**A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC**

**ABSTRACT**

Over the past few decades, the art of secretly embedding and communicating digital data has gained enormous attention because of the technological development in both digital contents and communication. The imperceptibility, hiding capacity, and robustness against attacks are three main requirements that any video steganography method should take into consideration. In this article, a robust and secure video steganographic algorithm in Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) domains based on the Multiple Object Tracking (MOT) algorithm and Error Correcting Codes (ECC) is proposed. The secret message is preprocessed by applying both Hamming and Bose, Chaudhuri, and Hocquenghem (BCH) codes for encoding the secret data. First, motion-based MOT algorithm is implemented on host videos to distinguish the regions of interest in the moving objects. Then, the data hiding process is performed by concealing the secret message into the DWT and DCT coefficients of all motion regions in the video depending on foreground masks. Our experimental results illustrate that the suggested algorithm not only improves the embedding capacity and imperceptibility but also it enhances its security and robustness by encoding the secret message and withstanding against various attacks.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **TITLE** | **PPAGE**  **NO.** |
|  | **ABSTRACT**  **LIST OF FIGURES**  **LIST OF ABBREVIATIONS** |  |
| **1.** | **CHAPTER 1 : INTRODUCTION**  1.1 GENERAL  1.1.1THE IMAGE PROCESSING SYSTEM  1.1.2 IMAGE PROCESSING FUNDAMENTAL  1.1.3 Image types  1.1.4 APPLICATIONS of image processing  1.2 OBJECTIVE  1.3EXIXTING SYSTEM  1.3.1 EXIXTING SYSTEM DISADVANTAGES  1.3.2 LITERATURE SURVEY  1.4PROPOSED SYSTEM  1.4.1 ADVANTAGES |  |
| **2.** | **CHAPTER 2 : PROJECT DESCRIPTION**  2.1 INTRODUCTION  2.2 MOTION-BASED MOT STAGE  2.3 DATA EMBEDDING STAGE  2.4 DATA EXTRACTION STAGE  2.5 APPLICATIONS |  |
| **3.** | **CHAPTER 3 : SOFTWARE SPECIFICATION**  3.1 SOFTWARE REQUIREMENTS  3.2 INTRODUCTION  3.3 features of matlab  3.4.1 INTERFACING WITH OTHER LANGUAGES  3.4 The MATLAB System  3.4.1 DESKTOP TOOLS  3.4.2 ANALYZING AND ACCESSING DATA  3.4.3 PERFORMING NUMERIC COMPUTATION |  |
| **4.** | **CHAPTER4 : IMPLEMENTATION**  4.1 GENERAL  4.2 CODE IMPLEMENTATION  4.3SCREENSHOTS |  |
| **5.** | **CHAPTER 5 : CONCLUSION &REFERENCES**  5.1 CONCLUSION  5.2 REFERENCES |  |

‘

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO** |
| 1.1 | A BLOCK DAIGRAM FOR IMAGE PROCESSING SYSTEM |  | |
| 1.2 | BLOCK DIAGRAM OF FUNDAMENTAL SEQUENCE INVOLVED IN AN IMAGE PROCESSING SYSTEM |  | |
| 1.3 | IMAGE PROCESSING TECHNIQUES |  | |
| 1.4 | BLOCK DIAGRAM OF THE PROPOSED SYSTEM |  | |

**CHAPTER I**

**INTRODUCTION**

* 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.

**1.1.1 THE 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

1. Microdensitometer
2. Flying spot scanner
3. Image dissector
4. Videocon camera
5. 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 n

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 thepreprocessing 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

FIG1.3: IMAGE PROCESSING TECHNIQUES

**MAGE 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.

**APPLICATIONS OF DIGITAL IMAGE PROCESSING:**

Digital image processing has a broad spectrum of applications, such as remote sensing via satellites and other spacecrafts, image transmission and storage for business applications, medical processing, radar, sonar and acoustic image processing, robotics and automated inspection of industrial parts.

**MEDICAL APPLICATIONS:**

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.

**SATELLITE IMAGING:**

Images acquired by satellites are useful in tracking of earth resources; geographical mapping; prediction of agricultural crops, urban growth and weather; flood and fire control; and many other environmental applications. Space image applications include recognition and analysis of objects contained in image obtained from deep space-probe missions.

**COMMUNICATION:**

Image transmission and storage applications occur in broadcast television, teleconferencing, and transmission of facsimile images for office automation, communication of computer networks, closed-circuit television based security monitoring systems and in military communications**.**

**RADAR IMAGING SYSTEMS:**

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

**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.

**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.

* 1. **OBJECTIVE:**

Embedding efficiency, hiding capacity, and robustness are the three major requirements incorporated in any successful steganographic method.

