DCT-BASED IMAGE WATERMARKING FOR SECURING IMAGES

by

S.KAWSIKA RAJA
Reg. No. 71203405006
of
Kumaraguru College of Technology, Coimbatore 641 006

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree

of

MASTER OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING
JUNE 2005

BONAFIDE CERTIFICATE

Certified that this project report entitled "**DCT-BASED IMAGE WATERMARKING FOR SECURING IMAGES**" is the bonafide work of **Mr. S. KAWSIKA RAJA**, who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

GUIDE

HEAD OF THE DEPARTMENT

The candidate with **University Register No. 71203405006** was examined by us in the project viva-voce examination held on 25/06/05

INTERNAL EXAMINER

EXTERNAL EXAMINER 25/6/05

ABSTRACT

A DCT based image-watermarking algorithm is described, where the original image is embedded into the cover image (i.e. the image which contains the original image). This is achieved by inserting the watermark into the sub images obtained through sub sampling, and the original image is secured through the password. Watermarking is a Process of inserting hidden information in an image by introducing modifications to its pixel with minimum perceptual distribution. In this project, a fixed number of highest magnitude pixels are randomly perturbed, so that the watermark is placed to the perceptually significant components of the image.

Even though the method is quite robust against signal manipulations, the original image must be present for watermark recovery. Recently, the pursuit of a scheme that doesn't need the original image during watermark recovery has become a topic of intense research. In DCT-Based watermark calculating the correlation between the watermark sequence and the highest magnitude pixels of the watermark image can identify scheme the watermark. In addition to this the password is obtained from the user and it is encrypted and it is hided into the cover image along with the original image.

Watermarking is a key process in the protection of copyright ownership of electronic data (image, videos, audio ...). Image can be easily duplicated and distributed without the owner's consent. Digital watermarks have been proposed as a way to tackle this issue. A digital watermark is invisible information (i.e. text, image or signature) embedded inside an image to show authenticity and ownership. In addition to this the original image is embedded into another image along with the encrypted password.

கருத்துச் சுருக்கம்

டி.சி.டி. அடிப்படையிலான நீர்க்குறிமச் செயல் வழிப்பாடு மெய்பிம்பம் சூழ்பிம்பத்துள் உள்ளிடாகத் தரப்படும் வரையறையாகும். உப மாதிரித் தோகையில் பெறப்படும் உப பிம்பங்களில் நீர்க் குறியிடுவதால் இது சாத்தியமாகிறது. மெய்பிம்பம் ரகசியச் சொல்லால் மீட்கப்படுகிறது. நீர்க் குறிமமாவது ஒரு பிம்பத்தில் அதன் பிக்ஸலில் மிக்குறை ஏற்புப் பரவல் அடிப்படையிலான மாற்றங்கள் வாயிலாக மறைபொருள் தகவலை உட்புகுத்தும் செயல்முறை ஆகும். இந்த ஆய்வில் குறிப்பிட்ட எண்ணிக்கையிலான மி வீச்சுள்ள பிக்ஸல்கள் பரவலாகத் தேர்வு செய்யப்பட்டு முக்கியத்துவமிக்க பிம்பத்தின் பாகங்கள் நீர்க் குறியிடப்பட்டுள்ளன.

இந்த வழிமுறை சமிக்ஞை வினைகளுடன் விரைவாயிருப்பினும், மெய்பிம்பம் நீர்மக்குறிம மீட்சிக்கு அவசியமாகிறது. மெய்பிம்பத்தின் தேவையற்ற நீர்மக்குறிம மீட்சி, நுண்ணாய்விலுள்ள ஒரு விஷயமாகும். டி.சி.டி. அடிப்படையிலான நீர்க் குறிம வழிப்பாட்டில், நீர்க்குறி காணப்படுகிறது. ரகசியக் குறியீடு பயனாளியிடமிருந்து பெறப்பட்டு, மறைதிரிக்கப்பட்டு சூழ்பிம்பத்துள் வைக்கப்படுகிறது.

நீர்மக்குறிமம் மின் தகவலுக்கான காப்புரிமையாகும். பிம்பம் பரதியெடுக்கப்பட்டுப் பரப்பப்படும் நிலை டிஜிட்டல் நீர்மக்குறியீடு கட்புலனாகாத, பிம்பத்துள் மறைந்திருக்கும் உரிம அடையாளமாகும். இத்துடன், மெய்பிம்பம் மறைதிரிக்கப்பட்ட் ரகசியச் சொல்லுடன் மற்றொரு பிம்பத்தின் உள்ளீடாகிறது.

ACKNOWLEDGEMENT

I express my profound gratitude to our esteemed Correspondent, **Professor Dr. K. Arumugam, B.E. (Hons), M.S. (USA), M.I.E.,** for giving me this great opportunity to pursue this course.

I thank, **Dr. K.K. Padmanabhan, B.Sc. (Engg), M.Tech., Ph.D.,** Principal, Kumaraguru College of Technology, Coimbatore, for being a constant source of inspiration and providing me with the necessary facilities to work on this project.

I express my sincere thanks to **Dr.S.Thangasamy, Ph.D.,** Professor and Head of Department of Computer Science and Engineering, for his support and encouragement throughout the project.

I would like to express my special thanks to **Mr. R. Dinesh, M.S. (Wisconsin),** Assistant Professor and Project Coordinator, Department of Computer Science and Engineering for his valuable suggestions and guidance.

I express my deep sense of gratitude and gratefulness to my guide **Mrs.N.Suganthi, M.E.,** Senior Lecturer, Department of Computer Science and Engineering, for his supervision, tremendous patience, active involvement and guidance.

I would like to convey my honest thanks to all **members of staff** of the Department for their unlimited enthusiasm, friendship and experience from which I have greatly benefited.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

The recent growth in the computer networks, and more specifically, the www, has allowed the multimedia data such as image, to be easily distributed over the internet. However, many publishers may be reluctant to show their work on the internet due to lack of security. Image can be easily duplicated and distributed without the owner's consent.

Digital watermarks have been proposed as a way to tackle this issue. A digital watermark is an invisible signature embedded inside an image to show authenticity and ownership.

An efficient digital watermark should be perceptually invisible to prevent obstruction of the original image. It should be statistically invisible to prevent detection, and it should also be robust to many image manipulations, such as filtering, additive noise, and compression.

Several watermarking techniques have been proposed. Some methods embed the watermark in the spatial domain of the image. Other watermarking techniques use transform methods, such as Fast Fourier Transform [FFT], Discrete Cosine Transform [DCT], or the Wavelet Transform [WT] to embed the watermark.

This project report proposes the Discrete Cosine Transform [DCT] technique only for embedding the watermark in an image. Algorithms proposed for watermarking may be either in image intensity domain or in frequency domain.

1.1. PROBLEM DEFINITION:

A DCT based image watermarking algorithm is described, where the original image is hided into another image (i.e. cover image) and it is secured through the encrypted password. Obviously the original image is obtained by decrypting the password and then extracting from the cover image.

Watermarking is a Process of inserting hidden information in an image by introducing modifications to its pixel with minimum perceptual distribution. *Watermarking* is a key process in the protection of copyright ownership of electronic data (image, videos, audio ...).

1.2. PROPOSED SOLUTION

We propose the system, in the encoder, the original image is decomposed according to the source image, and the sub images are transfer via DCT. The coefficient are efficient in pairs to verify whether they are appropriate for insertion, if two coefficient are different in amplitudes, they will not be modified this is to avoid causing excessive distortion. Using decoder we can obtain the original image.

There are two types of watermarking, namely visible and invisible watermarking. Visible watermarks are especially useful for conveying an immediate claim of ownership. Invisible watermarks, on the other hand, are more of an aid in catching the thief than discouraging the theft in the first place.

In this project the invisible watermarking technique is used, where the image is fully hided in another image using DCT and they are obtained using decoding method when ever they are needed.

Watermarking is not restricted to just retaining information of the author in the work, there are various other purposes for which watermarking may be incorporated into an object. Some of them are:

a) ***Copyright Protection:*** For the protection of intellectual property, the data owner can embed a watermark representing the copyright information in his data.

b) ***Fingerprinting:*** To trace the source of illegal copies, the owner can use a fingerprint technique. This requires the owner to embed different information onto copies of the work provided to different customers. The information embedded can be a serial number, customer id etc.

c) ***Data Authentication:*** Introducing fragile watermarks into the data can help to ensure that the data is not processed or modified in anyway by the user.

d) ***Data Hiding:*** Watermarking may be used to embed longer bits of information in the data or in the image.

e) ***Image Hiding:*** Watermarking may be used to embed the image into another image.

f) **Steganography:** Spies in other countries have to use local channels to communicate. They should assume that the channel will be monitored. Hence the message should be sent in such a way that the presence of the message is not revealed. Watermarking methods meant for such hidden transmission ensure that the presence of the message in the image is undetectable. These methods are not robust as robust watermarks are easily detectable

A watermarking algorithm that is used in embedding information should have a lot mutually exclusive property. The watermarking method should not require the original image to be present during the watermark extraction phase. The watermarking method then falls in the category of ``blind" watermarks. Blind watermarks usually lack robustness

The watermark must have considerably large amount of information capacity. The watermark capacity should not depend on the size of the image or channels in the image. An application might have several kilobytes of information to be embedded in a small image. Ideally the size of the image should not decide the amount of information that can be put into the image.

CHAPTER 2

2. LITERATURE SURVEY

The literature survey that is done in the phase one is given below. The survey is mainly concentrated one the watermarking techniques and Discrete Cosine Transform (DCT).

2.1 WATERMARKING

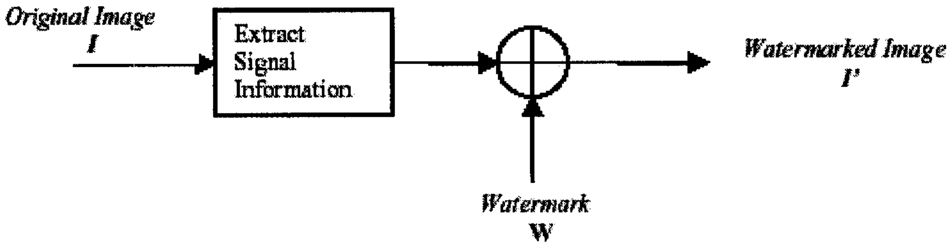
The enormous popularity of the World Wide Web in the early 1990's demonstrated the commercial potential of offering multimedia resources through the digital networks. Since commercial interests seek to use the digital networks to offer digital media for profit, they have a strong interest in protecting their ownership rights. Digital watermarking has been proposed as one way to accomplish this.

A digital watermark is a digital signal or pattern inserted into a digital image. Since this signal or pattern is present in each unaltered copy of the original image, the digital watermark may also serve as a digital signature for the copies. A given watermark may be unique to each copy (e.g., to identify the intended recipient), or be common to multiple copies (e.g., to identify the document source). In either case, the watermarking of the document involves the transformation of the original into another form.

