**COLOR IMAGE WATERMARKING FOR RGB IMAGES AUTHENTICATION AND COPYRIGHT PROTECTION**

**ABSTRACT**

We propose a robust image watermarking technique for the copyright protection based on Discrete Wavelet Transform (DWT). In this technique a multi-bit watermark is embedded into the low frequency sub-band of a cover image by using alpha blending technique. The insertion and extraction of the watermark in the color cover image is found to be efficient than other transform techniques. We demonstrate that the watermarks generated with the proposed algorithm are invisible and visible and the quality of watermarked image and the recovered image are improved. The image quality of the extracted watermark is analyzed by comparing with the input image by using statistical parameters such as peak-signal-to-noise-ratio (PSNR) and mean square error (MSE).

**CHAPTER I**

**INTRODUCTION-IMAGE PROCESSING**

* 1. **GENERAL**

The term digital image refers to processing of a two dimensional picture by a digital computer. In a broader context, it implies digital processing of any two dimensional data. A digital image is an array of real or complex numbers represented by a finite number of bits. An image given in the form of a transparency, slide, photograph or an X-ray is first digitized and stored as a matrix of binary digits in computer memory. This digitized image can then be processed and/or displayed on a high-resolution television monitor. For display, the image is stored in a rapid-access buffer memory, which refreshes the monitor at a rate of 25 frames per second to produce a visually continuous display.

**1.1.1THE IMAGE PROCESSING SYSTEM**

**DIGITIZER**

**MASS STORAGE**

**HARD COPY DEVICE**

**DISPLAY**

**IMAGE PROCESSOR**

**DIGITAL COMPUTER**

**OPERATOR CONSOLE**

**FIG 1.1 BLOCK DIAGRAM FOR IMAGE PROCESSING SYSTEM**

* **DIGITIZER**

A digitizer converts an image into a numerical representation suitable for input into a digital computer. Some common digitizers are

* Microdensitometer
* Flying spot scanner
* Image dissector
* Videocon camera
* Photosensitive solid- state arrays.
* **IMAGE PROCESSOR**

An image processor does the functions of image acquisition, storage, preprocessing, segmentation, representation, recognition and interpretation and finally displays or records the resulting image. The following block diagram gives the fundamental sequence involved in an image processing system.

**PROBLEM DOMAIN**

**KNOWLEDGE**

**BASE**

**SEGMENTATION**

**PREPROCESSING**

**IMAGE ACQUISITION**

**RECOGNITION & INTERPRETATION**

**REPRESENTATION & DESCRIPTION**

**RESULT**

**FIG 1.2 BLOCK DIAGRAM OF FUNDAMENTAL SEQUENCE INVOLVED IN AN IMAGE PROCESSING SYSTEM**

As detailed in the diagram, the first step in the process is image acquisition by an imaging sensor in conjunction with a digitizer to digitize the image. The next step is the preprocessing step where the image is improved being fed as an input to the other processes. Preprocessing typically deals with enhancing, removing noise, isolating regions, etc. Segmentation partitions an image into its constituent parts or objects. The output of segmentation is usually raw pixel data, which consists of either the boundary of the region or the pixels in the region themselves. Representation is the process of transforming the raw pixel data into a form useful for subsequent processing by the computer. Description deals with extracting features that are basic in differentiating one class of objects from another. Recognition assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects. The knowledge about a problem domain is incorporated into the knowledge base. The knowledge base guides the operation of each processing module and also controls the interaction between the modules. Not all modules need be necessarily present for a specific function. The composition of the image processing system depends on its application. The frame rate of the image processor is normally around 25 frames per second.

* **DIGITAL COMPUTER**

Mathematical processing of the digitized image such as convolution, averaging, addition, subtraction, etc. are done by the computer.

* **MASS STORAGE**

The secondary storage devices normally used are floppy disks, CD ROMs etc.

* **HARD COPY DEVICE**

The hard copy device is used to produce a permanent copy of the image and for the storage of the software involved.

* **OPERATOR CONSOLE**

The operator console consists of equipment and arrangements for verification of intermediate results and for alterations in the software as and when require. The operator is also capable of checking for any resulting errors and for the entry of requisite data.

* + 1. **IMAGE PROCESSING FUNDAMENTAL**

Digital image processing refers processing of the image in digital form. Modern cameras may directly take the image in digital form but generally images are originated in optical form. They are captured by video cameras and digitalized. The digitalization process includes sampling, quantization. Then these images are processed by the five fundamental processes, at least any one of them, not necessarily all of them.

**IMAGE PROCESSING TECHNIQUES**

This section gives various image processing techniques.

Image Enhancement

Image Restoration

Image Analysis

Image Compression

Image Synthesis

**FIG 1.3: IMAGE PROCESSING TECHNIQUES**

* **IMAGE ENHANCEMENT**

Image enhancement operations improve the qualities of an image like improving the image’s contrast and brightness characteristics, reducing its noise content, or sharpen the details. This just enhances the image and reveals the same information in more understandable image. It does not add any information to it.

* **IMAGE RESTORATION**

Image restoration like enhancement improves the qualities of image but all the operations are mainly based on known, measured, or degradations of the original image. Image restorations are used to restore images with problems such as geometric distortion, improper focus, repetitive noise, and camera motion. It is used to correct images for known degradations.

* **IMAGE ANALYSIS**

Image analysis operations produce numerical or graphical information based on characteristics of the original image. They break into objects and then classify them. They depend on the image statistics. Common operations are extraction and description of scene and image features, automated measurements, and object classification. Image analyze are mainly used in machine vision applications.

* **IMAGE COMPRESSION**

Image compression and decompression reduce the data content necessary to describe the image. Most of the images contain lot of redundant information, compression removes all the redundancies. Because of the compression the size is reduced, so efficiently stored or transported. The compressed image is decompressed when displayed. Lossless compression preserves the exact data in the original image, but Lossy compression does not represent the original image but provide excellent compression.

* **IMAGE SYNTHESIS**

Image synthesis operations create images from other images or non-image data. Image synthesis operations generally create images that are either physically impossible or impractical to acquire.

**1.1.3 Image types**

There are several ways of encoding the information in an image.

1. Binary image
2. Grayscale image
3. Indexed image
4. True color or RGB image

* **Binary image**

Each pixel is just blackor white. Since there are only two possible values for each pixel (0, 1), we only need one bitper pixel.

* **Grayscale image**

Each pixel is a shade of gray, normally from 0 (black) to 255(white). This range means that each pixel can be represented by eight bits, or exactly one byte. Other grayscale ranges are used, but generally they are a power of 2.

* **Indexed image**

An indexed image consists of an array and a color map matrix. The pixel values in the array are direct indices into a color map. By convention, this documentation uses the variable name X to refer to the array and map to refer to the color map.

* **True Color or RGB image**

Each pixel has a particular color; that color is described by the amount of red, greenand bluein it. If each of these components has a range 0–255, this gives a total of 2563different possible colors. Such an image is a “stack” of three matrices; representing the red, greenand bluevalues for each pixel. This means that for every pixel there correspond 3 values.

**1.1.4 APPLICATIONS of image processing**

Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods. Here is a short list just to give some indication of the range of image processing applications.

* **DOCUMENT PROCESSING**

It is used in scanning, and transmission for converting paper documents to a digital image form, compressing the image, and storing it on magnetic tape. It is also used in document reading for automatically detecting and recognizing printed characteristics.

* **Medicine**

Inspection and interpretation of images obtained from X-rays, MRI or CAT scans, analysis of cell images, of chromosome karyotypes. In medical applications, one is concerned with processing of chest X-rays, cineangiograms, projection images of transaxial tomography and other medical images that occur in radiology, nuclear magnetic resonance (NMR) and ultrasonic scanning. These images may be used for patient screening and monitoring or for detection of tumors’ or other disease in patients.

