The Study and Implementation of Natural User Interface using Kinect

Akhil Guliani, *Member, IEEE*

***Abstract*—In this paper Natural user interface (NUI), its adaptations, and its implementations are presented. NUI is a human-machine interface which has evolved from the way humans interact with each other and their surrounding environment. Gesture, color, speech recognition are used in NUI to identify the user and his actions, which can subsequently be interpreted to perform the relevant task. The task can be any of the various applications of NUI such as control of heavy machinery, robot, computer, etc. During the study it was found that the best way to approach the design of NUI is using a depth capturing system, an image capturing system and an audio capturing system. During the study a combination of the three systems was found to be the best solution and is found in form of a commercial sensor called the Microsoft Kinect sensor. The sensor is also advantageous as it provides access to raw digital data streams and is equipped with a configurable audio digital signal processor (DSP). The control of machines requires a communication interface between the controlling unit and the machine. The TI MSP430 Launchpad toolkit is used for this purpose. The toolkit is economically optimal, as it includes a programmer, on board USB, two MSP430 microcontrollers and a free integrated development environment (IDE). The main aim of this work is to develop, test and demonstrate a robot platform that works on gesture based, color based and speech based NUI. The paper comprises of the study on basic user interface primarily focusing on NUI. The approach to find suitable sensors, their libraries, actuator interface, communication interface and their implementation is presented. Finally the result of the testing of the developed robot platform is demonstrated in this work.**

***Index Terms*—Kinect, NUI, MSP430 Launchpad**

I. Introduction

THE field of User interfaces has always been at at the leading edge of technological revolution with the invention of switches in the early days the control panels used to be the symbols of modern technology. Then with the advent of computers in the early 70's command line interfaces became synonymous with modern technology. Then came graphical user interfaces thanks to Xerox and popularized by Apple we came to associate computers and modern interfaces with menus and windows. Now the world is ready for another revolution in the field with the advent of Natural User Interfaces(NUI).

A natural user interface is a user interface designed to reuse existing skills; these include gestures, speech and colors; for interacting directly with content; such as accessing files and driving motors in our case. What humans can accomplish with a gesture, a sound or a color is amazing; by holding out a hand, we can stop a group of people from approaching a dangerous situation; by waving we can invite people into a room. Without a touch, we can direct the actions of others, simply through gestures or voice. By flashing a color we can control the flow of traffic on a road. The goal of NUI is to provide a user experience in which the interaction comes naturally, while interacting with the technology, and that the interface itself is natural or as if not present allowing for a more comfortable and human interaction.

With the evolution of Natural User Interface (NUI) we will be able to use the same gestures, sounds and colors that we use to interact among ourselves to control machines such as a robot, computers, etc. Such Interfaces have wide-ranging applications such as developing aids for the hearing impaired; enabling very young children to interact with computers; designing techniques for forensic identification; recognizing sign language; medically monitoring patients' emotional states or stress levels; detecting lies; navigating and/or manipulating in virtual environments and real environment; monitoring automobile drivers alertness/drowsiness levels, etc.

In this paper we develop such an interface which would be able to understand simple gestures, colors and sounds, and translate them into a physically observable action. The work done previously by Shotton et al. , François Rioux, and Mitra et al.[1], [2], [3], [4], [5] were instrumental in the development of this paper. With the study of the various techniques and technical documents we arrived at a conclusion that for achieving our goal we would require an image capturing system, a motion capturing system, a sound capturing system and a system for converting the digital events generated by the previous systems. After considerable market research we decided to use Microsoft's Kinect sensor as the primary sensor for capturing the data related to gestures, on further inquiry it was found that it was also capable of capturing sound and color images and provide them as raw data for manipulation to the developer. The system for converting the digital signal into physical signals was a bit challenging to select due to the multitude of options available at our disposals. Finally we chose to use Texas Instrument's (TI) MSP430 LaunchPAD as the interface between the computer and the actuator elements. L293d was the designated motor driver used in the actuation system.

The paper provides information about the work done theoretically and practically during my final project at Netaji Subhas Institute of Technology, Delhi. The various systems mentioned in the last paragraph were developed and integrated to achieve our objectives and are discussed in brief in the following sections.