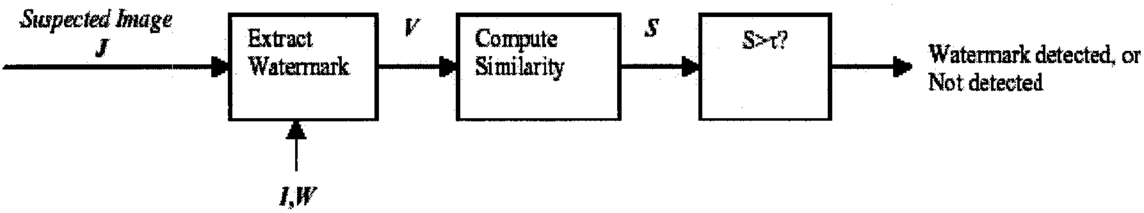
A digital watermark is a subtle signal embedded in host data causing an imperceptible change to the host data - be it is an image, video or audio. The easiest to implement is an image watermark, which invisibly embeds a message in an image using a password.

A digital watermark can be an identification code carrying out information (an author's signature, a company logo, etc...) about copyright owner, the creator of the work, the authorized consumer and so on. It is permanently embedded into digital data for copyright protection and checking if the data has been modified.

Watermark Transmission:



Watermark Detection:



2.1.1. THE PURPOSE OF DIGITAL WATERMARKS

Two types of digital watermarks may be distinguished, depending upon whether the watermark appears visible or invisible to the casual viewer. Visible watermarks are used in much the same way as their bond paper ancestors, where the opacity of paper is altered by physically stamping it with an identifying pattern. This is done to mark the paper manufacturer or paper type. One might view digitally watermarked documents and images as digitally "stamped".

Invisible watermarks, on the other hand, are potentially useful as a means of identifying the source, author, creator, owner, and distributor or authorized consumer of a document or image. For this purpose, the objective is to permanently and unalterably mark the image so that the credit or assignment is beyond dispute. In the event of illicit usage, the watermark would facilitate the claim of ownership, the receipt of copyright revenues, or the success of prosecution.

Watermarking has also been proposed to trace images in the event of their illicit redistribution. Whereas past infringement with copyrighted documents was often limited by the unfeasibility of large-scale photocopying and distribution, modern digital networks make large-scale dissemination simple and inexpensive.

Digital watermarking makes it possible to uniquely mark each image for every buyer. If that buyer then makes an illicit copy, the illicit duplication may be convincingly demonstrated.

2.1.2. VISIBLE VS. INVISIBLE WATERMARKS

Visible and invisible watermarks both serve to deter theft but they do so in very different ways. Visible watermarks are especially useful for conveying an immediate claim of ownership. The main advantage of visible watermarks, in principle at least, is that they virtually eliminate the commercial value of the document to a would-be thief without lessening the document's utility for legitimate, authorized purposes.

A familiar example of a visible watermark is in the video domain where CNN and other television networks place their translucent logo at the bottom right of the screen image. Invisible watermarks, on the other hand, are more of an aid in catching the thief than discouraging the theft in the first place.

2.1.3. WATERMARKING REQUIREMENTS:

a) Perceptual Transparency: In most application, the watermark inserted should not affect the quality of the cover image or data and hence remain undetectable. The watermark should go unnoticed as long as the data is not compared with the original data. This requirement also arises from the fact that perceptible signals are much easier to remove and also do not have the built in advantage of stealth.

b) Robustness*:* Robustness is a measure of the ability of the embedding algorithm to introduce the watermark in such a way that it is retained in the image despite several stages of image processing.

The image may be filtered (high-pass or low-pass or median) rotated, translated, cropped, scaled etc... as part of image processing. A good watermarking algorithm embeds the watermark in the spatial or frequency regions of the image, which would be least affected by such processing.

c) Security: Security of a watermarking technique can be judged the same way as with encryption techniques. Assuming that the unauthorized parties know the algorithm used for embedding, the security of the algorithm lies in the selection of key. Thus the algorithm is truly secure if knowing the exact algorithm to embed and extract data does not help an unauthorized party in actually recovering the data from the watermarked image.

d) Unambiguity: The retrieval of a watermark should unambiguously identify the owner. In the accuracy of owner identification should degrade gracefully under attacks.

e) Tamper-resistance: The embedded watermark must be resistant to tampering through collusion by comparing multiple copies of the media embedded with different watermarks.

f) Oblivious vs Non-Oblivious: In applications such as copyright protection and data monitoring the watermark extraction algorithms can use the original un-watermarked data to find the watermark. This is called non-oblivious watermarking. In other applications, such as copy protection and indexing the watermark extraction algorithms cannot access the un-watermarked image. This significantly raises the difficulty of extraction. Such methods are called as oblivious watermarking algorithms.

2.2. DISCRETE COSINE TRANSFORM

A watermarking algorithm has two stages: watermarking casting and detection. By means of watermark casting a specific code assigned to the owner is embedded in the image. In the detection stage the algorithm identifies the given code.

A watermarked image can be processed by means of various image transformations and processing algorithms which may be able to destroy, intentionally or not, the digital watermark. The standard still image compression algorithm is JPEG.

JPEG is based on the minimization of the energy in the Discrete Cosine Transform (DCT) domain. In the case of lossy compression, the image suffers information loss in the high frequency domain. The following figure represents the watermark embedding process.

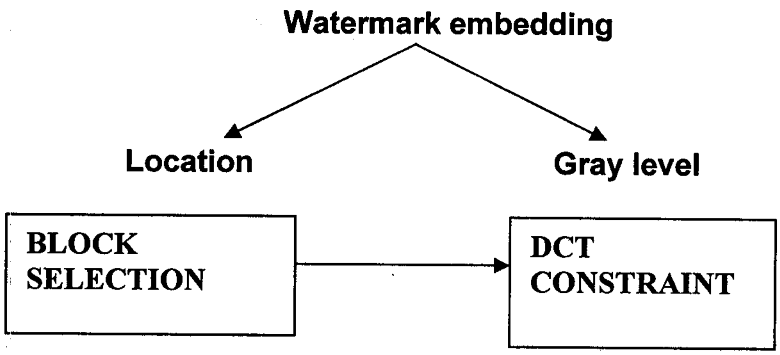


Fig 1: The processing block for watermark embedding.

JPEG is based on the minimization of the energy in the Discrete Cosine Transform (DCT) domain. In the case of lossy compression, the image suffers information loss in the high frequency domain.

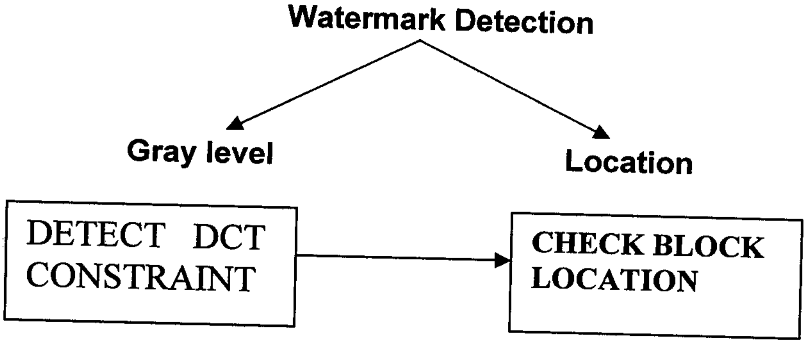


Fig 2: The processing block for watermark detection.

The watermarking algorithm consists of two steps. The first step is to select the blocks and modifying highest magnitude pixels such that they fulfill a given constraint. The second is the detection stage where we check for DCT constraint and then the respective block location. These two steps are shown in the above diagram.

2.3. RELATED WORKS

In this project, I concentrate on hiding the image inside another image. This is done with the help of DCT based watermarking method. The image watermarking algorithm can be classified into two categories: spatial-domain techniques and frequency-domain techniques.

The spatial-domain techniques directly modify the intensities or color values of some selected pixels while the frequency-domain modifies the values of some transformed coefficients.

The simplest spatial-domain image watermarking techniques is to embed a watermark in the least significant bits (LSBs) of some randomly selected pixels. The watermark is actually invisible to human eyes. However, the watermark can be easily destroyed if the watermarked image is low-pass filtered or JPEG compressed.

To increase the security of the watermark, Matsui and Tanaka proposed a method that uses a secret key to select the locations where a watermark is embedded. Voyatzis and Pitas used a toral automorphism approach to scramble the digital watermark before a watermark is inserted into an image.

To increase the robustness of the watermark, many approaches have been proposed to modify some properties of selected pixels or blocks. Darven and Scott proposed a fractal-based steganography method to embed the watermark. The frequency-domain techniques first transform an image into a set of frequency domain coefficients. The transformation may be DCT, Fourier transform, or wavelet transform etc... The watermark is then embedded in the transformed coefficients of the image such that the watermark is less invisible and more robust to some image processing operations.

Finally, the coefficients are inverse-transformed to form the watermarked image. The frequency sensitivity of the human visual system can be used to ensure that the watermark is invisible and more robust to any attacks. In this paper, we will propose DCT image watermarking technique that is robust to common image processing operations. The proposed approach utilizes the sensitivity of the human visual system to adaptively modify the intensities of some pixels in a block. The modification of pixel intensities depends on the content of a block. If the contrast of the block is large (e.g., anedge block), the intensities can be changed greatly without introducing any distortion to human eyes. On the other hand, if the contrast is small (e.g., a smooth block), the intensities can only be tuned slightly.

CHAPTER 3

3. LINE OF ATTACK

3.1 IMAGES

The term 'image' is used here to refer to graphic representations of real-world objects. Images have been produced for thousands of years, using different materials and media. The development of photography provided the first means of capturing realistic representations of reality of paper. The development of moving pictures on film allowed the creation of representations that moved. Television allowed such pictures to be transmitted.

Still and moving images can now be stored and transmitted in digital form. This allows images to be stored, transmitted and manipulated as computer data. Still and moving images can be integrated seamlessly with other computer applications. The combination of different media, such as text and images, and possibly sound and moving images, is an example of multimedia.

This document is concerned primarily with incorporating still images in computer applications. A distinction can be drawn between the use of images for screen display and those intended for printed output. Different considerations apply in each of these applications. In some cases there will be some overlap in requirements. The emphasis here will be on incorporating images intended for screen display, rather than for printed output.

3.1.1 IMAGE TYPES

Images can be stored in many formats. Three fundamental types of images can be considered: vector, bitmap, and moving images.

1) Vector images are stored as a sequence of drawing instructions.
2) Bitmap images are stored as a mosaic matrix of picture elements.
3) Moving images are stored as a sequence of pictures used to give the illusion of motion.

3.1.2. VECTOR IMAGES

Vector images define an image as a set of instructions composed of graphic objects or primitives. Strictly speaking, vectors are lines, but vector graphics generally include ellipses, arcs, curves, filled areas and in many cases text.

Vector graphics are well suited to computer handling, because they can be mathematically manipulated. This enables individual elements to be scaled and stretched. Vector graphics will always display or print at the best resolution for any output device.

