**ABSTRACT**

Implementation of Gesture Recognition System aims at expanding the dimensions of gesture navigation to next level. The system works by providing a live video of hand as an input, which then analyses the position and angle to judge which task is to be performed. Cursor movements and clicking operations can be easily performed by hand gestures.

It is an easy way to interact with the system and comfortable as well. You do not have to go near to your display to control the cursor and perform clicking operations you can just sit back and perform your task.

**Introduction**

# **Introduction**

Laptops, Desktops and other windows and android devices which uses cursor has become a significant part of our everyday lives. They contains different kinds of applications and our personal data like pictures, mails, messages, remainders etc.

This system can help in different aspects and does not require any additional hardware; just with a webcam, we can implement Gesture Recognition system. The only thing it requires is a software, which is compatible with the platform we are working on.

Now we are even using smart television which supports different types of operating systems like windows and android and they support web cams therefore Gesture Recognition is a very useful technology for them as when we use remotes they are very uncomfortable as speech recognition support is only for few applications otherwise we have to type everything using remote that's kind of very time consuming and boring.

## **1.1 Gesture Recognition**

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field include emotion recognition from face and hand gesture recognition. Users can use simple gestures to control or interact with devices without physically touching them. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques. 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.

**Introduction to Computer vision**

**Introduction to Computer Vision**

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

**2.1 Applications of Computer Vision**

Applications range from tasks such as industrial machine vision systems which, say, inspect bottles speeding by on a production line, to research into artificial intelligence and computers or robots that can comprehend the world around them. The computer vision and machine vision fields have significant overlap. Computer vision covers the core technology of automated image analysis which is used in many fields. Machine vision usually refers to a process of combining automated image analysis with other methods and technologies to provide automated inspection and robot guidance in industrial applications. In many computer vision applications, the computers are pre-programmed to solve a particular task, but methods based on learning are now becoming increasingly common. Examples of applications of computer vision include systems for:

* Automatic inspection, e.g., in manufacturing applications;
* Assisting humans in identification tasks, e.g., a species identification system;
* Controlling processes, e.g., an industrial robot;
* Detecting events, e.g., for visual surveillance or people counting;
* Interaction, e.g., as the input to a device for computer-human interaction;
* Modeling objects or environments, e.g., medical image analysis or topographical modeling
* Navigation, e.g., by an autonomous vehicle or mobile robot; and
* Organizing information, e.g., for indexing databases of images and image sequences.

**2.2 Typical Tasks**

Each of the application areas described above employ a range of computer vision tasks; more or less well-defined measurement problems or processing problems, which can be solved using a variety of methods. Some examples of typical computer vision tasks are presented below.

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

**2.2.1 Recognition**

The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. Different varieties of the recognition problem are described in the literature:

Object recognition (also called object classification) – one or several pre-specified or learned objects or object classes can be recognized, usually together with their 2D positions

* in the image or 3D poses in the scene. Blippar, Google Goggles and LikeThat provide stand-alone programs that illustrate this functionality.
* Identification – an individual instance of an object is recognized. Examples include identification of a specific person's face or fingerprint, identification of handwritten digits, or identification of a specific vehicle.
* Detection – the image data are scanned for a specific condition. Examples include detection of possible abnormal cells or tissues in medical images or detection of a vehicle in an automatic road toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interesting image data which can be further analyzed by more computationally demanding techniques to produce a correct interpretation.

Currently, the best algorithms for such tasks are based on convolutional neural networks. An illustration of their capabilities is given by the ImageNet Large Scale Visual Recognition Challenge; this is a benchmark in object classification and detection, with millions of images and hundreds of object classes. Performance of convolutional neural networks, on the ImageNet tests, is now close to that of humans. The best algorithms still struggle with objects that are small or thin, such as a small ant on a stem of a flower or a person holding a quill in their hand. They also have trouble with images that have been distorted with filters (an increasingly common phenomenon with modern digital cameras). By contrast, those kinds of images rarely trouble humans. Humans, however, tend to have trouble with other issues. For example, they are not good at classifying objects into fine-grained classes, such as the particular breed of dog or species of bird, whereas convolutional neural networks handle this with ease.

Several specialized tasks based on recognition exist, such as:

* Content-based image retrieval – finding all images in a larger set of images which have a specific content. The content can be specified in different ways, for example in terms of similarity relative a target image (give me all images similar to image X), or in terms of
* high-level search criteria given as text input (give me all images which contains many houses, are taken during winter, and have no cars in them).