II. System Hardware Components

The system hardware components are the most critical part of the project. They include everything from the sensor to the actuator. In our project the key hardware components are the sensor used to acquire the physical information about the user and the actuator interface which sends the relevant commands to the motors for proper operation of the system. The following sections take up the individual components of the hardware one by one and give in brief details about them relevant to our work.

A. Microsoft Kinect Sensor

Kinect, as shown in figure 1, is a motion sensing input device by Microsoft for the Xbox 360 video game console and Windows PCs. It enables users to control and interact with the computer without the need to touch a controller, through a natural user interface using gestures and spoken commands. It is a general purpose and low-cost 3D input device with audio input capabilities.[1], [6], [7], [8], [9], [10]

Figure 1. The Microsoft Kinect Sensor with its various streams identified

1) Sensor Specifications: This hardware is a box with two cameras that makes use of infra-red (IR) illumination to obtain depth data, color images and sound. A custom chip processes the data to provide a depth field that is correlated with the color image. That is the software can match each pixel with its approximate depth. The preprocessed data is fed to the machine via a USB interface in the form of a depth field map and a color image. The representative diagram of sensor operation is given in figure 2.[7]

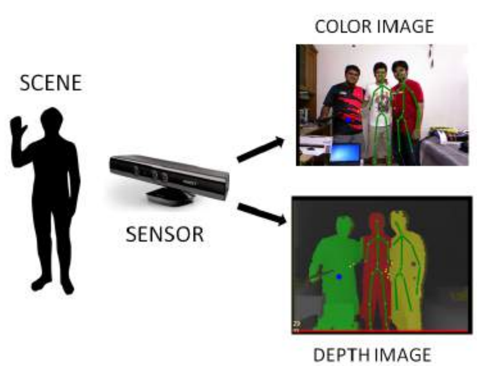


Figure 2. The data captured by the Kinect sensor as a color image and a color coded depth image

2) Software Interfacing Options: There are two options for using the Kinect for applications;

- OpenNI and NITE middleware ; is an open source initiative setup by PrimeSense. All of the necessary APIs available as a project called OpenNI with drivers for Windows and Ubuntu. NITE middleware is used for motion and skeleton tracking. [7]
- Microsoft SDK and Kinect for Windows API : Through it you can access the following sensor data in your application:
  - Audio data streamed out by the audio stream.
  - Color image data and depth image data streamed out by the color and depth streams.

  In addition to the hardware capabilities, the Kinect software runtime implements:
  - A software pipeline that can recognize and track a human body.
  - Tracks the skeleton of up to two people in front of the camera in real-time.
  - Integration with the Microsoft Speech APIs to implement a speech recognition engine .
  - A tight integration with the Face Tracking SDK which makes it possible to track human faces.

3) Advantages of the Kinect sensor:

- The microphone array in the Kinect sensor is intended to enable multiple scenarios, including:
  - Speech recognition
  - Raw voice data access
  - Sound source position tracking
- It processes color and depth data to identify up to six human figures in a segmentation map, which is a bitmap with pixel values corresponding to the index of the person in the field-of-view who is closest to the camera at that pixel position.
- It provides a depth field map which allows for the use of simple algorithms to recognize objects. This data can help to reconstruct 3D models and guide robots.
- Hands free input device; allows development of gesture based NUI which helps in reducing the spread of contagion by medium of touch and therefore improves health and productivity levels in the workplace.

4) Disadvantages of the Kinect sensor: The disadvantages of the Kinect sensor are listed below:

- Expensive in monetary terms, but if used in a proper manner,for example tracking human skeletons or building a 3D-map of the environment, justifies the cost.
- The sensor is required to be kept a distance of 5ft away from the user for optimal use. This makes it ineffective for deployment in small spaces, but can be taken as an advantage in places where such a distance is essential, like in an operation theater or in an industrial environment for operating heavy machinery.
- The sensor is not designed to be used under direct sun light.
- The sensor cannot track objects coated with infrared absorbing and reflecting materials.