Original vector images can be created in application programs. Such applications are often called drawing packages and are sometimes said to be object-oriented. Vector images are used extensively in Computer Aided Design. They are also highly suitable for certain graphic applications.

3.1.3. BITMAP IMAGES

Bitmapped images or bitmaps represent an image as a grid of picture elements or *pixels*, each with a value mapping a particular value of color and intensity.

Any image can be represented in bitmap form, although this is not necessarily always efficient. Bitmaps are generally employed to represent real world images. These can be acquired using either a scanner or a video frame grabber.

Existing bitmap images can be modified using photo retouching packages. Image processing functions can be used to modify such characteristics as sharpness, contrast, and color balance.

Original bitmap images can also be created in software applications. These are often called painting packages. There are certain limitations in creating original images in bitmap form:

- A bitmap image is stored as a pattern of pixels that can be difficult to modify.
- A bitmap image is effectively tied to the resolution at which it was created.
- A bitmap image of high resolution can require a great deal of storage.

When creating original artwork images it is often preferable to create images in vector form, possibly converting to a bitmap image if required.

Resolution

The quality of a bitmap reproduction is a factor of the resolution of the bitmap. The resolution of a bitmap image is a product of the dimensions of the bitmap in pixels and the amount of information that is stored for each pixel. The required resolution for a particular image is related to the way in which it will be reproduced.

Color resolution

The amount of information stored for each pixel determines its *pixel depth* or *color depth*. This is measured in the number of binary digits or bits employed to encode each pixel.

The following table illustrates the number of discrete values available for some common color depths:

Bits per pixel	Total colors
1	2
2	4
4	16
8	256
16	32,768
24	16,777,216

Depending upon the amount of information that is stored for each pixel, bitmaps can represent either bi-level, grey scale or color images.

Bi-level bitmaps

Bi-level bitmaps store a single bit for each pixel. This is sometimes called a flat bitmap. A bi-level bitmap allows the representation of only two values of brightness or color information, typically black and white, although any two colors might equally be displayed.

Grey scale bitmaps

Grey-scale bitmaps store more than one bit for each pixel, making more than one value available for each point in the image. This is sometimes referred to as a *deep* bitmap. In a grey-scale bitmap, each value represents a known level of luminosity or brightness. This allows the representation of a number of tonal values. Usually these will be shades of grey, but they could equally be presented as shades of any color.

Storing 2 bits per pixel allows 4 values, while 4 bits allow 16 values. A monochrome image of 6 bits per pixel allows 64 levels of grey which is about the minimum required to achieve a photographic effect.

A monochrome image of 8 bits per pixel offers 256 levels of grey, allowing continuous tonal variations to be represented without *contouring*, since the eye is unable to discriminate this many shades of grey.

3.2 WATERMARKING

3.2.1 WATERMARK EMBEDDING

In the proposed approach, the embedded watermark must be invisible to human eyes and robust to most image processing operations. To meet these requirements, a bit of binary pixel value (0 or 1) is embedded in a block of the host image.

Before insertion, the host image is decomposed into N´N blocks. Depending on the contrast of a block, pixels in the block are adaptively modified to maximize robustness and guarantee invisibility.

The position or block for embedding is selected by a pseudo-random number generator. The value of *k* is the encrypted password. Let B be the selected block, and *g*max, *g*min, and *g*mean represent the maximal, minimal, and average intensities of the block, respectively. That is,

gmax = max (*b*ij, 0<= i, j < N),

*g*min = min (*b*ij, 0<= i, j < N),

where *b*ij represents the intensity of the (i, j)-th pixel in block B.

3.2.2 WATERMARK EXTRACTION

The extraction of a watermark is similar to the embedding process while in a reverse order. In the proposed algorithm, the extraction of a watermark must make reference to the original host image. First, we use the seed value, *k*, to generate a sequence of positions or blocks where the watermark is embedded.

For each selected position, let B and B' represent the corresponding blocks of the original host image and watermarked image, respectively. Compute the sum of pixel intensities, *S*o and *S*w, of B and B'. The retrieved watermark bit value *b*w is determined by the following rule:

*b*w = 1 if *S*w > *S*o,

*b*w = 0 if *S*w £ *S*o.

The extracted watermark bit values, *b*w's, are then inversely permutated to get the reconstructed watermark.

3.3 IMAGE WATERMAKING ALGORITHM IN THE DCT DOMAIN

In the section we present the image watermarking algorithm in the DCT domain (Discrete CosineTransform). Digital watermarking has been presented as a new method for copyright protection. It is a technique to insert imperceptible (and hard to remove) data into digital image, audio or video, so that it allows ownership to be established or a buyer to be identified.

The common digital image watermarking techniques are spatial or transform-based, which are based on spread-spectrum communications. Most of transform-domain watermarking methods are based on Discrete Cosine Transforms (DCT). Both watermark structure and embedding strategy affect robustness of image watermarks. Where watermarks should be embedded in DCT domain in order for the invisible watermarks to be robust?

Where Watermarks Should be Embedded?

In the DCT domain, watermarks should be embedded in those coefficients that meet the following requirements in order for the watermarks to be invisible and robust:

1) having large perceptual capacity that allows strong watermarks to be embedded without perceptual distortion
2) Changing little with common image processing and noise corruption.

DCT WATERMARK EMBEDDING PROCESS

The following procedure indicates how the pixels values in the original image are embedded into the cover image using DCT.

W: watermark to be embedded (original image).

X: sequence of pixel values in cover image.

Y: pixel values in watermarked image.

a=scaling factor: Determines the intensity of the watermark.

$$Y(i) = X(i)(1 + aW)$$

The scaling factor 'a' is a critical system parameter.

- If 'a' is too small, the image is not distorted but the robustness of the scheme is low.
- If 'a' is too large the image is distorted but the robustness of the scheme is high.

DCT WATERMARK EXTRACTION PROCESS

The following procedure indicates how the pixels values in the original image are extracted from the cover image using DCT.

W^*: extracted watermark (original image).

X: sequence of pixel values in cover image.

Y: pixel values in watermarked image.

a=scaling factor: Determines the intensity of the watermark.

$$W^* (i) = (1/a) * (Y (i) / X (i)) -1$$

3.4. ENCRYPTING THE PASSWORD

An original image is known as the plaintext, while the coded message is called the cipher text. The process of converting the plaintext to cipher text is known as encryption; restoring the plaintext from the cipher text is known as decryption. There are various methods that are involved in this mechanism. In this project the hash algorithm and encryption algorithms are used.

HASH ALGORITHM

Hash algorithm uses the hash function that maps a variable length data block or message (M) into a fixed length value called a hash code H(M). The function is designed in such a way that, when protected, it provides an authenticator to the data or message. The following hash algorithms are used. A variation on the message authentication code is on-way hash function.

They are meant for digital signature applications where a large message has to be "compressed" in a secure manner before being signed with the private key. All three algorithms take a message of arbitrary length and produce a 128-bit message digest.

1) Message Digest 4 (MD 4)
2) Message Digest 5 (MD 5)
3) Secured Hash Algorithm (SHA)

Secure Hash Algorithm takes a message of less than 264 bits in length and produces a 160-bit message digest, whereas MD4 and MD5 were aimed at 32-bit machines.

ENCRYPTION ALGORITHM

The conversion of plaintext or data into unintelligible form by means of a reversible translation, based on a translation table or algorithm. The following methods are used in this project.

1) DES
2) Triple DES

The most widely used encryption scheme is based on the Data Encryption Standard (DES). In this method the data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.

CHAPTER 4

4. DETAILS OF METHODOLOGY EMPLOYED

In this project the image watermarking algorithm is based on constraints in the Discrete Cosine Transform (DCT) domain. An image watermarking algorithm has two stages: signature casting (embedding) and signature detection. In the first stage it embeds an identifying label in the image. This is recognized in the second stage.

The proposed algorithm has two processing steps. In the first step certain pixel blocks are selected using a set of parameters while in the second step a watermark is embedded in the selected blocks. Two different constraint rules are suggested for the parametric modification of the DCT. The first one embeds a linear constraint among certain selected highest magnitude pixels and the second defines circular detection regions according to the given parameters. The watermarks cast by the proposed algorithm are resistant to JPEG compression and filtering.

The watermarks are detected only from the watermarked image and the watermark code respectively without any need to have the original image. The algorithm which embeds circular DCT detection regions provides better results when compared to the algorithm which embeds a linear constraint in the DCT coefficients. The watermarks embedded by the proposed algorithm are also robust to certain image cropping or other distortions as extracting lines from the image. In transform coding, the signal is mapped from one domain (spatial) into the transform domain.

Each transform

- Decorrelates the pixels
- Results in energy compaction
- Concentrates the energy in the low-frequency zone of the transform domain.

4.1. EMBEDDING THE ORIGINAL IMAGE USING DCT

- Create a temporary bitmap from the cover image
- Declare a pixel array to copy pixels form the bitmap
- Initialize the 'bit buffer' which contains the bits that has to be embedded
- Insert the metadata bits at the start of the bit buffer
- Metadata contains the information about file size and encrypted password
- Compute the average difference in color intensity, which is used to embed the original image
- Compute capacity using

  C = int (log (avg) / log (2))
- The original data bits are added at end of the bit buffer after comparing with the capacity of the cover image
- Select the bit to be embedded (W(i)) from the bit buffer
- Select the pixel coordinates (X (i,j)) from the pixel array in order to embed to data bit

  $Y(i,j) = X(i,j)\,(1 + a\,W(i))$

  a=scaling factor: Determines the intensity of the watermark.
- Update the temporary bitmap (Y(i,j)) with the pixel array
- Create the watermark image from the temporary bit map
- Compute the image capacity and then save the Image

Detection of such an embedded message would seem to be quite difficult. (An advantage DCT has over other transforms is the ability to minimize the block-like appearance). The following are the two main things that have to be maintained during the embedding process.

4.2. EXTRACTING THE ORIGINAL IMAGE USING DCT

- Create a temporary bitmap from the watermarked Image
- Declare a 'pixel array' to copy the pixels from the bitmap
- Initialize the 'bit buffer' which contain the bits that are extracted
- Select the pixel co-ordinates Y(i,j) from pixel array
- Compute the average difference in color intensity, which is used to extract the original image
- The average color intensity of the cover image X (i,j) is passed in order to extract the original image

  $W^* (i) = (1/a) * (Y (i,j) / X (i,j)) -1$
- Store the extracted Metadata bits and data bits W*(i) in the bit buffer
- The meta data bits are added at the starting of the bit buffer and after that the data bits are added
- Select the bits from the bit buffer in order to construct the original image
- Update the temporary bitmap with the pixel array
- Create the cover image from the temporary bitmap
- Compute the original image from the data bits stored in the bit buffer
- Save the original image.

4.3. ENCRYPTION ALGORITHMS

Data Encryption Standard (DES)

DES was designed to use a 64-bit key to encrypt and decrypt 64-bit blocks of data using a cycle of permutations, swaps, and substitutions. Encryption and decryption use the same key. A block to be encrypted is subjected to an initial permutation, then to a complex key-dependent computation, and then to a final permutation. The initial and final permutations take the 64-bit block and change the position of each bit in a pre-determined manner. The final permutation is the reverse of the initial permutation.