* 1. **EXISTING SYSTEM**

DCT/DST-based data hiding algorithm for HEVC intra-coded frames where the block DCT and DST coefficient characteristics are investigated to locate the transformed coefficients that can be perturbed without propagating errors to neighboring blocks.

**1.3.1 EXISITING SYSTEM DISADVANTAGES:**

* By utilizing the preprocessing stages to include the manipulation on both secret messages and cover videos earlier to the embedding stage in order to enhance the security and robustness of the steganographic method.
* Using a portion of each video frame as regions of interest for the concealing process, the imperceptibility of stego videos will improve. Accordingly, we track multiple moving objects in video. Since it is very challenging for hackers to recognize the position of the hidden message in video frames because the hidden message is only concealed into moving objects, which changes over time from one frame to another, it is necessary to preserve the security and robustness of embedded data.
  + 1. **LITERATURE SURVEY:**

1. **“A Tube-and-Droplet-based Approach for Representing and Analyzing Motion Trajectories”, by Weiyao Lin, Yang Zhou, Hongteng Xu.**

Trajectory analysis is essential in many applications. In this paper, we address the problem of representing motion trajectories in a highly informative way, and consequently utilize it for analyzing trajectories. Our approach first leverages the complete information from given trajectories to construct a thermal transfer field which provides a contextrich way to describe the global motion pattern in a scene. Then, a 3D tube is derived which depicts an input trajectory by integrating its surrounding motion patterns contained in the thermal transfer field. The 3Dtube effectively: 1) maintains the movement information of a trajectory,2 )embeds the complete contextual motion pattern around a trajectory, 3) visualizes information about a trajectory in a clear and unified way. We further introduce a droplet-based process. It derives a droplet vector from a 3D tube, so as to characterize the high-dimensional 3D tubeinformation in a simple but effective way. Finally, we apply our tubeand- droplet representation to trajectory analysis applications includingtrajectory clustering, trajectory classification & abnormality detection, and 3D action recognition. Experimental comparisons with state-of-theart algorithms demonstrate the effectiveness of our approach.

1. **“Hierarchical Convolutional Features for Visual Tracking”, by Chao Ma, and Jia-Bin Huang.**

Visual object tracking is challenging as target objects often undergo significant appearance changes caused by deformation, abrupt motion, background clutter and occlusion. In this paper, we exploit features extracted from deep convolutional neural networks trained on object recognition datasets to improve tracking accuracy and robustness. The outputs of the last convolutional layers encode the semantic information of targets and such representations are robust to significant appearance variations. However, their spatial resolution is too coarse to precisely localize targets. In contrast, earlier convolutional layers provide more precise localization but are less invariant to appearance changes. We interpret the hierarchies of convolutional layers as a nonlinear counterpart of an image pyramid representation and exploit these multiple levels of abstraction for visual tracking. Specifically, we adaptively learn correlation filters on each convolutional layer to encode the target appearance. We hierarchically infer the maximum response of each layer to locate targets. Extensive experimental results on a largescale benchmark dataset show that the proposed algorithm performs favorably against state-of-the-art methods.

1. **“A New Video Steganography Algorithm Based on the Multiple Object Tracking and Hamming Codes”, by Ramadhan J. and Mstafa.**

In the modern world, video steganography has become a popular option for secret data communication. The performance of any steganography algorithm is based on the embedding efficiency, embedding payload, and robustness against attackers. In this paper, we propose a new video steganography algorithm based on the multiple object tracking algorithm and Hamming codes. The proposed algorithm includes four different stages. First, the secret message is preprocessed, and Hamming codes (*n*, *k*) are applied in order to produce an encoded message. Second, a motion-based multiple object tracking algorithm is applied on cover videos in order to identify the regions of interest of the moving objects. Third, the process of embedding 3 and 6 bits of the encoded message into the 1 LSB and 2 LSBs of RGB pixel components is performed for all motion regions in the video using the foreground mask. Fourth, the process of extracting the secret message from the 1 LSB and 2 LSBs for each RGB component of all moving regions is accomplished. Experimental results of the proposed video steganography algorithm have demonstrated a high embedding efficiency and a high embedding payload.

1. **“A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image”, by Khan Muhammad1 & Muhammad Sajjad.**

Image Steganography is a thriving research area of information security where secret data is embedded in images to hide its existence while getting the minimum possible statistical detectability. This paper proposes a novel magic least significant bit substitution method (M-LSBSM) for RGB images. The proposed method is based on the achromatic component (I-plane) of the hue-saturation-intensity (HSI) color model and multi-level encryption (MLE) in the spatial domain. The input image is transposed and converted into an HSI color space. The I-plane is divided into four sub-images of equal size, rotating each sub-image with a different angle using a secret key. The secret information is divided into four blocks, which are then encrypted using an MLE algorithm (MLEA). Each sub-block of the message is embedded into one of the rotated subimages based on a specific pattern usingmagic LSB substitution. Experimental results validate that the proposed method not only enhances the visual quality of stego images but also provides good imperceptibility and multiple security levels as compared to several existing prominent methods.

