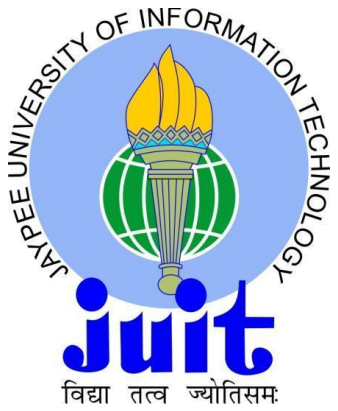
CONTROLLING MOUSE CURSOR USING HAND GESTURE RECOGNITION IN MATLAB

Submitted in partial fulfillment of the Degree of

Bachelor of Technology



May – 2014

Name of Students - Ashish Thakur (101090)

And Enrollment. No. Milind Singh Chauhan (101091)

Ashish Thakur (101099)

Name of supervisor - Ms. Pragya Gupta

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Certificate

This is to certify that project report entitled "**CONTROLLING MOUSE CURSOR USING HAND GESTURE RECOGNITION IN MATLAB**", submitted by **ASHISH THAKUR (101090), MILIND SINGH CHAUHAN (101091) and ASHISH THAKUR (101099)** in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date:

26th May 2014

Ms. PRAGYA GUPTA

ASSISTANT PROFESSOR

ACKNOWLEDGEMENT

We hereby express our sincere thanks to respected Prof. Dr. Sunil Bhooshan for giving us the chance to be the part of this project. We indebted to our project mentor Ms. Pragya Gupta who always guided us and helped us whenever needed and for her constant motivation and support during the course of our work. We truly appreciate and value her esteemed guidance and encouragement from the beginning to the end of this project work. We are indebted to her for having helped us shape the problem and providing insights towards the solution. We will be failing in our duty if we do not mention the lab staff and administrative staff of this department for their timely help. We would like to thank all whose direct and indirect support helped us completing our project in time.

Date: 15 May,2014

Name of student:

Ashish Thakur(101090)

Milind Singh Chauhan(101091)

Ashish Thakur(101099)

CONTENTS

LIST OF FIGURES

FIGURE NO. **PAGE NO.**

CHAPTER 1

SUMMARY

Abundant amount of input devices are used to interact with the computer world or more precisely saying to digital world and very less through gestures made by body movements. Concepts of assistive technology are one of them used for controlling the input from mouse movements, like by detecting the eye movements of a user with the help of eye tracking system, hand gestures, etc. In our work, we have tried to control mouse cursor movement and click events using a camera based on colour detection technique. Here real time video has been captured using a Web-Camera. The user wears coloured tapes to provide information to the system. Individual frames of the video are separately processed. The processing techniques involve an image subtraction algorithm to detect colours. Once the colours are detected the system performs various operations to track the cursor and performs control actions, the details of which are provided below. No additional hardware is required by the system other than the standard webcam which is provided in every laptop computer. The sections below will present the project in greater detail.

CHAPTER 2

INTRODUCTION

Since the computer technology continues to grow up, the importance of human computer interaction is enormously increasing. The keyboard and mouse are currently the main interfaces between man and computer. In other areas where 3D information is required, such as computer games, robotics and design, other mechanical devices such as roller-balls, joysticks and data-gloves are used.

Humans communicate mainly by vision and sound, therefore, a man-machine interface would be more intuitive if it made greater use of vision recognition. . Another advantage is that the user not only can communicate from a distance, but need have no physical contact with the computer

However, this technology is still not cheap enough to be used in desktop systems. Creating a virtual human computer interaction device such as mouse or keyboard using a webcam and computer vision techniques can be an alternative way for the touch screen. In this study, finger tracking based a virtual mouse application has been designed and implemented using a regular webcam.

The visual system chosen was the recognition of hand gestures. The amount of computation required to process hand gestures is much greater than that of the mechanical devices, however standard desktop computers are now quick enough to make this project hand gesture recognition using computer vision — a viable proposition.

The motivation was to create an object tracking application to interact with the computer, and develop a virtual human computer interaction device. It is software that allows users to give mouse inputs to a system without using an actual mouse. To the extreme it can also be called as hardware because it uses an ordinary web camera.

A gesture recognition system could be used in any of the following areas:

1. Man-machine interface: using hand gestures to control the computer mouse and/or keyboard functions. An example of this, which has been implemented in this project, controls various keyboard and mouse functions using gestures alone.
2. 3D animation: Rapid and simple conversion of hand movements into 3D computer space for the purposes of computer animation.
3. Visualisation: Just as objects can be visually examined by rotating them with the hand, so it would be advantageous if virtual 3D objects (displayed on the computer screen) could be manipulated by rotating the hand in space.
4. Computer games: Using the hand to interact with computer games would be more natural for many applications.
5. Control of mechanical systems (such as robotics): Using the hand to remotely control a manipulator.

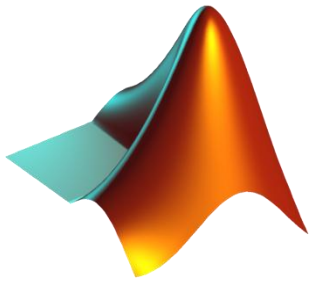
This project which uses web camera works with the help of different image processing techniques. In this the hand movements of a user is mapped into mouse inputs. A web camera is set to take images continuously. The user must have a particular color in his hand so that when the web camera takes image it must be visible in the image. This color is detected from the image pixel and the pixel position is mapped into mouse input. Depending upon the size of the image taken by camera various scaling techniques are used because the pixel position in the image will not have a correspondence with screen resolution. In this study, a red color pointer has been used for the object recognition and tracking. Left and the right click events of the mouse would be covered in the 8^{th} semester.

“Many researchers in the human computer interaction and robotics fields have tried to control mouse movement using video devices. However, all of them used different methods to make a clicking event. One approach, by Erdem et al, used fingertip tracking to control the motion of the mouse. A click of the mouse button was implemented by defining a screen such that a click occurred when a user’s hand passed over the region. Another approach was developed by Chu-Feng Lien. He used only the finger-tips to control the mouse cursor and click. His clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. Paul et al, used still another method to click. They used the motion of the thumb (from a ‘thumbs-up’ position to a fist) to mark a clicking event thumb. Movement of the hand while making a special hand sign moved the mouse pointer.”

Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse.

Gesture recognition enables humans to communicate with the machine (HMI) and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at the computer screen so that the cursor will move accordingly. This could potentially make conventional input devices such as mouse, keyboards and even touch-screens redundant.

Gesture recognition can be conducted with techniques from computer vision and image processing.



CHAPTER 3

TECHNICAL DETAILS

This Hand Gesture Recognition project is developed using MATLAB software. MATLAB stands for matrix laboratory. MATLAB is a powerful and high-performance language for technical computing. MATLAB integrates programming, computation, and visualization in user friendly environment where problems & solutions are presented in common mathematical notation.

MATLAB is uses for:

- Math and calculation
- Development of algorithms
- Prototyping, modeling and simulation
- Data analysis, exploration and visualization
- Scientific and engineering graphics
- Application/software development

MATLAB is an interactive system which allows us to solve many technical computing problems, especially those problems involved with vector and matrix formulations. This because MATLAB's basic data element is an array which does not require dimensioning. This would save user's time that takes to write a program if compared with non-interactive language such as Fortran or C.

MATLAB already developed and evolved over a period of years by researchers and is still developing. In study environment, MATLAB is advanced courses in engineering, science and mathematics and is a standard instructional tool for introductory. In industry, MATLAB's tool is widely use for high-productivity research, development and analysis.

Another thing that makes MATLAB so special is its Toolboxes. Toolboxes are MATLAB features a family of application-specific solutions. Toolboxes allow user to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Examples of Toolboxes available are fuzzy logic, neural network, signal processing, image processing, wavelets, image acquisition and many more.

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java.

You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

MATLAB can call functions and subroutines written in the C programming language or 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" (for MATLAB executable).

Libraries written in Perl, Java, ActiveX or .NET can be directly called from MATLAB, and many MATLAB libraries (for example XML or SQL support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with a MATLAB toolbox which is sold separately by MathWorks, or using an undocumented mechanism called JMI (Java-to-MATLAB Interface), (which should not be confused with the unrelated Java Metadata Interface that is also called JMI).

As alternatives to the MuPAD based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to Maple or Mathematica.

Libraries also exist to import and export MathML.

MATLAB is a programming language developed by MathWorks. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job.

Most MATLAB scripts and functions can be run in the open source programme octave. This is freely available for most computing platforms.

GNU Octave and LabVIEW MathScript are systems for numerical computations with an m-file script language that is mostly compatible with MATLAB. Both alternatives can replace MATLAB in many circumstances. While a good deal of the content of this book will also apply to both Octave and LabVIEW MathScript, it is not guaranteed to work in exactly the same manner. Differences and comparison between MATLAB and Octave are presented in Comparing Octave and MATLAB.