For the key-dependent computation, a key schedule function takes the 64-bit key and from it creates a set of sixteen 48-bit key blocks. Each of these is used in turn at each of the 16 rounds of the cipher function, which alters the plaintext in accordance with the specifications of an S-Box.

TRIPLE DES

The Data Encryption Standard (DES) was developed by an IBM team around 1974 and adopted as a national standard in 1977. Triple DES is a minor variation of this standard. It is three times slower than regular DES but can be billions of times more secure if used properly. Triple DES enjoys much wider use than DES because DES is so easy to break with today's rapidly advancing technology.

Triple DES was the answer to many of the shortcomings of DES. Since it is based on the DES algorithm, it is very easy to modify existing software to use Triple DES. It also has the advantage of proven reliability and a longer key length that eliminates many of the shortcut attacks that can be used to reduce the amount of time it takes to break DES.

Triple DES is simply another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. In Private Encryptor, you simply type in the entire 192-bit (24 character) key rather than entering each of the three keys individually. The Triple DES DLL then breaks the user provided key into three sub keys, padding the keys if necessary so they are each 64 bits long.

The procedure for encryption is exactly the same as regular DES, but it is repeated three times. Hence the name Triple DES. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key.

The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse. Like DES, data is encrypted and decrypted in 64-bit chunks. Unfortunately, there are some weak keys that one should be aware of: if all three keys, the first and second keys, or the second and third keys are the same, then the encryption procedure is essentially the same as standard DES. This situation is to be avoided because it is the same as using a really slow version of regular DES.

4.4. HASH ALGORITHMS

MD4

MD4 was developed by Rivest in 1990. The message is padded to ensure that its length in bits plus 64 is divisible by 512. A 64-bit binary representation of the original length of the message is then concatenated to the message. The message is processed in 512-bit blocks and each block is processed in three distinct rounds. It is worthy briefly discussing MD4 because MD5 shares the design goals of MD4. The following goals were listed:

Security: There is the usual requirement for a hash code, namely, that it be computationally infeasible to fond two messages that have the same message digest. MD4 has secured against brute-force attacks because of the length of the digest.

Speed: The algorithm is intended to be fast on 32-bit architectures. Thus, the algorithm is based on a simple set of primitive operations on 32-bit words.

Simplicity and Compactness: This algorithm is simple to describe and simple to program, without requiring large programs and substitution tables. These characteristics not only have obvious programming advantages but are desirable from a security point of view, because a simple algorithm is more likely to receive the necessary critical review.

Secure Hash Algorithm

Secure Hash Algorithm, a message digest algorithm developed by the NSA for use in the Digital Signature standard, FIPS number 186 from NIST. SHA is an improved variant of MD4 producing a 160-bit hash. SHA is one of two message digest algorithms available in IPSEC. The other is MD5. Some people do not trust SHA because it was developed by the NSA. There is, as far as we know, no cryptographic evidence that SHA is untrustworthy, but this does not prevent that view from being strongly held.

Secure Hash Algorithm takes a message of less than 264 bits in length and produces a 160-bit message digest. Secure Hash Algorithm. Computes a condensed representation of a message or a data file. When a message of any length is input, the SHA-1 produces a 160-bit output called a message digest.

The message digest can then be input to the Digital Signature Algorithm (DSA), which generates or verifies the signature for the message. The creator of the digital signature and the verifier of the digital signature must use the same hash algorithm.

A one-way hash function that creates a 20-byte (160-bit) hash or message digest to authenticate packet data. SHA is more resistant to attacks than MD5, but slower to compute. The term "SHA-0" is used to refer to the first version, "SHA-1" is used to refer to the second version, and the term "SHA" is used to refer to the pair of algorithms in the abstract.

4.5. SOFTWARE USED FOR IMPLEMENTATION

Visual Basic .NET 2003

Visual Basic .NET 2003 is the second release of Visual Basic .NET, building on the high productivity and outstanding performance of the first release. Using a single programming model, Visual Basic .NET 2003 enables you to easily create rich desktop applications for Microsoft Windows® and powerful Web applications. This release also includes integrated support for the creation of applications for wireless, Internet-enabled hand-held devices. You can accomplish these capabilities using the Visual Basic programming skills that you already have.

Unless otherwise noted, all of the features listed below are available in every member of the Visual Studio family product line which enables Visual Basic .NET development including: Visual Basic .NET 2003 Standard, Visual Studio .NET 2003 Professional, Visual Studio .NET 2003 Enterprise Developer, and Visual Studio .NET Enterprise Architect.

- Powerful Windows-based Applications in Less Time
- Direct Access to the Platform
- COM Interoperability
- Full Object-Oriented Constructs
- Reuse Existing Investments
- Upgrade Wizard
- Flexible, Simplified Data Access
- Improved Coding
- Web-based Applications

Reasons to Adopt Visual Basic .NET 2003

While there are countless reasons to adopt VB.NET 2003, the following are some of the reasons.

NEW—Enhanced Visual Studio .NET IDE

Visual Basic .NET 2003 provides developers with the award-winning Visual Studio .NET integrated development environment (IDE), which now includes faster startup time, enhanced Smart Listing for faster and more accurate coding, flexible Task Lists, property editors, forms designers, and much more.

NEW—Improved Debugging with IntelliSense

IntelliSense is now available within the Immediate Window for providing assistance while debugging applications. In addition, the Visual Basic .NET IDE now offers a simplified Debug Window to provide only the most pertinent information for quickly debugging Visual Basic .NET applications.

Build Applications for Windows

The Microsoft Windows® Forms Designer included in Visual Basic .NET 2003 is an enhanced version of the forms designer that Visual Basic developers have been using for years. Features include control anchoring and docking to eliminate the need for complex resize code, an in-place menu editor to deliver WYSIWYG menu creation, the tab order editor to provide rapid application development (RAD) organization of controls, and visual inheritance.

Build Applications for the Web

Visual Basic .NET 2003 delivers "Visual Basic for the Web." Using Web Forms, you can easily build true thin-client Web-based applications that intelligently render on any browser and on any platform. Programming with Web Forms combines the RAD experience of Visual Basic 6.0 forms with the easy deployment and maintenance of Web-based applications. The enhanced HTML editor delivers IntelliSense statement completion for HTML tags, and the separation of user interface (UI) from code enables more efficient team-based development.

Trouble-free Deployment of Windows-based Applications

Visual Basic .NET and the .NET Framework simplify Windows-based application deployment and help to make "DLL Hell" and component versioning issues a thing of the past. XCOPY deployment enables developers to install a Windows-based application simply by copying files to a directory. With Visual Basic .NET and auto-download deployment, Windows-based applications can be installed and executed simply by pointing a Web browser to a URL.

NEW—Target the Improved Microsoft .NET Framework 1.1

The .NET Framework version 1.1 provides numerous enhancements over the .NET Framework 1.0, including better scalability and performance and managed providers for Oracle and ODBC database connectivity. To ensure the highest level of compatibility, the .NET Framework 1.1 can be installed side-by-side with the .NET Framework 1.0.

NEW—Enhanced Upgrade Technology

Visual Basic .NET 2003 developers can now leverage even more of their existing investments in code and skills. The improved upgrade wizard enables developers to migrate up to 95 percent of existing code to Visual Basic .NET. The upgrade wizard is now available in Visual Basic .NET 2003 Standard edition and Visual Studio .NET 2003 Professional, Visual Studio .NET 2003 Enterprise, and Visual Studio .NET 2003 Enterprise Architect editions.

Powerful, Flexible Data Access

Visual Basic .NET provides developers with both the ActiveX® Data Objects (ADO) data access programming model for backward-compatibility, plus XML-based ADO.NET. With ADO.NET, developers gain access to more powerful components, such as the DataSet control and a strongly typed programming model that provides IntelliSense statement completion for data access code.

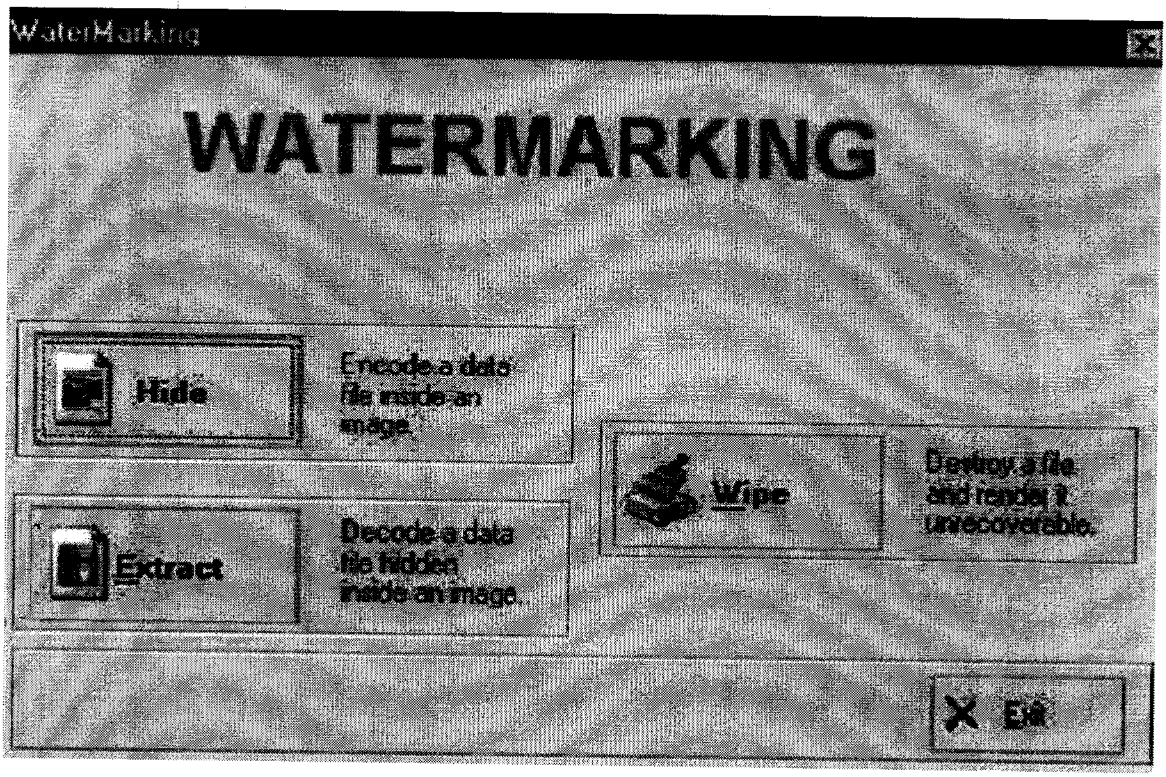
Native Support for XML Web Services

Visual Basic .NET enables developers to easily deploy and consume XML Web services from within any application. Web services reduce development time by enabling software integration with applications on any platform.

