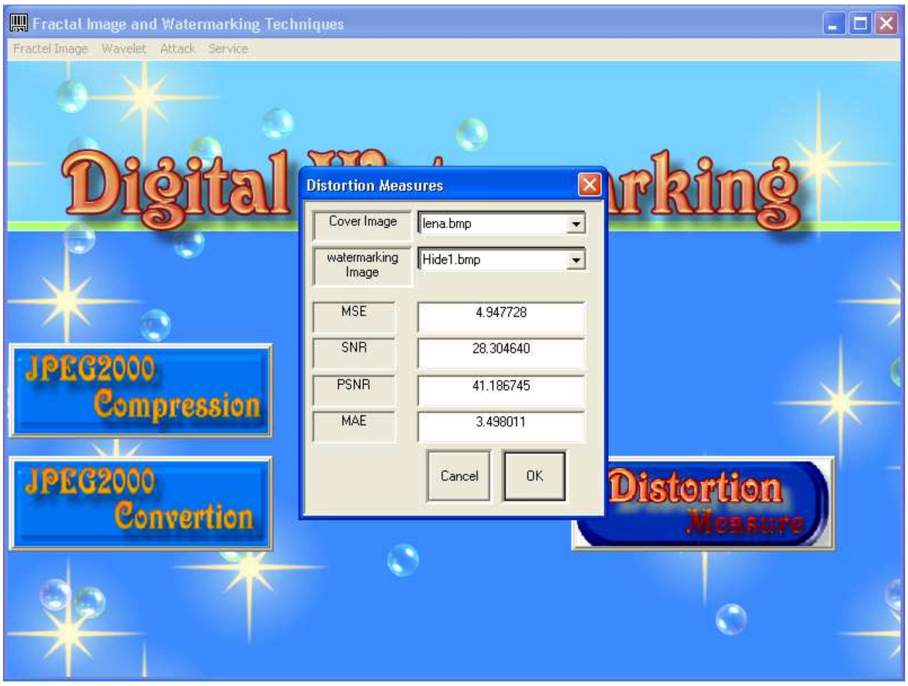


Figure (4.16) JPEG2000 Converter

Distortion measures are used to compare the two images by computing the errors using Mean Square Error (MSE), Signal to Noise Ration (SNR), Peak Signal to Noise Ratio (PSNR), Mean Absolute Error (MAE), as shown in figure (4.17).



Cover Image: lena.bmp

watermarking Image: Hide1.bmp

MSE: 4.947728

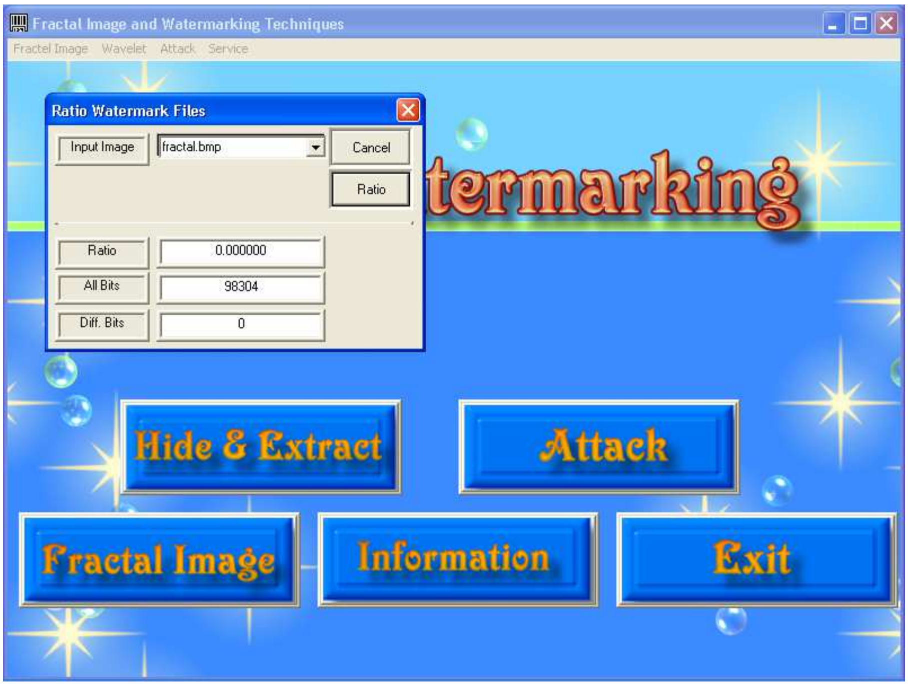
SNR: 28.304640

PSNR: 41.186745

MAE: 3.498011

Figure (4.17) Distortion Measure

The ratio of distortion bits in the fractal watermark image was determined by comparing extracted watermark image with the original image, some results are shown in figure (4.18).