* **Industry**

Automatic inspection of items on a production line, inspection of paper samples.

* **DEFENSE/INTELLIGENCE**

It is used in reconnaissance photo-interpretation for automatic interpretation of earth satellite imagery to look for sensitive targets or military threats and target acquisition and guidance for recognizing and tracking targets in real-time smart-bomb and missile-guidance systems.

* **RADAR IMAGING SYSTEM**

Radar and sonar images are used for detection and recognition of various types of targets or in guidance and maneuvering of aircraft or missile systems.

* **Agriculture**

Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops, inspection of fruit and vegetables distinguishing good and fresh produce from old.

**CHAPTER 2**

**INTRODUCTION-WATERMARKING**

Digital image watermarking has become one of the most widely accepted solutions to this problem. Algorithms for digital watermarking embed into the original media specific identifying information (the “digital watermark”), such as a logo, fingerprint, or serial number.

**1.1 Introduction**

Watermarking has been widely used for proof of ownership and copyright protection; however, it has also been applied to applications such as broadcast monitoring, data integrity verification, and image indexing and labeling. Digital watermarking has been investigated for the last several decades and is a mature field of research. However, current efforts try to improve its performance, as new requirements and challenges posed by new applications motivate the need for continued research in this area.

One new requirement is the ability of a watermark to survive multiple, consecutive attacks. Over the last five years, MMS and online sharing of digital imagery on sites such as Facebook, Twitter, and other social and traditional media outlets, have become standard ways of disseminating photos. Typically, the original image is subjected to various forms of manual and/or automated processing before being posted, particularly when using popular sharing software such as Instagram. Such photos are typically resized (resulting in slight blurring), then cropped, then contrast-adjusted, then rotated (e.g., to make horizon lines horizontal), and then compressed via JPEG (resulting in blurring and blocking). Furthermore, if the images are transmitted over lossy communication channels, further degradation may result from transcoding, packet loss, and/or attempted corrections for such loss. The ability to survive multiple consecutive attacks was not a mainstream requirement 10-15 years ago, but it is clearly important today.

**1.1.1 Logo watermarking**

One popular thrust in digital image watermarking is logo watermarking, in which the watermark is itself an image, typically a small and user-specified image representing a symbol or a trademark. There has been significant research devoted to logo watermarking over the last 15 years. Invisible logo watermarking further requires that the watermark is visually imperceptible when placed within the host image, yet is visually recognizable when extracted. The primary advantage of logo watermarking is the ability to visually compare the extracted logo with the original logo during verification. It is well-known that the human visual system is unmatched in its ability to perform visual recognition, even for small thumbnail-sized images. Thus, even an untrained viewer can often effortlessly determine whether the extracted logo matches the original logo.

Logo watermarking has to overcome multiple challenges. One challenge is that the extracted watermark must be visually meaningful in order to facilitate a visual comparison, as opposed to simply catering to a present/not-present decision. Thus, the watermark must be able to survive attacks such that the visual quality of the extracted logo gracefully degrades as the attack intensity increases. Another challenge to logo watermarking and some other non-logo watermarking algorithms, such as copy protection codes, stems from the fact that the logo (or watermark in general) is specified by the end-user and not by the watermarking algorithm; thus, the algorithm designer cannot choose which data to embed. A successful logo watermarking algorithm must be able to handle a variety of logos, which largely restricts the algorithm from employing approaches which are tailored to particular data patterns.

Binary logo watermarking has also been performed in the discrete cosine transform (DCT) and discrete wavelet transform (DWT) domains. In addition to binary logo watermarking, algorithms for invisible logo watermarking have also been designed to operate with grayscale logos, and there has been significant research devoted to both categories.

**1.1.2 Applications of watermarking**

Invisible logo watermarking is particularly useful for more recent applications, such as watermarking for auto- mated quality monitoring of multimedia transmission for embedding QR codes in images and for embedding hospital and/or calibration logos in medical imagery. In particular, watermarking has recently emerged as a promising approach to no-reference image and video quality assessment. No-reference quality assessment algorithms seek to estimate the quality of an image/video without having access to the original, undistorted image/video, which is a challenging research area that has traditionally relied on the use of statistical image features. Watermarking has shown recent promise for this task, and further improvements could be realized via the use of logo watermarking, which would allow the use of a full-reference quality assessment algorithm applied to the original and extracted logos.

**1.1.3 Properties of Watermarks**

There are a number of desirable characteristics that a watermark should exhibit. These include that it be difficult to notice, robust to common distortions of the signal, resistant to malicious attempts to remove the watermark, Support a sufficient data rate commensurate with the application, and allow multiple watermarks to be added and that the decoder be scalable. Difficult to notice the watermark should not be noticeable to the viewer nor should the watermark degrade the Quality of the content. We had used the term imperceptible” and this is certainly the ideal.

Early work on watermarking focused almost exclusively on designing watermarks that were imperceptible and therefore often placed watermark signals in perceptually insignificant regions on the content. However, other properties of a watermark conflict with this choice.

**CHAPTER 3**

**PROJECT METHODOLOGIES**

**3.1 OBJECTIVE**

* Digital watermarking is now a relatively focused technique aimed at providing a reliable way to authenticate images or protect copyrights protection.
* In the existing mode such as spatial domain, the watermark is not properly embedded and extracted and it leads to improper protection of secret information.
* The existing spatial domain is transformed into frequency domain using discrete wavelet transform.
* In our proposed work the watermark is usually embedded invisibly in the images to avoid attracting the attention of malicious attackers. We are using color image watermarking, which has high efficiency compared to the traditional grayscale image watermarking.

**3.2 EXISTING SYSTEM**

* The visible watermarking was implemented in early stages.
* Pixel addition based techniques such that the pixel values of the original image and the watermark are combined.
* In existing papers, Watermarking was done in spatial domain. The spatial domain is the normal image space, in which a change in position in image directly projects to a change in position in space. Ex.-Least Significant bit (LSB) method.

**3.3 PROBLEM IDENTIFICATION**

The embedding and extraction of logo image invisibly into the original input image is very important in the watermarking aspects. Embedding and extraction of logo image into the original input image and the watermarked image respectively, helps the user to protect their personal data’s and other authenticated devices from the hackers. Use of efficient and better robust algorithm such as Discrete Wavelet Transform (DWT) gives accurate classification results.

**3.3.1 EXISTING SYSTEM DISADVANTAGES**

* Due to the less robustness of existing watermarking algorithms, the watermarked image looks in different color, so the people can easily identify that some kind of watermark is added.
* Furthermore, image quality may be degraded by the watermark
* The rate–distortion performance is low, and there is a leakage of statistical information.
* The problem in existing scheme is that data is highly sensitive to noise and is easily destroyed.
  1. **PROPOSED SYSTEM**

In this work, we are going to embed the logo into the host image to generate the watermarked image and to detect the hidden watermark (i.e., logo) from the watermarked image. This concept can be achieved using watermarking techniques. We have implemented a robust image watermarking technique for the logo detection based on Discrete Wavelet Transform (DWT). In this technique a multi-bit watermark (the logo) is embedded into the low frequency sub-band of an input image by using alpha blending technique. Then IDWT is applied to combine alpha blended image with the other sub-bands and high frequency coefficients to form the watermarked biometric image. Here we are going to implement invisible watermarking; hence the logo will be added with the input image to form the watermarked image. Then at the decryption stage, by using Alpha Blending Extraction Technique we are successfully extracting the watermark content (i.e. logo) present in the watermarked image. We demonstrate that the watermark generated with the proposed algorithm is invisible and the quality of watermarked image and the recovered image are improved. The quality of the extracted image is analyzed by using statistical parameters such as Peak-Signal-to-Noise-Ratio (PSNR) and Mean Square Error (MSE).