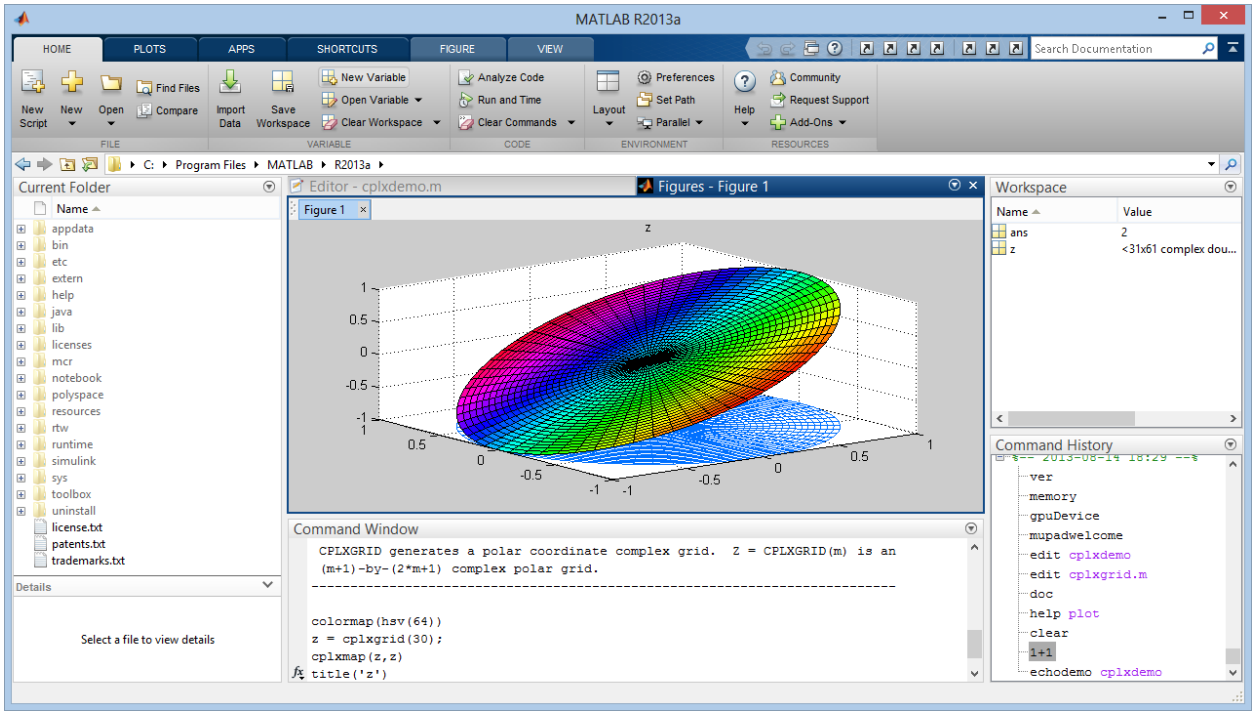


FIGURE 3.1

CHAPTER 4

IMAGE PROCESSING

Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing are also possible. Here image processing is in the sense that we are splitting each pixel of the image into RGB components. In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in the first place) is referred to as imaging.

Image processing refers to processing of a 2D picture by a computer. Basic definitions:

An image defined in the "real world" is considered to be a function of two real variables, for example, a(x,y) with a as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y).

Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories:

- Image Processing (image in -> image out)
- Image Analysis (image in -> measurements out)
- Image Understanding (image in -> high-level description out)

An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions.

Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition.

Sequence of image processing:

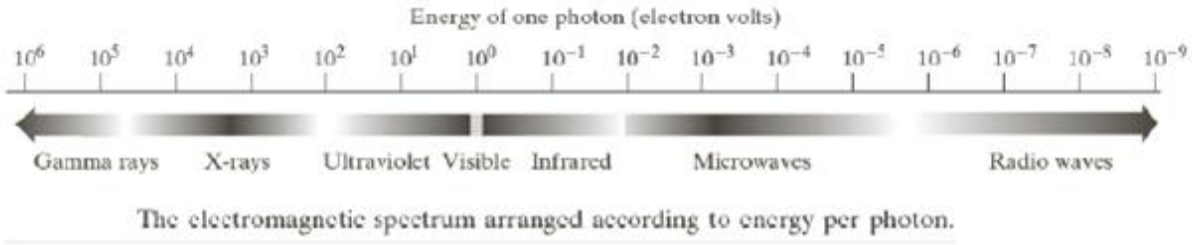
Most usually, image processing systems require that the images be available in digitized form, that is, arrays of finite length binary words. For digitization, the given Image is sampled on a discrete grid and each sample or pixel is quantized using a finite number of bits. The digitized image is processed by a computer. To display a digital image, it is first converted into analog signal, which is scanned onto a display.

Closely related to image processing are computer graphics and computer vision. In computer graphics, images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies. Computer vision, on the other hand, is often considered high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans).

In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or real-time multi-asset portfolio trading in finance.Before going to processing an image, it is converted into a digital form. Digitization includes sampling of image and quantization of sampled values. After converting the image into bit information, processing is performed. This processing technique may be Image enhancement, Image restoration, and Image compression.

Fields that use image Processing:

- Gamma-Ray Imaging
- X-Ray Imaging
- Imaging in the Ultraviolet Band
- Imaging in the Visible and Infrared Bands
- Imaging in the Microwave Band
- Radio Band

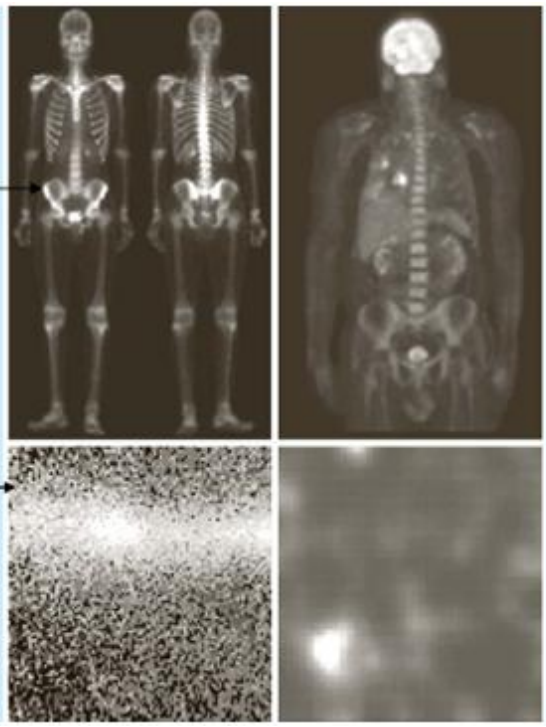


The electromagnetic spectrum arranged according to energy per photon.

FIGURE 4.1

Gamma-Ray Imaging:

a: A radioactive isotope is injected, which emits positrons as it decays; when a positron meets an electron, both are annihilated and two gamma rays are generated.
b: Positron-Emission Tomography
c: Natural gamma ray source superheated stationary gas cloud in the constellation of Cygnus.
d: An image of gamma radiation from a valve in a nuclear reactor.