1. **“Dual-Level Security based Cyclic18 Steganographic Method and its Application for Secure Transmission of Keyframes during Wireless Capsule Endoscopy”, by Khan Muhammad and Muhammad Sajjad.**

In this paper, the problem of secure transmission of sensitive contents over the public network Internet is addressed by proposing a novel data hiding method in encrypted images with dual-level security. The secret information is divided into three blocks using

a specific pattern, followed by an encryption mechanism based on the three-level encryption algorithm (TLEA). The input image is scrambled using a secret key, and the encrypted sub-message blocks are then embedded in the scrambled image by cyclic18 least significant bit (LSB) substitution method, utilizing LSBs and intermediate LSB planes. Furthermore, the cover image and its planes are rotated at different angles using a secret key prior to embedding, deceiving the attacker during data extraction. The usage of message blocks division, TLEA, image scrambling, and the cyclic18 LSB method

results in an advanced security system, maintaining the visual transparency of resultant images and increasing the security of embedded data. In addition, employing various secret keys for image scrambling, data encryption, and data hiding using the cyclic18 LSB method makes the data recovery comparatively more challenging for attackers. Experimental results not only validate the effectiveness of the proposed framework in terms of visual quality and security compared to other state-of-the-art methods, but also

suggest its feasibility for secure transmission of diagnosticallyimportant keyframes to healthcare centers and gastroenterologists during wireless capsule endoscopy.

* 1. **PROPOSED SYSTEM:**

• Design of an optimal wideband bandpass filter (WBBF) for enhancement of bright lesions.

• Differential Evolution (DE) based contrast enhancement to automatically set the gain and the bandwidth of the WBBF to make the system adaptive for different types of images based on their characteristics.

• Maximization of mutual information (MI) of the maximum matched filter response (MFR) and the maximum Laplacian of Gaussian response (LoGR) in 2 dimensional (2D) feature space using DE which, to the best of ou knowledge, is not explored earlier in lesion detection.

• Significantly improved results over the existing methods when evaluated on a large set of images on different openaccess online databases.

**BLOCK DIAGRAM**

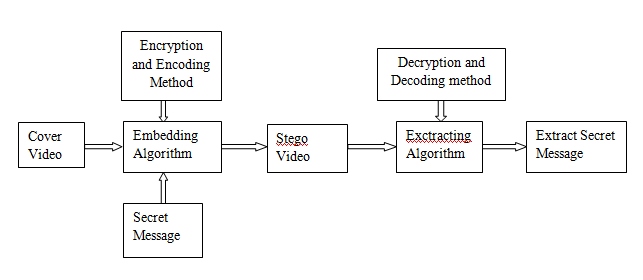
****

Fig: Block Diagram of the proposed system

**1.4.1 ADVANTAGES**

* Applying encryption methods and ECC such as Hamming codes and BCH codes to encode the hidden message earlier to the concealing stage will produce a secure and robust steganographic algorithm.
* Transforming video frames into frequency domain such as DWT and DCT transformations will improve the robustness of the steganographic method against attacks, hence preserving imperceptibility of stego videos.

**MODULES:**

* **1. MOTION-BASED MOT STAGE**
* **2. DATA EMBEDDING STAGE**
* **3. DATA EXTRACTION STAGE**

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 INTRODUCTION**

In spite of the fact that the Internet is utilized as a medium to access desired information, it has also opened a new door for attackers to obtain precious information of other users with little effort [1]. Steganography has functioned in a complementary capacity to offer a protection mechanism that hide communication between an authorized transmitter and its recipient. Steganography is defined as the art of concealing secret information in specific carrier data, establishing covert communication channels between official parties [2, 3]. Subsequently, a stego object (steganogram) should appear the same as an original data that has a slight change of the statistical features. The primary objective of the steganography is to eliminate any suspicion to the transmission of hidden messages and provide security and anonymity for legitimate parties. The simplest way to observe the steganogram’s visual quality is to determine its accuracy, which is achieved through the Human Visual System (HVS). The HVS cannot identify slight distortions in the steganogram, thus avoiding suspiciousness [4]. However, if the size of the hidden message in proportion with the size of the carrier object is large, then the steganogram’s degradation will be visible to the human eye resulting in a failed steganographic method.

**MODULE 1:**

**2.2 MOTION-BASED MOT STAGE**

The motion-based MOT algorithm has been previously explained in Section 4. The process of identifying the moving objects in the video frames must be carried out when motion object regions are utilized as host data. This process is achieved by detecting each moving object within an individual frame, and then associating these detections throughout all of the video frames. The background subtraction method is applied to detect the moving objects based on the GMM. It also computes the differences between consecutive frames that generate the foreground mask. Then, the Kalman filter is employed to predict estimation trajectory of each moving region.