CHAPTER 5

5. RESULTS OBTAINED

5.1. HIDING THE IMAGE



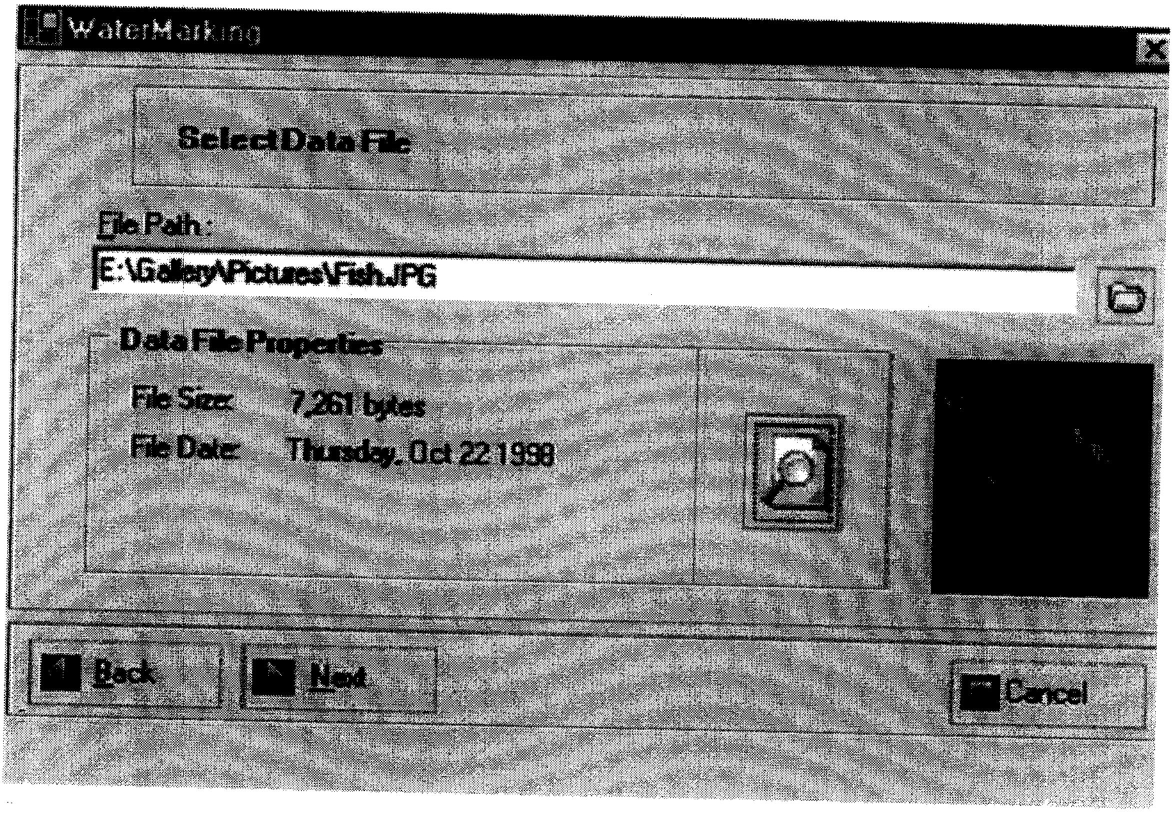
Encode a data file inside an image.

Decode a data file hidden inside an image.

Destroy a file and render it unrecoverable.

Description: The above figure shows the first form in the project, which indicates the three basic operations namely Hide, Extract and Wipe. If we click those buttons the corresponding operations would be carried on. The hide operation is continued after clicking the hide button.

5.1.1 SELECTING THE ORIGINAL IMAGE



File Path : E:\Gallery\Pictures\Fish.JPG

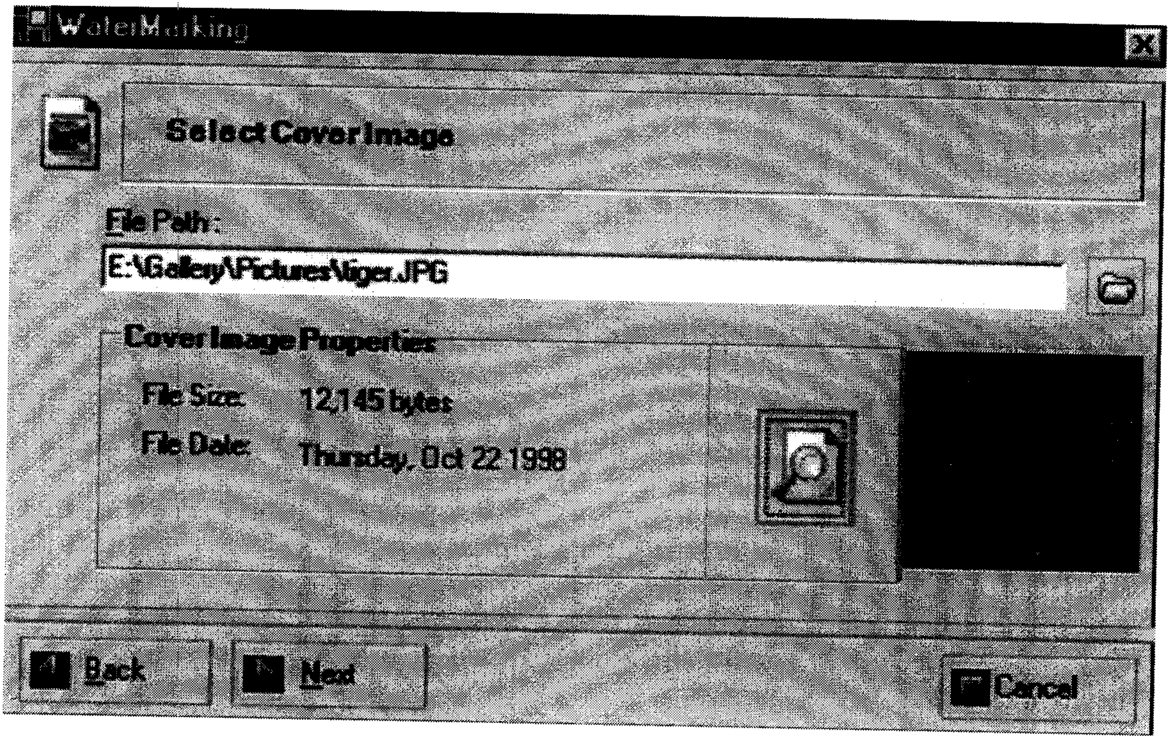
Data File Properties

File Size: 7,261 bytes

File Date: Thursday, Oct 22 1998

Description: The above figure comes after clicking the Hide button. In this the data file is selected. So the image selected in this form is considered as the original image. The file size of this image is noted.

5.1.2. SELECTING THE COVER IMAGE



File Path: E:\Gallery\Pictures\tiger.JPG

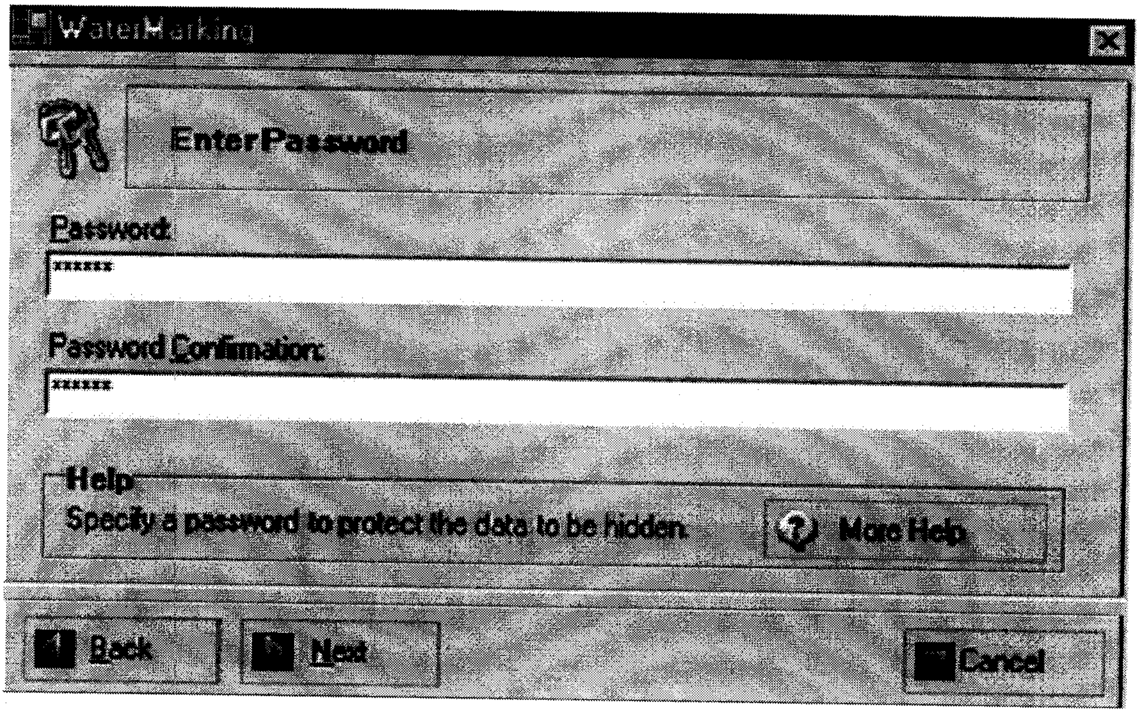
Cover Image Properties

File Size: 12,145 bytes

File Date: Thursday, Oct 22 1998

Description: The above figure comes after clicking the next button. In this the cover image is selected. So the image selected in this form is considered as the cover image and to this image the original image is embedded. The file size of this image is noted.

5.1.3. GETTING THE PASSWORD



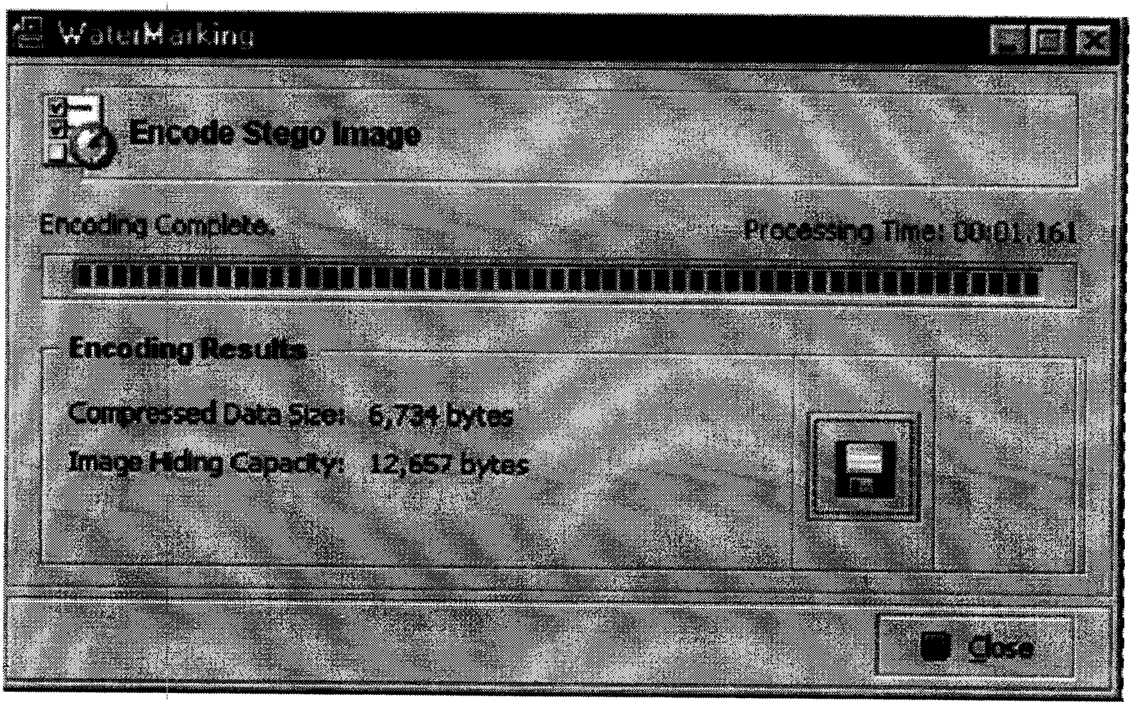
Password: ******

Password Confirmation: ******

Specify a password to protect the data to be hidden.

