Ease of Accessibility Using Microsoft Kinect

By
Patel Prashantkumar Navinchandra
(120170107010)
Solanki Rom Prakash
(120170107002)
Gopani Vatsal Biren
(120170107054)

DEPARTMENT OF COMPUTER ENGINEERING
VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE CHANDKHEDA

Ease of Accessibility Using Microsoft Kinect

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Engineering

By
Patel Prashantkumar Navinchandra
(120170107010)
Solanki Rom Prakash
(120170107002)
Gopani Vatsal Biren
(120170107054)

DEPARTMENT OF COMPUTER ENGINEERING
VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE CHANDKHEDA

Declaration

This is to certify that

i. The project comprises my/our original work towards the degree of bachelor of Engineering in Computer Engineering at Vishwakarma Government Engineering College, Chandkheda, under the Gujarat Technological University, Ahmedabad and has not been submitted elsewhere for a degree.

ii. Due acknowledgement has been made in the text to all other material used.

- Patel Prashantkumar Navinchandra (120170107010)
- Solanki Rom Prakash (120170107002)
- Gopani Vatsal Biren (120170107054)

Certificate

This is to certify that the Project entitled "Ease of accessibility using Microsoft Kinect" submitted by following student**s**, towards the partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Engineering of Vishwakarma Government Engineering College, Chandkheda, under the Gujarat Technological University, Ahmedabad, is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

1. Patel Prashantkumar Navinchandra (120170107010)
2. Solanki Rom Prakash (120170107002)
3. Gopani Vatsal Biren (120170107054)

External Guide

Name: Karan J. Pujara

Designation: CEO

Organization: Creative Cube Infoweb Ltd. (Studentdesk.in)

Internal Guide

Name: Prof. Uday A. Yadav

Designation: Assistant Professor

Organization: Vishwakarma College

Head of Department

Prof. M.T.Savaliya,

Associate Professor

Computer Engineering Department,

Vishwakarma government engineering college,

Chandkheda, Ahmedabad.

Signature of External Examiner

Abstract

Brining you a new technology that moves one step forward from Graphical User Interface and provides gesture recognition as a key feature for the control of computers instead of conventional hardware devices like mouse or keyboard. GUI has made user experience more convenient than the previous CLI but technology keeps advancing and so is the need to bring new and exciting as well as amusing ways of accessing such resources that ease the life of those using it. We introduce you a new way of controlling computer with gestures that enable you to control the mouse with the movement of your hands and access various computer applications using gesture recognition. We are using Microsoft Kinect which is a gaming device that uses some of core fundamentals of various computer vision algorithms to captures gestures more efficiently. With the help of the device's powerful hardware features and SDK, we can use gesture recognition and its depth sensing capabilities to control various hardware features such as mouse (track pad), keyboard and touch interface. This system relieves you from the orthodox use of hardware resources and provides an innovative and enjoyable way of using computer with the palm of your hand which will be first of its kind.

Acknowledgement

I would like to place on record my deep sense of gratitude to Prof. M.T. Savaliya, HOD-Dept. of Computer Engineering, Vishwakarma Government Engineering College, Gandhinagar for his generous guidance, help and useful suggestions.

I express my sincere gratitude to Prof. Uday A. Yadav, Assistant Professor Dept. of Computer Engineering, Vishwakarma Government Engineering College, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

I am extremely thankful to Mr. Karan J. Pujara, CEO, Creative Cube Infoweb Pvt. Ltd. (Studentdesk.in), Ahmedabad, for providing me infrastructural facilities to work in, without which this work would not have been possible.

- Patel Prashantkumar Navinchandra (120170107010)
- Solanki Rom Prakash (120170107002)
- Gopani Vatsal Biren (120170107054)

Contents

1. Introduction

1.1 Project summary

With the intension to increase the accessibility of the current system and ease of accessing the computer with the help of Microsoft Kinect we provide a way of controlling the computer with Gesture recognition as a key feature. With the help of the device's powerful hardware features and SDK, we can use gesture recognition and its depth sensing capabilities to control various hardware features such as mouse (track pad), keyboard and touch interface. We provide a way of controlling mouse movement with the motion of our hand and accessing various computer applications by gesture recognition. It will also improve significant control for those who have disability in using their fingers properly. We intent to build a system where using Microsoft Kinect, we will make an interface to control the computer. The system will be first of its kind and has the capability to detect full body gestures to control various hardware functions mentioned above.

1.2 Objective

Development in technology is intended to make daily life easier. Computers are part of our daily life and hence improving the use of computer will ultimately make the life easy. One way to accomplish this is to improve the user interface. In the beginning of the computer era, only command line interface was there which was not user friendly. Then Graphical user interface evolved and that brought the computer to the use for general people. Now it is time to change it again by gesture user interface. Now to implement this, we are using Microsoft Kinect which is a device invented for gaming. It has two cameras 1. IR camera and 2. RGB camera. By use of these cameras and its own image processing capabilities, we can take the user interface to another level. We are implementing the system to recognize full body 3D gestures and then use it to control the computer components such as mouse and keyboard.

We intend to use the images provided by the devices for training of various gestures and using such gestures for accessing various computer application. Moreover we also focused on using the real time controlling of mouse tracking based on hand movements using device's image processing capabilities.

1.3 Scope

This project has wide scope. It is as follows.

- Free hand drawing for various designing purpose
- Use of hardware mouse and hardware keyboard is eliminated.
- Physically disabled people who cannot operate computer with their hands can operate it through gestures.
- Model Designers like car designers can use this system to improve ease in their designing process.
- This system will lead to a new era of PC gaming.
- Free hand drawing for various designing purpose
- Amusing interaction capabilities to control computer will provide a superior environment for children to explore knowledge and use the system.
- Long durability of hardware with comparison to conventional hardware
- Depth sensing capability of this system can be used in many means. Most of them are yet to explore

1.4 Technology Used

- Microsoft Kinect's depth sensing technology
- Support Vector Machine (supervised machine learning) for image classification
- .NET architecture for app development
- Accord.NET in C# as a base machine learning library
- Emgu CV image processing library (.NET wrapper for Open CV image processing platform)
- Open NI for Microsoft Kinect (Open source driver and set of library in order to use Kinect in Open Source Framework)
- LIBSVM (a standard support vector machine library for testing purpose only)

1.5 Hardware Software Used

Hardware

- Microsoft Kinect

Software

- Visual Studio

2. Literature Review

2.1 General Terminology

2.1.1 Kinect

Kinect (codenamed Project Natal during development) is a line of motion sensing input devices by Microsoft for Xbox 360 and Xbox One video game consoles and Windows PCs. Based around a webcam-style add-on peripheral, it enables users to control and interact with their console/computer without the need for a game controller, through a natural user interface using gestures and spoken commands.

Microsoft Kinect is a first of its kind device which not only provide the user with RGB/CMYK frame but also provides the approximate depth of the surrounding objects from the lens of the camera. The device is equipped with various modern sensors and high resolution camera which makes it unique and quite use full for the image processing and computer vision tasks.

Currently there are two versions of Microsoft Kinect.

(i) Kinect V1
(ii) Kinect V2

Our project is based on the Kinect V1, so we will explain all the details about the V1 device.

As the image shows, Kinect V1 has RGB camera, 3D depth Sensors, Multi Array Mic and a motorized tilt. All of these individual components make up the whole Kinect device and each one performs a unique task.

The purpose of the Kinect was to provide an easy interface for playing XBOX 360/One Games. The Kinect developer team has released all the APIs and related documentation for the access of device's core features. Since then the device has captured the interest of the computer scienceints and researches to implements complex image processing task on the Kinect. Previously one of the major issue in the field of computer vision was to get the approximate depth information from the 3D environment. There are number of algorithms which were developed to fulfill this Job such as

a. Z- depth algorithm.
b. PTAM algorithm

Even though these algorithms were quite useful, they lack the performance. Kinect however provides the depth image of all the frame with millimeter precision and with 30FPS -depending on the devices' version.

These features attracted various group of scientists and researchers to use the Kinect for all kinds of computer vision problem. The new term called "RGB-D" sensor was introduced.

We are also utilizing the same feature in our project to implement a robust gesture recognition system which is highly portable and efficient.

2.1.2 Image

Images are general pictures that human eyes recognize easily. But computers cannot interpret them as we do. They understand image in the form of 2-D matrices. There are multiple formats to store images and each one of them has its individual peculiar characteristics. But in general all of them stores values of pixels as the member of 2D matrices. Previously all the images format was designed to store only 2D information i.e. no depth data. However recently image formats have been evolved to store the depth of the pixels with a newer RGB-D format.

In the conventional RGB images, each pixel is associated with its corresponding color data in the matrix. But in the RGB-D a 3D array not only stores the color value of each pixel but also stores the depth value of each pixel.

2.1.3 Computer Vision

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image. Understanding in this context means the transformation of visual images (the input of retina) into descriptions of 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. Computer vision has also been described as the enterprise of automating and integrating a wide range of processes and representations for vision perception.

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 to the construction of computer vision systems.

Computer Vision tries to do what a human brain does with the retinal data, that means understanding the scene based on image data. That mainly involves segmentation, recognition and reconstruction (3D) and these work together to give us the scene understanding. Computer Vision employs image processing and learning as well as some of the other mathematical methods (i.e. Vibrational Methods, Combinatorial approaches) to do the aforementioned tasks.

However, Image processing, is mainly focused on processing raw images without giving any knowledge feedback on them. For example, if you want to do a semantic image segmentation (A computer vision task) you might apply some filtering on the image during the process (an image processing task) or try to recognize the objects in the scene (learning task).