**PROPOSED SYSTEM BLOCK DIAGRAM**

1. **WATERMARK EMBEDDING**

**DWT**

**INPUT IMAGE (ORIGINAL)**

**WATER-MARKED IMAGE**

**ALPHA BLENDING EMBEDDING TECHNIQUE**

**IDWT**

**DWT**

**WATERMARK (SECRET IMAGE)**

1. **WATERMARK EXTRACTION**

**WATERMARKED IMAGE**

**DWT**

**DWT**

**ALPHA BLENDING EXTRACTION TECHNIQUE**

**EXTRACTED WATERMARK SECRET IMAGE**

**IDWT**

**INPUT IMAGE (ORIGINAL)**

**FIG: BLOCK DIAGRAM OF PROPOSED SYSTEM**

**DETAILED DESIGN FOR WATERMARK EMBEDDING**.

**LL**

**DECOMPOSED IMAGE**

**LOW FREQUENCY COEFFICIENTS**

**DISCRETE WAVELET TRANSFORM (DWT)**

**INPUT FACE IMAGE**

**HH**

**LL**

**HL**

**HIGH FREQUENCY COEFFICIENTS**

**WATERMARKED IMAGE**

**ALPHA BLENDING EMBEDDING TECHNIQUE**

**INVERSE DISCRETE WAVELET TRANSFORM**

**WATERMARK (LOGO)**

**DISCRETE WAVELET TRANSFORM (DWT)**

**3.4.1 PROPOSED ADVANTAGES**

* Output image quality is high.
* It is not sensitive to noise.
* Performance measurements (PSNR) value is high.
* Error rate (MSE) is low.

**3.4.2 PROPOSED TECHNIQUES**

* DISCRETE WAVELET TRANSFORM (DWT) AND INVERSE DWT
* ALPHA BLENDING EMBEDDING TECHNIQUE
* ALPHA BLENDING EXTRACTION TECHNIQUE

**CHAPTER 4**

**LITERATURE SURVEY**

**[1] Klimis Ntalianis and Nicolas Tsapatsoulis, “Remote Authentication via Biometrics: A Robust Video-Object Steganographic Mechanism Over Wireless Networks,” in IEEE Transactions on Emerging Topics In Computing, Jan. 2015.**

This paper proposes a robust authentication mechanism based on semantic segmentation, chaotic encryption and data hiding. Assuming that user X wants to be remotely authenticated; initially X’s video object (VO) is automatically segmented, using a head and body detector. Next, one of X’s biometric signals is encrypted by a chaotic cipher. Afterwards the encrypted signal is inserted to the most significant wavelet coefficients of the VO, using its Qualified

Significant Wavelet Trees (QSWTs). QSWTs provide both invisibility and significant resistance against lossy transmission and compression, conditions that are typical in wireless networks. Finally, the Inverse Discrete Wavelet Transform (IDWT) is applied to provide the stego-object (SO).

**DRAWBACK**

* Computational complexity is high.
* There are chances of loss of original data while decrypting, since biometric signals are encrypted by high level transform.
* While Transmission over wireless networks, there may be chances of loss of data.
* DWT algorithm is used, which already exists.

**[2] Mehran Andalibi and Damon M. Chandler,, “Digital Image watermarking via adaptive logo texturization,” in IEEE Transactions on Image processing, Dec. 2015.**

Here, logo is considered as the watermark. The logo should be embedded into the original input image to form the watermarked image using DWT algorithm and the watermark (logo) was decrypted using IDWT. Thus Digital Image watermarking is done.

**DRAWBACK**

* DWT algorithm is used, which already exists.
* Not so robust.

**[3] Huang Lidong, Zhao Wei, Wang Jun and Sun Zebin, “Combination of contrast limited adaptive histogram equalization and discrete wavelet transform for image enhancement,” in IET Transactions on Image processing, 2015, Vol. 9, Issue no. 10, pp. 908–915**

This study presents a novel image enhancement method, named CLAHE-discrete wavelet transform (DWT), which combines the CLAHE with DWT. The new method includes three main steps: First, the original image is decomposed into low-frequency and high-frequency components by DWT. Then, the authors enhance the low-frequency coefficients using CLAHE and keep the high-frequency coefficients unchanged to limit noise enhancement. This is because the high-frequency component corresponds to the detail information and contains most noises of original image. Finally, reconstruct the image by taking inverse DWT of the new coefficients.

**DRAWBACK**

* Watermarking was not done, only image enhancement was done.
* It faces the contrast overstretching and noise enhancement problems.

**[4] Xinpeng Zhang*, Member, IEEE*, Yanli Ren, Liquan Shen, Zhenxing Qian, and Guorui Feng, “Compressing Encrypted Images with Auxiliary information,” IEEE Transactions on Image processing, vol. 16, no. 5, pp. 1327–1336, Aug 2014.**

This paper proposes a novel scheme of compressing encrypted images with auxiliary information. The content owner encrypts the original uncompressed images and also generates some auxiliary information, which will be used for data compression and image reconstruction. At receiver side, the principal image content can be reconstructed using the compressed encrypted data and the secret key.

**DRAWBACK**

* The proposed compression approach is compatible with the modulo-256 addition encryption, but is not suitable for other encryption methods, such as standard stream cipher or AES/DES.
* Low ratio/distortion performance.

**[5] W. Hong and M. Hang, “Robust Digital Watermarking Scheme for Copy Right Protection”, IEEE Trans. Signal Process, vo.l2, pp. 1- 8, 2006.**

W. Hong et al. proposed a robust digital watermarking scheme for copyright protection of digital images based on sub-sampling. The watermark is a binary image, which is embedded in discrete transform coefficient of the host image and not used in the original image. In this scheme, they had used chaotic map in watermarked image. However the result of watermark image is good and robust to attack.

**DRAWBACK**

* Reconstructed image quality is low.
* Watermark is only a binary image.

**[6] Blossom Kaur, Amandeep Kaur, Jasdeep Singh, “Steganographic Approach for hiding Image in DCT Domain”, International Journal of Advances in Engineering & Technology, July 2011.**

Blossom et al. proposed a DCT based watermarking scheme which provides higher resistance to image processing attacks such as JPEG compression, noise, rotation, translation etc. In this approach, the watermark is embedded in the mid frequency band of the DCT blocks carrying low frequency components and the high frequency sub band components remain unused. Watermark is inserted by adjusting the DCT coefficients of the image and by using the private key. Watermark can then be extracted using the same private key without resorting to the original image

**DRAWBACK**

* Not so robust.
* Reconstructed image quality is low.
* DCT is traditional algorithm.

**[7] Baisa L. Gunjal, R.R. Manthalkar, An overview of transform domain robust digital image watermarking algorithms, Journal of Emerging Trends in Computing and Information Sciences, 2010.**

Compared to spatial domain techniques, frequency-domain watermarking techniques proved to be more effective with respect to achieving the imperceptibility and robustness requirements of digital watermarking algorithms. Commonly used frequency-domain transforms include the Discrete Wavelet Transform (DWT), the Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT). However, DWT has been used in digital image watermarking more frequently due to its excellent spatial localization and multi-resolution characteristics, which are similar to the theoretical models of the human visual system. Further performance improvements in DWT-based digital image watermarking algorithms could be obtained by increasing the level of DWT.