Description: The above figure comes after clicking the next button. In this the password is obtained from the user. So the password obtained in this form is encrypted and along with the original image it is embedded into the cover image.

5.1.4. ENCODING THE ORIGINAL IMAGE



Encode Stego Image

Encoding Complete.

Processing Time: 00:01.161

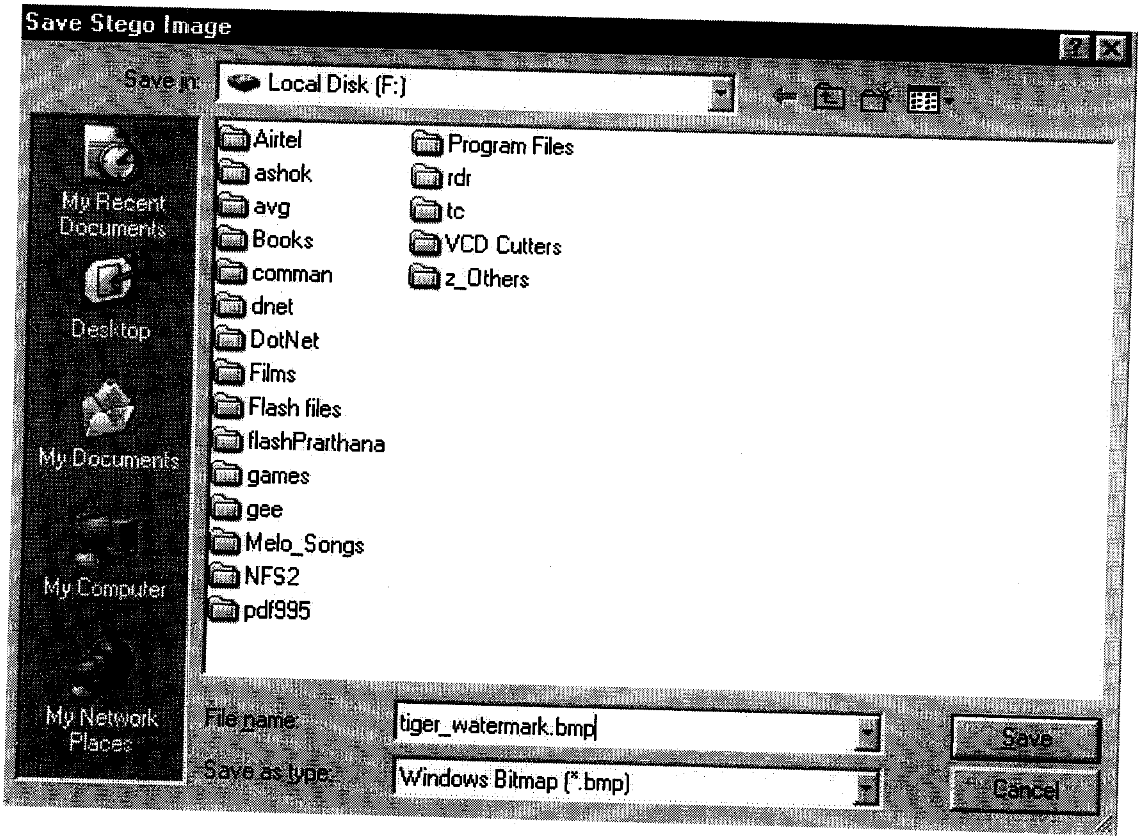
Encoding Results

Compressed Data Size: 6,734 bytes

Image Hiding Capacity: 12,657 bytes

Description: The above figure comes after clicking the next button. In this the selected original image is embedded into the cover image along with the encrypted password.

5.1.5. SAVING THE WATERMARKED IMAGE

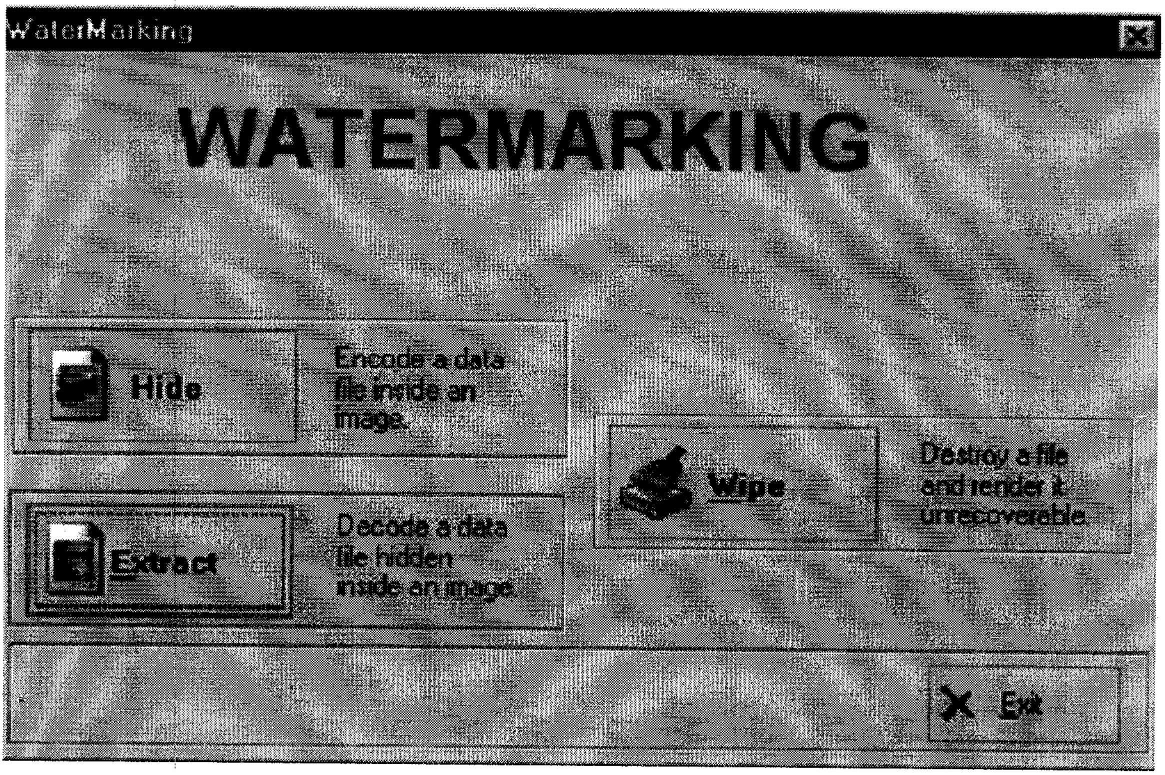


File name: tiger_watermark.bmp

Save as type: Windows Bitmap (*.bmp)

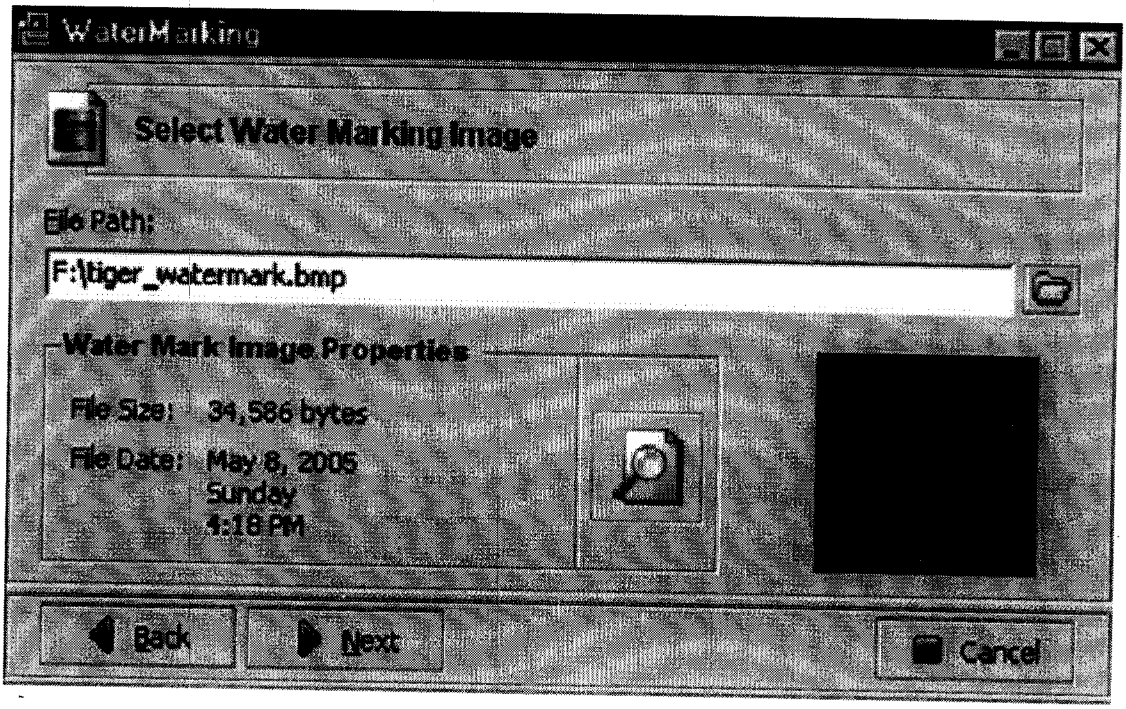
Description: The above figure comes after clicking the close button. In this the watermarked image is saved in the computer. Watermarked image is nothing but the image which contains the original image inside the cover image.

5.2 EXTRACTION OF ORIGINAL IMAGE


WaterMarking
WATERMARKING
Hide
Encode a data file inside an image.
Extract
Decode a data file hidden inside an image.
Wipe
Destroy a file and render it unrecoverable
Exit

Description: The above figure shows the first form in the project, which indicates the three basic operations namely Hide, Extract and Wipe. If we click those buttons the corresponding operations would be carried on. The extract operation is continued after clicking the Extract button.