Sub-domains of computer vision include

(i) scene reconstruction
(ii) event detection
(iii) video tracking
(iv) object recognition
(v) object pose estimation
(vi) learning
(vii) indexing
(viii) motion estimation
(ix) image restoration

In our project we have worked on few subdomains described above like Object recognition, event detection and learning.

2.1.4 Machine Learning

Machine learning is a subfield of computer science and statistics that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. In 1959, Arthur Samuel defined machine learning as a "Field of study that gives computers the ability to learn without being explicitly programmed". Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from example inputs in order to make data-driven

predictions or decisions expressed as outputs, rather than following strictly static program instructions.

Machine learning is closely related to and often overlaps with computational statistics; a discipline which also focuses in prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms is infeasible. Example applications include spam filtering, optical character recognition (OCR), search engines and computer vision. Machine learning is sometimes conflated with data mining, where the latter sub-field focuses more on exploratory data analysis and is known as unsupervised learning.

Machine learning can be implemented via two methods:

a. Supervised learning
b. Unsupervised learning

Supervised learning

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way (see inductive bias).

Unsupervised learning

Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to preprocess data.

We have used both supervised and unsupervised learning methods for in our project for different purpose. For example, we have used Minimal sequential algorithm as a supervised learning algorithm for hand detection and unsupervised learning algorithm for gesture recognition.

2.2 Approached for Image recognition and classification

There are multiple approaches which can be used for image recognition and Image classification. In our project we have used image recognition for the hand state detection (Open/Close) and Image recognition for Gesture detection.

In this section we will discuss various high performance approaches for image recognition and classification.

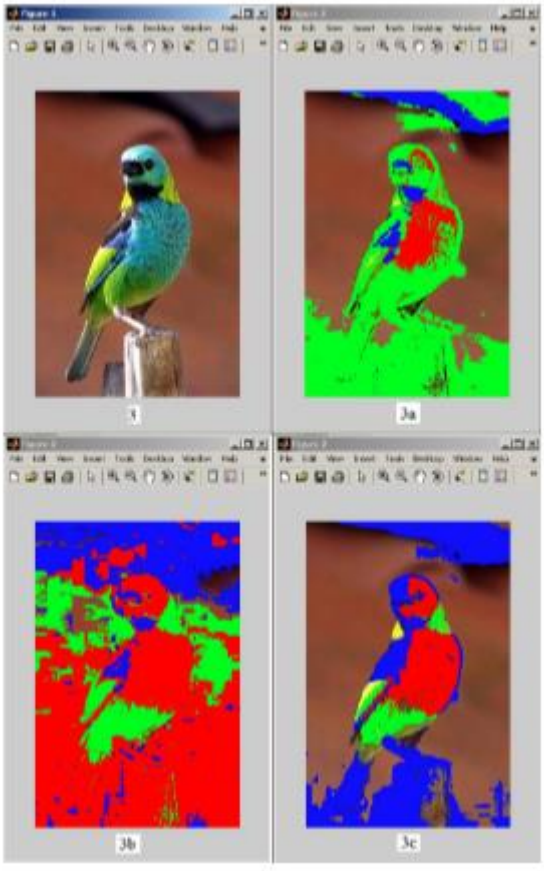
I. Image Processing Based approach

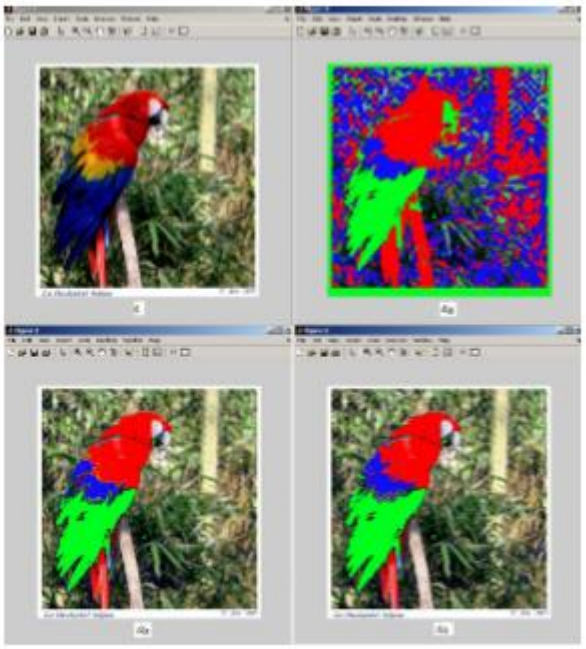
In the image processing based approach is used to detect and classify certain kinds of images. Although there are various flaws in this approach but it can be helpful in some cases for performance optimization point of view.

For example, we can use this approach for some images where we need to extract certain feature of the image and then compare them with preexisting sets of rules in order to classify or recognize the images.

Following section provides some example to clarify the concept and then we will discuss some flaws of this approach and talk about how to rectify them.

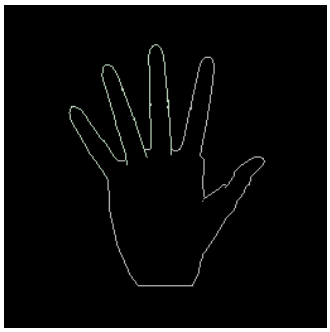
Take the example of the Images below.



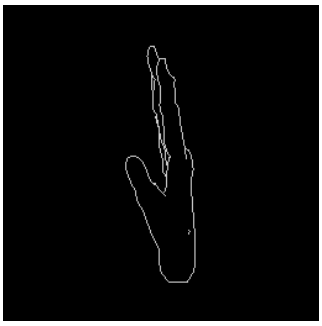


As we can see in the above images that we have applied simple image processing task of feature extraction using binarizaton of the pixel to highlight the contrast color. We can now use this features to compare them with the preexisting sets of rules to classify them into appropriate groups of pictures.

Now we will take another example from which we will discuss its limitations and possible solution. Consider the pictures given in below.



Open Hand with considerable convex hull



Open hand with minimal convex hull.

As we can see that we have two pictures of the open hand, and we are trying to classify them based on their image feature characteristic. The first picture has a large amount of noticeable convex hull between the fingers but the second one has almost null convex hull.

Now if we develop an algorithm which classify the images of hands based on their convex hull values then we can see that we have two contradictory results for the same picture taken with different angle.

This is where the limitation of the image processing lies when it comes to classify images. This can rectify using the learning based approach. We have presented here several of them here.

II. Learning Based approached
   a. Hidden Markov Model

Hidden Markov Model is a Supervised learning machine which is a regression based system. A hidden Markov model (HMM) is a statistical Markov model in which the

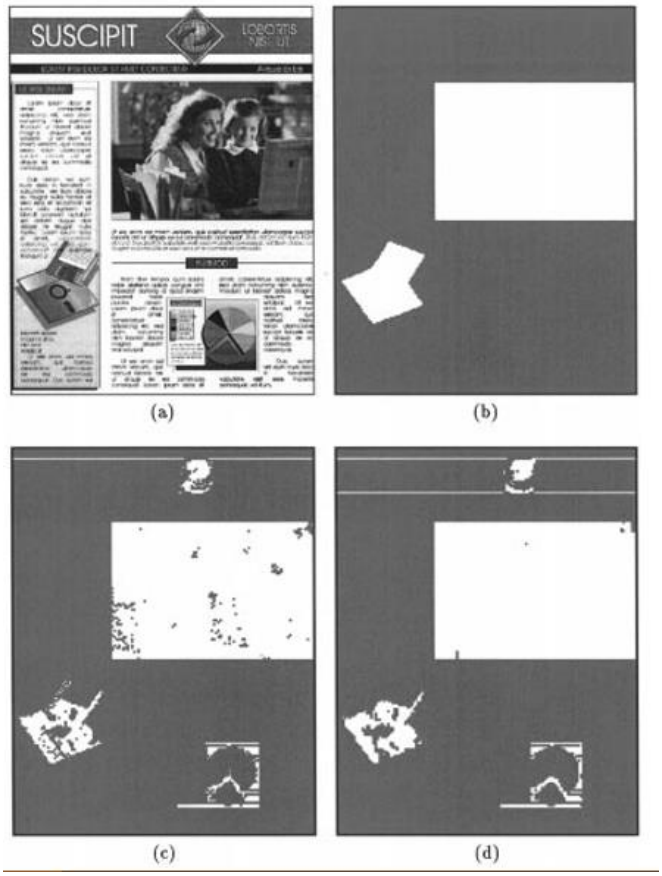
system being modeled is assumed to be a Markov process with unobserved (hidden) states. A HMM can be presented as the simplest dynamic Bayesian network. The mathematics behind the HMM were developed by L. E. Baum and coworkers. It is closely related to an earlier work on the optimal nonlinear filtering problem by Ruslan L. Stratonovich, who was the first to describe the forward-backward procedure.

In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but the output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by an HMM gives some information about the sequence of states. The adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; the model is still referred to as a 'hidden' Markov model even if these parameters are known exactly.

When it comes to classify images based on HMM we have to take certain factors into consideration for optimum use of the Markov model. Some of the algorithms which are used in HMM are as below

a. Baum-Welsh Algorithm
b. Viterbi Algorithm
c. Forward-backward algorithms

HMM is very popular for Face recognition and face detection problems, Because of its seamless implementation and efficiency. Basically what HMM does is that it uses simple image processing to extract features or some object. We can used SIFT features or HAAR classifier with HMM to provide more accurate model. HMM then creates the Markov chain with the vectors of all those recognized features. Any supervised algorithm mentioned above can be then used to compare the data and classify the image. The following image is self-explanatory based on HMM theory.



b. Artificial Neural Network

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning.