**MODULE 2:**

**2.3 DATA EMBEDDING STAGE**

In entire video frames, the host data of our proposed method is the motion objects that are considered as regions of interest. By using the motion-based MOT algorithm, the process of detecting and tracking the motion regions over all video frames are achieved. The regions of interest altered in each video frame is dependent on the number and the size of the moving objects. In every frame, 2D-DWT is implemented on RGB channels of each motion region resulting LL, LH, HL, and HH subbands. In addition, 2D-DCT is also applied on the same motion regions generating DC and AC coefficients. Thereafter, the secret messages are concealed into LL, LH, HL, and HH of DWT coefficients, and into DC and AC of DCT coefficients of each motion object separately based on its foreground mask.

**MODULE 3:**

**2.4 DATA EXTRACTION STAGE**

video is separated into a number of frames through the receiver side, and then two secret keys are obtained from the non-motion region of the first video frame. To predict trajectories of motion objects, the motion-based MOT algorithm is applied again by the receiver. Then, 2D-DWT and 2D-DCT are employed on the RGB channels of each motion object in order to create LL, LH, HL, and HH subbands, and DC and AC coefficients, respectively. Next, the extracting process of the embedded data is achieved by obtaining the secret messages from LL, LH, HL, HH, DC, and AC coefficients of each motion region over all video frames based on the same foreground masks used in the embedding stage. The extracted secret message is decoded by Hamming and BCH (7, 4), and then decrypted to obtain the original message.

**PROPOSED SYSTEM TECHNIQUE EXPLANATION**

A robust and secure video stegano graphic algorithm in Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) domains based on the Multiple Object Tracking (MOT) algorithm and Error Correcting Codes (ECC) is proposed. The secret message is preprocessed by applying both Hamming and Bose, Chaudhuri, and Hocquenghem (BCH) codes for encoding the secret data. First, motion-based MOT algorithm is implemented on host videos to distinguish the regions of interest in the moving objects. Then, the data hiding process is performed by concealing the secret message into the DWT and DCT coefficients of all motion regions in the video depending on foreground masks.

**2.6 APPLICATIONS:**

* Computer vision is one of the fastest emerging fields in computer science. The detection and tracking of moving objects within the computer vision field has recently gained significant attention.
* The modification of high frequency coefficients does not have an impact on the video quality.

**CHAPTER 3**

**SOFTWARE SPECIFICATION**

**3.1 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.

**3.2 SOFTWARE REQUIREMENTS**

* MATLAB 7.14 Version

**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

**3.3 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" \o "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.

**3.4 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" \o "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" \o "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.

**3.4.1 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" \o "MuPAD) based Symbolic Math Toolbox available from Math Works, MATLAB can be connected to [Maple](http://en.wikipedia.org/wiki/Maple_(software)) or [Mathematical](http://en.wikipedia.org/wiki/Mathematica).

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

**3.5 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).**

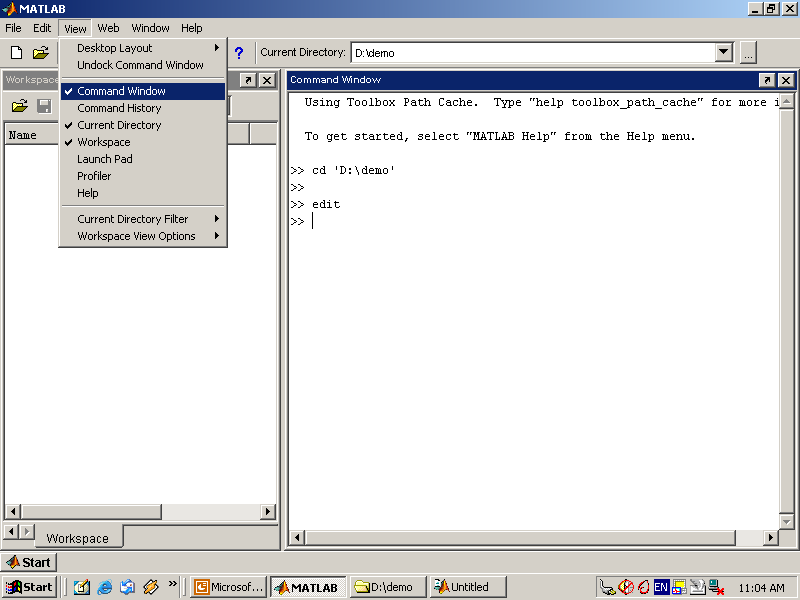
This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

**3.5.1 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,!emacsmagik.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 help browser 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**