Figure 1: Content Based Retrieval [1]

* Pose estimation – estimating the position or orientation of a specific object relative to the camera. An example application for this technique would be assisting a robot arm in retrieving objects from a conveyor belt in an assembly line situation or picking parts from a bin.
* Optical character recognition (OCR) – identifying characters in images of printed or handwritten text, usually with a view to encoding the text in a format more amenable to editing or indexing (e.g. ASCII).
* 2D Code reading Reading of 2D codes such as data matrix and QR codes.
* Facial recognition
* Shape Recognition Technology (SRT) in people counter systems differentiating human beings (head and shoulder patterns) from objects

**2.2.2 Motion analysis**

Several tasks relate to motion estimation where an image sequence is processed to produce an estimate of the velocity either at each points in the image or in the 3D scene, or even of the camera that produces the images. Examples of such tasks are:

* **Egomotion** – determining the 3D rigid motion (rotation and translation) of the camera from an image sequence produced by the camera.
* **Tracking** – following the movements of a (usually) smaller set of interest points or objects (e.g., vehicles or humans) in the image sequence.
* **Optical flow** – to determine, for each point in the image, how that point is moving relative to the image plane, i.e., its apparent motion. This motion is a result both of how the corresponding 3D point is moving in the scene and how the camera is moving relative to the scene.

**2.2.3 Scene reconstruction**

Given one or (typically) more images of a scene, or a video, scene reconstruction aims at computing a 3D model of the scene. In the simplest case the model can be a set of 3D points. More sophisticated methods produce a complete 3D surface model. The advent of 3D imaging not requiring motion or scanning, and related processing algorithms is enabling rapid advances in this field. Grid-based 3D sensing can be used to acquire 3D images from multiple angles. Algorithms are now available to stitch multiple 3D images together into point clouds and 3D models.

**2.2.4 Image restoration**

The aim of image restoration is the removal of noise (sensor noise, motion blur, etc.) from images. The simplest possible approach for noise removal is various types of filters such as low-pass filters or median filters. More sophisticated methods assume a model of how the local image structures look like, a model which distinguishes them from the noise. By first analysing the image data in terms of the local image structures, such as lines or edges, and then controlling the filtering based on local information from the analysis step, a better level of noise removal is usually obtained compared to the simpler approaches. An example in this field is inpainting.

**2.3 System Methods**

The organization of a computer vision system is highly application dependent. Some systems are stand-alone applications which solve a specific measurement or detection problem, while others constitute a sub-system of a larger design which, for example, also contains sub-systems for control of mechanical actuators, planning, information databases, man-machine interfaces, etc. The specific implementation of a computer vision system also depends on if its functionality is pre-specified or if some part of it can be learned or modified during operation. Many functions are unique to the application. There are, however, typical functions which are found in many computer vision systems.

**Image acquisition** – A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or colour images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

**Pre-processing** – Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method. Examples are

* Re-sampling in order to assure that the image coordinate system is correct.
* Noise reduction in order to assure that sensor noise does not introduce false information.
* Contrast enhancement to assure that relevant information can be detected.
* Scale space representation to enhance image structures at locally appropriate scales.
* Feature extraction – Image features at various levels of complexity are extracted from the image data. Typical examples of such features are
* Lines, edges and ridges.
* Localized interest points such as corners, blobs or points.
* More complex features may be related to texture, shape or motion.

**Detection/segmentation** – At some point in the processing a decision is made about which image points or regions of the image are relevant for further processing. Examples are

* Selection of a specific set of interest points
* Segmentation of one or multiple image regions which contain a specific object of interest.
* Segmentation of image into nested scene architecture comprised foreground, object groups, single objects or salient object parts (also referred to as spatial-taxon scene hierarchy)

**High-level processing** – At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example:

* Verification that the data satisfy model-based and application specific assumptions.
* Estimation of application specific parameters, such as object pose or object size.
* Image recognition – classifying a detected object into different categories.
* Image registration – comparing and combining two different views of the same object.
* Decision making Making the final decision required for the application, for example:
* Pass/fail on automatic inspection applications
* Match / no-match in recognition applications
* Flag for further human review in medical, military, security and recognition applications

**2.4 Hardware**

There are many kinds of computer vision systems, nevertheless all of them contain these basic elements: a power source, at least one image acquisition device (i.e. camera, ccd, etc.), a processor as well as control and communication cables or some kind of wireless interconnection mechanism. In addition, a practical vision system contains software, as well as a display in order to monitor the system. Vision systems for inner spaces, as most industrial ones, contain an illumination system and may be placed in a controlled environment. Furthermore, a completed system includes many accessories like camera supports, cables and connectors.