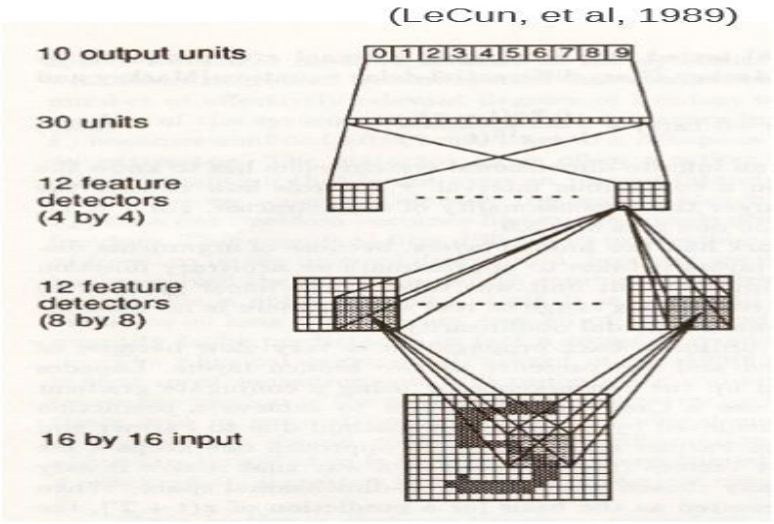
For example, a neural network for handwriting recognition is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by a function (determined by the network's designer), the activations of these neurons are then passed on to other neurons. This process is repeated until finally; an output neuron is activated. This determines which character was read.

Artificial Neural network used 3 models to receive signals from other neurons, processes (integrates) incoming signals, and to send the processed signal to other neurons.

- Deterministic Model
- Stochastic model

We will take an example of how to recognize the hand-written ZIP codes. Although the process itself is very complex and long but we will summarize the whole process.

Let us take the 16*16 array with the neuron weight ranges from (-1,1). For 10 units of hand written digits. We will take 3 hidden Layer and use each one of them to further process. We will reduce the parameters by weight sharing on the first hidden layer. We will use the hidden layer two for the same task. We will apply the learning algorithm and then use the back-propagation learning accelerated with quasi-newton rule. Studies shows that a neural network with above configuration has 99% training accuracy and 95% Testing accuracy.



c. Support Vector machine (SVM)

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked

for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

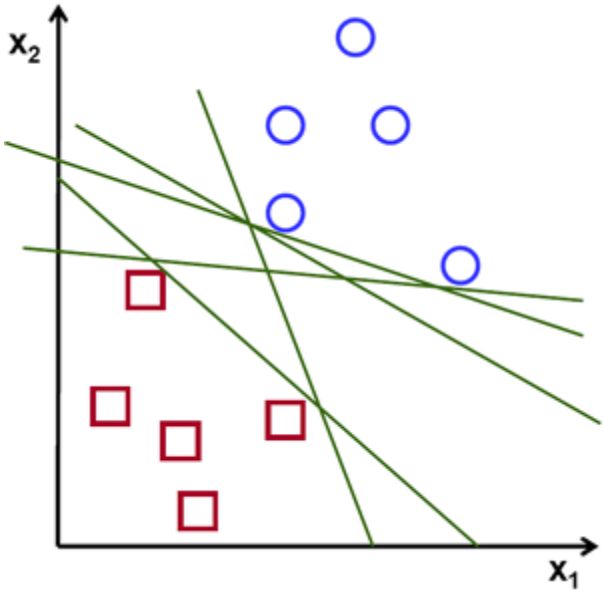
In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, a supervised learning is not possible, and an unsupervised learning is required, that would find natural clustering of the data to groups, and map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering and is often used in industrial applications either when data is not labeled or when only some data is labeled as a preprocessing for a classification pass.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

In which sense is the hyperplane obtained optimal? Let’s consider the following simple problem:

For a linearly separable set of 2D-points which belong to one of two classes, find a separating straight line.



In our project we have used SVM for both hand detection and gesture recognition. We used Multiclass Support vector machine for gesture recognition system. We used sequential minimal optimization as the supervised algorithm. The pseudo code of the Sequential minimal optimization is given below.

```
target = desired output vector
 point = training point matrix
 procedure takeStep(i1,i2)
 if (i1 == i2) return 0
 alph1 = Lagrange multiplier for i1
 y1 = target[i1]
 E1 = SVM output on point[i1] - y1 (check in error cache)
 s = y1*y2
 Compute L, H via equations (13) and (14)
 if (L == H)
 return 0
 k11 = kernel(point[i1],point[i1])
 k12 = kernel(point[i1],point[i2])
 k22 = kernel(point[i2],point[i2])
 eta = k11+k22-2*k12
 if (eta > 0)
 {
 a2 = alph2 + y2*(E1-E2)/eta
 if (a2 < L) a2 = L
 else if (a2 > H) a2 = H
 }
 else
 {
```

```
 Lobj = objective function at a2=L
 Hobj = objective function at a2=H
 if (Lobj < Hobj-eps)
 a2 = L
 else if (Lobj > Hobj+eps)
 a2 = H
 else
 a2 = alph2
 }
 if (|a2-alph2| < eps*(a2+alph2+eps))
 return 0
 a1 = alph1+s*(alph2-a2)
 Update threshold to reflect change in Lagrange
multipliers
 Update weight vector to reflect change in a1 & a2, if SVM
is linear
 Update error cache using new Lagrange multipliers
 Store a1 in the alpha array
 Store a2 in the alpha array
 return 1
endprocedure
procedure examineExample(i2)
 y2 = target[i2]
 alph2 = Lagrange multiplier for i2
 E2 = SVM output on point[i2] - y2 (check in error cache)
 r2 = E2*y2
 if ((r2 < -tol && alph2 < C) || (r2 > tol && alph2 > 0))
 {
 if (number of non-zero & non-C alpha > 1)
 {
 i1 = result of second choice heuristic (section 2.2)
 if takeStep(i1,i2)
 return 1
 }
11
 loop over all non-zero and non-C alpha, starting at a
random point
 {
 i1 = identity of current alpha
 if takeStep(i1,i2)
 return 1
 }
 loop over all possible i1, starting at a random point
 {
 i1 = loop variable
 if (takeStep(i1,i2)
 return 1
```

```
 }
 }
 return 0
endprocedure
main routine:
 numChanged = 0;
 examineAll = 1;
 while (numChanged > 0 | examineAll)
 {
 numChanged = 0;
 if (examineAll)
 loop I over all training examples
 numChanged += examineExample(I)
 else
 loop I over examples where alpha is not 0 & not C
 numChanged += examineExample(I)
 if (examineAll == 1)
 examineAll = 0
 else if (numChanged == 0)
 examineAll = 1
 }
```

2.3 Our Approach

One of the main components of this system- Microsoft Kinect is a device which has capabilities to provide various outputs like skeleton stream, depth stream and the color stream of the visible environment. In order to build the hand gesture recognition, we needed to make sure that we can recognize the hand i.e. separating hands from the skeleton and then manipulating that result so that robust hand detection can be done with less effort and maximum optimization.

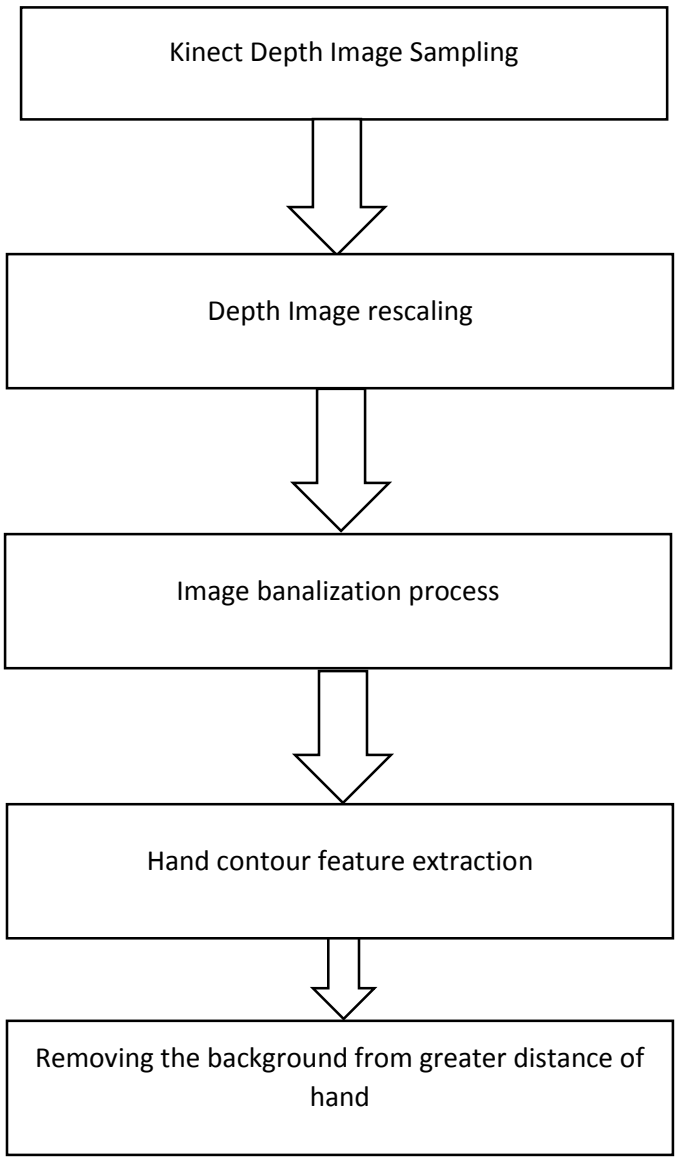
We used the depth sensing capabilities of the device to carry out the given task. Entire processes of segmenting hands from the body is shown in the following lines.