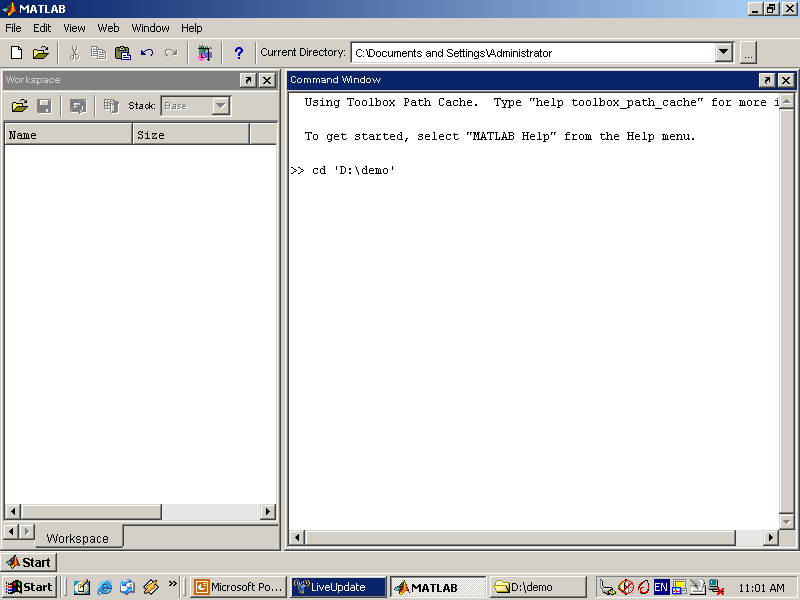
Click the print button in the toolbar.

* **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.

**3.5.2 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.

**3-D Plotting and Volume Visualization**

MATLAB provides functions for visualizing 2-D matrices, 3-D scalar, and 3-D vector data. You can use these functions to visualize and understand large, often complex, multidimensional data. Specifying plot characteristics, such as camera viewing angle, perspective, lighting effect, light source locations, and transparency.

3-D plotting functions include:

* Surface, contour, and mesh.
* Image plots.
* Cone, slice, stream, and isosurface.

**3.5.3 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 4**

**IMPLEMENTATION**

**4.1 GENERAL**

Matlab is a program that was originally designed to simplify the implementation of numerical linear algebra routines. It has since grown into something much bigger, and it is used to implement numerical algorithms for a wide range of applications. The basic language used is very similar to standard linear algebra notation, but there are a few extensions that will likely cause you some problems at first.

**4.2 CODE IMPLEMENTATION**

% % % A Robust and Secure Video Steganography Method in DWT-DCT Domains

% % % % % % % Based on Multiple Object Tracking and ECC

clc;

clear all;

close all;

warning('off','all');

%% Motion Object Detection and Region Extraction

folder = dir('View\_001\\*.jpg');

for x = 1:length(folder)

f = folder(x).name;

images{x,:} = imread(fullfile('View\_001\',f));

end

load a1.mat;

ij = 1;l = 1;k = 1;

[z z1] = size(images);

if mask(i,j) == 1

b = j;

break;

end

end

end

D = 2\*(floor(cc.Centroid(2) - b));

%%>>>>>>>>>>>>>>>>>>>>>>>>>>>SPATIAL CALIBRATION<<<<<<<<<<<<<<<<<<<<<<<<<<<

d1 = floor(D/10); %Average radius of the OD(optic disk).

d2 = floor(D/360); %Size of the smallest MA(Microa-neurysms).

d3 = floor(D/28); %Size of the largest HE(Hemorrhages).

%%>>>>>>>>>>>>>>>>>>>>>>>>>>>>PREPROCESSING<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

%% step:1--------->>>>>>illumination Equalization<<<<<<<-------------------

R1 =im(:,:,1);

G1 =im(:,:,2);

B1 =im(:,:,3);

h1 =fspecial('average',d1);

hm1 = imfilter(R1,h1);hm2 = imfilter(G1,h1);hm3 = imfilter(B1,h1);

hmm = cat(3,hm1,hm2,hm3);

avg = mean2(im);

ill\_eqa = im + (avg - (im.\*hmm));

figure,subplot(2,2,1);imshow(ill\_eqa);title('ILLUMINATION');

%%step:2---------->>>>>>>>>>>>Denoising<<<<<<<<<<<<------------------------

R2 = ill\_eqa(:,:,1);

G2 = ill\_eqa(:,:,2);

B2 = ill\_eqa(:,:,3);

h2 =fspecial('average',d2);

hm12 = imfilter(R2,h2);hm22 = imfilter(G2,h2);hm32 = imfilter(B2,h2);

dn = cat(3,hm12,hm22,hm32);

subplot(2,2,2);imshow(dn);title('DENOSING');

%%step:3---------->>>>>>Adaptive Contrast Equalization<<<<<<---------------

R3 = dn(:,:,1);