5.2.1 SELECTING THE WATERMARKED IMAGE



File Path:
F:\tiger_watermark.bmp

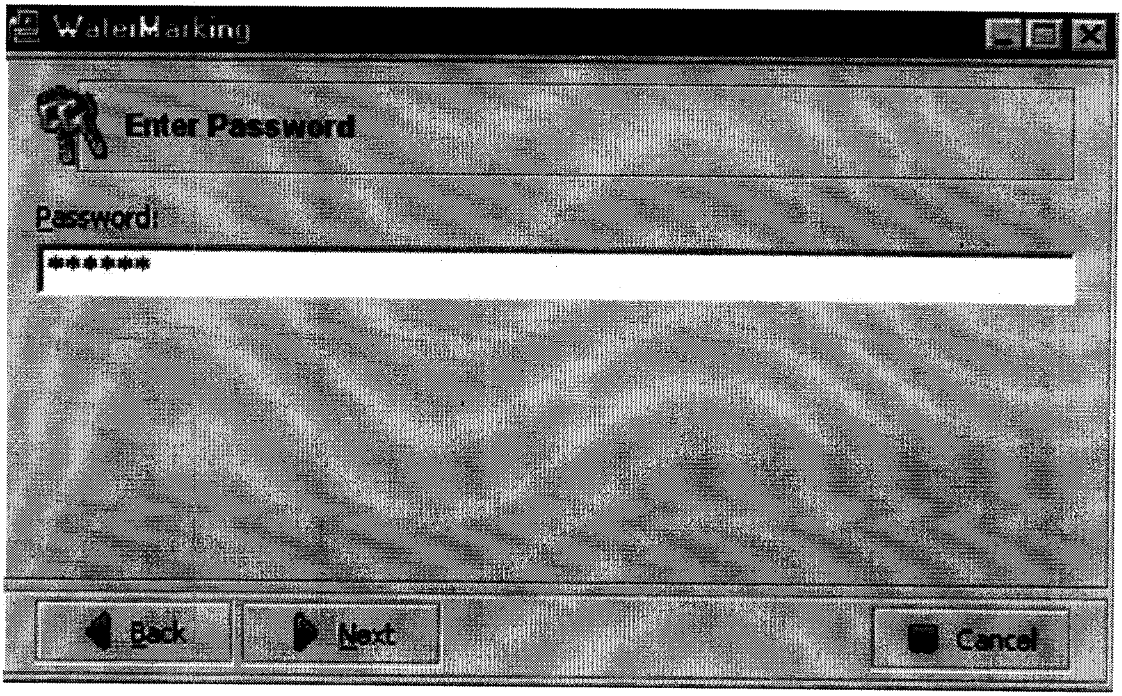
Water Mark Image Properties

File Size: 34,586 bytes

File Date: May 8, 2005
Sunday
4:18 PM

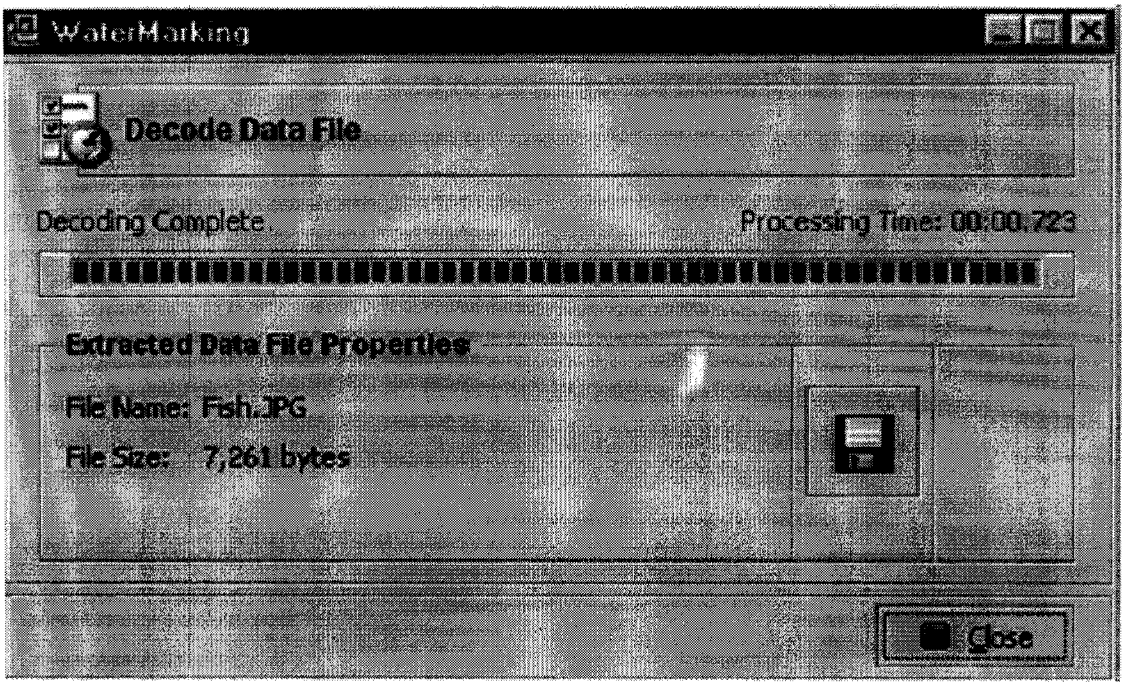
Description: The above figure comes after clicking the Extract button. In this the watermarked image is selected. So the image selected in this form is considered as the watermarked image and from this image the original image is extracted. Here in the watermarked image we can notice that the file size of the cover image is increased after embedding the original image.

5.2.2. GETTING THE PASSWORD



Description: The above figure comes after clicking the next button. In this the password is obtained from the user. So the password obtained in this form is decrypted and compared to the password given in hide module. If the comparison is satisfied the further process is continued. Else the correct password is obtained.

5.2.3. THE ORIGINAL IMAGE IS EXTRACTED



Decoding Complete

Processing Time: 00:00.723

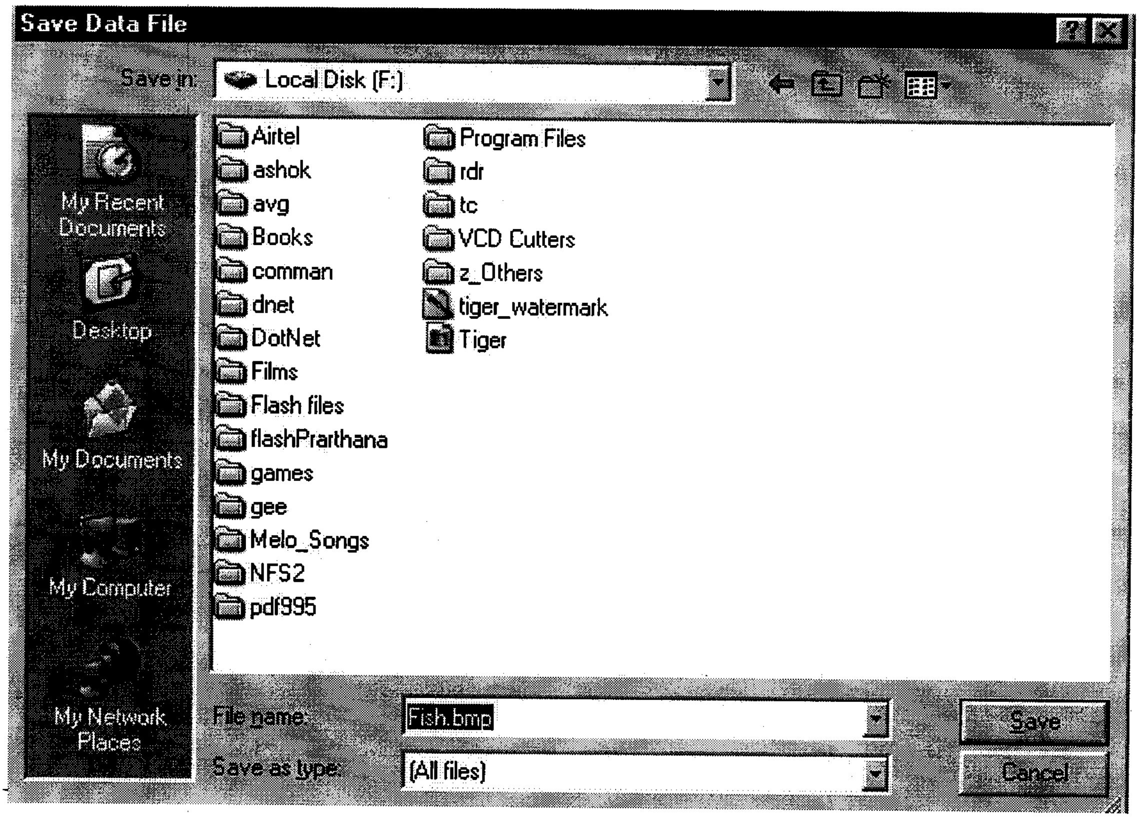
Extracted Data File Properties

File Name: Fish.JPG

File Size: 7,261 bytes

Description: The above figure comes after clicking the next button. In this the selected original image is extracted from the watermarked image.

5.2.4. ORIGINAL IMAGE IS SAVED



Save Data File

Save in: Local Disk (F:)

File name: Fish.bmp

Save as type: (All files)

Description: The above figure comes after clicking the close button. In this the selected original image is extracted from the watermarked image and it is stored in the computer.

CHAPTER 6

6. CONCLUSION AND FUTURE OUTLOOK

An image watermark algorithm is presented together with the results to demonstrate its capabilities and limitations. The algorithm is based on a random perturbation of the highest magnitude pixels corresponding to different sub images. The proposed method achieves the goal by inserting a watermark sample to highest magnitude pixels, with each perturbed differently. Thus a new embedding strategy is proposed with the highest magnitude pixels to place watermark in order to improve the robustness of the watermark, while al the previous methods avoid doing so. The same principle can be applied to other transforms, such as Fourier or wavelet.

Many potential applications exist for digital watermarking. Artists and photographers could mark their images to secure ownership rights. Publishing companies who commercially distribute their images could watermark them to prevent unauthorized distribution. Watermarking could also apply to other multimedia data such as audio and video. Compact disks and digital video disks are extremely susceptible to bootlegging via the internet. Digital watermarks might take part in diminishing this potential underground market.

APPENDIX

Authentication

A process used to verify the integrity of transmitted data.

Bitmap

Bitmapped images or bitmaps represent an image as a grid of picture elements or *pixels*, each with a value mapping a particular value of color and intensity.

Bi-level bitmap

Bi-level bitmaps store a single bit for each pixel. This is sometimes called a flat bitmap. A bi-level bitmap allows the representation of only two values of brightness or color information, typically black and white, although any two colors might equally be displayed.

Color resolution

The amount of information stored for each pixel determines its *pixel depth* or color depth. This is measured in the number of binary digits or bits employed to encode each pixel.

Copyright Protection

It is the protection of intellectual property; the data owner can embed a watermark representing the copyright information in his data.