The Kinect provides the IR image of the environment which is the pixel by pixel representation of the depth data. We assume a virtual foreground plane at the distance of the left hand from the camera. Now the main idea is behind this logic is that anything that is in front of the plane should be visible to the camera and anything with the distance greater than the camera is neglected as a dark pixel so we can get a perfect picture of the

hand. Image binarization process is must in order to carry out this task. After this process is completed we are left with nothing but the picture of our hand- assuming that the hand is in front of the body and there are no other objects between the hands and the lens of the camera.

After the image of the hand is obtained we need to extract the ROI (Region of interest). We used the chain approximation method for extracting all the contours from the images. Fortunately, we will have only one contour from the image which will be then used to recognize the gesture of the hand.

This process is summarized in below.



(A) Depth Image from Kinect

(B) Sampling of depth Image with respect to human Skelton points

(C) Extracting Images of the hand from the Depth Data (D) Binariezing Image

(E) Contour Extraction from the Given Binarized Image.

1. Gesture Recognition

   The process of gesture recognition in this project is compartmentalized into two sub processes.

   (I) Fist detection
   (II) Hand gesture Recognition

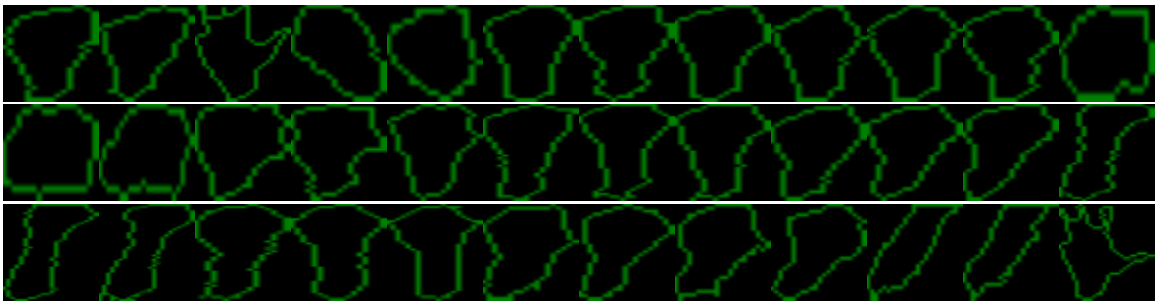
   Fist detection

   Fist detection is a necessary part of this research based project. We are using fist as the standardize gesture to emulate the mouse movements. For example, when the used has his hands open he can navigate through the screen using his hands movements. But when a user is desired to click on a certain item from the screen. The using closes his hands and then he can emulate the “Mouse click” event.

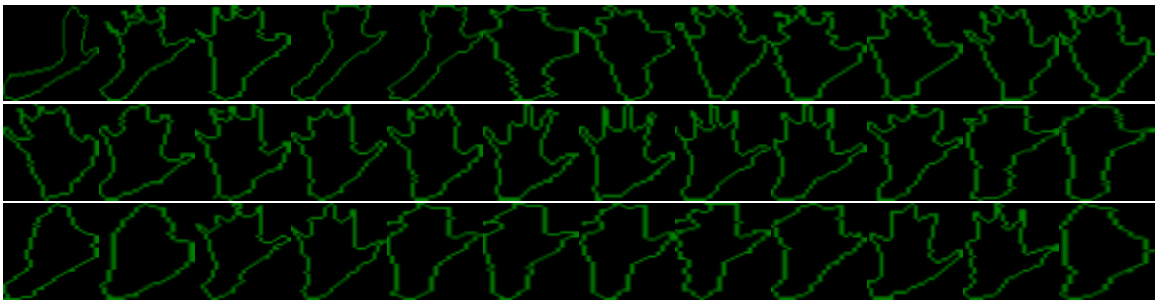
   As we have discussed in above about various methods of image classification. We can use one of the above described method named “Support Vector Machine” for Image classification- in our case fist detection.

   We used the training data provided by Boston University to testing out the hypothesis. After successful implementation of the SVM in the C# environment we used Accord.NET library with SVM and Sequential minimal optimization for training hand images.

   We used total of 4688 training images of combined Open and Closed hands, then we used 3540 Images for testing purpose. We got fairly accurate results with our testing dataset with 95% of accuracy. The given figure gives the example



   Images of close hands.



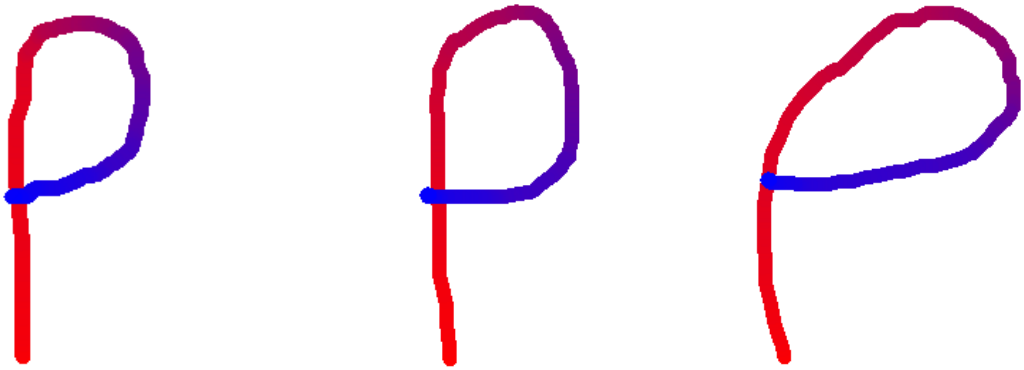
Images of Open Hands.

Note that all of these images were captured at the rate of 30 Frames per second using Microsoft Kinect.

Hand Gesture Recognition

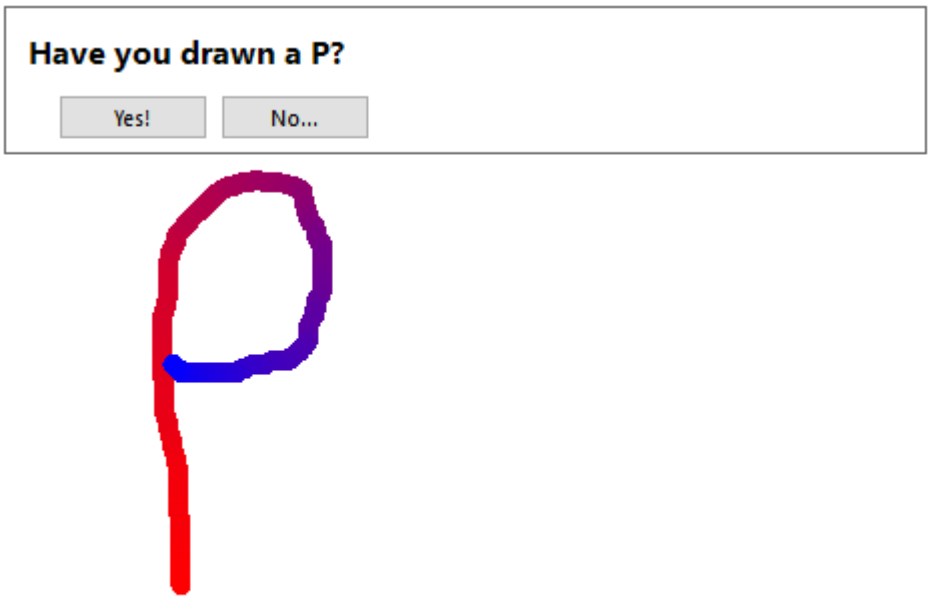
Hand gesture recognition is the way of recognizing various hand movements done by humans. We used multiclass Support Vector machine to classify all the gestures we wanted to perform.

We trained each identical gestures with 3 suitable alternatives and then train them using multiclass support vector machine. We have given the example of that below



“P” Sample-1 “P” Sample-2 “P” Sample -3

And when we finally train the dataset then we can get the result like this.



This is how we can create a gesture recognition system using robust multiclass support vector machine.

3. System Analyses

a. Study of current system

The conventional system consists of graphical user interface as an interface for controlling various aspects of computer. However, there seems to be nothing wrong with the current usability and as we know, GUI is currently the most efficient way of interacting with computer, we intend to provide a new approach i.e. Gesture User Interface. It does not alleviate any of the problems related to current system but it surely does provide an innovative and amusing way of interacting with computer. The idea is to provide something new and advancement of technology instead of improving and focusing the current technology. Gesture recognition approach will provide numerous features and it will ease the accessibilities of computers by removing various hardware dependencies e.g., mouse.

We have searched through various resources to know if there exists a system much like our but there is none. Hence ours is first of its kind.

3.2 Problem and weakness of current system & requirement of new system

There are no problems with current system. There is no significant improvement required in the current system though technology advances and so is the desire to find innovative and new ways of interacting with the various resources of new technologies. Gesture User Interface is just a new and pioneering approach rather than upgrading the current system. We can consider Gesture User Interface as an alternative to classical graphical user interface rather than an enhancement of the same.

However, Graphical User Interface requires dependencies on multiple hardware, the use of Gesture User Interface will surely eliminate that need. Moreover, depth sensing capabilities and 3D gesture recognition are surely some of the features that cannot be used using conventional graphical user interface. Not only this, but it provides more efficient access to computer resources for handicapped persons. Hence this system will provide incredible interface to use which will lead to unique user experience.

3.3 Feasibility study

Security:

There are no security issues concerned with the system. No personal details are required to operate this system and hence the system will not lead to any security risk which will lead to any fatal information loss.