G3 = dn(:,:,2);

B3 = dn(:,:,3);

h3 = fspecial('average',d3);

hm13 = imfilter(R3,h3);hm23 = imfilter(G3,h3);hm33 = imfilter(B3,h3);

Iace = cat(3,hm13,hm23,hm33);

R31 = std2(R3);G31 = std2(G3);B31 = std2(B3);

so = cat(3,R31,G31,B31);

is = std2(so);

Ice = (dn + 1 .\*(dn .\*(1 - Iace))/is);

subplot(2,2,3);imshow(Iace);title('EQUALIZATION');

%%step:4--------->>>>>>Gamma/Color Normalization<<<<<<<<<------------------

% hgamma = vision.GammaCorrector(2.0,'Correction','Gamma');

% y = step(hgamma,Iace);

% imshow(y);

r3 = im2double(Ice(:,:,1));m1 = mean2(r3);s1 = std2(r3);f1min =( m1 - (s1))/10;f1max =3\*(m1 + (s1))/10;

g3 = im2double(Ice(:,:,2));m2 = mean2(g3);s2 = std2(g3);f2min =( m2 - (s2))/10;f2max =3\*(m2 + (s2))/10;

b3 = im2double(Ice(:,:,3));m3 = mean2(b3);s3 = std2(b3);f3min =( m3 - (s3))/10;f3max =3\*(m3 + (s3))/10;

j1 = imadjust(r3,stretchlim(r3),[f1min f1max]);

j2 = imadjust(g3,stretchlim(g3),[f2min f2max]);

j3 = imadjust(b3,stretchlim(b3),[f3min f3max]);

p = cat(3,j1,j2,j3);mask = im2bw(mask);

Ip1 = j1.\*mask;Ip2 = j2.\*mask;Ip3 = j3.\*mask;

Ip = cat(3,Ip1,Ip2,Ip3);subplot(2,2,4);,imshow(p);title('COLOR NORMALIZATION');

%%>>>>>>>>>>>>>>>>>>>>OPTIC DISC REMOVAL<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

mm = Ip(:,:,2); %Extracted green channel

m4 = mean2(mm);

m5 = std2(mm);

figure,imshow(mm,[]);

%// Invert the green

if isinteger(mm)

z = intmax(class(mm))-mm;

elseif isfloat(mm)

z = 1 - mm;

elseif islogical(mm)

z = ~mm;

end

figure,imshow(z,[]);title('Optic Disc Removal');

se = strel('ball',85,85);

% adahist= histeq(z); % Structuring Element

% gopen = imopen( adahist,se); % Morphological Open

% godisk = adahist - gopen; % Remove Optic Disk

% figure,imshow(godisk,[]);title('Optic Disk Removed');

%%>>>>>>>>>>>>>>>>>>>>>>CANDIDATE EXTRACTION<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

xx = imregionalmin(z,8);

figure,subplot(1,2,1);imshow(xx);title('Reginal Minima');

% j = imgradient(xx);figure,imshow(j);

MM = bwareaopen(xx,20);subplot(1,2,2);imshow(MM);title('Finding Candidate');

% % % % % %-------->>>>>>DYNAMIC SHAPE FEATURES<<<<<<<<-------- % % % % % %

lesion\_dec = regionprops(MM,'All');

figure,imshow(im1);title('Output');

xxxx = 3\*ones(1,length(lesion\_dec));

hold on

for i = 1:length(lesion\_dec)

viscircles(lesion\_dec(i).Centroid,5,'Color','b');

end

hold off

n = numel(im1);

Istop = floor(avg - 0.5); % flooding level

Rarea = zeros(length(lesion\_dec),length(Istop));

Elong = zeros(length(lesion\_dec),length(Istop));

Ecc = zeros(length(lesion\_dec),length(Istop));

Circ = zeros(length(lesion\_dec),length(Istop));

Rect = zeros(length(lesion\_dec),length(Istop));

Sol = zeros(length(lesion\_dec),length(Istop));

for i = 1:Istop

for k =1:length(lesion\_dec)

n1 = numel(lesion\_dec(k).PixelList());

Rarea(k,i) = n1 ./ n;

a = lesion\_dec(k).BoundingBox(1,3:4);

Elong(k,i) = 1 -(a(1)/a(2));

Ecc(k,i) = lesion\_dec(k).Eccentricity();

Circ(k,i) = (lesion\_dec(k).Perimeter.^ 2)./ 4\*pi\*(lesion\_dec(k).Area());

Rect(k,i) = a(1).\*a(2);

Sol(k,i) = lesion\_dec(k).Solidity();

end

end

k = 1;