Most computer vision systems use visible-light cameras passively viewing a scene at frame rates of at most 60 frames per second (usually far slower).

**Feasibility study**

**Feasibility**

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

* What are the user’s demonstrable needs and how does a candidate system meet them?
* What resources are available for given candidate system?
* What are the likely impacts of the candidate system on the organization?
* Whether it is worth to solve the problem?

During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this.

**3.1 Steps in feasibility analysis:**

The steps involved in the feasibility analysis are:

* Form a project team and appoint a project leader
* Prepare system flowcharts.
* Enumerate potential proposed system.
* Define and identify characteristics of proposed system.
* Determine and evaluate performance and cost effective of each proposed system.
* Weight system performance and cost data.
* Select the best-proposed system.
* Prepare and report final project

**3.2** **Technical feasibilty**

This is a study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. The key questions to be addressed are:

* Can the work for the project be done with current equipment existing software technology & available personnel.
* Can the system be upgraded if further developments occur and requirements increase.
* If new technology is needed then what can be developed.

Technical feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirement.

**3.3 Operational feasibilty**

It is mainly related to human organizations and political aspects. The points to be considered are:

* What changes will be brought with the system?
* What organization structures are disturbed?
* What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

This system is operationally feasible as it very easy for the end users to operate it. It only needs basic information about the Windows platform.

**System design and working**

**Aim**

The Aim of the system is to locate a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality. Video tracking can be a time consuming process due to the amount of data that is contained in video. Adding further to the complexity is the possible need to use object recognition techniques for tracking, a challenging problem in its own right.

**4.1 Objective**

The objective of video tracking is to associate target objects in consecutive video frames. The position and the orientation of the tracked object can be used to develop mathematical models to perform the movement of the cursor, clicking operations, typing, dragging etc.

**4.2 Algorithms**

To perform tracking an algorithm analyzes sequential video frames and outputs the movement of targets between the frames. There are a variety of algorithms, each having strengths and weaknesses. Considering the intended use is important when choosing which algorithm to use. There are two major components of a visual tracking system: target representation and localization, as well as filtering and data association.

**Target representation and localization** is mostly a bottom-up process. These methods give a variety of tools for identifying the moving object. Locating and tracking the target object successfully is dependent on the algorithm. For example, using blob tracking is useful for identifying human movement because a person's profile changes dynamically. Typically the computational complexity for these algorithms is low.

**The following Algorithm has been used in the system to perform localization and tracking operations:**

**4.2.1 Suzuki contour and tracking algorithm:**

Contour tracking is the detection of object boundary (e.g. active contours or Condensation algorithm). Contour tracking methods iteratively evolve an initial contour initialized from the previous frame to its new position in the current frame.

Let I be a binary digital image with M×N pixels, where the coordinate of the top-leftmost pixel is (0,0)and that of the bottom-rightmost pixel is (M−1,N−1). In I, a pixel can be represented as P=(x,y),x=0,1,2,⋯,M−1,y=0,1,2,⋯,N−1. Most contour-tracing algorithms use a tracer T(P,d) with absolute directional information d∈{N,NE,NW,W,SW,S,SE,E,NE}, and they have the following basic sequence:

1. The tracer starts contour tracing at the contour of an object after it saves the starting point along with its initial direction.
2. The tracer determines the next contour point using its specific rule of following paths according to the adjacent pixels and then moves to the contour point and changes its absolute direction.
3. If the tracer reaches the start point, then the trace procedure is terminated.

To determine the next contour point, which may be a contour pixel or pixel corner, the tracer detects the intensity of its adjacent pixel Pr and the new absolute direction dr for Pr by using relativedirectioninformationr∈{front,front−left,left,rear−left,rear,rear−right,right,r∈{front−right}.