Reliability:

The system is highly reliable. User can make it to engage with either nearest user or most active user and many more. It will even remove distortion by some obstacles to some extent. The device was trained with millions of possible skeletal and hence it can easily remove unwanted disturbance with help of its machine learning algorithms.

Portability:

Portability is not a big issue in this system. The setup needs to use one USB port of the system which is generally easily available on most of the systems. Other than this, user must install Kinect drivers from Microsoft's official website. These drivers are essential to run the services provided by Kinect. In general, all the technical requirements are easy to accomplish. The only problem is, this works only on windows machines. In other operating systems like Linux or Mac OS, the device does not work efficiently as its support and development is very much limited in those cases.

Extensibility:

The system is extensible and adding new features are easy to add in the existing system. Our system works on .NET frame work and uses MVC architecture. Hence adding functionality to existing system is not a problem until and unless it is conflicting with the existing system. So, considering the fact that most of the features will not affect each other, the system is fairly extensible.

Economic Feasibility:

Economic Feasibility is not an issue with our system. The device we are using to get enhanced and better gesture recognition is costly for middle class or lower middle class audience. Its market price is 10K INR. But we are not targeting the same audience. For our product, the target audience

is either higher middle class or upper class people, or people who are in need of this kind of system who will be ready to pay anything for the same. Our system is first of its kind and far more futuristic. Hence, economically it is feasible. We are also offering other schemes that can reduce the price a little and help the consumer which is plausible.

Maintainability:

Regular maintenance is not required in our system. In most of the cases, user will use it as people use television. Fix for once and use it for long time. Hence there are less chances of damaging the hardware which is most critical. Although the hardware is delicate and hence enough care must be taken to clean it on regular interval from dust and any physical damage must be avoided for long life span of hardware.

4. Project Management

4.1 Project planning and scheduling

4.1.1 Project development approach

In order to build our current system, we are using iterative development approach because our system is highly dynamic that cannot be developed by single time programming approach i.e. waterfall approach. At various stages in our project development life cycle, we need to carry out various testing, consider the performance and accuracy, check for most suitable solution as well as integrate various modules. It takes immense amount of intermediate testing sessions on each small module and then the combination of them will be tested thoroughly. Few times, the process led to some unexpected changes in system's core design as the libraries and SDK tools are constantly changing and their support can be limited sometimes. Moreover we found that some algorithms were better than the others but had their tradeoffs in terms of performance and accuracy. Selecting the best possible approach was therefore crucial to the output of testing and selecting the best plausible algorithm among the many.

4.1.2 Project plan

Basic idea was to map user's movements to a virtual hardware and actions of virtual hardware would be mapped to the real hardware. But there was not a single approach to do this. Hence, we developed our own strategy and planned our way to achieve our desired goal. Basic idea of what we planned to achieve could be summarized in the following steps.

1. Track the skeleton of the body to get the position of hand so that it can be further used to get co-ordinates for mouse pointer.
2. Identify how to control mouse movements corresponding to hand movements.
3. Implement mapping of hand movements with mouse co-ordinates.
4. Determine how to recognize gesture based on hand position.
5. Find an algorithm fast and accurate enough for image classification in order to implement gesture.

6. Select and implement best possible algorithm for high performance and accuracy for image classification.
7. Implement image classification algorithm to set triggers for start and stop reading for recognizing gestures and other activities.
8. Implement gesture recognition.
9. Deploy application.

4.1.3 Schedule representation

To start with our concerned plan as we described, the first step to track the skeleton of the body to get the position of the hand so it can be further used to get co-ordinates for mouse pointer. Kinect provides the powerful SDKs and framework itself to recognize and tracking skeleton. It uses IR camera and RGB camera which processes the data stream to identify the human body in each frame and then gives skeleton of the human. We decided to work with Kinect at a performance that gives us skeleton stream at the rate of 30 frames per second.

Next task was to identify how to control mouse movements with hand movements. Extracting the information provided by skeletal stream, we mapped co-ordinates of the hand with the X-Y co-ordinates of mouse and set mouse to the corresponding position on the screen. Using the depth sensing capability, we even implemented the mouse click feature and drag components feature.

Afterwards we needed a specific way for determining gestures. Various approaches were considered including continuous background service running on server or on local machine which will continuously check for mouse movements and if a gesture is recognize then trigger corresponding event, using a panel to draw the gesture etc. But the above approaches had many issues like, if the user does not intend to draw a gesture but happens to move the mouse in corresponding way, it may unwillingly fire the action. So finally we decided to recognize a gesture only when hand is in particular state so that user will have the control of triggering actions.

In order to achieve the solution for above considered problem, our primary motive was to determine the state of the hand i.e. whether the hand is open or close. Unfortunately, Kinect V1 does not provide the implicit support to determine such a state as it can identify only one point in the hand i.e. palm. So we needed an algorithm for image classification.

There were various algorithms and approaches available like Classical Computer Vision, Hidden Markov Model, Artificial Neural Network, Machine learning etc. All these approaches have several algorithms that we could use to fulfill our requirements. We tried few algorithms from all these approaches but each of those algorithms had their own benefits, tradeoffs as well as limitations and none of them were able to accurate and fast enough to suit our needs. Because of low performance and high complexity of almost all of the above approaches, we finally decided to work with Support Vector Machine algorithm for image classification of Machine learning category.

SVM uses the supervised machine learning and hence it needed to be trained. While training, it puts every image on an n dimensional hyper plane and uses an imaginary plane to classify the images. The most significantly distinct images are put far from the plane whereas the images with less distinct images stay close to the plane. This is called as vector and it can stored as trained machine state. Hence once the machine is trained, it can be used to classify images based on stored vector. To implement the above algorithm in .net platform, we first tried to use python library Emgu.CV as it works on python which is very fast when it comes to execution. But that did not work well due to cross platform application support issues. Hence we finally implemented it using Accord.NET. This is the C# library that can be used for many image processing applications.

Using these tools and techniques, we trained the Support Vector Machine with tons of images. And to our surprise, it was so fast along with the superior accuracy of consistently maintaining 95%. But what we used for training and testing were clipped images of hand which was not easy to extract from the data stream that Kinect provides us. We used Kinect's depth sensing capability to eliminate background noise and kept only the hand in the foreground. Then we used contour to extract only the hand from the whole frame and also used clipping method to get only edges of hand. Hence all the problems regarding the image classification were resolved and we got around 94% accuracy on real time testing.

Next step was to identify the way to recognize gesture. For that we used a panel that can be opened from any window. The hand movements made after the panel is opened are supervised and co-ordinates are stored in an array. This array is again processed by an algorithm which determines the appropriate gesture is any and returns it to the main process which triggers corresponding actions.

4.2 Risk Management

4.2.1 Risk Analyses

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Many problems can plague a software project. A risk is a potential problem—it might happen, it might not. But, regardless of the outcome, it's a really good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur.

4.2.2 Risk Planning

As far as our project is concerned we have done detailed risk analysis. And it can be summarised as below.

Risk	**Probability**	**Effects**
Bug and performance limitations of used algorithms	Moderate	Moderate
Hardware failure	High	Serious
Hardware and Software's minimum availability requirement	Moderate	Moderate
Expensive Hardware	Moderate	Moderate
Power failure	High	Moderate
Not Suitable environment condition in order to recognize body parts	Medium	Serious

Risk planning process is considered when each of the key risk has been identified. Risk reduction Strategy is used as abatement procedure. This involves planning ways to contain the damage due to a risk.

Risk	Management
Environment Condition	As Infrared sensor is provided and It can be used to recognize body parts without availability of Light. This problem can be managed to some extent
Power Failure	To reduce the risk, UPS facility is used for backup storage.
Schedule risk	To reduce this risk, we are going to complete our project according to our schedule.
Performance	We have used best practice currently available in order to control the hardware without any failure in operation thus improving performance of the program.

4.2.3 Risk Identification

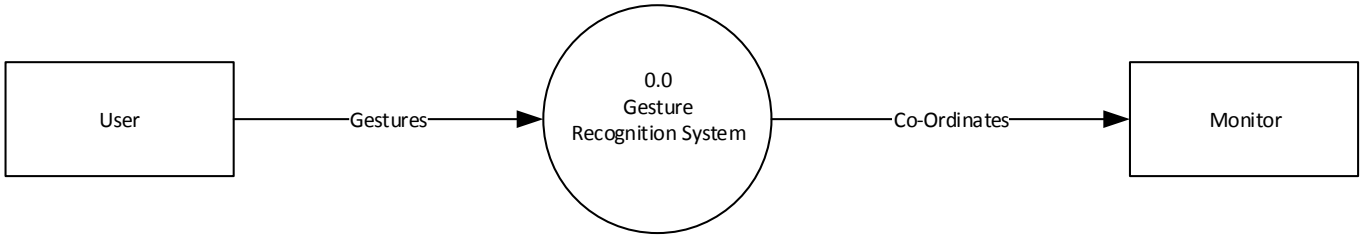
During the project plan we have considered all the proactive which we have think we will face during the project period. Here we have listed the risks which we have considered during the project plan:

- Possibility that the components are not available during the project period.
- Possibility that operating system is not compatible with the device.
- Possibility that the hardware resources are not available during the project period.
- Possibility that SDK components and CV algorithms do not provide possible output as we have expected that it would provide.
- Possibility that performance is not up to the mark that we expected it would be.