M = 4 + 6 \* (k + 4);

x = 1:Istop;

feature = zeros(length(lesion\_dec),M);

for c = 1:length(lesion\_dec)

p1 = polyfit(x,Rarea(c,:),1);

y1 = polyval(p1,x);

y1e = sqrt(mean2(((Rarea(c,:)) - y1).^2));

a1 = mean(Rarea(c,:)); aa1 = median(Rarea(c,:));f\_area = ([p1(1),p1(2),y1e,a1,aa1]);

p2 = polyfit(x,Elong(c,:),1);

y2 = polyval(p2,x);

y2e = sqrt(mean2(((Elong(c,:) - y2).^2)));

a2 = mean(Elong(c,:)); aa2 = median(Elong(c,:));f\_Elong = ([p2(1),p2(2),y2e,a2,aa2]);

p3 = polyfit(x,Ecc(c,:),1);

y3 = polyval(p3,x);

y3e = sqrt(mean2(((Ecc(c,:) - y3).^2)));

a3 = mean(Ecc(c,:)); aa3 = median(Ecc(c,:));f\_Ecc = [p3(1),p3(2),y3e,a3,aa3];

p4 = polyfit(x,Circ(c,:),1);

y4 = polyval(p4,x);

y4e = sqrt(mean2(((Circ(c,:) - y4).^2)));

a4 = mean(Circ(c,:)); aa4 = median(Circ(c,:));f\_Circ = [p4(1),p4(2),y4e,a4,aa4];

p5 = polyfit(x,Rect(c,:),1);

y5 = polyval(p5,x);

y5e = sqrt(mean2(((Rect(c,:) - y5).^2)));

a5 = mean(Rect(c,:)); aa5 = median(Rect(c,:));f\_Rect = ([p5(1),p5(2),y5e,a5,aa5]);

p6 = polyfit(x,Sol(c,:),1);

y6 = polyval(p6,x);

y6e = sqrt(mean2(((Sol(c,:) - y6).^2)));

a6 = mean(Sol(c,:)); aa6 = median(Sol(c,:));f\_Sol =([p6(1),p6(2),y6e,a6,aa6]);

feature(c,:) = [f\_area,f\_Elong,f\_Ecc,f\_Circ,f\_Rect,f\_Sol,m1,m2,m3,m4];

end

feature = mean(feature);

load Feature;

train = ones(88,1);

train(17:end) = 2;

rng(1); % For reproducibility

BaggedEnsemble = TreeBagger(200,Feature,train,'OOBPrediction','On',...

'Method','classification');

Yfit = predict(BaggedEnsemble,feature);

aaa=str2num(cell2mat(Yfit));

disp('RANDOM FOREST CLASSIFIER RESULT');

if aaa==1

msgbox('Normal');

disp('Normal');

else

msgbox('Abnormal');

disp('Abnormal');

end

**4.3 SNAPSHOTS:**

1. **INPUT & MOT**
2. **EMBEDDED OUTPUTS**
3. **Recover Watermark image and data**

**CHAPTER 5**

**CONCLUSION AND REFERENCES**

**5.1 CONCLUSION**

A robust and secure video steganography method in DWT-DCT domains based on MOT and ECC is proposed in this paper. The proposed algorithm is three-fold: 1) the motion-based MOT algorithm, 2) data embedding, and 3) data extraction. The performance of our suggested method is verified via extensive experiments, demonstrating the high embedding capacity with an average HR of 3.40% and 3.46% for DWT and DCT domains, respectively. An average PSNR of 49.01 and 48.67 dBs for DWT and DCT domains are achieved leading to a better visual quality for the proposed algorithm when compared to existing methods of the literature. The proposed algorithm has utilized MOT and ECC as the preprocessing stages which in turn provides a better confidentiality to the secret message prior to embedding phase. Moreover, through experiments from different perspectives, the security and robustness of the method against various attacks have been confirmed. In our future work, we will apply our algorithm in some other frequency domains such as curvelet transform for further improving the efficiency, visual quality, and security.

**5.2 REFERENCES:**

[1] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "A secure and improved self-embedding algorithm to combat digital document forgery," *Signal Processing,* vol. 89, pp. 2324-2332, 2009.

[2] X.-y. Wang, C.-p. Wang, H.-y. Yang, and P.-p. Niu, "A robust blind color image watermarking in quaternion Fourier transform domain," *Journal of Systems and Software,* vol. 86, pp. 255-277, 2013.

[3] M. Sajjad*, et al.*, "Mobile-cloud assisted framework for selective encryption of medical images with steganography for resource-constrained devices," *Multimedia Tools and Applications,* vol. 76, pp. 3519-3536, 2017.

[4] M. S. Subhedar and V. H. Mankar, "Current status and key issues in image steganography: A survey," *Computer Science Review,* vol. 13–14, pp. 95-113, 2014.

[5] S. Islam, M. R. Modi, and P. Gupta, "Edge-based image steganography," *EURASIP Journal on Information Security,* vol. 2014, pp. 1-14, 2014.