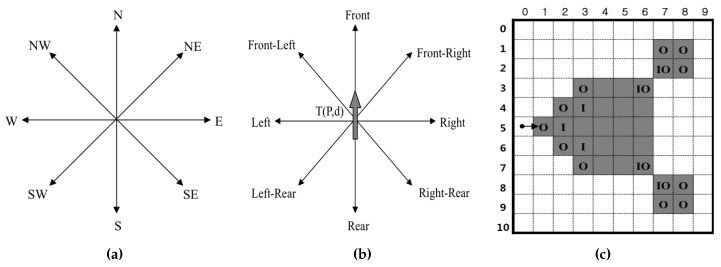


Figure 2: Directions and types of contour pixels. (a) Absolute direction d; (b) relative direction r; (c) types of contour pixels: inner corner pixel (I), outer corner pixel (O) and inner-outer corner pixel (IO).[2]

* 1. **Working:**
* **STEP 1:** Convert image from BGR(Blue, Green, Red) to HSV(Hue, Saturation, Value)



Figure3: BGR color space on right and HSV on left.[3]

The HSV image is also known as Threshold image. The transformation is done by using filters with white space representing the desired image to be tracked.

* **STEP 2:** Filter the color of interest between a MIN and MAX Threshold

What is Thresholding?

The simplest segmentation method

Application example: Separate out regions of an image corresponding to objects which we want to analyze. This separation is based on the variation of intensity between the object pixels and the background pixels.

To differentiate the pixels we are interested in from the rest (which will eventually be rejected), we perform a comparison of each pixel intensity value with respect to a threshold (determined according to the problem to solve).

Once we have separated properly the important pixels, we can set them with a determined value to identify them (i.e. we can assign them a value of (black), (white) or any value that suits your needs).

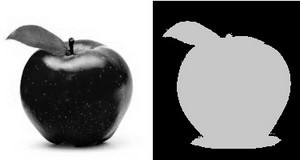


Figure 4: Thresholding operations[4]

* **STEP 3:** Performing Morphological Operations

Two morphological operations are performed: DILATE and ERODE

1. **ERODE:** “Erodes” into white space. Making it smaller or non existent
2. **DILATE:** “Dilates” white space. Making it larger.

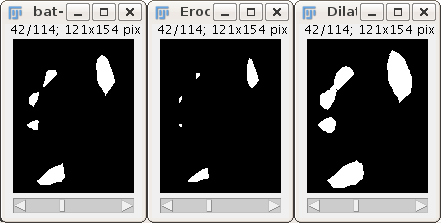


Figure 5: Erode and Dilute operations[5]

* **STEP 4:** Find Contours
* **INPUT :** Binary Image
* **OUTPUT:** A vector of Contours

Moments Method

* **INPUT:** Vector of Contours
* **OUTPUT:** Coordinates of Largest Contour.

**4.4 Motivation:**

The motivation for this project lies in developing an environment which could be capable of controlling the machines with hand gestures .Gestures, in general, could be hand gestures or face gestures. By recognizing various gestures on the screen, desired operations can be performed.

**4.5 System Modules**:

The implementation of gesture recognition system can be divided into 3 major parts:

1. Segmenting image from the rest of the environment (sensing):

* This module contains steps for segmenting the image from the environment. The segmentation of the image is performed by converting the RGB image to HSV images, which is suitable for intensity varying images.
* V in HSV stands for Value which is a measure for intensity values of same color.
* Various Thresholds are provided for segmenting a particular wavelength from a set of wavelengths by using opencv’s inrange() function
* The inrange() works by assigning a minimum and a maximum values to the Hue, Saturation , and values and assigning it to the particular image under working.
* The images are then converted into black and white binary images, which reduces the overhead of processing three different matrices.
* Various morphological operations are also performed which assists in segmentation of the images and providing a noise free image.
* Once the image has been segmented, a pure white image of the hand is observed which is ready to be tracked.

1. Image Processing (processing):

* The binary image is then used to perform the necessary operations.
* The binary image is operated upon by counter tracking algorithm.
* The algorithm works by storing and updating the pixel values and the corresponding locations of the pointer.
* After each step, the surrounding pixels are evaluated for pixel values and again an updation to the locations is performed.
* The procedure is continued until all the pixel values are circled once and a complete counter has been tracked.
* The counter has been given a single color so that it distinguishes from the other objects in the surroundings
* A convex hull, which is the smallest set of points possible to cover the counter completely, has been drawn.
* A convex hull is basically a set of points such that, a line drawn between any two points in this set will completely lie inside the set.
* This technique helps in finding out the defect points in the graph.
* The defect points are the points, which define the gap between the convex hull and the counter of the tracking object.
* An array of Fingertips has been defined which stores the number of such defects calculated foe the particular counter.