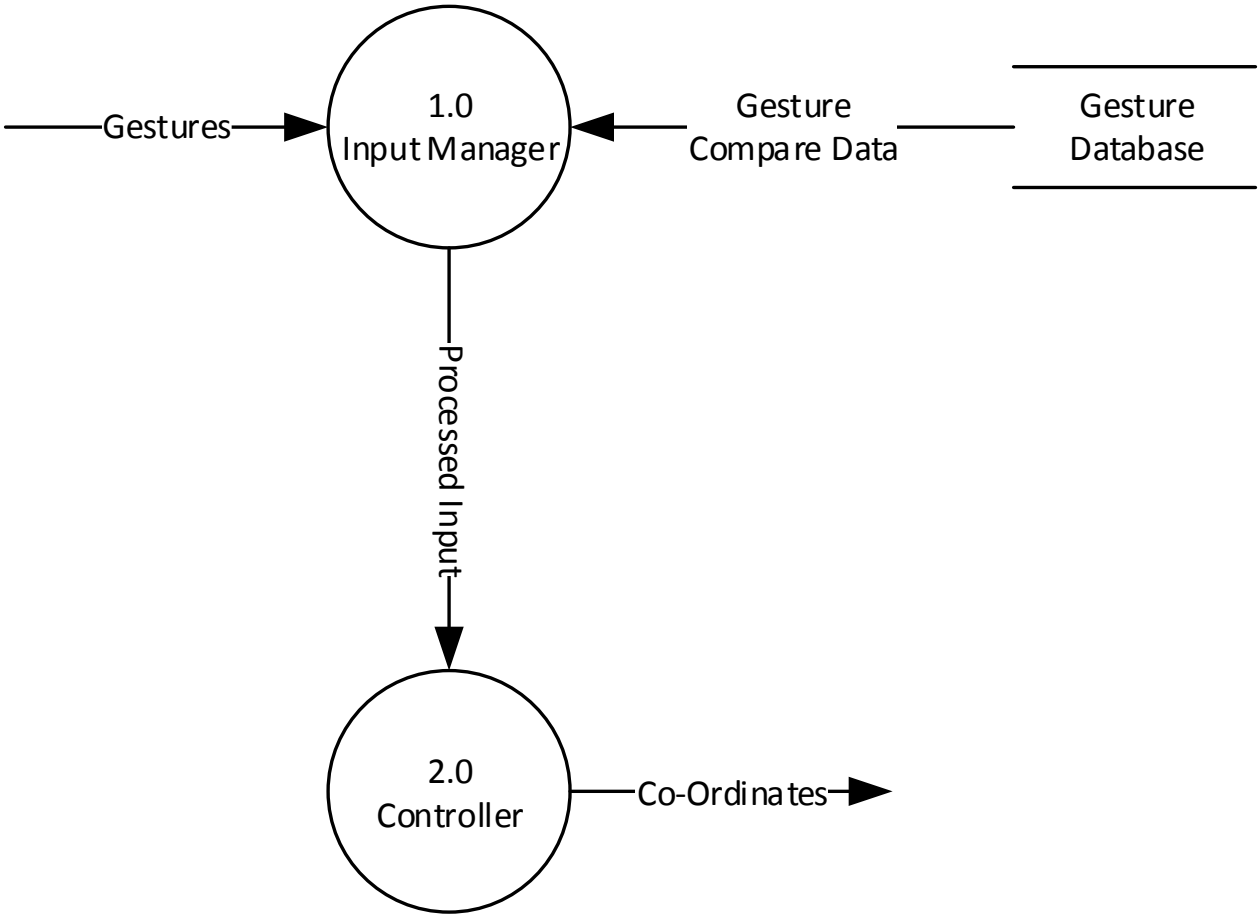
5. System Modeling

5.1 Dataflow diagrams

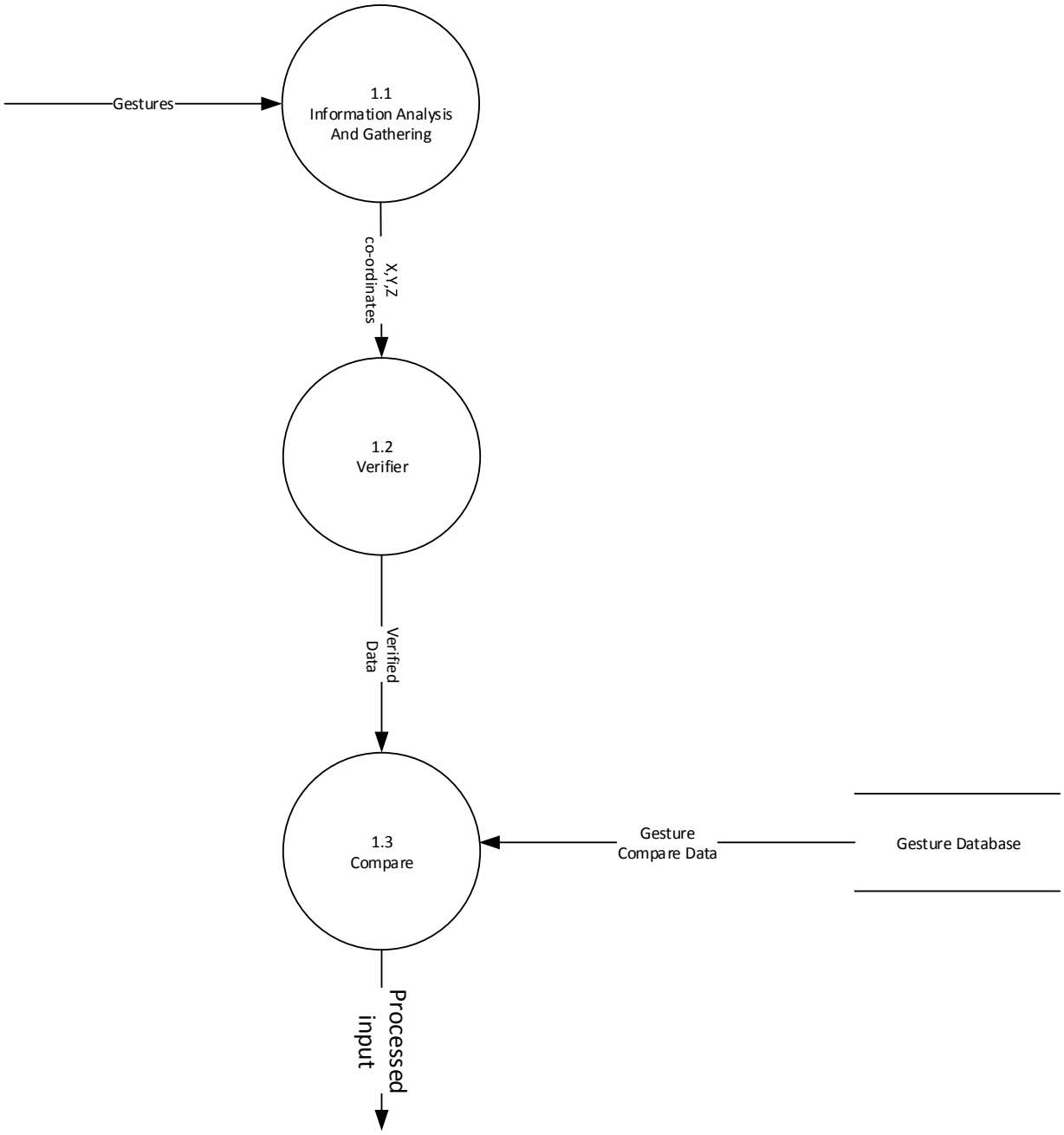
5.1.1 Context level diagram

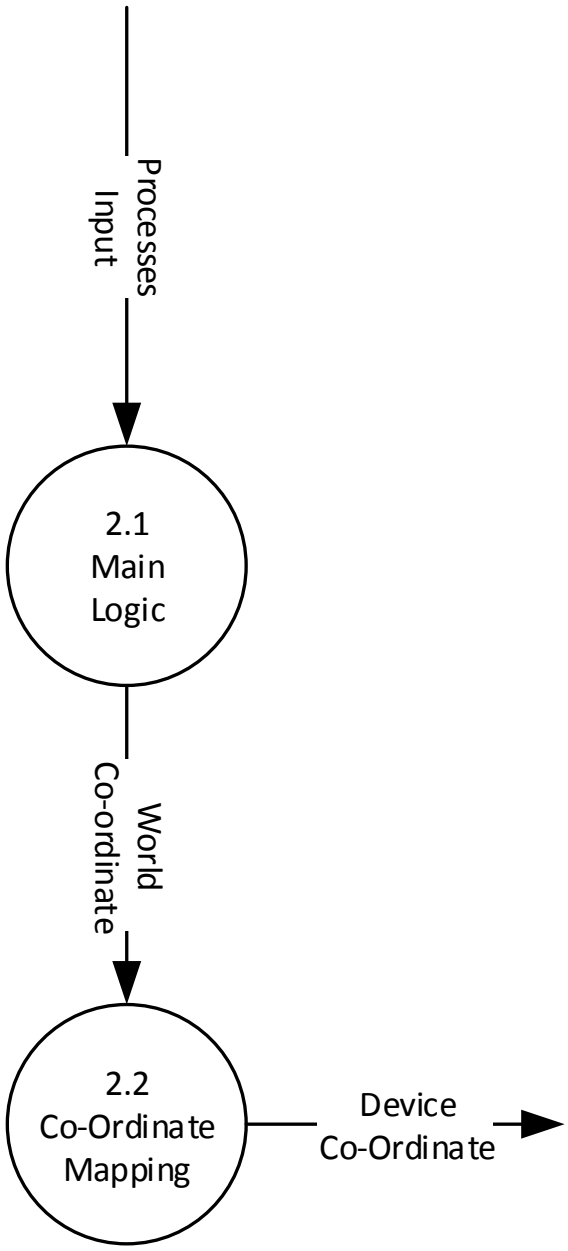


5.1.2 Level 1 Data Flow Diagram

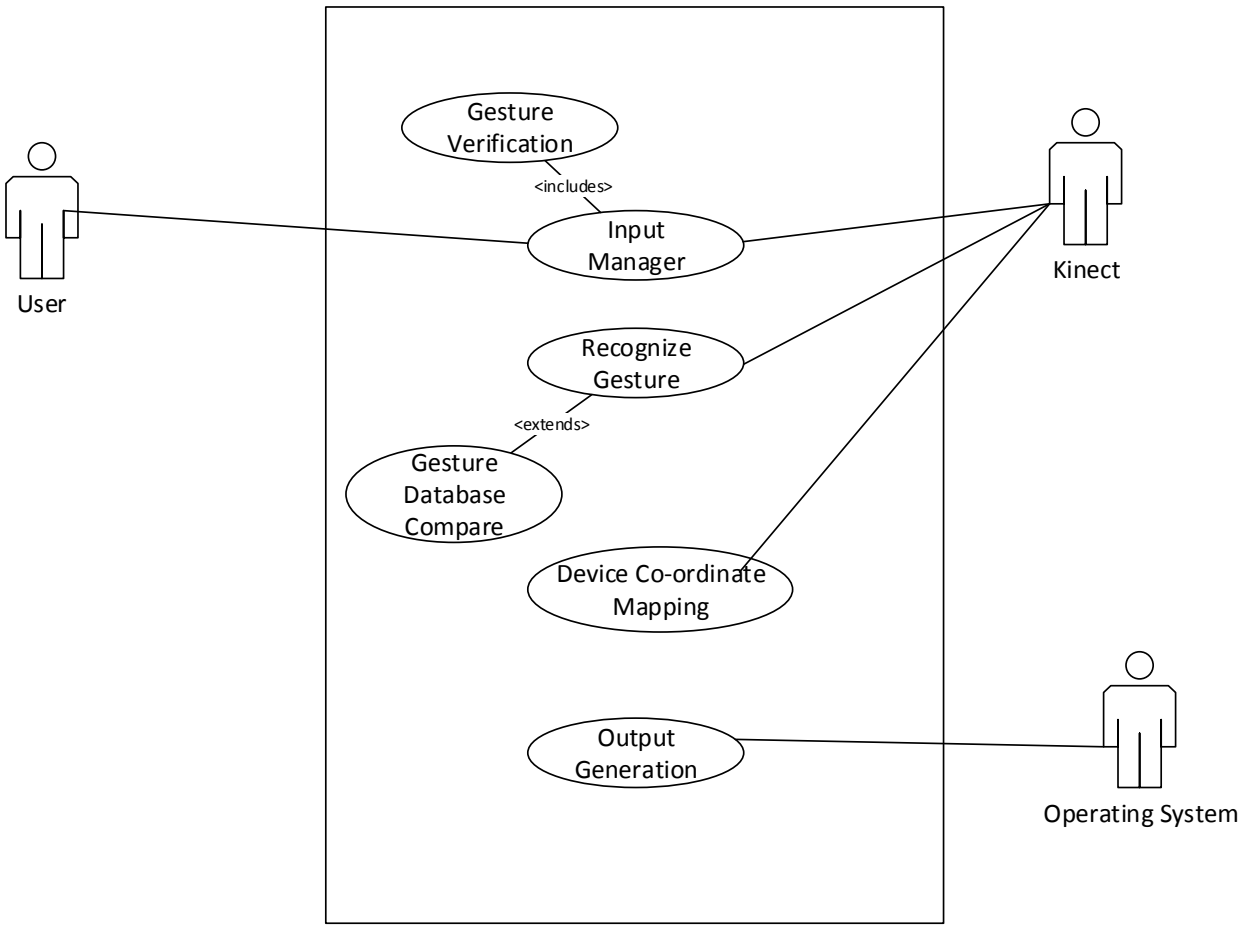


5.1.3 Level 2 Data flow diagram




Processes Input
2.1 Main Logic
World Co-ordinate
2.2 Co-Ordinate Mapping
Device Co-Ordinate

5.2 Use case Diagram



Use case	Input Manager
Summary	Manages Input in the form of Video and Audio streams
Actors	User, Kinect
Precondition	Kinect must be activated
Description	User perform body movements in order to control the system. Kinect gets input through the Input Manager.
Post Condition	
Exception	

Use case	Gesture Verification
Summary	Verification of Gesture
Actors	
Precondition	Input manager must me done first.