5) Applications of the Kinect sensor: The three aspects of Kinect have various applications follows:

- Hands free input for computer.
- Allows for Speech recognition.
- Allows for Human skeletal tracking.
- Allows for Speaker identification and tracking his voice.
- Provides us with a Depth Map of the observable area, which can be used to make 3D scanner.
- Allows for facial recognition software based applications
- Allows for color recognition.

B. Texas Instruments MSP430 LaunchPad

LaunchPad is an easy-to-use development tool intended for beginners and experienced users alike for creating micro-controller based applications. LaunchPad has an integrated DIP target socket that supports up to 20 pins, allowing MSP430 Value Line devices to be dropped into the LaunchPad board. An on-board flash emulation tool allows direct interface to a PC for easy programming, debugging, and evaluation. Software development environments for writing and debugging software are free. LaunchPad can be used to create interactive solutions due to its on-board push buttons, LED's, and extra input/output pins for easy integration of external devices.[11]Figure 3 shows the actual MSP430 launchpad

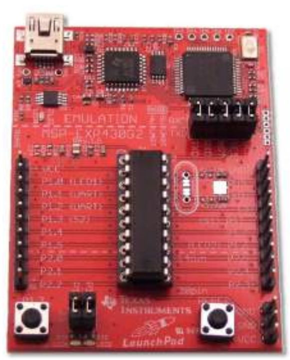


Figure 3. The MSP430 Launchpad device

1) Advantages of Using LaunchPad: The Launchpad has the following advantages:

- Its low cost; at $4.3 for the whole development environment inclusive of two micro-controllers, a development board, an IDE and a burner.
- Its easy to develop for; with TI providing ample documentation and a quick start series makes life simpler for the developer.
- It uses USB to serial communication layer to provide for quick and efficient interfacing between PC side applications and the micro-controller side firmware.
- Has a lot of on-Chip peripherals like A/D and D/A converter, a temperature sensor, oscillator, upto two 8-pin I/O ports etc making development even easier.

2) Disadvantages: The lack of value-line microcontrollers with programmable memory above 2048 bytes; making it useful only for minimalistic embedded applications.

C. Actuator

1) The Motor & its Driving Circuit: We used Geared DC motors of rating 24V, 0.6A no-load. We used a H-bridge based motor driver. An H bridge is an electronic circuit that enables a voltage to be applied across a load in either forward or backward direction. This circuit allows the user to run the DC motors forwards and backwards. H bridges are available as integrated circuits, or can be built from discrete components. The H-bridge we are using is an integrated circuit called the L293D.

III. Gesture Recognition

A gesture is a part of body language to undergo non-verbal communication in which visible bodily actions communicate particular messages, either in place of speech or together and in parallel with spoken words. Gestures include movement of the hands, face, or other parts of the body. Gestures differ from physical non-verbal communication that does not communicate specific messages, such as purely expressive displays, proxemics, or displays of joint attention. Gestures allow individuals to communicate a variety of feelings and thoughts, from contempt and hostility to approval and affection.[3], [4], [6], [12], [13], [14], [15], [16], [17], [18]

A. Types of Gestures

These can be:

- Single handed Gestures : Its a form of Dynamic gesture and are used for implementing unistroke gestures that uses just one hand to express the natural interface. These are simple to execute but cover a small range of gestures.
- Double handed Gestures : Its a form of Dynamic gesture and are used for implementing multistroke gestures which requires various angular and spatial movements of both the hands. These are more complex but carry out a wide range of gestures.
- Waving Gesture : They are dynamic gesture, for eg. hello and goodbye. These are standard gestures used in our day to day life.
- Posture : The term posture is refers to the appearance of the body. Posture can provide important information on nonverbal communication and emotions. Psychological studies have shown the effects of body posture on emotions. It can signal both the enduring characteristics of a person (character, temperament, etc.), and his or her current emotions and attitudes. Therefore, posture can be considered in the context of a given situation, and independently of it. Posture is a form of static gesture.
- Proxemics : It is the study of measurable distances between people as they interact with one another. Depending upon the distance between the two interacting bodies one can deduce the type of proxemics gesture thus can get the respective actions with respect to NUI.