Number of Fingertips = defect points of the counter traced

* Hence, the fingertips array is basically storing a set of defect points in the counters.
* Once the fingertips and the centre of the counter has been successfully calculated, a line connecting these two points is drawn in order to clearly show how many defect points are actually there on the screen.
* Angles between lines are also measured in case the number of fingertips is equal to one.
* This angle measurement helps in performing either the left click operation, or the right click operation.

1. The actual working (Actuators):

* The coordinates of the tracked object are stored in an vector type array which guides the cursor movement operations of the cursor.
* The clicking operations can be performed by noting down the angles between the lines, and invoking a particular function on fulfillment of certain conditions.
* We have used a NoofFingertips() function in order to assign different functionalities according to the number of fingertips visible on the screen.
* Various other functionalities can also be deployed by invoking Microsoft’s events such as, MOUSEEVENTF\_SCROLL or MOUSEEVENTF\_LEFTDOWN.

**4.6 Features:**

* Cursor movement operations can be done
* Left click operation can be performed
* Right click operations can be performed
* The system can work in dim or no light
* Can be used to operate the system without physically touching it

**Software requirement specification**

**Software Requirement Specification**

Software Requirement Specification (SRS) is the starting point of the software development activity. It is a complete description of the behavior of a system which is to be developed. The SRS document enlists all necessary requirements for project development. To derive the requirements we need to have clear and thorough understanding of the product which is to be developed. This is prepared after detailed communication with project team and the customer.

A SRS is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations.

**5.1 Purpose**

The purpose of this document is to describe the external behaviour of the System Requirments specifications

SRS defines and describes the operations, interfaces, performance, and quality assurance requirements of the System. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The objective of this project is to provide a friendly environment to maintain the data and to predict the future data for an organization.

**5.2 Characteristics of SRS:**

* **Correct -** An SRS is correct if, and only if, every requirement stated therein is one that the software shall meet. Traceability makes this procedure easier and less prone to error.
* **Unambiguous** - An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. As a minimum, this requires that each characteristic of the final product be described using a single unique term.
* **Verifiable** – It is verifiable if there exists some finite cost-effective process with which a person or machine check whether software product meets requirements.
* **Consistent** - Consistency refers to internal consistency. If an SRS does not agree with some higher-level document, such as a system requirements specification, then it is not correct. An SRS is internally consistent if, and only if, no subset of individual requirements described in it conflict.
* **Modifiable** – SRS is said to be modifiable if its structure and style are such that any changes to the requirements can be made easily, completely and consistently while retaining the structure and style.
* **Traceable** – SRS is said to be traceable if the origin of each of its requirements is clear and it facilitates the referencing of each requirement in future enhancement.
* **Ranked for importance or stability** – SRS is ranked for importance or stability if each requirement in it has an identifier to indicate either the importance or stability of that particular requirement.
  1. **Functional Requirements**

Gesture Recognition

* Able to recognize images.
* Various clicking and movement operations can be performed.
* Real time analysis of images could be done
* No need of mouse or keyboard anymore.

## **5.4 Non Functional Requirements**

**Performance Requirements:**

Application must respond within milliseconds. The user must use the required option to get the information of the users.

**Availability:**

The application can be used by anyone with a little knowledge of images

**Maintainability:**

Since we are using C++ software to support our application no maintenance is very easy and economical also.

**Portability:**

The project is built using C++ and can be run on any device which uses C++ environment.

**Safety Requirements:**

It is better to use the antivirus and keep on checking for the latest updates of the application.

**Security Requirements:**

The application will prompt the user for upgrading and downloading new features updated by the developer.