Description	Takes the data from input manager and verify it in order to find whether it is valid input or not. If any distortion is found, then it discards the data.
Post condition	
Exception	

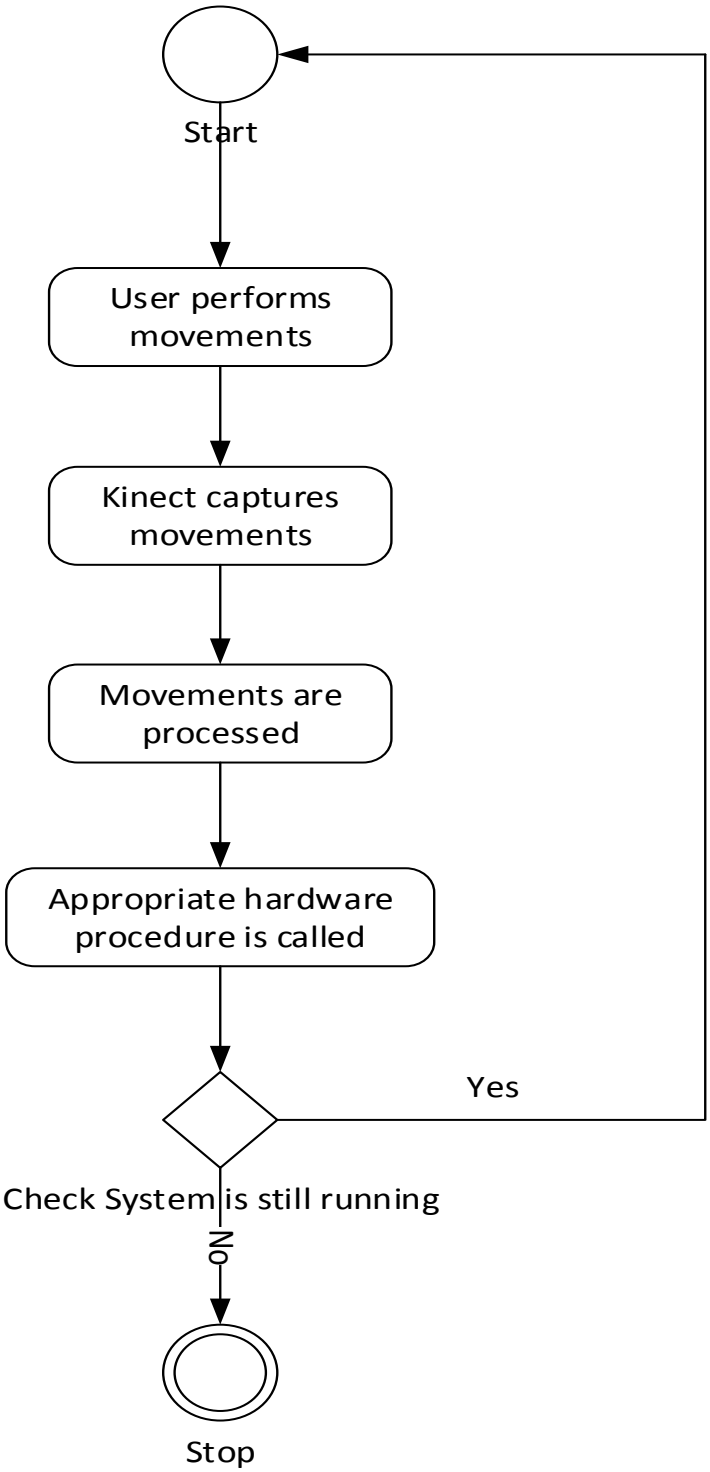
Use case	Recognize Gesture
Summary	Gestures are recognized
Actors	Kinect
Precondition	
Description	Kinect recognizes gesture and identifies it properly.
Post condition	Gesture must be matched with database to identify custom gestures.
Exception	

Use case	Gesture Database Compare
Summary	Compares Gesture with internal database
Actors	Kinect
Precondition	Gesture must have been recognized first
Description	The recognized gesture is matched with the database. If the same pattern is found and is already assigned to a specific task, then the appropriate action must be taken. Otherwise normal routine is to be followed.
Post condition	
Exception	