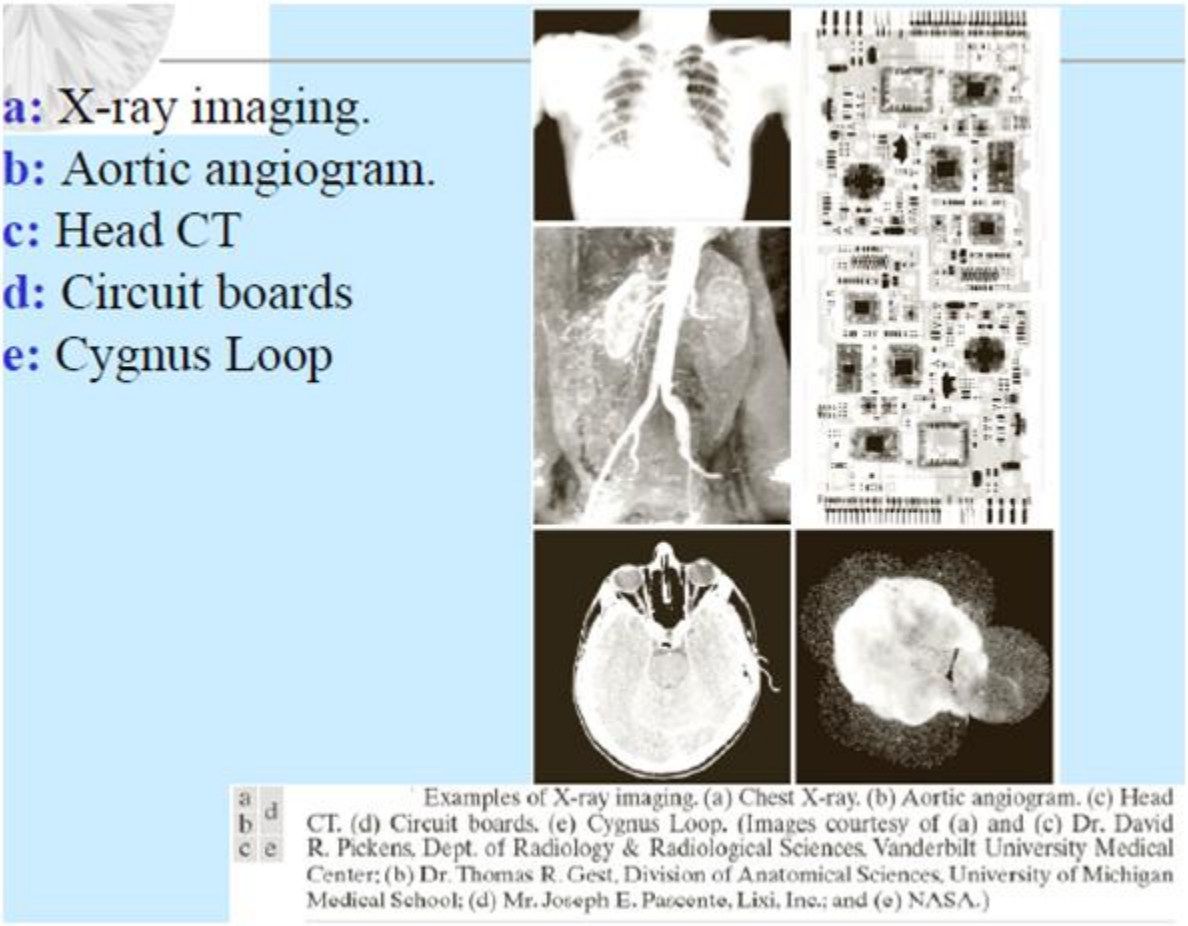


a b
c d

Examples of gamma-ray imaging. (a) Bone scan. (b) PET image. (c) Cygnus Loop. (d) Gamma radiation (bright spot) from a reactor valve. (Images courtesy of (a) G.E. Medical Systems, (b) Dr. Michael E. Casey, CTI PET Systems, (c) NASA, (d) Professors Zhong He and David K. Wehe, University of Michigan.)

FIGURE 4.2

X-Ray Imaging:



a b c d e

Examples of X-ray imaging. (a) Chest X-ray. (b) Aortic angiogram. (c) Head CT. (d) Circuit boards. (e) Cygnus Loop. (Images courtesy of (a) and (c) Dr. David R. Pickens, Dept. of Radiology & Radiological Sciences, Vanderbilt University Medical Center; (b) Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School; (d) Mr. Joseph E. Pascente, Lixi, Inc.; and (e) NASA.)

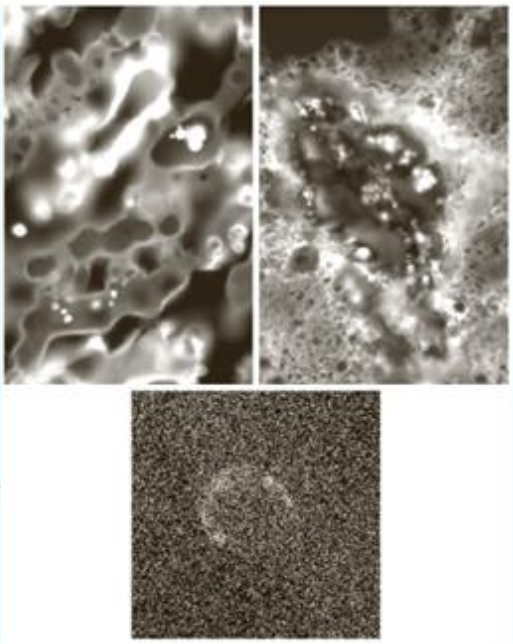
FIGURE 4.3

Imaging in the Ultraviolet Band:

Ultraviolet imaging Microphotography; normal corn and corn infected by parasites.

Visible radiation is excited by UV light (fluorescence)

Cygnus Loop in the high-UV band

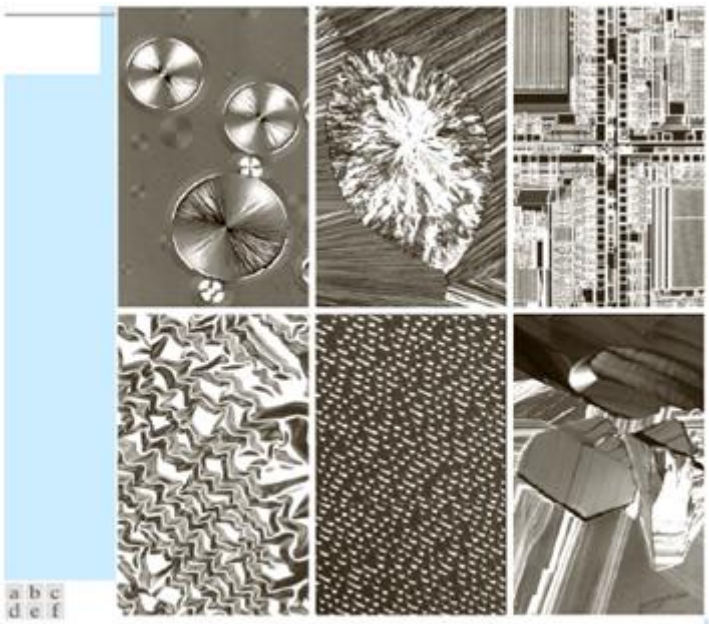


a b
c

Examples of ultraviolet imaging.
(a) Normal corn.
(b) Smut corn.
(c) Cygnus Loop.
(Images courtesy of (a) and (b) Dr. Michael W. Davidson, Florida State University, (c) NASA.)

FIGURE 4.4

Imaging in the Visible and Infrared Bands:



a b c
d e f

Examples of light microscopy images. (a) Taxol (anticancer agent), magnified 250×. (b) Cholesterol—40×. (c) Microprocessor—60×. (d) Nickel oxide thin film—600×. (e) Surface of audio CD—1750×. (f) Organic superconductor—450×. (Images courtesy of Dr. Michael W. Davidson, Florida State University.)

FIGURE 4.5

Imaging in the Microwave Band:

Spaceborne radar image of mountains in southeast Tibet. (Courtesy of NASA.)

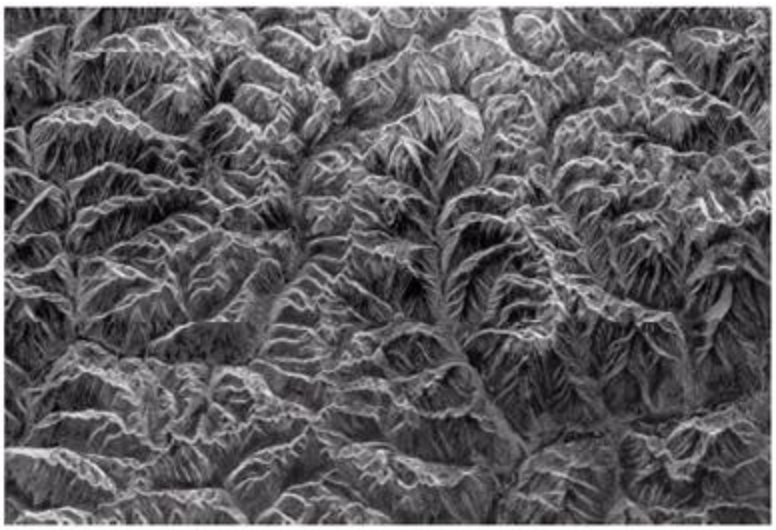
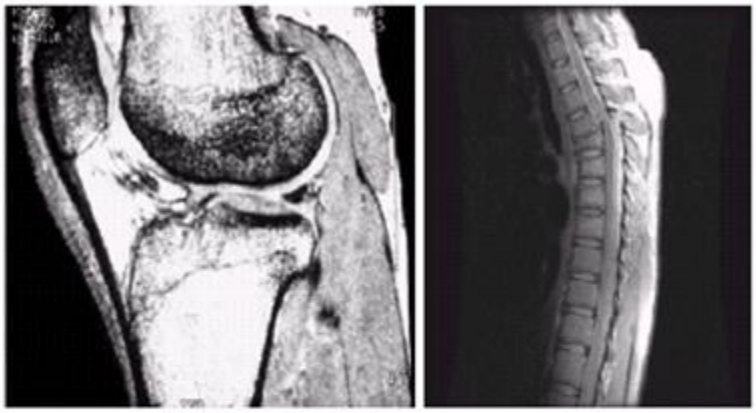


FIGURE 4.6

Radio Band:

Magnetic Resonance Imaging

(MRI): nuclei with nonzero magnetic moment will align with a strong magnetic field, and resonate with a time varying component of the field. After the time-varying component is removed, the exponential decay time of the re-alignment is measured and used to develop image contrast between different tissues.



a b

MRI images of a human (a) knee, and (b) spine. (Image (a) courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School, and (b) Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

FIGURE 4.7

Image enhancement:

It refers to accentuation, or sharpening, of image features such as boundaries, or contrast to make a graphic display more useful for display & analysis. This process does not increase the inherent information content in data. It includes gray level & contrast manipulation, noise reduction, edge crispening and sharpening, filtering, interpolation and magnification, pseudo coloring, and so on.