**[8] X. Xia, C. Boncelet, and G. Arce, A Multiresolution Watermark for Digital Images, Proc. IEEE Int. Conf. on Image Processing, Oct. 1997.**

Xia et.al proposed a watermarking scheme based on the Discrete Wavelet Transform (DWT). The watermark, modeled as Gaussian noise, was added to the middle and high frequency bands of the image. The decoding process involved taking the DWT of a potentially marked image. Sections of the watermark were extracted and correlated with sections of the original watermark. If the cross-correlation was above a threshold, then the watermark was detected. Otherwise, the image was decomposed into finer and finer bands until the entire, extracted watermark was correlated with the entire, original watermark. The problem with the proposed method is that this technique is susceptible to geometric attacks.

**[9] Bhatnagar, G. and Raman, B., A new robust reference watermarking scheme based on DWT-SVD, Elsevier B.V. All rights reserved 2008.**

G. Bhatnagar et al, presented a semi-blind reference watermarking scheme based on discrete wavelet transform (DWT) and singular value decomposition (SVD) for copyright protection and authenticity. Their watermark was a gray scale logo image. For watermark embedding, their algorithm transformed the original image into wavelet domain and a reference sub-image is formed using directive contrast and wavelet coefficients. Then, their algorithm embedded the watermark into reference image by modifying the singular values of reference image using the singular values of the watermark.

**[10] Barni M, Bartolini F, Piva, An Improved Wavelet Based Watermarking Through Pixelwise Masking, IEEE transactions on image processing, 2001.**

M. Barni et al. have developed an improved wavelet-based watermarking through pixel-wise masking. It is based on masking watermark according to characteristics of HVS. The watermark is adaptively added to the largest detail bands. The watermark weighing function is calculated as a simple product of data extracted from HVS model. The watermark is detected by correlation. The proposed method is robust to various attacks but this method is complex than other transform technique.

**[11] D. Kundur and D. Hatzinakos, Digital Watermarking using Multiresolution Wavelet Decomposition, Proceedings, IEEE International Conference Acoustic, Speech, Signal Processing, 1998.**

Kundur et al. decomposed binary logo through DWT. The watermark is scaled by a salience factor, computed on a block by block basis, depending on local image noise sensitivity. It is then repeatedly added to the subbands of DWT decomposition of host image. Visual masking is thus exploited upto only block resolution. A binary code is embedded by suitably quantizing the coefficients of detail bands. For watermark recovery, the embedded binary code is estimated by analyzing coefficients quantization. Once the code is estimated, it is correlated and result is compared to a threshold chosen on the basis of a given false positive probability.

**CHAPTER 5**

**MODULE DESCRIPTION**

**5.1 MODULE NAMES**

* Pre-processing.
* Watermark embedding
* Watermark extraction
* Performance analysis

**5.2 MODULE DESCRIPTIONS**

**PREPROCESSING**

It describes the individual functionalities performed during preprocessing phase.

Images such as Lena, Mandrill, Barbara and pepper which are available default in Matlab are used as inputs.

* 1. Matlab images are taken as the input, which is in jpg and png format.
  2. The input image contains different levels and distributions of texture to demonstrate the capability for invisible watermarking.

**RESCALING**

3. Since the calculation of DWT values needs a square matrix, images needs to have a standard size thus resizing of image is performed.

1. Both Original Input and Logo Images are resized.

**CHANNEL SEPARATION**

1. After Resizing image will be transformed as of RGB towards individual components, Red, Green and blue evaluated by way of Channel separation.
2. We can convert RGB image into grayscale image.

**Applying DWT**

1. For color image watermarking, DWT is performed on each individual R, G, and B components.
2. DWT is performed on the Red Channel.
3. DWT is performed on the Green Channel.
4. DWT is performed on the Blue Channel
5. For Grayscale image watermarking DWT is performed on grayscale image.

**INPUT IMAGE**

**RESCALING**

**RGB TO CHANNEL CONVERSION**

**WATERMARK**

**Fig. 3.2 Level1 DFD for Preprocessing**

**WATERMARK EMBEDDING**

1. In this process we propose an approach of hiding the watermark (logo) into the input image and watermarked image/encrypted image is formed.
2. First the host image (original input image) is taken as the input and DWT (Discrete Wavelet Transform) is applied to the image which decomposes image into low frequency and high frequency components.
3. Second the watermark image (logo image) is taken as the input and DWT (Discrete Wavelet Transform) is applied to the image which decomposes image into low frequency and high frequency components.
4. The output of the DWT is LL, HL, LH and HH band.
5. LL-Approximation Image, HL-Horizontal Image, LH-Vertical Image and HH-Diagonal Image.
6. The LL part of the watermark is embedded into the LL part of the input image using alpha blending embedding technique.
7. Then IDWT is applied to combine alpha blended image with the other sub-bands and high frequency coefficients to form the watermarked image.

**ORIGINAL INPUT IMAGE**

**DWT**

**WATAERMARKED IMAGE**

**ALPHA BLENDING EMBEDDING TECHNIQUE**

**IDWT**

**WATERMARK (LOGO)**

**DWT**

**Fig. 3.3 Level1 DFD for Watermark Embedding**

**WATERMARK EXTRACTION**

1. In this process we propose an approach of decrypting/extracting the watermark (logo) from the watermarked image/encrypted image.
2. First the watermarked image (combined image) is taken as the input and DWT (Discrete Wavelet Transform) is applied to the image which decomposes image into low frequency and high frequency components.
3. Second the original image is taken as the input and DWT (Discrete Wavelet Transform) is applied to the image which decomposes image into low frequency and high frequency components.
4. The output of the DWT is LL, HL, LH and HH band.
5. LL-Approximation Image, HL-Horizontal Image, LH-Vertical Image and HH-Diagonal Image.
6. The LL part of the watermark is extracted from the LL part of the watermarked image using alpha blending extraction technique.
7. Then IDWT is applied to combine alpha blended image with the other sub-bands and high frequency coefficients to recover the watermark image (logo).

**WATERMARKED IMAGE**

**DWT**

**IDWT**

**ALPHA BLENDING EXTRACTION TECHNIQUE**

**WATERMARK**

**INPUT IMAGE**

**DWT**

**Fig. 3.4 Level1 DFD for Watermark Extraction**

**PERFORMANCE EVALUATION**

1. Experiments were made by comparing original input images and logo images.
2. Finally obtained result (extracted logo) is compared with the actual watermark logo (before embedding).
3. The performance factor for the proposed algorithm is evaluated by PSNR, MSE and NCC values.
4. PSNR-Peak Signal to Noise Ratio.
5. MSE-Mean Squared Error
6. NCC-Normalized Cross Correlation

**PSNR**

**WATERMARK (LOGO) BEFORE EMBEDDING**

**MSE**

**PERFORMANCE EVALUATION**

**WATERMARK (LOGO) AFTER EXTRACTION**

**Fig. 3.5: Level 1 DFD for Classification**

**CHAPTER 6**

**SOFTWARE SPECIFICATION**

**6.1 GENERAL**

This paper proposes a novel nonrigid inter-subject multichannel image registration method which combines information from different modalities/channels to produce a unified joint registration. Multichannel images are created using co-registered multimodality images of the same subject to utilize information across modalities comprehensively. Contrary to the existing methods which combine the information at the image/intensity level, the proposed method uses feature-level information fusion method to spatio-adaptively combine the complementary information from different modalities that characterize different tissue types, through Gabor wavelets transformation and Independent Component Analysis (ICA), to produce a robust inter-subject registration.