Input Image: fractal.bmp

Ratio: 0.000000

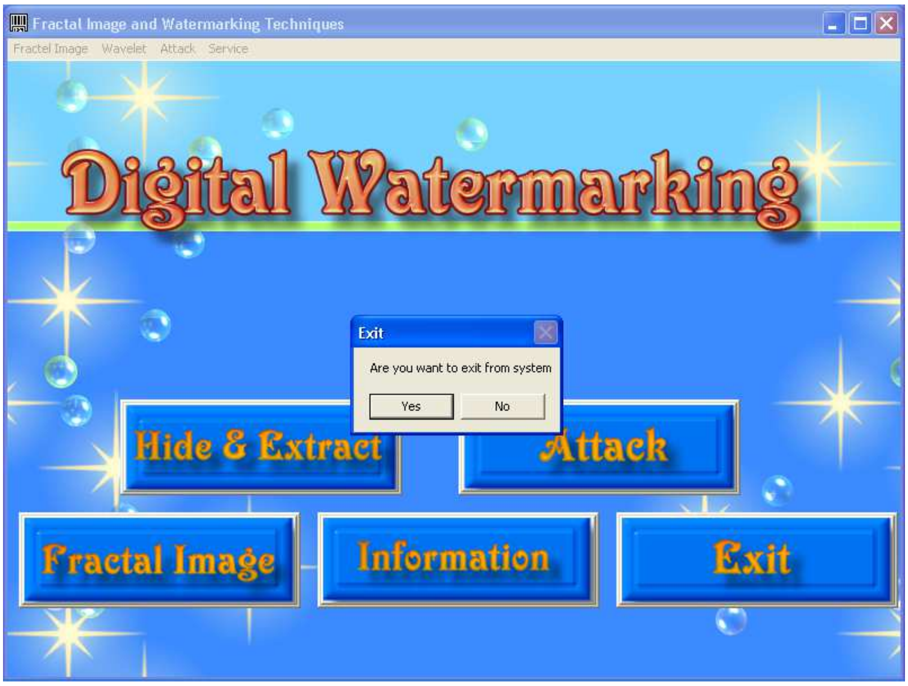
All Bits: 98304

Diff. Bits: 0

Figure (4.18) The Ratio of Watermark Bits.

4.7.5 Exit Command

This command will be used to exit from the System. As shown in Figure (4.19):



Are you want to exit from system

Figure (4.19) Exit from the System

References

[Abd01] **Abd Al-Rehman Fingan**, “Fractal Image Synthesizing”, M.Sc, Thesis, Informatics Institute for Postgraduate studies, Baghdad, IRAQ, 2002.

[Ali02] **Ali M. Al-Aofy**, “Color image compression using Haar Wavelet transform”, M.Sc. Thesis, Informatics Institute for Postgraduate studies, Baghdad, IRAQ, 2002.

[Ali04] **Ali Kadhim Musa**, “Watermark Applications in Color Images Using Wavelet Transforms”, Ph.D. Thesis, Informatics Institute for Postgraduate studies, Baghdad, IRAQ, 2004.

[BaL94] **Batty and Longley**, “Fractal Cities”, Academic Press Inc., ISBN 0-12-4555-70-5,1994.
Site: http://www.casa.ucl.ac.uk/publications/b_fractalcities.htm

[Bar99] **Mauro Barni**, “ A DWT based technique for spatial-frequency masking of digital Signatures”, proceeding of the 1st SPIE Annual Symposium, Electronic Imaging 99 security and watermarking of multimedia contents, USA, 1999.
Site: http://lportal.acm.org/citation.cfm?id=635342

[Bat94] **Batty and Longley**, “Fractals can be called the Geometry of chaos”, Academic Press Inc., 1994.

[Bol95] **Boland F. M., Ruanaidh O. J., Dantzenberg C.**, “Watermarking Digital Images for Copyright Protection”, Proceeding of the 5th International

Conference on Image Processing and its applications, Edinburgh, IEE conference, IEEE, 1995.
Site: http://www.sims.berkeley.edu/courses/is290-1/f96/watermarking.html

[Bov96] **Bovill Carl**, "Fractal Geometry in Architecture and Design",Spekturen der Wissenschaft, 1996.