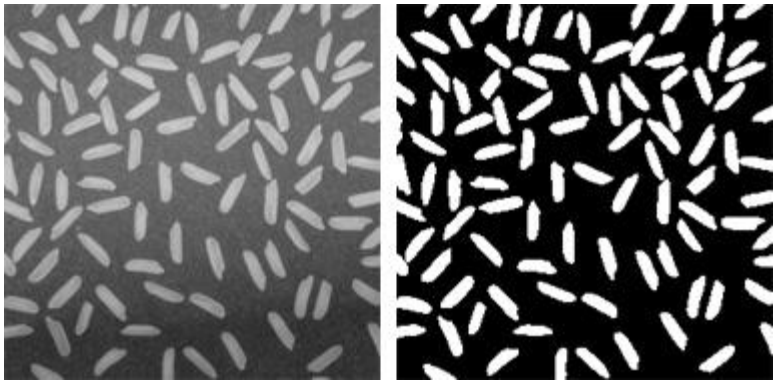


FIGURE 4.8: *Correcting nonuniform illumination with morphological operators.*

Image restoration:

It is concerned with filtering the observed image to minimize the effect of degradations. Effectiveness of image restoration depends on the extent and accuracy of the knowledge of degradation process as well as on filter design. Image restoration differs from image enhancement in that the latter is concerned with more extraction or accentuation of image features.

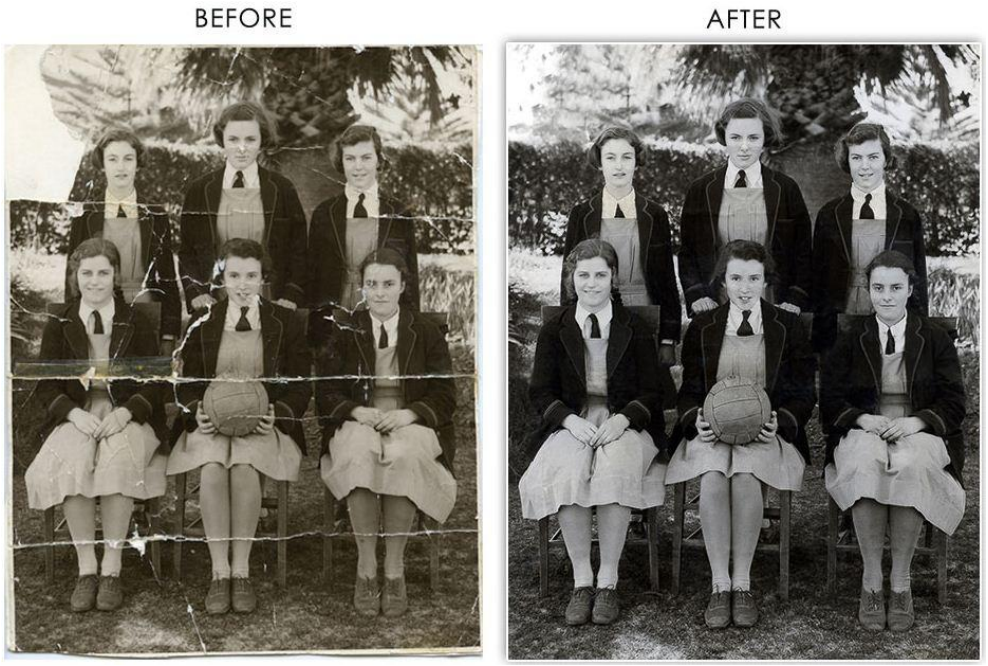


FIGURE 4.9: *The image is being restored and the degraded part is being restored to create a smooth image.*

Image compression:

It is concerned with minimizing the number of bits required to represent an image. Application of compression are in broadcast TV, remote sensing via satellite, military communication via aircraft, radar, teleconferencing, facsimile transmission, for educational & business documents, medical images that arise in computer tomography, magnetic resonance imaging and digital radiology, motion, pictures, satellite images, weather maps, geological surveys and so on.

- Text compression – CCITT GROUP3 & GROUP4
- Still image compression – JPEG
- Video image compression - MPEG

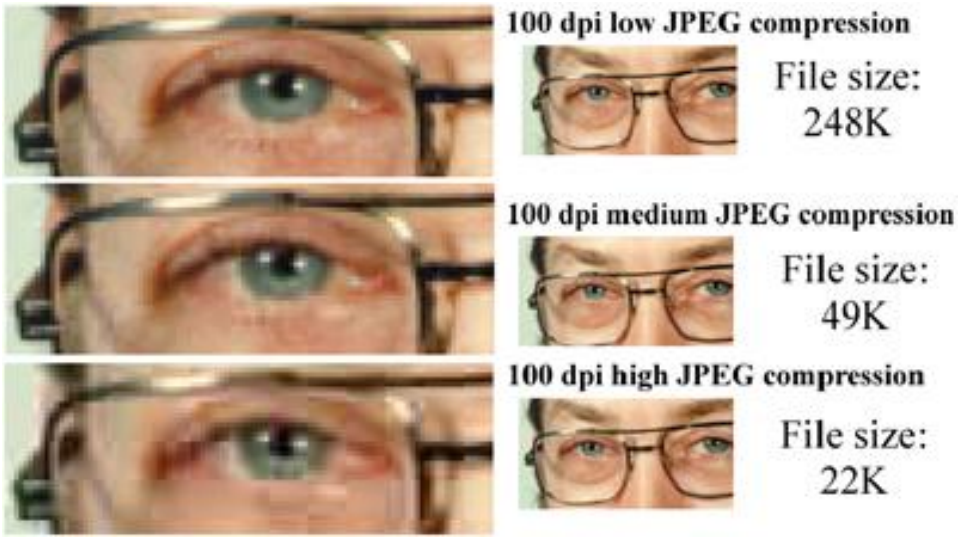


FIGURE 4.10: *An image has been compressed into smaller sizes but the essential some features of the image is being affected by the compression.*

Image Analysis:

Image analysis is the extraction of meaningful information from images; mainly from digital images by means of digital image processing techniques. Image analysis tasks can be as simple as reading bar coded tags or as sophisticated as identifying a person from their face. In this project we do various image analyzing techniques. The main thing done is the color detection. At first we receive an image from the web cam. Then each pixel is retrieved from the image and extracts the red, green and blue values (RGB) from each pixel. Now we can easily detect a particular color since all the colors are combinations of RGB values. Here we just to try to detect only Red Green and Blue colors. This is done by traversing through the image, retrieving each pixel, extracting RGB values and then comparing the color values to the RGB values. In java the getRGB( ) method is used for extracting a pixel from a image.

Image Distortion:

If the camera's visual axis is not perpendicular to the floor plane, a given gesture would appear different depending on the position and yaw of the hand (a given length in one area of the frame would appear longer or shorter in another area of the frame). This is termed projective distortion. Also, if the camera lens is of poor quality then the straight sides of a true square in the frame would appear curved. This is termed radial distortion.