**6.2 SOFTWARE REQUIREMENTS**

* MATLAB 8.3 Version R2014a

**MATLAB**

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Typical uses include:

* Math and computation
* Algorithm development
* Modeling, simulation, and prototyping
* Data analysis, exploration, and visualization
* Scientific and engineering graphics.
* Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN

**6.3 INTRODUCTION**

**MATLAB** (**mat**rix **lab**oratory) is a [numerical computing](http://en.wikipedia.org/wiki/Numerical_analysis) environment and [fourth-generation programming language](http://en.wikipedia.org/wiki/Fourth-generation_programming_language). Developed by [Math Works](http://en.wikipedia.org/wiki/MathWorks), MATLAB allows [matrix](http://en.wikipedia.org/wiki/Matrix_(mathematics)) manipulations, plotting of [functions](http://en.wikipedia.org/wiki/Function_(mathematics)) and data, implementation of [algorithms](http://en.wikipedia.org/wiki/Algorithm), creation of [user interfaces](http://en.wikipedia.org/wiki/User_interface), and interfacing with programs written in other languages, including [C](http://en.wikipedia.org/wiki/C_(programming_language)), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), and [Fortran](http://en.wikipedia.org/wiki/Fortran).

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the [MuPAD](http://en.wikipedia.org/wiki/MuPAD) [symbolic engine](http://en.wikipedia.org/wiki/Computer_algebra_system), allowing access to [symbolic computing](http://en.wikipedia.org/wiki/Symbolic_computing) capabilities. An additional package, [Simulink](http://en.wikipedia.org/wiki/Simulink), adds graphical multi-domain simulation and [Model-Based Design](http://en.wikipedia.org/wiki/Model_based_design) for [dynamic](http://en.wikipedia.org/wiki/Dynamical_system) and [embedded systems](http://en.wikipedia.org/wiki/Embedded_systems).

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of [engineering](http://en.wikipedia.org/wiki/Engineering), [science](http://en.wikipedia.org/wiki/Science), and [economics](http://en.wikipedia.org/wiki/Economics). MATLAB is widely used in academic and research institutions as well as industrial enterprises.

MATLAB was first adopted by researchers and practitioners in [control engineering](http://en.wikipedia.org/wiki/Control_engineering), Little's specialty, but quickly spread to many other domains. It is now also used in education, in particular the teaching of [linear algebra](http://en.wikipedia.org/wiki/Linear_algebra) and [numerical analysis](http://en.wikipedia.org/wiki/Numerical_analysis), and is popular amongst scientists involved in [image processing](http://en.wikipedia.org/wiki/Image_processing). The MATLAB application is built around the MATLAB language. The simplest way to execute MATLAB code is to type it in the Command Window, which is one of the elements of the MATLAB Desktop. When code is entered in the Command Window, MATLAB can be used as an interactive mathematical [shell](http://en.wikipedia.org/wiki/Shell_(computing)). Sequences of commands can be saved in a text file, typically using the MATLAB Editor, as a [script](http://en.wikipedia.org/wiki/Shell_script) or encapsulated into a [function](http://en.wikipedia.org/wiki/Functional_programming), extending the commands available.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

**6.4 FEATURES of matlab**

* High-level language for technical computing.
* Development environment for managing code, files, and data.
* Interactive tools for iterative exploration, design, and problem solving.
* Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration.
* 2-D and 3-D graphics functions for visualizing data.
* Tools for building custom graphical user interfaces.
* Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java™, COM, and Microsoft Excel.

MATLAB is used in vast area, including signal and image processing, communications, control design, [test and measurement](http://www.mathworks.in/applications/t_m), financial modeling and analysis, and computational. Add-on toolboxes (collections of special-purpose MATLAB functions) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB can be used on personal computers and powerful server systems, including the [Cheaha](http://docs.uabgrid.uab.edu/wiki/Cheaha) compute cluster. With the addition of the Parallel Computing Toolbox, the language can be extended with parallel implementations for common computational functions, including for-loop unrolling. Additionally this toolbox supports offloading computationally intensive workloads to [Cheaha](http://docs.uabgrid.uab.edu/wiki/Cheaha) the campus compute cluster.MATLAB is one of a few languages in which each variable is a matrix (broadly construed) and "knows" how big it is. Moreover, the fundamental operators (e.g. addition, multiplication) are programmed to deal with matrices when required. And the MATLAB environment handles much of the bothersome housekeeping that makes all this possible. Since so many of the procedures required for Macro-Investment Analysis involves matrices, MATLAB proves to be an extremely efficient language for both communication and implementation.

**6.4.1 INTERFACING WITH OTHER LANGUAGES**

MATLAB can call functions and subroutines written in the [C programming language](http://en.wikipedia.org/wiki/C_(programming_language)) or [FORTRAN](http://en.wikipedia.org/wiki/Fortran). A wrapper function is created allowing MATLAB data types to be passed and returned. The dynamically loadable object files created by compiling such functions are termed "[MEX-files](http://en.wikipedia.org/wiki/MEX_file)" (for **M**ATLAB **ex**ecutable).

Libraries written in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), [ActiveX](http://en.wikipedia.org/wiki/ActiveX) or [.NET](http://en.wikipedia.org/wiki/.NET_Framework) can be directly called from MATLAB and many MATLAB libraries (for example [XML](http://en.wikipedia.org/wiki/XML) or [SQL](http://en.wikipedia.org/wiki/SQL) support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with MATLAB extension, which is sold separately by Math Works, or using an undocumented mechanism called JMI (Java-to-Mat lab Interface), which should not be confused with the unrelated Java that is also called JMI.

As alternatives to the [MuPAD](http://en.wikipedia.org/wiki/MuPAD) based Symbolic Math Toolbox available from Math Works, MATLAB can be connected to [Maple](http://en.wikipedia.org/wiki/Maple_(software)) or [Mathematica](http://en.wikipedia.org/wiki/Mathematica).

Libraries also exist to import and export [MathML](http://en.wikipedia.org/wiki/MathML).

* **Development Environment**
* Startup Accelerator for faster MATLAB startup on Windows, especially on Windows XP, and for network installations.
* [Spreadsheet Import Tool](http://www.mathworks.in/videos/matlab/new-spreadsheet-import-tool-in-r2011b.html?type=shadow) that provides more options for selecting and loading mixed textual and numeric data.
* Readability and navigation improvements to warning and error messages in the MATLAB command window.
* [Automatic variable and function renaming](http://www.mathworks.in/videos/matlab/new-automatic-variable-and-function-renaming-in-r2011b.html?type=shadow) in the MATLAB Editor.
* **Developing Algorithms and Applications**

MATLAB provides a high-level language and development tools that let you quickly develop and analyze your algorithms and applications.

* **The MATLAB Language**

The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution. With the MATLAB language, you can program and develop algorithms faster than with traditional languages because you do not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for ‘for’ loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code.

At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, [object-oriented programming](http://www.mathworks.in/products/matlab/object_oriented_programming.html) (OOP), and debugging features.

MATLAB lets you execute commands or groups of commands one at a time, without compiling and linking, enabling you to quickly iterate to the optimal solution. For fast execution of heavy matrix and vector computations, MATLAB uses processor-optimized libraries. For general-purpose scalar computations, MATLAB generates machine-code instructions using its JIT (Just-In-Time) compilation technology.

This technology, which is available on most platforms, provides execution speeds that rival those of traditional programming languages.