**5.5 Overall description**

* **Planned approach towards working**: The working in the organization will be well planned and organized. The data will be stored efficiently with optimal disk space consumption in data stores which will help in retrieval of information as well as its storage under resource constraints.
* **Accuracy**: The level of accuracy in the proposed system will be higher. All operations would conform to integrity constraints and correctness and it will be ensured that whatever information is received at or sent from the centre is accurate.
* **Reliability**: The reliability of the proposed system will be high due to the above mentioned reasons. This comes from the fact that only the data which conforms to the accuracy clause would be allowed to commit back to the disk.
* **No redundancy**: In the proposed system it will be ensured that no repetition of information occurs; neither on a physical storage nor on a logical implementation level. This economizes on resource utilization in terms of storage space. Also even in case of concurrent access no anomalies occur and consistency is maintained. In addition to all this, principles of normalization have been endeavored to be followed.
* **Ease of operation**: The system should be simplistic in design and use. It is such that it can be easily developed within a short period of time and can conform to the financial and resource-related constraints of the organization.

**5.6 Phases of software development**

Software Engineering deals with various tools, methods and procedures required for controlling the complexity of software development, project management and its maintenance. Object-oriented development emphasizes on using programming languages with certain unique capabilities for real world object modeling. Object model is the conceptual framework for object-oriented development.

The four major elements of this model are Encapsulation, Abstraction, Modularity and Hierarchy.

Software systems pass through two principal phases during their lifecycle.

* The development phase
* The operation and maintenance phase

Software development passes through various phases. They include

**Program Definition:** The first stage in the development process is understanding the problem in question and its requirements. Requirements include the context in which the problem arouses, functionality expected from the system and system constraints.

**Analysis:** Analysis phase delivers requirement specification. The system specification serves as an interface between the design and the implementer as well as between the implementer and the user.

**Design:** Design is the process of mapping system requirements defined during analysis to an abstract representation of a specific system implementation. Since the whole system may be complex the main design objective is decomposition. The system is divided into modules and their interactions. The modules may be then further decomposed into sub modules and procedures until each module can be implemented easily.

**Coding or Implementation:** Once the specification and design of the software is over, the choice of the programming language remains as one of the most critical aspect in producing reliable software.

**Testing:** Testing is the process of evaluating a system or system components by manual or automated means to verify that it satisfies the specified requirements.

**5.7 System requirements**

### **5.7.1** **HARDWARE REQUIREMENTS**

**On Developer Side**

Processor : Dual core or above.

RAM : 4 GB

Hard disk : 40 GB or above.

Keyboard : NOT REQUIRED ANYMORE

**On Client Side**

Device : C++ Enabled environment

### **5.7.2 SOFTWARE REQUIREMENTS**

Development Kit : Visual Studio Express, openCV

Languages : C++

Platform : Windows

**5.8 UML diagrams**

**5.8.1 Use case diagram**

35 | P a g e
12 UML Diagrams
Use Case Diagram
Figure 12.1 Use case diagram
 

Figure 6: Use case diagram

**5.8.2 Activity diagram**

37 | P a g e
Flow Diagram
Figure 12.3 Flow diagram
 

Figure 7: Activity diagram

**Conclusion & future scope**

**Conclusion and future scope**

A gesture recognition system has been created which controls mouse movements and clicking operations with the help of contour tracking algorithm. Movement of hand has been recorded and the image has been processed and filtered to specify the gestures. The cursor movements are performed using the coordinates of the counter centre on the window. As the centre moves, the coordinates of the centre point are updated and the corresponding movements of mouse cursor are performed. The counter tracking algorithm is necessary to segment a particular section of the image from the others. This helps the system in tracking only that object which is being countered. Binary images on the other hand, provides efficient execution environment as binary images are easily stored and the overload of processing three different matrices (red, green, blue) is reduced.

This technology is the future. Infact its already been started to implemented in some gaming consoles such as PS4 & XBox 360. As we are living in a Modern era age of smart televisions with powerful operating systems such as Android, IOS, Windows etc. are at our doorsteps so this gesture recognition can help in controlling these devices.

The gesture Recognition System could be further extended to autonomous robots where the goal is to track a particular object in a particular environment. It could be used with underwater diving robots in order to track a specific ancient stone, or any other artifacts. It could also be used in night vision cameras in order to track objects in night. Appropriate changes to the hardware may be done to provide these functionalities.

**References**

**References**

[1] Content Retrieval

- https://en.wikipedia.org/wiki/File:SRT\_Shape\_Recognition\_Technology.png

[2] Directions and types of contour pixels.

- https://www.researchgate.net?Pixels-b-The-Shape-and-Corner-Codes-of-its-Contour-Corners.png

[3] BGR color space on right and HSV on left.