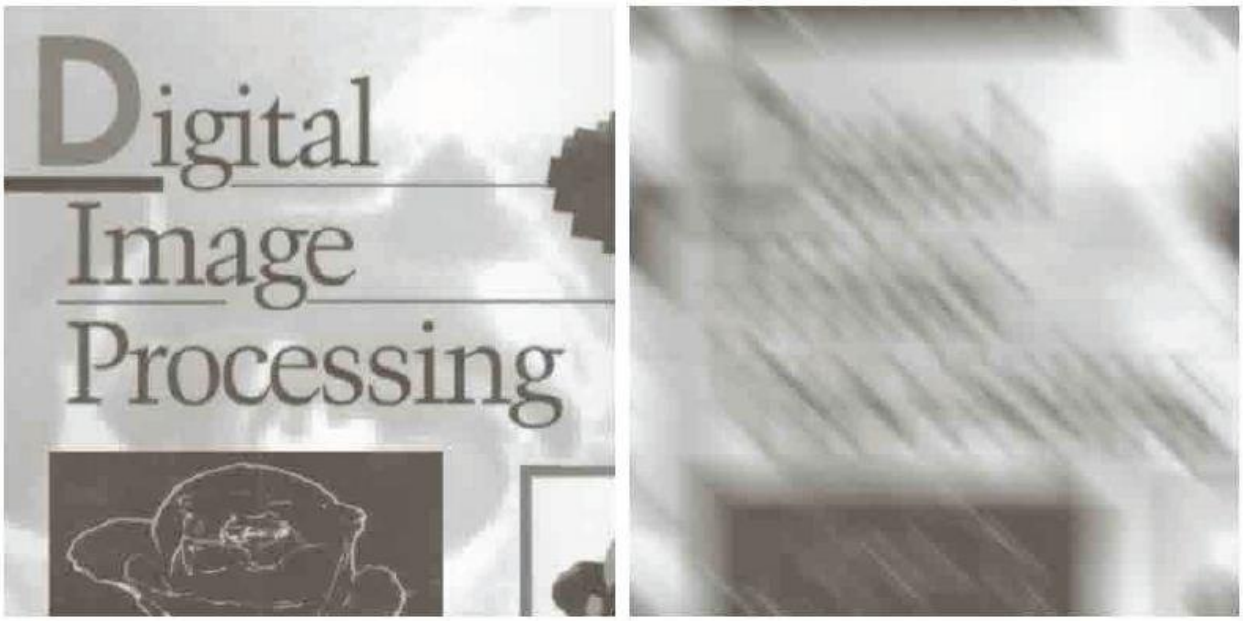


FIGURE 4.11

(a) Original Image *(b) Distorted image using blurring*

Image Acquisition Toolbox

Image Acquisition Toolbox enables you to acquire images and video from cameras and frame grabbers directly into MATLAB and Simulink. You can detect hardware automatically and configure hardware properties. Advanced workflows let you trigger acquisition while processing in-the-loop, perform background acquisition, and synchronize sampling across several multimodal devices. With support for multiple hardware vendors and industry standards, you can use imaging devices ranging from inexpensive Web cameras to high-end scientific and industrial devices that meet low-light, high-speed, and other challenging requirements.

Key Features

- Support for industry standards, including DCAM, Camera Link, and GigE Vision
- Support for common OS interfaces for webcams, including Direct Show, QuickTime, and video4linux2
- Support for a range of industrial and scientific hardware vendors
- Multiple acquisition modes and buffer management options
- Synchronization of multimodal acquisition devices with hardware triggering
- Image Acquisition app for rapid hardware configuration, image acquisition, and live video previewing
- Support for C code generation in Simulink

Together, MATLAB, Image Acquisition Toolbox, and Image Processing Toolbox (and, optionally, Computer Vision System Toolbox™) provide a complete environment for developing customized imaging solutions. You can acquire images and video, visualize data, develop processing algorithms and analysis techniques, and create Ui's and apps. The image acquisition engine enables you to acquire frames as fast as your camera and PC can support for high speed imaging. In addition, you can use Image Acquisition Toolbox with Simulink and Computer Vision System Toolbox to model and simulate real-time embedded imaging systems.

Image Acquisition Toolbox simplifies the acquisition process by providing a consistent interface across operating systems, hardware devices, and vendors. The toolbox provides multiple ways to access hardware devices from MATLAB and Simulink: the Image Acquisition Tool, a programmatic interface in MATLAB, and a block for Simulink. Each workflow provides access to camera properties and controls while enabling you to solve different types of problems with the strengths of each environment.

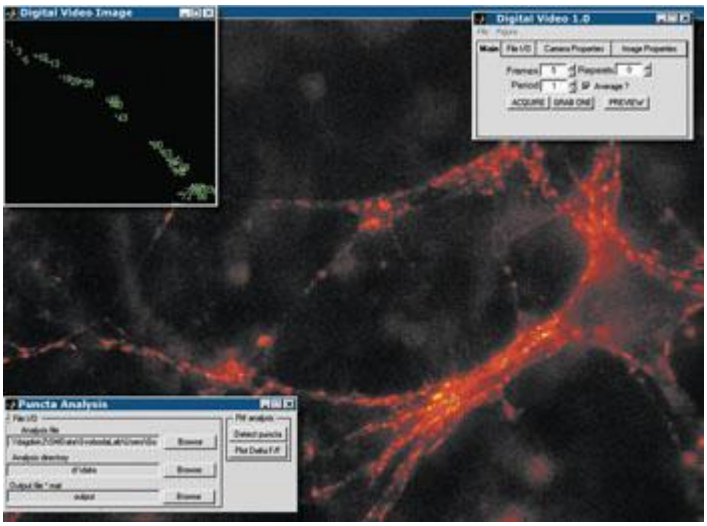


FIGURE 4.12: *Image Acquisition Toolbox application that acquires and analyzes images of central synapses to monitor synaptic transmission over time. GUIs (insets, upper right and lower left) enable researchers to tune acquisition and processing parameters. Image courtesy of Polugruto, T.A.; Tervo, D.G.; and Svoboda, K.; Howard Hughes Medical Institute/Cold Spring Harbor Labs.*

Image Acquisition in MATLAB

Image Acquisition Toolbox provides an app and functions and a programmatic interface to help you work with image acquisition hardware in MATLAB. You can automate repetitive tasks, create workflows combined with tasks such as image processing, and create standalone executables that acquire images and video with MATLAB Compiler™. The toolbox enables you to customize the acquisition process to include integrating image processing functionality to identify objects, enhance imagery, or construct mosaics and panoramic views as the data is acquired.

FIGURE 4.13: *Image of blister packs acquired by Image Acquisition Toolbox. Image Processing Toolbox analyzes the data and marks broken pills.*

Acquiring Image Data

Image Acquisition Toolbox supports several modes, including background acquisition and continuous acquisition, while processing the acquired data. The toolbox automatically buffers data into memory, handles memory and buffer management, and enables acquisition from an ROI. The image acquisition engine is designed to acquire imagery as fast as your camera and computer can support, enabling analysis and processing of high-speed imaging applications.Data can be acquired in a wide range of data types, including signed or unsigned 8-, 16-, and 32-bit integers and single- or double-precision floating point. The toolbox supports any color space provided by the image acquisition device including RGB, YUV, or grayscale. Raw sensor data in a Bayer pattern can be automatically converted into RGB data.

Computer Vision System Toolbox

Computer Vision System Toolbox provides algorithms, functions, and apps for the design and simulation of computer vision and video processing systems. You can perform object detection and tracking, feature detection and extraction, feature matching, stereo vision, camera calibration, and motion detection tasks. The system toolbox also provides tools for video processing, including video file I/O, video display, object annotation, drawing graphics, and compositing. Algorithms are available as MATLAB functions, System objects, and Simulink blocks.

For rapid prototyping and embedded system design, the system toolbox supports fixed-point arithmetic and automatic C-code generation.

Feature Detection, Extraction, and Matching

Computer Vision System Toolbox provides a suite of feature detectors and descriptors. Additionally, the system toolbox provides functionality to match two sets of feature vectors and visualize the results.

When combined into a single workflow, feature detection, extraction, and matching can be used to solve many computer vision design challenges, such as image registration, stereo vision, object detection, and tracking.

Feature Detection and Extraction

A feature is an interesting part of an image, such as a corner, blob, edge, or line. Feature extraction enables you to derive a set of feature vectors, also called descriptors, from a set of detected features. Computer Vision System Toolbox offers capabilities for feature detection and extraction that include:

- Corner detection, including Shi & Tomasi, Harris, and FAST methods
- BRISK, MSER, and SURF detection for blobs and regions
- Extraction of BRISK, FREAK, SURF, and simple pixel neighborhood descriptors
- Histogram of Oriented Gradients (HOG) feature extraction
- Visualization of feature location, scale, and orientation

FIGURE 4.14: *SURF (left), MSER (center), and corner detection (right) with Computer Vision System Toolbox. Using the same image, the three different feature types are detected and results are plotted over the original image.*

Feature Matching

Feature matching is the comparison of two sets of feature descriptors obtained from different images to provide point correspondences between images. Computer Vision System Toolbox offers functionality for feature matching that includes:

- Configurable matching metrics, including SAD, SSD, and normalized cross-correlation
- Hamming distance for binary features
- Matching methods including Nearest Neighbor Ratio, Nearest Neighbor, and Threshold
- Multicore support for faster execution on large feature sets

FIGURE 4.15: *Detected features indicated by red circles (left) and green crosses (right). The yellow lines indicate the corresponding matched features between the two images.*

Statistically robust methods like RANSAC can be used to filter outliers in matched feature sets while estimating the geometric transformation or fundamental matrix, which is useful when using feature matching for image registration, object detection, or stereo vision applications.

Feature-Based Image Registration

Image registration is the transformation of images from different camera views to use a unified co-coordinate system. Computer Vision System Toolbox supports an automatic approach to image registration by using features. Typical uses include video mosaicking, video stabilization, and image fusion.

Feature detection, extraction, and matching are the first steps in the feature-based automatic image registration workflow. You can remove the outliers in the matched feature sets using RANSAC to compute the geometric transformation between images and then apply the geometric transformation to align the two images.