* **Development Tools**

MATLAB includes development tools that help you implement your algorithm efficiently. These include the following:

**MATLAB Editor**

Provides standard editing and debugging features, such as setting breakpoints and single stepping

**Code Analyzer**

Checks your code for problems and recommends modifications to maximize performance and maintainability

**MATLAB Profiler**

Records the time spent executing each line of code

**Directory Reports**

Scan all the files in a directory and report on code efficiency, file differences, file dependencies, and code coverage

**Designing Graphical User Interfaces**

By using the interactive tool GUIDE (Graphical User Interface Development Environment) to layout, design, and edit user interfaces. GUIDE lets you include list boxes, pull-down menus, push buttons, radio buttons, and sliders, as well as MATLAB plots and Microsoft ActiveX® controls. Alternatively, you can create [GUIs](http://www.mathworks.in/discovery/matlab-gui.html) programmatically using MATLAB functions.

**6.5 The MATLAB System**

The MATLAB system consists of five main parts:

* **Development Environment**.

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, and browsers for viewing help, the workspace, files, and the search path.

* **The MATLAB Mathematical Function Library**.

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

* **The MATLAB Language**.

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

* **Handle Graphics**.

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

* **The MATLAB Application Program Interface (API).**

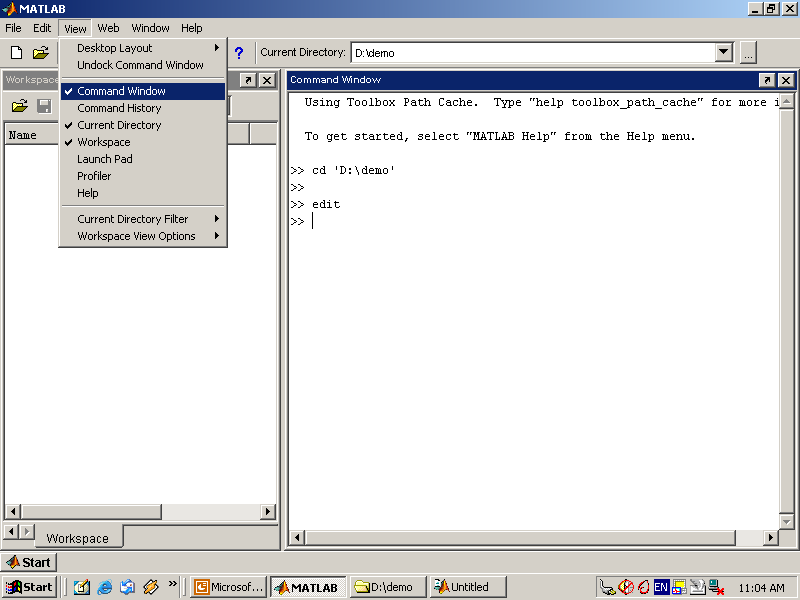
This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

**6.5.1 DESKTOP TOOLS**

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

* Current Directory Browser
* Workspace Browser
* Array Editor
* Editor/Debugger
* Command Window
* Command History
* Launch Pad
* Help Browser

**Command Window**



Use the Command Window to enter variables and run functions and M-files.

* **Command History**

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

* **Running External Programs**

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

* **Launch Pad**

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

* **Help Browser**

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

* **Help Navigator**

Use to Help Navigator to find information. It includes:

* **Product filter**

Set the filter to show documentation only for the products you specify.

* **Contents tab**

View the titles and tables of contents of documentation for your products.

* **Index tab**

Find specific index entries (selected keywords) in the MathWorks documentation for your products.

* **Search tab**

Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

* **Favorites tab**

View a list of documents you previously designated as favorites.

* **Display Pane**

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

* **Browse to other pages**

Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

* **Bookmark pages**

Click the Add to Favorites button in the toolbar.

* **Print pages**

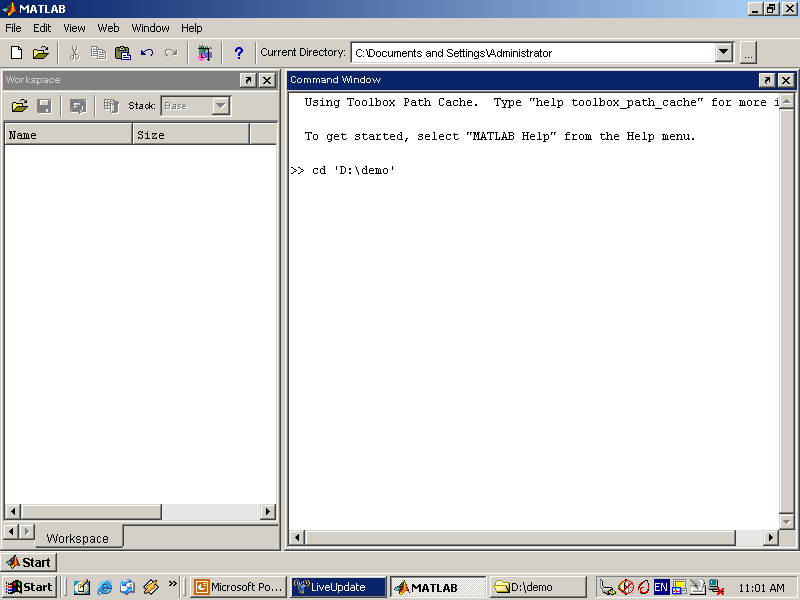
Click the print button in the toolbar.

* **Find a term in the page**

Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

**Current Directory Browser**

****

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

**Search Path**

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

* **Workspace Browser**

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

* **Array Editor**

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

* **Editor/Debugger**

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic textediting, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

**6.5.2 ANALYZING AND ACCESSING DATA**

MATLAB supports the entire data analysis process, from acquiring data from external devices and databases, through preprocessing, visualization, and numerical analysis, to producing presentation-quality output.

* **Data Analysis**

MATLAB provides interactive tools and command-line functions for data analysis operations, including:

* Interpolating and decimating
* Extracting sections of data, scaling, and averaging
* Thresholding and smoothing
* Correlation, Fourier analysis, and filtering
* 1-D peak, valley, and zero finding
* Basic statistics and curve fitting
* Matrix analysis

**Data Access**

MATLAB is an efficient platform for accessing data from files, other applications, databases, and external devices. You can read data from popular file formats, such as Microsoft Excel; ASCII text or binary files; image, sound, and video files; and scientific files, such as HDF and HDF5. Low-level binary file I/O functions let you work with data files in any format. Additional functions let you read data from Web pages and XML.

**Visualizing Data**

All the graphics features that are required to visualize engineering and scientific data are available in MATLAB. These include 2-D and 3-D plotting functions, 3-D volume visualization functions, tools for interactively creating plots, and the ability to export results to all popular graphics formats. You can customize plots by adding multiple axes; changing line colors and markers; adding annotation, Latex equations, and legends; and drawing shapes.

**2-D Plotting**

Visualizing vectors of data with 2-D plotting functions that create:

* Line, area, bar, and pie charts.
* Direction and velocity plots.
* Histograms.
* Polygons and surfaces.
* Scatter/bubble plots.
* Animations.

**6.5.3 PERFORMING NUMERIC COMPUTATION**

MATLAB contains mathematical, statistical, and engineering functions to support all common engineering and science operations. These functions, developed by experts in mathematics, are the foundation of the MATLAB language. The core math functions use the LAPACK and BLAS linear algebra subroutine libraries and the FFTW Discrete Fourier Transform library. Because these processor-dependent libraries are optimized to the different platforms that MATLAB supports, they execute faster than the equivalent C or C++ code.

MATLAB provides the following types of functions for performing mathematical operations and analyzing data:

* Matrix manipulation and linear algebra.
* Polynomials and interpolation.
* Fourier analysis and filtering.
* Data analysis and statistics.
* Optimization and numerical integration.
* Ordinary differential equations (ODEs).
* Partial differential equations (PDEs).
* Sparse matrix operations.

MATLAB can perform arithmetic on a wide range of data types, including doubles, singles, and integers.