- <https://i.stack.imgur.com/L6AmO.png>

[4] Thresholding operations

- http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html

[5] Erode and Dilute operations

- http://imagej.net/3D\_Binary\_Filters

[6] Suzuki Contour Tracking Algorithm

- http://www.imageprocessingplace.com/downloads\_V3/root\_downloads/tutorials/contour\_tracing\_Abeer\_George\_Ghuneim/alg.html

[7] Object Detection, Fundamentals and Case Studies

- **Bennamoun**, M., **Mamic**, G.J.

[8] Artificial Intelligence: A Modern Approach

* Stuart J. Rusell, Peter Norving

[9] The Elements of Statistical Learning

* Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie

[10] Foundations of Machine Learning

* Afshin Rostamizadeh, Ameet Talwalkar, and Mehryar Mohri

[11] Digital Image Processing

**-** Rafael C. Gonzalez

[12] Digital Image Processing

**-** Jayaraman

[13] Real-world Machine Learning

- Henrik Brink, Joesph W. Richards, and Mark Fetherolf

[14] Image Restoration

- http://nptel.ac.in/courses/117104069/chapter\_9/9\_1.html

[15] Scene Reconstruction

- http://vision.in.tum.de/research/imagebased\_3d\_reconstruction/multiviewreconstruction

[16] Morphological Operations

- https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm

[17] Motion Analysis

- https://www.kennedykrieger.org/research-training/centers-labs-cores/motion-analysis-laboratory