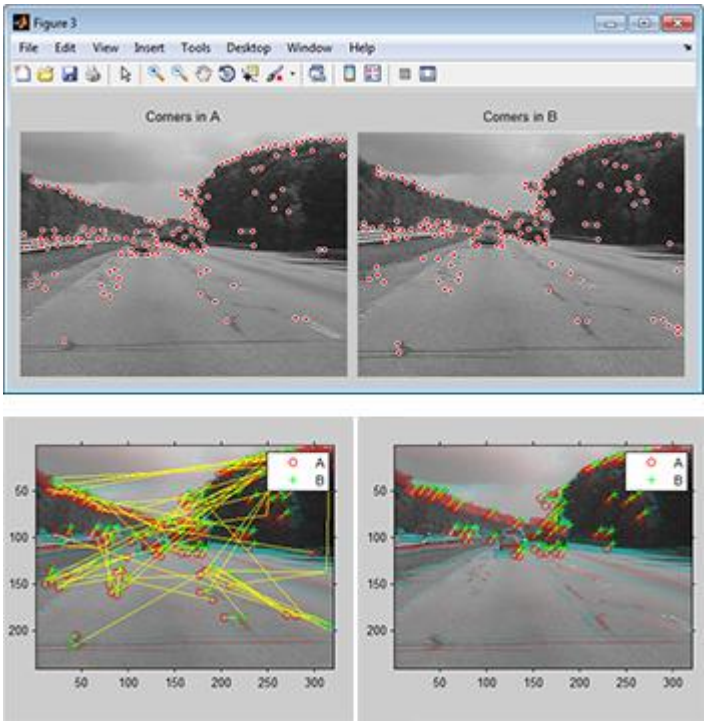


FIGURE 4.16: *Feature-based registration, used for video stabilization. The system toolbox detects interest points in two sequential video frames using corner features (top); the putative matches are determined with numerous outliers (bottom left), and outliers are removed using the RANSAC method (bottom right).*

Object Detection and Recognition

Object detection and recognition are used to locate, identify, and categorize objects in images and video. Computer Vision System Toolbox provides a comprehensive suite of algorithms and tools for object detection and recognition.

Object Classification

You can detect or recognize an object in an image by training an object classifier using pattern recognition algorithms that create classifiers based on training data from different object classes. The classifier accepts image data and assigns the appropriate object or class label.

FIGURE 4.17: *Face detection using Viola-Jones algorithm using a cascade of classifiers to detect faces*

FIGURE 4.18: ***People detector***
Detecting people using pretrained support vector machine(SVM) with histogram of oriented gradient (HOG) features

FIGURE 4.19: ***Text detection and optical character recognition (OCR)***
Recognizing text in natural images

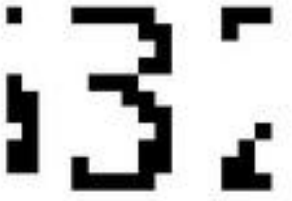


FIGURE 4.20: ***Digit classification***
Classifying digits using support vector machines (SVM) and HOG feature extraction

Motion-Based Object Detection

Motion-based object detection algorithms use motion extraction and segmentation techniques such as optical flow and Gaussian mixture model (GMM) foreground detection to locate moving objects in a scene. Blob analysis is used to identify objects of interest by computing the blob properties from the output of a segmentation or motion extraction algorithm such as background subtraction.

FIGURE 4.21: ***Detect moving objects with optical flow***

FIGURE 4.22: ***Detect cars using Gaussian Mixture models***

Feature-Based Object Detection

Feature points are used for object detection by detecting a set of features in a reference image, extracting feature descriptors, and matching features between the reference image and an input. This method of object detection can detect reference objects despite scale and orientation changes and is robust to partial occlusions.

FIGURE 4.23: *Reference image of object (left), input image (right); the yellow lines indicate the corresponding matched features between the two images.*

Training Object Detectors and Classifiers

Training is the process of creating an object detector or classifier to detect or recognize a specific object of interest. The training process utilizes:

- Positive images of the object of interest at different scales and orientations

- Negative images of backgrounds typically associated with the object of interest
- Nonobjects similar in appearance to the object of interest

The system toolbox provides an app to select and assign regions of interest (ROI) and label training images.

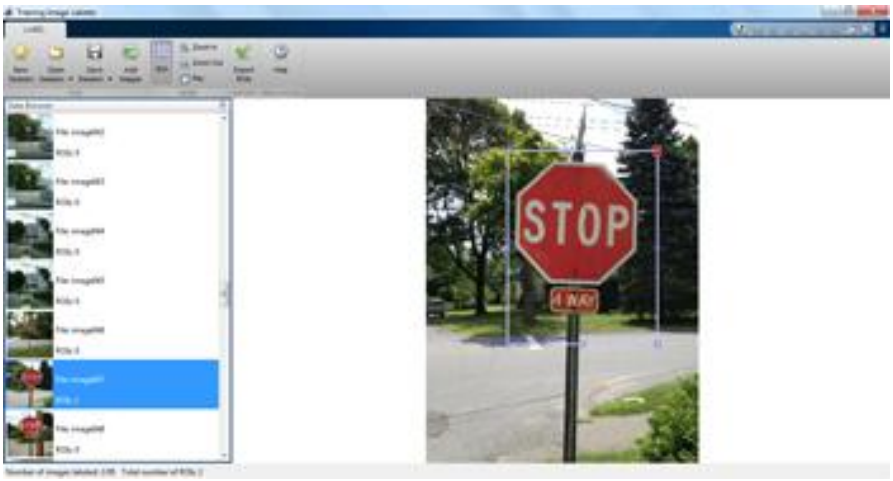


FIGURE 4.24: *Training image labeler app to select regions of interest (ROIs) in positive training images.*

The system toolbox provides functions to train a Viola-Jones object detector to locate any object of interest. An app to train a detector is available on File Exchange.

CHAPTER 5

MOTIVATION

The use of only basic input devices such as the mouse and keyboard is limiting the creativity and capabilities of the user. We want to expand the ways that people are able to interact with their computers. We want to enable all users to interact more naturally using simple hand gestures to control tasks. These hand gestures can be recognized in real time to allow for quick human-computer interaction. Hand gestures can be used for cursor control via a computer vision-based interface. Video-based hand tracking and gesture recognition is a powerful tool as it can provide a more natural interface to the computer in comparison to a traditional mouse.

Certain application ideas mentioned in modern technology websites, magazines, and even movies such as the Minority Report range from allowing people to point at objects on a display screen in a effort to utilize them, rotate objects or models by rotating their hands, or zooming in and out of figures by way of specific hand gestures.

This project is an application of the mentioned ideas and could be seen advantageous to bed stricken patients, or handicapped individuals with limited range or motion that are unable to get out of bed and sit at a desk for a extended period of time to operate a computer. Through the use of the program, individuals could now easily operate a PC's interface through the simple motion and gesture of their hand recorded by a video capture device connected to a computer from any location in their household. Target audience & Need for the application. Anyone with a computer and a camera would be should be able to take advantage of this method. This type of human and computer interaction would also benefit those that have difficulty using the normal input devices. For most users they will find that this is a more natural way to interact with their computers. This project is an application of the mentioned ideas and could be seen advantageous to bed stricken patients, or handicapped individuals with limited range or motion that are unable to get out of bed and sit at a desk for a extended period of time to operate a computer. Through the use of the program, individuals could now easily operate a PC's interface through the simple motion and gesture of their hand recorded by a video capture device connected to a computer from any location in their household.

CHAPTER 6

PREVIOUS APPROACHES

Hand gesture recognition for computer control is a popular research topic incomputer vision. The first research on in the area began in about 1992, when
it first became possible to grab images from a camera in real time, and thus enable effective human-computer interaction. Research in the area has typically fallen within the categories of applications for pointing, presenting, digital desktops, virtual workbenches, and virtual reality. In all cases, the motivation has been to create convenient, intuitive, powerful means of interaction with a computer that can serve as an alternative to the traditional keyboard and mouse.

Approaches to recognizing hand gestures have usually been divided into two types: model-based and view-based approaches. Model-based approaches use a 3D hand model for tracking, which has many more potential applications, but is a challenging task. The main problems are that it works well only for relatively slow hand motions, and in constrained environments (Stenger, 2006). View-based approaches, however, use pattern classification to identify hand features and determine the current hand pose. The main challenges in this case are to find how to segment the hand from the background, and how to determine which features to extract from the segmented region. Segmentation is often done using skin color detection and segmenting this from a uniform background.