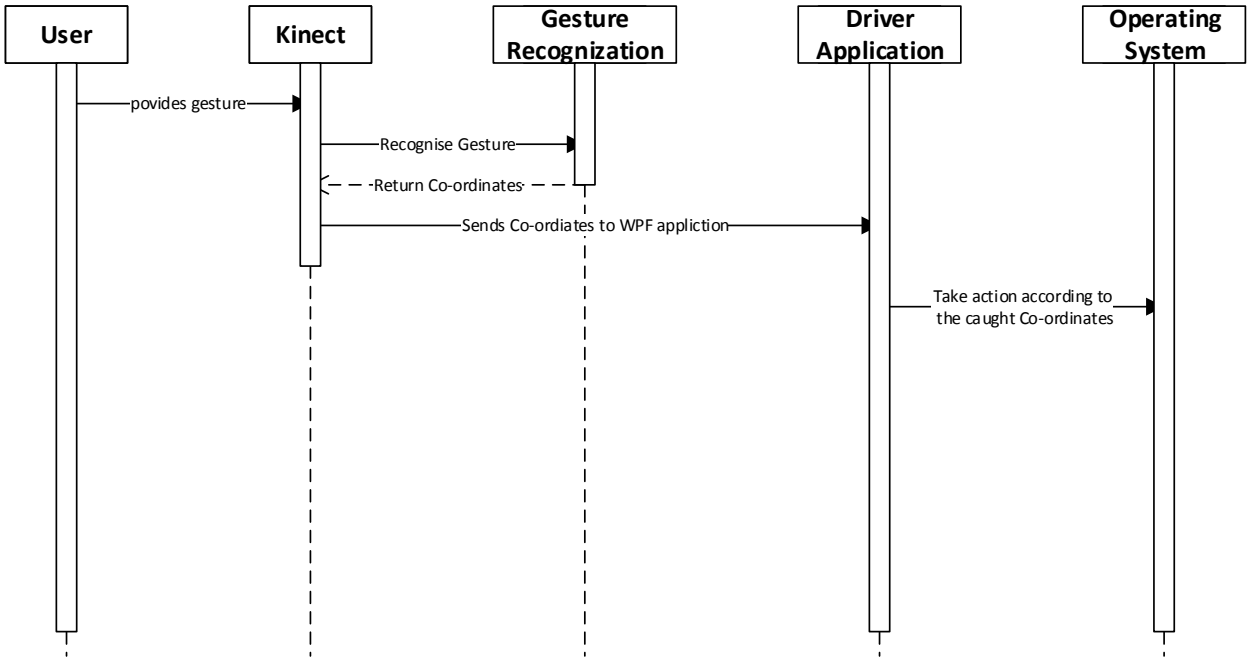
Use case	Device Co-ordinate mapping
Summary	Input co-ordinates are mapped with device co-ordinates.
Actors	Kinect
Precondition	No preset gesture is found
Description	When a normal routine is to be followed, the co-ordinates of the user's body are mapped to the world and view co-ordinates of the system. So that specific displacement of mouse can be applied.
Post condition	
Exception	

Use case	Output Generation
Summary	Output is generated and displayed on the screen
Actors	Operating System
Precondition	
Description	The co-ordinates which are mapped in the co-ordinate mapping are displayed here. In order to replicate user movements on screen, those co-ordinates are manipulated and processed for the specific monitor and then they are then displayed.
Post condition	
Exception	

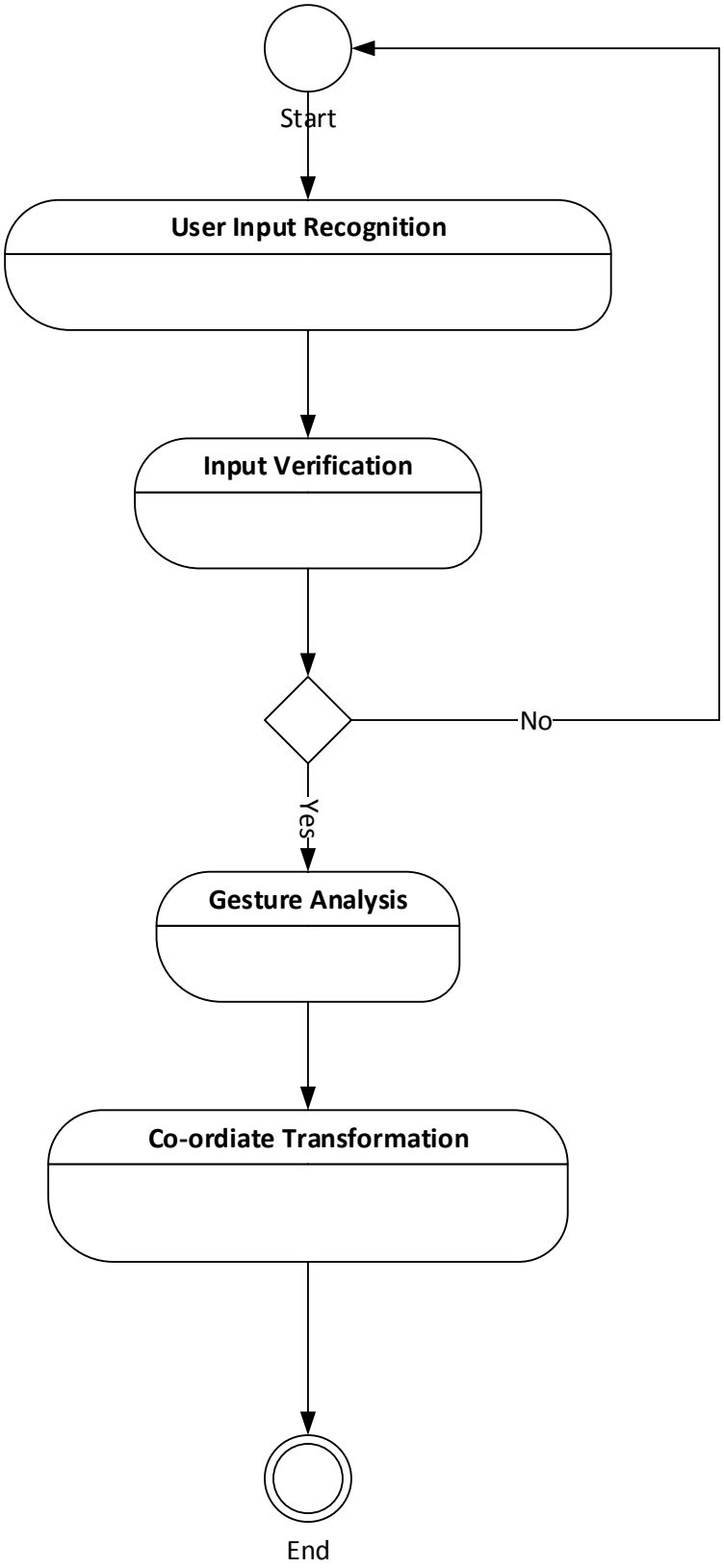
5.3 Activity Diagram



5.4 Sequence Diagram



5.5 State Transition Diagram



6. Limitations and Future enhancement

There is minor issue of accuracy when the mouse click event is triggered. The other disadvantage is that this system can be used only with windows platform. Hence Mac or Linux machines will not be able to run this system.

Future enhancements are possible because of its amazing features and endless applications. E.g. By mapping device co-ordinates with world co-ordinates and by using the depth sensing information any surface can be transformed into the touch enables surface. (Projector is required in order to project the screen information on the desired surface.) Low cost 3D model building. Equipment training using Kinect are some of the future enhancements that we think can be developed in this project.

Some other features can be added for some platform specific systems i.e. for windows 10 or higher, we can use voice recognition module in our system which will eventually use Cortana which is the best voice recognition system till date. Zooming feature is also one possible enhancement.

7. Conclusion

With the completion of our project, we are happy to say that the various features that we were planning to implement were finally been accomplished. Our system successfully recognizes the hand state as well as can track the hand movements with high accuracy of 93%. As a result gestures can be recognized with high efficiency and various events can be triggered based on the gestures. Moreover, as the control of the mouse movements can be mapped very fluently corresponding to our hand gestures, and various mouse events like clicking and dragging can be achieved, this system can successfully replace the functionality of mouse with some extra twist. This system is highly beneficial for designers and it provides very amusing and entertaining interface for general users who work with computer systems. The only hardware that is necessary with our system is Kinect. With its version 2, various new features are provided which gives us high possibilities for future enhancement.

8. Bibliography and references

No index entries found.

[1] Z. Ren and J. S. Yuan, “Robust part-based hand gesture recognition using kinect sensor,” IEEE Transactions On Multimedia, vol. 15, pp. 1110-1120, May, August 2013.

[2] H. Y. Tao and Y. L. Yu, “Finger tracking and gesture interaction with kinect,” IEEE 12th International Conference on Computer and Information Technology, pp. 214-218, 2012.

[3] X. Y. Wu, C. Yang, Y. W. Wang, H. Li, and S. M. Xu, “An intelligent interactive system based on hand gesture recognition algorithm and kinect,” Fifth International Symposium on Computational Intelligence and Design, 2012.

[4] Kinect for Windows document. [Online]. Available: http://en.wikipedia.org/wiki/Kinect

[5] Y. Zhang, S. Zhang, Y. Luo, and X. D. Xu, “Gesture trajectory identification and application based Kinect depth image,” Application research of computer, 2012, vol. 29, no. 9.

[6] S. Wu, F. Jiang, and D. B. Zhao, “Hand gesture recognition based on skeleton of point clouds”, 2012 IEEE fifth International Conference on Advanced Computational Intelligence (ICACI), pp. 566-569, October 2012.

[7] R. Miles, “Learn the kinect api,” O’Reilly Media, Inc. 2012.

[8] J. L. Raheja, A. Chaudhary, and K. Singal, “Tracking of fingertips andcentres of palm using Kinect,” presented at 2011 Third International Conference on Computational Intelligence, Modelling & Simulation.

[9] Opencv documentation. [Online]. Available: http://docs.opencv.org

[10] G. Bradski and A. Kaehler, “Learning OpenCV,” O’Reilly Media , Inc. 2008.

[11] W. Z. Chen, “Real-time palm tracking and hand gesture estimation based on fore-arm contour,” M. S. Thesis, Dept. Inform. Eng., National Taiwan University of Science and Technology, 2011.

[12] R. Fujiki, D. Arita, and R. I. Taniguchi, “Real-Time 3D hand shape estimation based on inverse kinematics and physical constraints,” presented at , Cagliari, Italy, September 6-8, 2005.