[Cha00] **Chaelynne M. Wolak**, " Digital Watermarking", School of Computer and Information Sciences, Nova South astern University, 2000.
Site: http:// www.itstudyguide.com/

[Chr00] **C. Christopolous, A. Skodras and T.Ebrahimi**, " The JPEG2000 still image coding system: an overview", IEEE trans. ,2000.
Site: http://jj2000.epfl.ch/jj-publications/papers/006.pdf

[Cox97] **Ingemar j. Cox, J. Kilian, F. T. Leighton and T. Shamoon**, " Secure Spread spectrum Watermarking for multimedia", IEEE Trans. On Image Processing, 1997.
Site: http:// www.ee.ucl.ac.uk/~icox

[Cox02] **Ingemar j. Cox, Matthew L. Miller and Jeffrey A. Bloom**, "Digital Watermarking", Morgan Kaufmann,2002.

[Cru00] **Diego Santa Cruz and Touradj Ebrahimi**, "A Study of JPEG2000 Still Image Coding Versus other standards", Proc. of the X European Signal Processing Conference Tampere, 2000.
Site: http://citeseer.ist.psu.edu/context/2060883/0

[Cyn04] **Cynthia Lanins**, " Fractal Dimension", 2004.
Site: http://math.rice.edu/~lanius/fractals/dim.html

[Dug98] **Rakesh Dugad**, "A new wavelet-based scheme for watermarking image", Proceeding of IEEE International Conference on image processing, ICIP 98, Chicago, IL, USA, 1998.

[Fer98] **Fernando Pere. Gonzalez and Juan R. Hernandez**, Dep. Technologies delas communications, university of Devigo, 1998.

[Fri98] **Fridrich Jiri**, "Application of Data Hiding in Digital Images", tutorial for ISPACS 98 Conference in Melborne, Australia, 1998.
Site: http://antonello.unime.it/atti/2004/c1a0401009.pdf

[Gra95] **A. Graps**, "An Introduction to Wavelets", IEEE Computational Science and Engineering, Vol.2, pt 2, p.p.1-18,1995.

[Hai04] **Haider J.**, "Wavelet-based gray in color image Steganography with Coefficient Reduction", University of Al-Nahrain ,Dep. Of Computer Science, Baghdad, IRAQ, 2004.

[Isa04] **Prof. Dr. Ing. Isar Alexandru**, "Watermarking in the Wavelet domain", University Politehnica, Dep. of Electronic and Telecommunications, 2004.

[ISO00] **ISO/IEC 15444-1:** " Information Technology-JPEG2000 image coding System", 2000.
Site: http:// www.altera.com/products/ip/dsp/image_video_processing/m-bar-jpeg_2000_enc.html

[Ivo02] **Ivo Mark**, "Generating Random Fractal Terrain", 2002.

[Jam02] **Jamal M. Abdul-Jabbar**, "Watermark Robustness Algorithms Using Error Correcting Codes on Compressed Image", Ph.D. Thesis, Informatics Institute for Postgraduate studies, Baghdad, IRAQ, 2002.

[Jen01] **Jens Palsberg, Sowmya Krishnaswamy, Minseok Kwon, Di. Ma., Qiuyun Shao and Yi. Zhang**, "Experience with Software Watermarking", CERIAS and Dep. Of Computer Science, Purdue University, 2001.
Site: http:// www.cs.ucla.edu/~palsberg/paper/acsac00.pdf

[Joh01] **Johnson N.F., Duric Zoran, Jajodia Sushil**, "Information Hiding: Steganography and Watermarking", Kluwer Academic Publishers, USA, 2001.
Site: http://www.jjtc.com/stegdoc/stegdoc.html

[Kat00] **S. Katzenbeisser and F.P. Petitcolas**, "Information Hiding for Steganography and Digital Watermarking", First Edition, Stefan Artech House, London, 2000.