**CHAPTER 7**

**ALGORITHM USED**

## PROPOSED ALGORITHM

**RGB TO CHANNEL CONVERSION**

Once the images are resized the image is splited to three individual colors as Red, Green and Blue**.**

**Algorithm:**

Step 1: Read an original image by specifying the file name and image format such jpg etc

Img = imread (‘filename.png’); % Read image

Step 2: specify individual color name as shown

Red = img (:,:, 1); % for red channel%

Green = img (:,:, 2); % for Green channel%

Blue = img (: , :, 3); % for Blue channel%

Step 3: Display the images using imshow function.

**Discrete Wavelet Transform**

The discrete wavelet transform (DWT) is the basic and simplest transform among numerous multiscale transform and other type of wavelet based fusion schemes are usually similar to the DWT fusion scheme.

DWT is the multiresolution description of an image the decoding can be processed sequentially from a low resolution to the higher resolution. The DWT splits the image into high and low frequency parts. The high frequency part contains information about the edge components, while the low frequency part is split again into high and low frequency parts. The high frequency components are usually used for watermarking since the human eye is less sensitive to changes in edges.

In two dimensional applications, for each level of decomposition, we first perform the DWT in the vertical direction, followed by the DWT in the horizontal direction. After the first level of decomposition, there are 4 sub-bands: LL1, LH1, HL1, and HH1.

Commonly used frequency-domain transforms include the Discrete Wavelet Transform (DWT), the Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT). However, DWT has been used in digital image watermarking more frequently due to its excellent spatial localization and multi-resolution characteristics, which are similar to the theoretical models of the human visual system. Further performance improvements in DWT-based digital image watermarking algorithms could be obtained by increasing the level of DWT.

**ALPHA BLENDING EMBEDDING TECHNIQUE**

Alpha Blending can be accomplished in image processing by blending each pixel from the first source image with the corresponding pixel in the second source image. According to the formula of the alpha blending the watermarked image is given by

WMI=K\*LL4 +Q \* WM2

Where,

WMI = low frequency component of watermarked image

LL4 = low frequency component of the original image obtained by 4-level DWT

WM4 = low frequency component of Watermark image

k, q = Scaling factors for the original image and watermark respectively.

**Inverse wavelet transform**

By combining the low frequency sub band (LL) with the high frequency subbands by applying IDWT, the output image will contain sharper edges than the watermarked image obtained. This is due to the fact that, the interpolation of isolated high frequency components in high frequency subbands and using the corrections obtained by adding high frequency subbands of DWT of the input image, will preserve more high frequency components after the interpolation than interpolating input image directly.

**ALPHA BLENDING EXTRACTION TECHNIQUE**

According to the formula of the alpha blending the recovered image is given by

RW = (WMI - k \* LL4)

Where

RW= Low frequency approximation of Recovered watermark,

LL4= Low frequency approximation of the original image

WMI= Low frequency approximation of watermarked image.

**PERFORMANCE EVALUATION**

Performance measures are employed towards verifying the consistency as well as effectiveness regarding the method. Computation about the performance metrics is derived from three factors that contain PSNR, MSE and NCC are as follows:-

**Mean Square Error (MSE)**

Mean Square Error (MSE) is used to analyze themeasures the average of the squares of the errors or deviations that is, the difference between the estimator and what is estimated.

**Peak Signal to Noise Ratio (PSNR)**

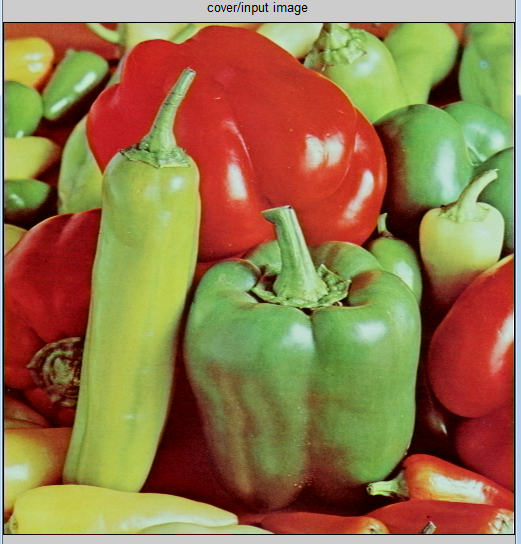
Peak Signal to Noise Ratio (PSNR) is generally used to analyze quality of image and video files in dB (decibels). PSNR calculation of two images, one original and an altered image, describes how far two images are equal.

**CHAPTER 8**

**IMPLEMENTATION RESULTS**

**8.1 SCREENSHOTS**

The figures showed below gives the step by step operational output for our proposed system.



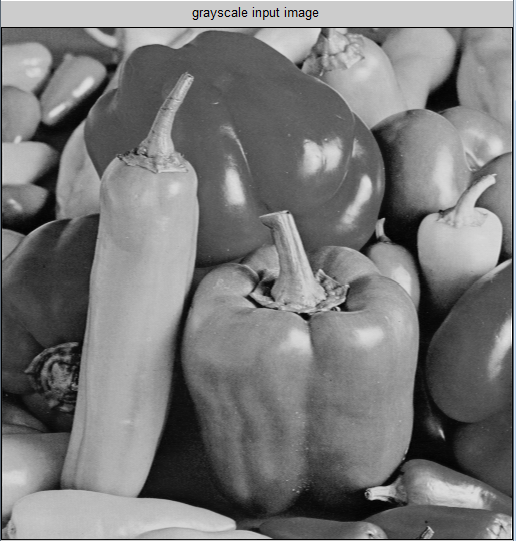
**Fig 8.1: Original Input Image**

The original image is resized to a standardized size since size of different images varies as shown in fig 6.2.



**Fig 6.2: Resized Image**

After Resizing image will be transformed as of RGB towards grayscale image evaluated by way of Channel separation as shown in fig 6.3.



**Fig 6.3: Channel Separation (grayscale image)**

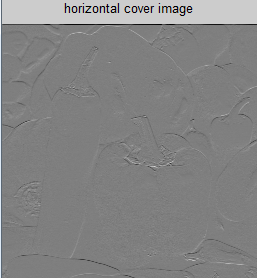
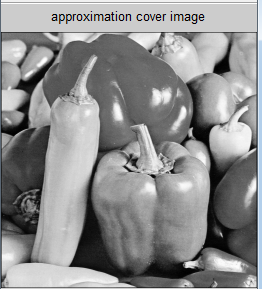


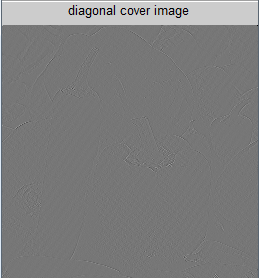
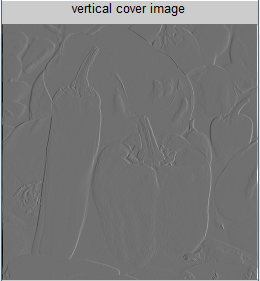
**Fig 6.4: Resized Logo image (secret image)**

****

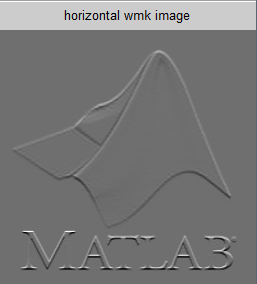
**Fig 6.5: grayscale logo image**

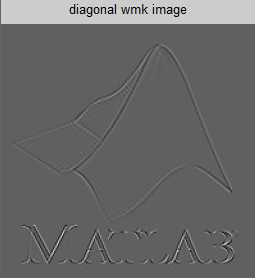
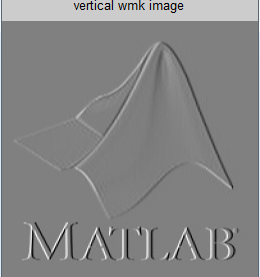
**8.2 Results for watermark embedding**