[18] Computer Vision

- https://hci.iwr.uni-heidelberg.de/compvis

[19] RGB Color Model

- http://whatis.techtarget.com/RGB-red-green-and-blue

[20] HSV Color Model

- http://infohost.nmt.edu/tcc/help/pubs/colortheory/web/hsv.html

# 



Fig-8 Image Segmentation

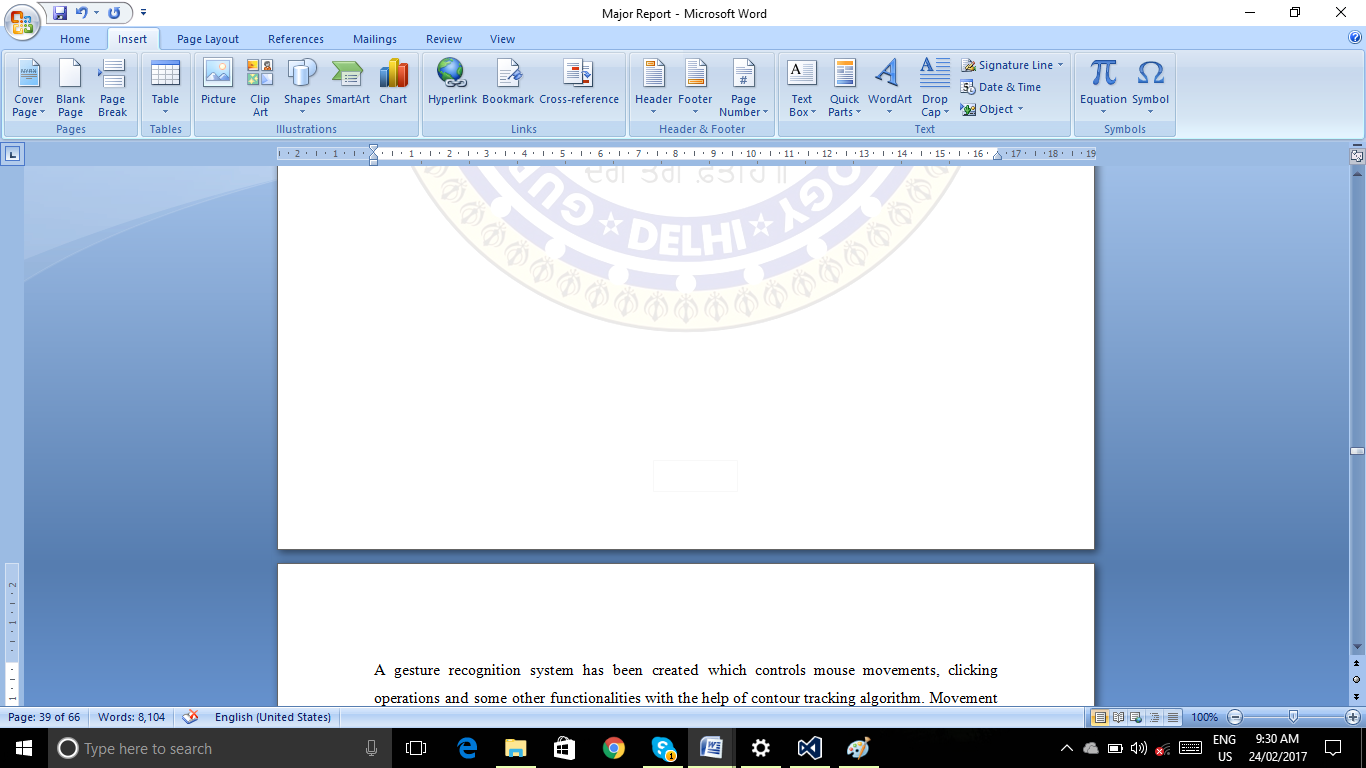


Fig-9 Scroll down

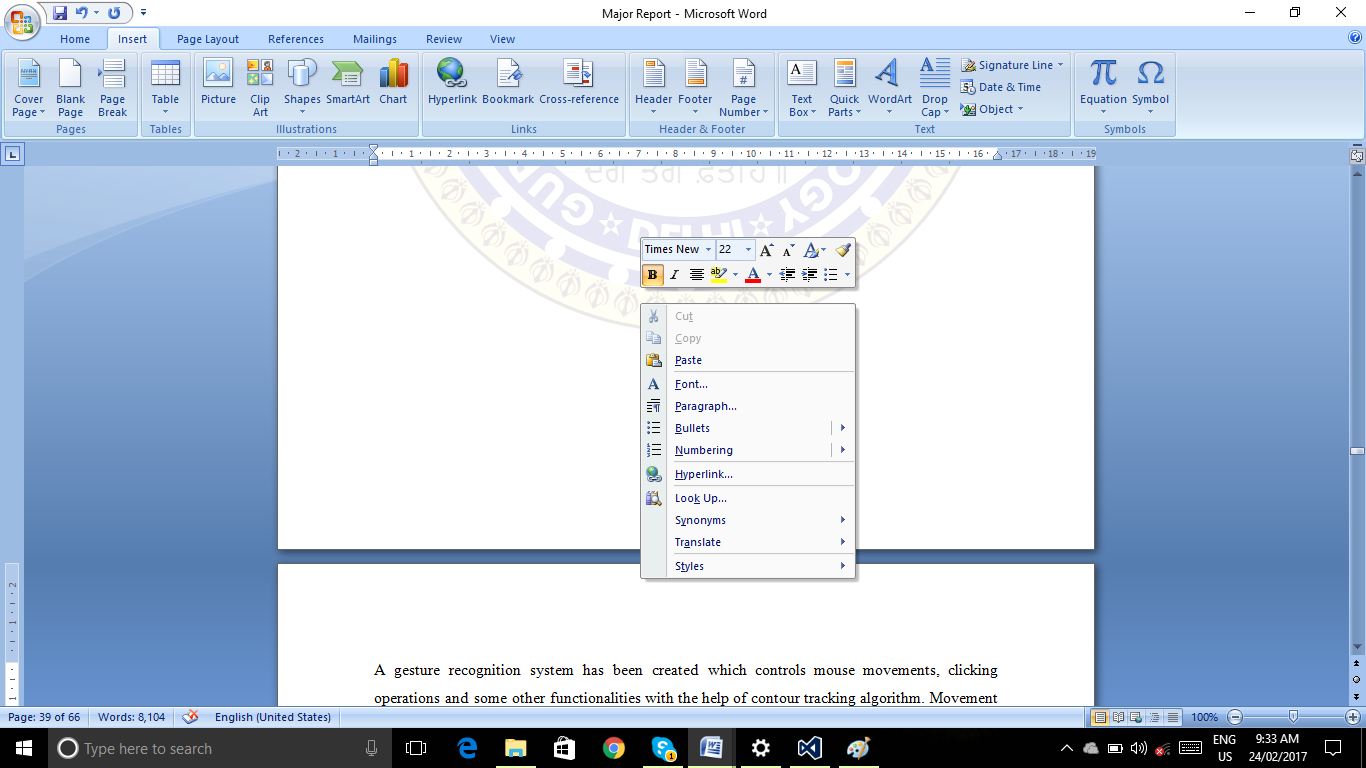
****

Fig-10 Right click