[Kat02] **S. Katzenbeisser and A.P. Petitcolas**, “Information Hiding Techniques for Steganography and Digital Watermarking”, Second Edition, Stefan Artech House, Inc., 2002.
Site: http://www.jjtc.com/pub/book99_ih.htm

[Kun97] **D.Kundur and D. Hatzinakos**, “A Robust Digital Image Watermarking Method Using Wavelet-based Fusion”, procedding of International Conference on Image Processing, 1997.
Site: http://www.ece.tamu.edu/~deepa/pdf/icip97a.pdf

[Kun98] **D. Kundur and D. Hatzinakos**, “Digital Watermarking Using Mutiresolution Wavelet Decomposition”, proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing, 1998.
Site: http://www.ece.tamu.edu/~deepa/pdf/icassp98.pdf

[Kut99] **M. Kutter and F. A. Petitcolas**, “A Fair Benchmark for image watermarking Systems”, Security and Watermarking of Multimedia content, 1999.
Site: http://www.petitcolas.net/fabien/publications/ei99-benchmark.pdf

[Lia02] **Liang J. X4 P. and Tran.T.D.**, “ A Universal Robust Low Frequency Watermarking Scheme”, Submitted to IEEE Transactions on Image Processing, May 2002.
Site: http://citeseer.csail.mit.edu/340428.html

[Lur03] **Lura Tech.**, “The JPEG2000 Source”, www.JPEG2000info.com. 2003.
[Man77] **Benoit Mandelbrot**, “The Fractal Geometry of Nature”,1977.

[Man97] **Mandelbrot B. B.**, “Fractal and Fractal Dimension”, Physical Review E. Statistical Physics, July 1997.
Site: http://www.math.yale.edu/mandelbrot/webbooks/wb_key.html

[Mat00] **MatLab Documentation**, “Wavelet Technique”, 2000.
Site:
http://www.mathworks.com/support/product/seemore.html?page=1&type=Documentation&product=ML

[Mee01] **Peter Meerwald**, “ A Survey of Wavelet domain Watermarking algorithms”, in Proceeding of SPIE, San Jose, C.A. USA, January 2001.
Site: http://www.cosy.sbg.ac.at/~pmeerw/watermarking/waveletSurvey/

[Mic01] **Michael J.Ostwald**, “Fractal Architecture”, Dep. Of Architecture Faculty of architecture, Uni. of Newcastle, Callaghan, Australia, 2001.
Site: http://www.nexusjournal.com/Ostwald-fractal.html

[Pau91] **Paul Bourke**, “ An Introduction to Fractal”, May, 1991.
Site: http://astronomy.swin.edu.au/~pbourke/fractals/fracintro/

[Pet01] **Peter Meerwald**, “Digital image Watermarking in the Wavelet Transform domain”, MSc. Thesis, Dep. of Scientific computing, University of Salzburg, Austria 2001.
Site: http:// www.cosy.sbg.ac.at/~pmeerw/

[Rob95] **Robert L. Devaney**, “Fractal Dimension”, the school of Wisdom, 1995.
Site: http://math.bu.edu/DYSYS/Chaos-game/node6.html

[Sal00] **David Salomon**, “ Data Compression the Complete reference”, Second Edition, 2000.
Site: http://www.ecs.csun.edu/~dsalomon/DC3adveris/DComp3Ad.html

[Vol01] **S. Voloshynovskiy, S. Pereira and T. Pun**, “Attacks on Digital Watermarks: Classification, Estimation-based attacks and Benchmarks”, IEEE communications, Magazine (Aug.2001), P.P. 118-126.
Site: http://www.lnt.de/~eggers/texte/IEEEcom2.pdf

[Wol04] **Wolfgang E. Lorenz**, “Fractal and Fractal Architecture”, Dep. Of Computer aided planning and architecture, Vienna University of technology, 2004.
Site: http://www.iemar.tuwien.ac.at/modul23/fractals/

Ministry of Higher Education
& Scientific Research
Al-Nahrain University

Image Watermarking System based on Wavelet Transform

A thesis submitted to the College of Science, University of Al-Nahrain in partial fulfillment of the requirements for the Degree of Master in Computer Science

By

Itemad Raheem Ali

Supervised by

Prof. Dr. Hilal M.Yousif

January *2006*

Thu Al-Heja *1426*

Acknowlegement

It is a great pleasure to seize this opportunity to thank first my supervisor Prof. Dr. Hillal M. Yousif for his support and sincere and continuous direction, Particular thanks for the Head of Department of computer Science Dr. Ban N. Al-Kallak, my father and teacher Dr. Taha S. Bashaga, staff, and employees.

I thank the staff of Al-Rafidain college university to the animation to complete this project.

I thank everybody who helped me in my work, my colleagues and friends who have offered advice and pointed out positive remarks.

Finally, spatial thanks to my family (particularly my brother Nehad and my husband qayce) for encouragement during the period of my studies.