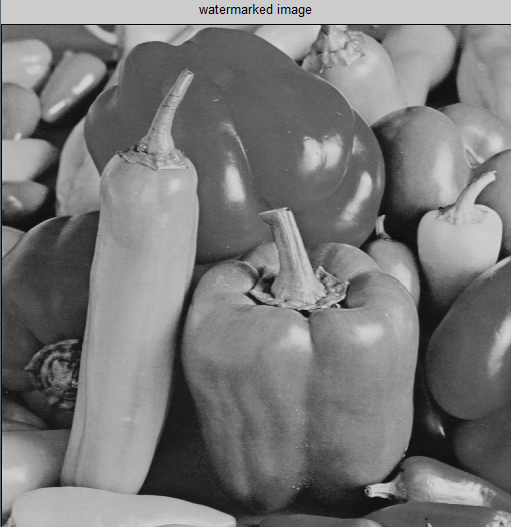


**Fig 6.5: DWT output of Input image**



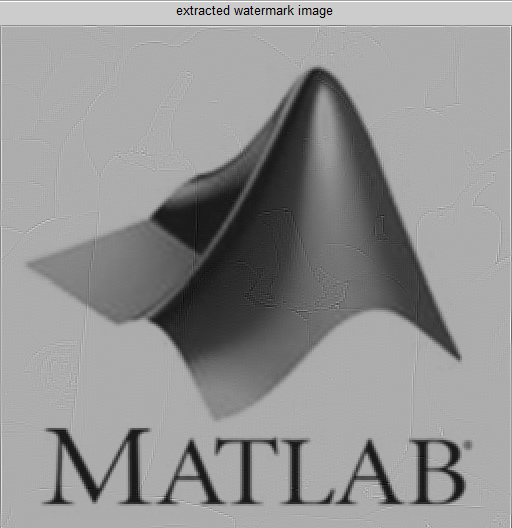


**Fig 6.6 DWT output of watermark image (logo)**



**Fig 6.7 Watermarked image**

**8.3 Results for watermark extraction**

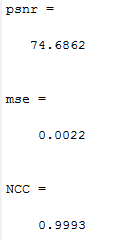
****

**Fig 8.7 Extracted logo image**

**6.5 Performance Evaluation**

The analysis is done for 4 to 6 images which are present default in MATLAB. Performance measure such as PSNR, MSE and NCC is determined. Results were satisfactory compared to existing method.

The performance parameters of the proposed system are evaluated as shown in table 6.1



**Table: 8.1 PSNR and MSE values for Proposed System**

|  |  |  |
| --- | --- | --- |
|  | PSNR | MSE |
| Proposed method | 74.6862 | 0.0022 |

**CHAPTER 9**

**CONCLUSION AND REFERENCES**

**9.1 CONCLUSION AND FUTURE WORK**

This work proposed an approach of combining the watermark (logo) into the input image using DWT and Alpha blending Embedding algorithm. The watermarked image/encrypted image is formed. Then we need to decrypt/extract the logo image from the watermarked image using DWT, Alpha blending Extraction and IDWT algorithm. We demonstrate that the watermarks generated with the proposed algorithm are invisible and can be visible sometimes and the quality of adulterated image and the recovered image are improved. The proposed method is compared with the existing watermarking methods by using statistical parameters such as peak-signal-to-noise-ratio (PSNR), Mean Square Error (MSE) and Normalized Cross Correlation (NCC).

In the future work experiments with more images were carried out and tested by considering various phenomenon’s such as exclusion of low resolution images and including the calculation of some more statistical parameters.

**9.2 REFERENCES**

[1] L. Yi-bo, X. Hong, and Z. Sen-yue, “The wrinkle generation method for facial reconstruction based on extraction of partition wrinkle line features and fractal interpolation,” in Proc. 4th Int. Conf. Image Graph*.*, Aug. 22–24, 2007, pp. 933–937.

[2] Y. Renner, J. Wei, and C. Ken, “Down sample-based multiple description coding and post-processing of decoding,” in Proc. 27th Chinese Control Conf., Jul. 16–18, 2008, pp. 253–256.

[3] H. Demirel, G. Anbarjafari, and S. Izadpanahi, “Improved motion based localized super resolution technique using discrete wavelet transform for low resolution video enhancement,” in Proc. 17th Eur. Signal Process. Conf., Glasgow, Scotland, Aug. 2009, pp. 1097–1101.

[4] Y. Piao, I. Shin, and H. W. Park, “Image resolution enhancement using inter-subband correlation in wavelet domain,” in Proc. Int. Conf. Image Process., 2007, vol. 1, pp. I-445–448.

[5] H. Demirel and G. Anbarjafari, “Satellite image resolution enhancement using complex wavelet transform,” IEEE Geoscience and Remote Sensing Letter, vol. 7, no. 1, pp. 123–126, Jan. 2010.

[6] C. B. Atkins, C. A. Bouman, and J. P. Allebach, “Optimal image scaling using pixel classification,” in Proc. Int. Conf. Image Process*.*, Oct. 7–10, 2001, vol. 3, pp. 864–867.

[7] W. K. Carey, D. B. Chuang, and S. S. Hemami, “Regularity-preserving image interpolation,” IEEE Trans. Image Process., vol. 8, no. 9, pp. 1295–1297, Sep. 1999.

[8] S. Mallat*,* A Wavelet Tour of Signal Processing, 2nd ed. New York: Academic, 1999.

[9] J. E. Fowler, “The redundant discrete wavelet transform and additive noise,”Mississippi State ERC, Mississippi State University, Tech. Rep. MSSU-COE-ERC-04-04, Mar. 2004.

[10] X. Li and M. T. Orchard, “New edge-directed interpolation,” IEEE Trans. Image Process., vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[11] K. Kinebuchi, D. D. Muresan, and R. G. Baraniuk, “Waveletbased statistical signal processing using hidden Markov models,” in Proc. Int. Conf. Acoustic., Speech, Signal Process., 2001, vol. 3, pp. 7–11.

[12] S. Zhao, H. Han, and S. Peng, “Wavelet domain HMT-based image super resolution,” in Proc. IEEE Int. Conf. Image Process., Sep. 2003, vol. 2, pp. 933–936.

[13] A. Temizel and T. Vlachos, “Wavelet domain image resolution enhancement using cycle-spinning,” Electron. Lett., vol. 41, no. 3, pp. 119–121, Feb. 3, 2005.

[14] A. Temizel and T. Vlachos, “Image resolution up scaling in the wavelet domain using directional cycle spinning,” J. Electron. Image. vol. 14, no. 4, 2005.

[15] G. Anbarjafari and H. Demirel, “Image super resolution based on interpolation of wavelet domain high frequency subbands and the spatial domain input image,” *ETRI J.*, vol. 32, no. 3, pp. 390–394, Jun. 2010.

[16] A. Temizel, “Image resolution enhancement using wavelet domain hidden Markov tree and coefficient sign estimation,” in Proc. Int. Conf. Image Process*.*, 2007, vol. 5, pp. V-381–384.

[17] J.D. van Ouwerkerk. Image super-resolution survey. Image and Vision Computing, 24:1039–1052, 2006.

[18] E. Vansteenkiste, D.Van der Weken, W. Philips, and E.E. Kerre. Evaluation of the perceptual performance of fuzzy image quality measures. In *proc. KES,* 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems, pages 623–630, 2006.

[19] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13:600–612, 2004.