For both approaches, tactics to help in recognizing hand gesture input include:

- using props or input devices, such as a pen or dataglove
- restricting the object information, as through a silhouette of the hand
- restricting the recognition situation, by using a uniform background
- restricting the set of gestures made, and using only one hand (Lenman, Bretzner & Thuresson, 2002)

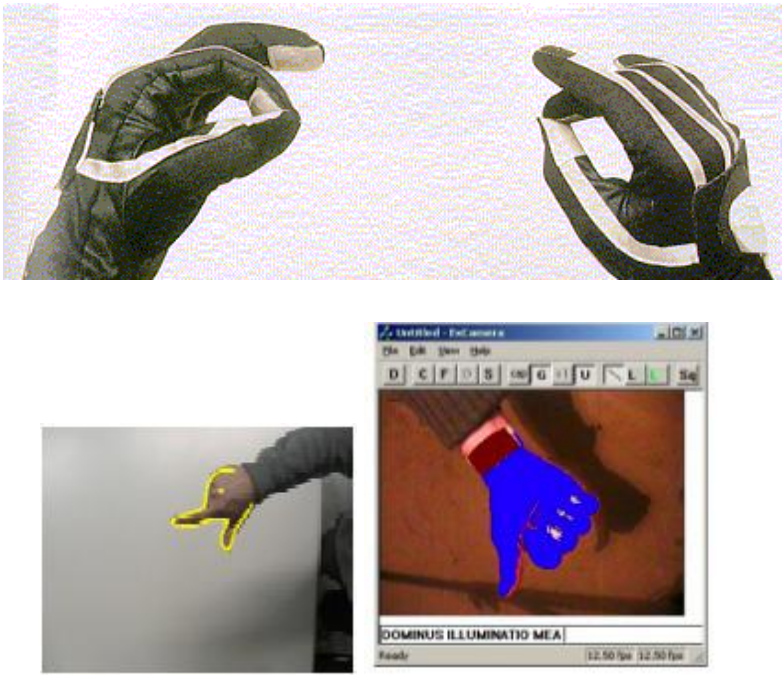


FIGURE 6.1: *Some previous research approaches, using data gloves, edge detection, and skin color segmentation.*

CHAPTER 7

CURSOR CONTROL

Several applications for cursor control have been created using 2D tracking. These methods all use a single camera to track a hand, and have typically used pointing gestures as the means of replacing the mouse. An early example is Finger Mouse (Quek, Mysliwjec & Zhao, 1995), which used a camera to view the hand from above and recognize hand gestures to control the mouse. One drawback of this system was that the user had to press the shift key with their other hand in order to 'click', making the system not completely gesture-based. Later applications found ways to solve this problem and create click gestures, such as O'Hagan (1997) who defined clicking as when two fingers were extended, and von Hardenberg and Berard (2001) who defined clicking as when a one second delay in hand movement

occurred. Research Goals and Objectives. The overall goal of this project was to develop a program implementing hand tracking and gesture recognition. The program will map the user's tracked hand motion to the computer's mouse cursor. The hand gesture will determine the state or action of the cursor. When a closed

fist hand gesture is placed in front of the video camera the cursor will act in a null state; however, when the closed fist transitions to an open hand a double left click action will be performed by the cursor on the tracked position of the hand. The user can vary the distance it places its hand from the camera with no effect on the ability of the program to perform tracking or gesture recognition. The user can also present the hand gesture at any degree of rotation in the x-y direction.

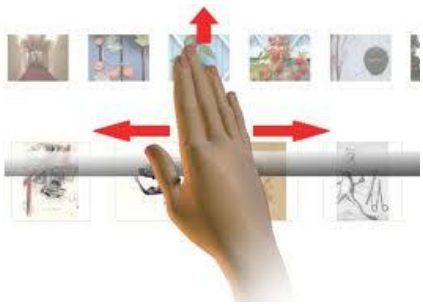


FIGURE 7.1

CHAPTER 8

FEATURES OF THE PROJECT

- This project will design and build a man-machine interface using a video camera to interpret one-handed sign language to control the mouse cursor of computer.

- Our interface can be used as a virtual mouse for controlling any Windows application.

- Hand gesture recognition is a subtopic in machine vision and it mainly aims to decrease the effort in human machine interactions. Although there are serious improvements for input devices in computer world, people still find the interaction uncomfortable. Especially moving, rotating, scrolling, zooming and selecting can be quite exhausting in long studies if they are needed to be used frequently.

CHAPTER 9

ALGORITHM

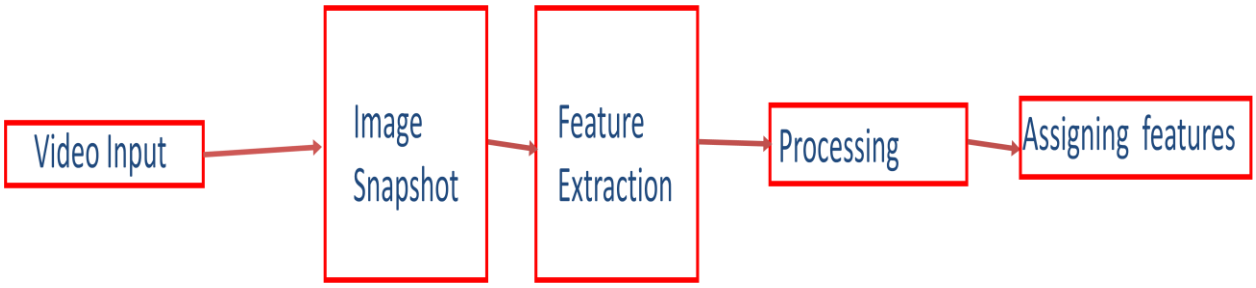


FIGURE 9.1

ALGORITHM

Video Input:

Taking video as input through webcam in Matlab. Inputing video in real time.

Snapshot:

Taking real time images in Matlab.(Snapshot)

Feature Extraction:

Extraction of colours from image the snapshot. Now if the part of image is of selected colour then it is taken as “1” and if the part is not of selected colour then it is taken as “0”. By this the image is converted into a Binary Image.

Processing:

Now the images are processed and noise is filtered out and we calculate the centroid of the detected region.

Assigning Features:

According to the position of centroid we found, it will be linked to the cursor position, left click, right click, double click and scrolling in the system.

CHAPTER 10

Code

```
import java.awt.Robot;
import java.awt.event.*
mouse = Robot;
%set threshold value for each color
 redthresh = 0.27;
 greenthresh = 0.14;
 bluethresh = 0.32;
%acquire camera name ,id
cam = imaqhwinfo;
cameraName = char(cam.InstalledAdaptors(end));
cameraInfo = imaqhwinfo(cameraName);
cameraId = cameraInfo.DeviceInfo.DeviceID(end);
cameraFormat =
char(cameraInfo.DeviceInfo.SupportedFormats(end));
%vid=videoinput('winvideo',1,'YUY2_640x480');
vid=videoinput(cameraName,cameraId,'YUY2_640x480');
set(vid,'ReturnedColorSpace','rgb');

preview(vid);
pause(2);

while 1
 im1=getsnapshot(vid);
 pause(0.05);
 %flip image for u
 %im1=flipdim(im1,2);
 im2=im1(:,:,1);
```

```
 ig2=im1(:,:,2);
 in2=im1(:,:,3);
 im3=rgb2gray(im1);
 im2=imsubtract(im2,im3);
 im2=medfilt2(im2);
 in2=imsubtract(in2,im3);
 in2=medfilt2(in2);
 ig2=imsubtract(ig2,im3);
 ig2=medfilt2(ig2);



im3 = im2bw(im2,redthresh);
in3 = im2bw(in2,bluethresh);
ig3 = im2bw(ig2,greenthresh);
imshow(flipdim(im3,2));
 pause(0.05);
 %detect boundaries
[B,L,N]= bwboundaries(im3,'noholes');
[P,Q,R]=bwboundaries(in3,'noholes');
[S,T,U]=bwboundaries(ig3,'noholes');
%find centroid
 a = regionprops(L,'centroid');
 b = regionprops(Q,'centroid');
 c = regionprops(T,'centroid');

 if N >= 1 && R==0 && U==0
  mouse.mouseMove(1600-
(a(1).Centroid(1,1)*(5/2)),(a(1).Centroid(1,2)*(5/2)-180));

 elseif N>=1 && R>=1  && U==0
        if (a(1).Centroid(1,1)-b(1).Centroid(1,1)>70)
```

```
                mouse.mousePress(16);
                pause(0.1);
                mouse.mouseRelease(16);
        elseif(a(1).Centroid(1,1)-b(1).Centroid(1,1)<-70)
                mouse.mousePress(4);
                pause(0.1);
                mouse.mouseRelease(4);
        end

 elseif N>=1 && R>=0  && U==1
                mouse.mousePress(16);
                pause(0.1);
                mouse.mouseRelease(16);
                pause(0.2);
                mouse.mousePress(16);
                pause(0.1);
                mouse.mouseRelease(16);

 elseif N>=1 && R>=0  && U==1
     if (c(1).Centroid(1,2)-b(1).Centroid(1,2)>2)
                mouse.mouseWheel(1);
                elseif(c(1).Centroid(1,2)-b(1).Centroid(1,2)>-2)
                mouse.mouseWheel(-1);
     end

 end
 flushdata(vid);
end
```

CHAPTER 11

FUTURE SCOPE

A gesture recognition system could be used in any of the following areas:

- Man-machine interface
- 3D animation
- Visualization
- Computer games
- Control of mechanical systems (such As robotics)

Man-machine interface

FIGURE 12.1

Through the use of gesture recognition, "remote control with the wave of a hand" of various devices is possible. The signal must not only indicate the desired response, but also which device to be controlled

Foregoing the traditional keyboard and mouse setup to interact with a computer, strong gesture recognition could allow users to accomplish frequent or common tasks using hand or face gestures to a camera. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building an interface between machines and humans than primitive text user interfacesor even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse.