Decryption

It is a method of translating the encrypted text or cipher text into original text or plain text. This method is also known as deciphering.

DES

Data Encryption Standard was designed to use a 64-bit key to encrypt and decrypt 64-bit blocks of data using a cycle of permutations, swaps, and substitutions.

Encryption

It is a method of translating the plain text into unintelligible form by some algorithm (cipher text). This method is also known as enciphering.

Frequency domain

The frequency-domain techniques first transform an image into a set of frequency domain coefficients. The transformation may be DCT, Fourier transform, or wavelet transform etc...

Grey-level bitmap

Grey-scale bitmaps store more than one bit for each pixel, making more than one value available for each point in the image. This is sometimes referred to as a *deep* bitmap.

Hash algorithm

Hash algorithm uses the hash function that maps a variable length data block or message (M) into a fixed length value called a hash code H(M).

Image

The term 'image' is used here to refer to graphic representations of real-world objects.

JPEG

Joint Photographic Experts Group is a compression standard for still color images and grey scale image, otherwise known as continuous-tone images.

Pixel

A pixel is made up of a triad. Pixels are arranged in an array of rows. Each row forms a scan line.

Password

A character string used to authenticate an identity. Knowledge of the password and its associated user ID is considered proof of authorization to use the capabilities associated with that user ID.

Resolution

The quality of a bitmap reproduction is a factor of the resolution of the bitmap. The resolution of a bitmap image is a product of the dimensions of the bitmap in pixels and the amount of information that is stored for each pixel.

Security

It is a method of providing safety to any image. The security can be provided by the means of password.

Spatial domain

The spatial-domain techniques directly modify the intensities or color values of some selected pixels.

Steganography

Steganography is the method of hiding any kind of information in an image. The information may be a text or image.

Triple DES

Triple DES is simply another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. The procedure for encryption is exactly the same as regular DES, but it is repeated three times.

Vector image

Vector images define an image as a set of instructions composed of graphic objects or primitives. Strictly speaking, vectors are lines, but vector graphics generally include ellipses, arcs, curves, filled areas and in many cases text.

Watermarking

A watermark is an invisible or visible signature embedded inside an image to show authenticity and ownership.

Sample Coding

```
Public Function Encode(ByVal PasswordHashMD5 As String, ByVal PasswordHashSHA As String) As Boolean

    Dim hBmp As Integer
    Dim Bits As Integer
    Dim C As Integer
    Dim E As Integer
    Dim MetadataCtr As Integer
    Dim ReadCtr As Integer
    Dim EmbedCtr As Integer
    Dim PixelCtr As Integer
    Dim Interval As Integer
    Dim Buffer As BitStream

    m_MetadataMaxPosX = m_ImageWidth - METADATA_BITS
    m_MetadataMaxPosY = m_ImageHeight - 1

    Do While (m_MetadataMaxPosX < 0)
      m_MetadataMaxPosX = m_ImageWidth + m_MetadataMaxPosX
      m_MetadataMaxPosY = m_MetadataMaxPosY - 1
    Loop


    If m_ImagePixels < 200 Then
      Interval = 1
    Else
      Interval = m_ImagePixels \ 100
    End If

    hBmp = Imager.CreateBitmapFromDIB(m_CoverImageDIB)


    InitializePRNG(PasswordHashMD5)

    m_MetadataPosC = Int(Rnd() * 3)
    m_MetadataPosX = Int(Rnd() * (m_MetadataMaxPosX + 1))
    m_MetadataPosY = Int(Rnd() * (m_MetadataMaxPosY + 1))

    Buffer = New BitStream

    MetadataCtr = 0
    ReadCtr = 0

    EmbedCtr = 0
    PixelCtr = 0
    ResetPixelPosition()

    For PixelCtr = 1 To m_ImagePixels

      If m_Abort Then
        m_Abort = False
        GoTo Cleanup
      End If


      If MetadataCtr < 1 Then

        SelectPixelChannel()

       If (m_PosC = m_MetadataPosC) And (m_PosY(m_PosC) =
m_MetadataPosY) And (m_PosX(m_PosC) = m_MetadataPosX) Then
          MetadataCtr = METADATA_BITS
          Buffer.InsertStringAtTop(m_DataFileChecksum)
          Buffer.InsertBitsAtTop(m_DataFileSize, FILESIZE_BITS)
          Buffer.InsertBitsAtTop(m_DataFileTime.dwLowDateTime,
FILEDATELOW_BITS)
          Buffer.InsertBitsAtTop(m_DataFileTime.dwHighDateTime,
FILEDATEHIGH_BITS)
          Buffer.InsertStringAtTop(m_DataFileName)
          Buffer.InsertStringAtTop(PasswordHashSHA)
        End If

      End If

      PerformCapacityEvaluation(C)

      If C > 0 Then

        If Buffer.Length < C Then

          If ReadCtr < m_DataFileSize Then

            ReadCtr = ReadCtr + 1
            Buffer.InsertByteAtEnd(m_DataFileBits(ReadCtr))

             Else


           Buffer.InsertByteAtEnd(Int(Rnd() * 256))
          End If

        End If
```

```
     Bits = Buffer.ExtractBits(C)

            E = PerformMinimumErrorReplacement(Bits, C)

            PerformErrorDiffusion(E)

            EmbedCtr = EmbedCtr + C

            If MetadataCtr > 0 Then MetadataCtr = MetadataCtr - C

        End If

        SelectPixelCoordinate()

        If (PixelCtr Mod Interval) = 0 Then

            RaiseEvent Progress(PixelCtr, m_ImagePixels)
            System.Windows.Forms.Application.DoEvents()
        ElseIf GetInputState() Then

            System.Windows.Forms.Application.DoEvents()
        End If

    Next PixelCtr

   If (PixelCtr Mod Interval) > 0 Then
        RaiseEvent Progress(PixelCtr, m_ImagePixels)
    End If

    m_waterImageDIB = Imager.CreateDIBFromBitmap(hBmp)

    m_ImageCapacity = (EmbedCtr - METADATA_BITS) \ 8

    Encode = (m_ImageCapacity >= m_DataFileSize)

    Imager.DeleteBitmap(hBmp)
    Erase m_ImageBits

End Function
```

CHAPTER 7

REFERENCES

[1] Cox.I.J, Miller.M.I, and Bloom.J.a, (2000), "Watermarking Applications and Properties", in IEEE International Conference on Information Technology, pp6-10.

[2] Adrian G. Bors and Ioannis Pitas, " Image watermarking using block site selection and DCT domain constraints", Department of Informatics University of Thessaloniki, Thessaloniki 540 06, Greece.

[3] Ken Cabeen and Peter Gent, "Image Compression and the Discrete Cosine Transform", Math 45, College of the Redwoods.

[4] R. Schyndel, A. Tirkel, and C. Osborne, "A Digital Watermark," Proc. IEEE Int. Conf. on Image Processing, Nov. 1994, vol. II, pp. 86-90.

[5] Wai C. Chu, "DCT-Based Image Watermarking Using Sub sampling", IEEE Transactions on Multimedia, Vol. 5, March 2003.

[6] Podilchuk.C.I and Delp.E.J. (2001), "Digital Watermarking Algorithms and Applications, IEEE Signal processing Magazine, vol.18,no 4, pp 33-46, July.

[7] N. F. Johnson and S. Katzenbeisser, "A survey of steganographic techniques", in S. Katzenbeisser and F. Peticolas(Eds.): Information Hiding, pp.43-78. Artech House, Norwood, MA, 2000.

[8] Weili Tang and Yoshinao Aoki, "A DCT-based coding of images in watermarking", Proceedings of the International Conference on Information, Communications and Signal Processing, Vol. 1, 1997, pp. 510-512.

[9] J J K. O Ruanaidh, W J. Dowling, and F M. Boland, "Watermaking igital images for copyrightprotection", IEE Proceedings: Vision, Image & Signal Processing, Vol. 143, No. 4, Aug. 1996, pp. 250-256.

[10] J. J. Chae and B. S. Manjunath, “A Technique for Image Data Hiding and Reconstruction without Host Image “,Department of Electrical and Computer Engineering University of California, Santa Barbara, CA 93106.

[11] *Ying Wang and Pierre Moulin*,” STEGANALYSIS OF BLOCK-DCT IMAGE STEGANOGRAPH”, University of Illinois at Urbana-Champaign Beckman Institute, CSL & ECE Dept.

[12] http://www.digimarc.com

ARULMIGU KALASALINGAM COLLEGE OF ENGINEERING

Accreditated by NBA-AICTE

Anandnagar, Krishnankoil-626190, Virudhunagar District, TN

Department of Instrumentation & Control Engineering

INTELLIGENT ENGINEERING

National Conference on "INTELLIGENT COMPUTING IN COMMUNICATION & AUTOMATION"

NCICCA -2005

1st - 2nd April, 2005

Phone:04563-289042,Fax:04563-289322,Email:ncicca_2005@yahoo.com, reg_ncicca2005@yahoo.com
Website:www.akce.ac.in

RGANISING COMMITTEE

hief Patron: Thiru T. Kalasalingam
Chairman,

atron : Thiru. K. Sridharan
Secretary

hair-Person: Dr. C. Thangaraj
Principal

DVISORY COMMITTEE

Dr. P. Kanagasapapathy, MIT
Dr. V. Abhaikumar, TCE
Dr. S. Stanley Johnson, KIT
Dr.T.K.Radhakrishnan, NIT
Dr. S. Radhakrishnan, AKCE
Dr. P. Subbaraj, AKCE
Dr. S. Arumugam, AKCE
Dr. V. Vasudevan, AKCE
Dr. D. Devaraj, AKCE
Prof. Sudhakar Gummadi, AKCE

RGANISING SECRETARY

Prof. S. Kannan,
Head, ICE Dept.
AKCE

RGANISING COMMITTEE

Mr. T. Sabapathy
Mr. P. Murugan
Mr. S. Murugan
Ms. K. Valarmathi
Mr. R. Arivalahan
Mr. B. Kannapiran
Mr. M. Pallikonda Rajasekaran
Mr. P. S. Godwin Anand
Ms. C. Ramya
Mr. G. Petchinathan
Ms. R. Pandiarajamani
Ms. B. Ambika

Ref No: AKCE/ICE/NCICCA/ 171 19-3-05

To:

MR. S. KAWSIKA RAJA

Sir/ Madam,

Sub: (Intimation of acceptance of paper-reg.,)

We are happy to inform you that your paper entitled DCT BASED IMAGE WATERMARKING FOR SECURING IMAGES has been accepted for presenting in Session III of NCICCA-2005 to be held on 1st and 2nd April 2005. Your paper will be published in the Proceedings which will be released at the time of Conference. The tentative schedule is enclosed for your reference. (TA/DA is not entertained). Kindly confirm your date of arrival before 28.03.05. If necessary, accommodation will be provided on payment of Rs.100/- at the time of registration.

e-mail id for correspondence : reg_ncicca2005@yahoo.com

Thanking You,

S. Kannan

Organizing Seceratary
NCICCA-2005

- In case of inconvenienc