[6] M. Hasnaoui and M. Mitrea, "Multi-symbol QIM video 2169-3536 (c) 2016 IEEE. Translations and content mining are permitted for academic research only.

[7] R. J. Mstafa and K. M. Elleithy, "A novel video steganography algorithm in the wavelet domain based on the KLT tracking algorithm and BCH codes," in *2015 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2015, pp. 1-7.

[8] K. Qazanfari and R. Safabakhsh, "A new steganography method which preserves histogram: Generalization of LSB++," *Information Sciences,* vol. 277, pp. 90-101, 2014.

[9] L. Guangjie, L. Weiwei, D. Yuewei, and L. Shiguo, "Adaptive Steganography Based on Syndrome-Trellis Codes and Local Complexity," in *2012 Fourth International Conference on Multimedia Information Networking and Security (MINES)*, 2012, pp. 323-327.

[10] A. Singh, B. Kumar, S. Singh, S. Ghrera, and A. Mohan, "Multiple watermarking technique for securing online social network contents using Back Propagation Neural Network," *Future Generation Computer Systems,* 2016.

[11] C. Rupa, "A Digital Image Steganography using Sierpinski Gasket Fractal and PLSB," *Journal of The Institution of Engineers (India): Series B,* vol. 94, pp. 147-151, 2013/09/01 2013.

[12] R. J. Mstafa and K. M. Elleithy, "Compressed and raw video steganography techniques: a comprehensive survey and analysis," *Multimedia Tools and Applications,* pp. 1-38, 2016.

[13] K. Muhammad*, et al.*, "A Secure Method for Color Image Steganography using Gray-Level Modification and Multi-level Encryption," *KSII Transactions on Internet and Information Systems (TIIS),* vol. 9, pp. 1938-1962, 2015.

[14] A. K. Singh, M. Dave, and A. Mohan, "Hybrid technique for robust and imperceptible multiple watermarking using medical images," *Multimedia Tools and Applications,* vol. 75, pp. 8381-8401, 2016.

[15] R. Zhang, V. Sachnev, and H. Kim, "Fast BCH Syndrome Coding for Steganography," in *Information Hiding*. vol. 5806, S. Katzenbeisser and A.-R. Sadeghi, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 48-58.

[16] C. Fontaine and F. Galand, "How Can Reed-Solomon Codes Improve Steganographic Schemes?," in *Information Hiding*. vol. 4567, T. Furon*, et al.*, Eds., ed: Springer Berlin Heidelberg, 2007, pp. 130-144.

[17] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information Sciences,* vol. 279, pp. 251-272, 2014.

[18] T. Yiqi and W. KokSheik, "An Overview of Information Hiding in H.264/AVC Compressed Video," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 24, pp. 305-319, 2014.

[19] R. J. Mstafa and K. M. Elleithy, "A high payload video steganography algorithm in DWT domain based on BCH codes (15, 11)," in *Wireless Telecommunications Symposium (WTS)*, 2015, pp. 1-8.

[20] P.-C. Chang, K.-L. Chung, J.-J. Chen, C.-H. Lin, and T.-J. Lin, "A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames," *Journal of Visual Communication and Image Representation,* vol. 25, pp. 239-253, 2014.

[21] X. Ma, Z. Li, H. Tu, and B. Zhang, "A data hiding algorithm for H. 264/AVC video streams without intra-frame distortion drift," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 20, pp. 1320-1330, 2010.

[22] Z. Shahid, M. Chaumont, and W. Puech, "Considering the reconstruction loop for data hiding of intra-and inter-frames of H. 264/AVC," *Signal, Image and Video Processing,* vol. 7, pp. 75-93, 2013.

[23] R. WANG, L. HU, and D. XU, "A Watermarking Algorithm Based on the CABAC Entropy Coding for H.264/AVC," *J. Comput. Inform. Syst.,* vol. 7, no. 6, pp. 2132–2141, 2011.

[24] Y. Liu, Z. Li, X. Ma, and J. Liu, "A robust data hiding algorithm for H.264/AVC video streams," *Journal of Systems and Software,* vol. 86, pp. 2174-2183, 2013.

[25] N. Ke and Z. Weidong, "A video steganography scheme based on H. 264 bitstreams replaced," in *4th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2013, pp. 447-450.

[26] M. A. Alavianmehr, M. Rezaei, M. S. Helfroush, and A. Tashk, "A lossless data hiding scheme on video raw data robust against H.264/AVC compression," in *2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)*, 2012, pp. 194-198.

[27] E. Prasad, "High Secure Image Steganography based on Hopfield Chaotic Neural Network and Wavelet Transforms," *International Journal of Computer Science & Network Security,* vol. 13, 2013.

[28] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 11, pp. 674-693, 1989.