Supervisor Certification

I certify that this thesis was prepared under supervision at the Department of Computer Science, College of Science, Al-Nahrain University, by Itemad Raheem Ali as a partial fulfillment of requirements for the Degree of Master in Computer Science.

Supervisor

Signature :
Name : Prof. Dr. Hilal M. Yousif.
Title : Professor.
Date : / / 2005.

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature :
Name :Dr. Ban N. Al-Kallak
Title : Head of the department of Computer Science, Al- Nahrain University.
Date : / / 2005.

Dedication

**To My Country,
Teachers, Family and all Friends who
assisted me in my project.**

((وَإن ليسَ للإنسان إلا ما سعى. وان سَعيَهُ سوفَ يُرى. ثمَ يجُزاه الجزاءَ الأوفى))

القرآن الكريم –سورة النجم ، أية ٣٩-٤١

((اللهم أغنيني بالعلم، وزيني بالحلم، وأكرمني بالتقوى))

الرسول محمد (صلى الله عليه وسلم)

((كل وعاء يضيق بما جعل فيه، إلا وعاء العلم، فانه يتسع))

الإمام علي بن أبي طالب (عليه السلام)

No.	***List of Abbreviations***	
1.	ACF	Auto Covariance Function
2.	CWT	Continuous Wavelet Transform
3.	DC	Discrete Cosine
4.	DCT	Discrete Cosine Transform
5.	DLA	Diffusion-Limited Aggregation Model
6.	DVD	Digital Versatile Disk
7.	EBCOT	Embedded Block Coding with Optimized Truncation
8.	HH	High High
9.	HL	High Low
10.	HLC	High Low Cover
11.	HHW	High High Watermark
12.	HLW	High Low Watermark
13.	HVS	Human Visual System
14.	IDWT	Inverse Discrete Wavelet Transform
15.	IHHC	Inverse High High Cover
16.	IHHW	Inverse High High Watermark
17.	IHLC	Inverse High Low Cover
18.	IHLW	Inverse High Low Watermark
19.	IFS	Iteration Function System
20.	JPEG2000	Joint Photographic Expert Group 2000

21.	LH	Low High
22.	LL	Low Low
23.	LHC	Low High Cover
24.	LLC	Low Low Cover
25.	LHW	Low High Watermark
26.	LLW	Low Low Watermark
27.	L-SYSTEM	Lidenmayer-System
28.	MAE	Mean Absolute Error
29.	MSE	Mean Square Error
30.	PSNR	Peak Signal to Noise Ratio
31.	SNR	Signal to Noise Ratio
32.	STFT	Short Time Fourier Transform
33.	W	Watermark

No.	***List of Symbol***	
1.	$E_k(m)$	Encryption Function
2.	m	Message
3.	m_c	Cipher Message
4.	$D_k(m)$	Decryption Function
5.	C_O	Original Work
6.	C_W	Watermarked Work
7.	I	Original Image
8.	I'	Distorted Image
9.	K	Key
10.	Ψ	Wavelet Function
11.	$f(t)$	Signal
12.	$H(w)$	Low pass filter
13.	$G(w)$	High pass filter
14.		

" Table of Contents "

وزارة التعليم العالي والبحث العلمي
جامعة النهرين
العراق - بغداد

نظام صورة العلامة المائية باستخدام التحويل المويجي

بحث مقدم إلى جامعة النهرين كجزء من متطلبات نيل شهادة الماجستير في علوم الحاسبات

من قبل الطالبة

اعتماد رحيم علي

بإشراف

أ. د. هلال محمد يوسف

ذو الحجة ١٤٢٦

كانون الثاني ٢٠٠٦

الاسم : اعتماد رحيم علي مرعي الربيعي

القسم: علوم الحاسبات

المرحلة: ماجستير

السنة: ٢٠٠٦-٢٠٠٧

العنوان: البلديات –حي ٩ نيسان- م ٧٤٤-ز٣-د٤٠

الهاتف: ٠٧٩٠١٨١٤٥٤١

البريد الالكتروني: weffee@yahoo.com

عنوان الاطروحة: Image Watermarking System based on Wavelet Transform
(نظام صورة العلامة المائية بالتحويل المويجي)

المشرف: أ.د. هلال محمد يوسف

المناقشين: د.لؤي ادور جورج - رئيساً
د.بان ذنون – عضواً
د. بشرى قاسم – عضواً

تاريخ المناقشة: الخميس ٤-٥-٢٠٠٦ الساعة ٩,٣٠ صباحاً