Gesture recognition enables humans to communicate with the machine (HMI) and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at the computer screen so that the cursor will move accordingly. This could potentially make conventional input devices such as mouse, keyboards and even touch-screens redundant.

Gesture recognition can be conducted with techniques from computer vision and image processing.

The literature includes ongoing work in the computer vision field on capturing gestures or more general human pose and movements by cameras connected to a computer.

3D animation

FIGURE 12.2

Rapid and simple conversion of hand movements into 3D computer space for the purposes of computer animation.

Visualization

FIGURE 12.3

Just as objects can be visually examined by rotating them with the hand, so it would be advantageous if virtual 3D objects (displayed on the computer screen) could be manipulated by rotating the hand in space

Computer games

FIGURE 12.4

Using the hand to interact with computer games would be more natural for many applications.

Control of mechanical systems

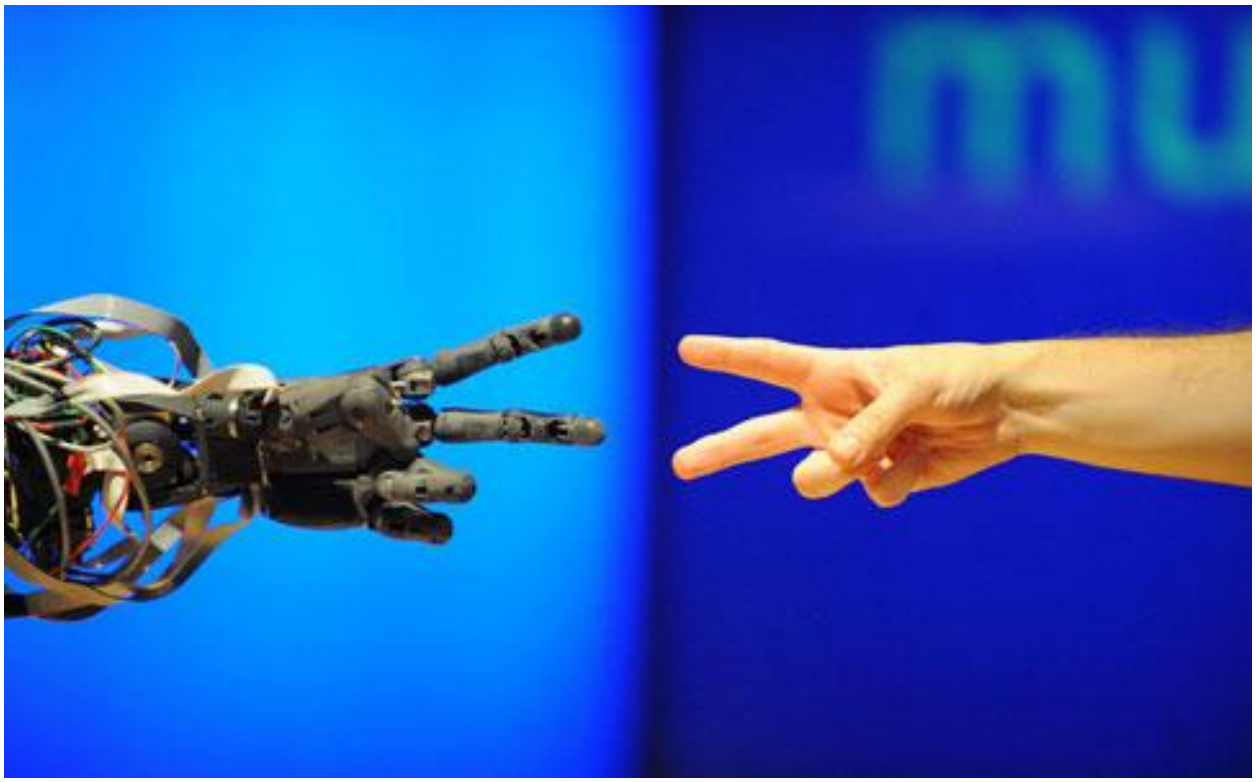


FIGURE 12.5

Using the hand to remotely control a manipulator. This concept can be used for controlling the movements of robots according to the gestures of the controller.

Conclusion

Hand gesture recognition system received great attention in the recent few years because of its manifoldness applications and the ability to interact with machine efficiently through human computer interaction.

In recent years a lot of research has been conducted in gesture recognition. The aim of this project was to develop an Gesture recognition system.

A primary goal is to create a system which can identify specific human gestures and use them to convey information or for device control.

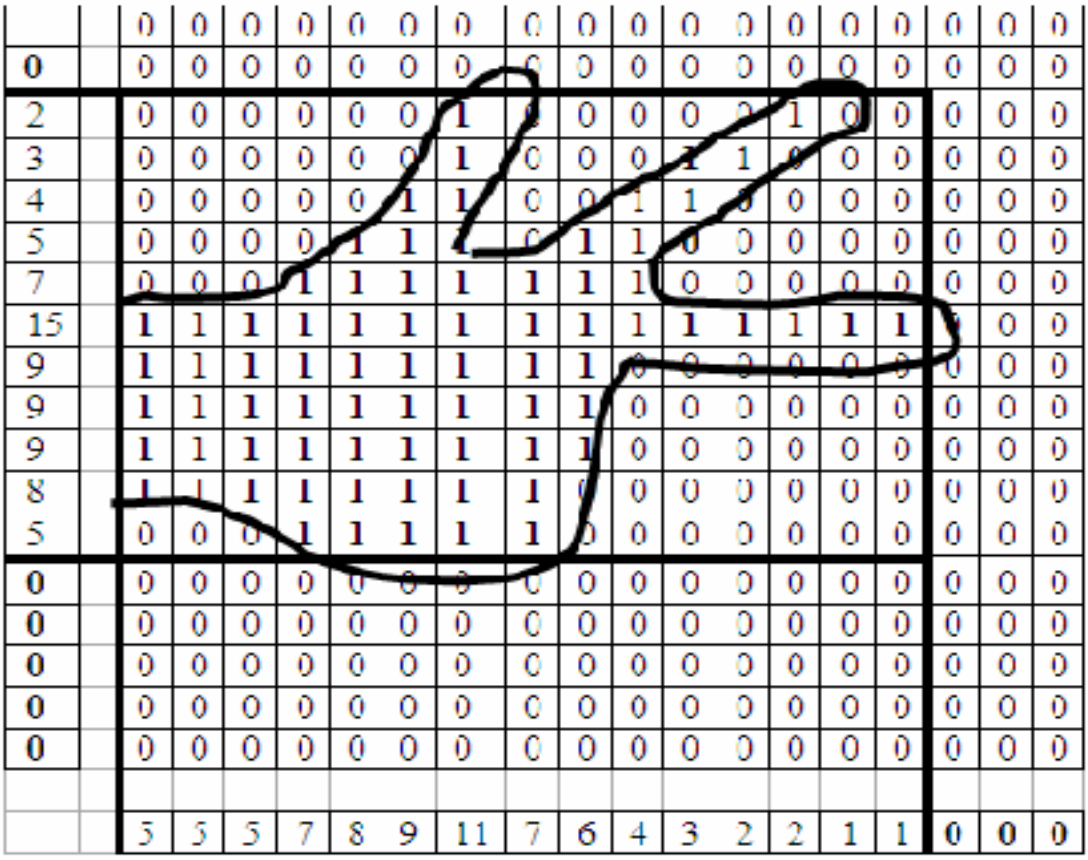


FIGURE 13.1

Till now we are successful in implementing the webcam with MATLAB and took current frame images by video acquisition.
From the images captured, we tracked the hand in real time and with the position of hand adjusted the location of mouse cursor.

- By adding more gestures, we can handle all My Computer operations like Cut, Copy, Paste and Undo etc.

- By integrating our system with voice recognition system we can embed it in ROBOTS.

- We can enhance our system to control PowerPoint application.

- We are also able to handle dynamic image processing and event handling accordingly.

REFERENCES

- Digital Image Processing
  - R. Gonzalez , R. Woods
- http://matlabnstuff.blogspot.in
- http://sprobotics.blogspot.in
- http://www.mathworks.in
- http://www.cmeri.res.in