B. Gesture Algorithms

Typically, the meaning of a gesture can be dependent on the following:

- spatial information: where it occurs;
- pathic information: the path it takes;and
- symbolic information: the sign it makes;

Using these parameters we can identify the critical points that help constitute a good gesture algorithm. A good gesture algorithm must have to:

- Be resilient to variations in sampling due to movement speed or sensing;
- Support optional and configurable rotation, scale, and position invariance;
- Require no advanced mathematical techniques (e.g., matrix inversions, derivatives, integrals);
- Be easily written in few lines of code;
- Be fast enough for interactive purposes;
- Allow developers and application end-users to "teach" it new gestures with only one example;
- Return the best matched list with sensible [0..1] scores that are independent of the number of input points;
- Provide recognition rates that are competitive with more complex algorithms used previously.

C. Importance of Gestures

Gestures are processed in the same areas of the brain as speech and sign language. It has been suggested that the pairing of gesture and meaning were adapted in human evolution "for the comparable pairing of sound and meaning as voluntary control over the vocal apparatus". As a result, it underlies both symbolic gesture and spoken language in the present human brain. Their common neuro-biological basis supports the idea that gesture and spoken language are two parts of the same system that underlies human communication. Hence to communicate effectively gestures play a vital role in expressing and communication of messages.

IV. Color Recognition

We humans perceive color when light enters the eye and is detected by the multitude of cells that make up the retina. The effect that light has on the individual cells is then sent to the brain where it is interpreted. The light entering the eye is the result of either light reflected by objects or light being emitted by objects.[19], [20]

In image processing we can digitize this light using a camera and each pixel behaves in similar manner as our retina cells. We can use this pixel information to design and develop color feature extractors to identify colors in a scene. Here we select an appropriate scale for each pixel, and extract color, identify the color of that pixel with respect to a selected scale. Using different algorithms to determine the dominant color feature. Using this extracted data describe the color distribution. The color data associated with the pixels is used to computationally determine the predominant color in localized areas of the image or the image as a whole. There are two popular method used for color extraction and identification; color Filtering Method and color Histogram Method.

A. Color Filtering Method

The process of extracting color values from pixels in an digital image and using these values to perform color segmentation in the image. color filtering is a powerful image processing tool finding important applications in the field of digital image analysis.

1) Advantages of color filtering : The advantages of color filtering are stated as follows:

- This can be used to generate three single color images (red, green and blue).
- Filtering allows us to infer the concentration of a single color in the image,
- Can be used in conjunction with other filters for image segmentation according to color.
- Can be used for Intuitive applications such as traffic light sensing.

2) Disadvantages of color filtering : The disadvantages of color filtering are listed as follows:

- Sensitive to Lighting conditions
- Thresholds need to be modified precisely for optimum results; making it dependent on the user set values found empirically without any standards.

B. Color Histogram Method

This method falls under statistical filtering. In these methods the pixel values are compared are compared with one another and values are modified according to the requirement. The mean of all the three RGB values is computed and a histogram is plotted showing the concentration of a particular color in different parts of the image.

1) Advantages of color histogram method: The advantages of color histogram method are listed below:

- It can be used for various applications such as dominant color finding.
- Can be used to identify color concentrations in different parts of the image.
- Pixel value comparisons are more accurate at dominant color prediction.
- Reliable and efficient for real time applications.

2) Disadvantages of color histogram method: The disadvantages of color histogram method are listed below:

- Suffers from lighting conditions variations.
- If two color means are very close, it makes the recognition of the dominant color difficult.

C. Applications of Color Recognition

Color recognition has the following applications;

- Searching color images by emotional concepts
- Image segmentation using skin color
- Traffic Light color sensing for applications such as a self-driving car.
- Image query system ,segmenting the images according to dominant color.
- Object avoidance by color coding the obstacles.

V. Methodology

The project was executed in the following manner:

- We started by studying user interfaces. The study resulted in the conclusion that all user interfaces have two important components; input and output. So we concluded that for developing our interface we need to select an appropriate input mechanism and establish a clearly visible output mechanism.
- Also during the study we established that for a robust implementation of the NUI, the input mechanism needed to be fast and reliable. We looked at touch screens, audio-based and vision-based systems, which are the popular input mechanisms for NUI.
- We found that touch based systems were fast and popular, whereas vision based systems were in early developmental stages and getting more popular. So during our research we decided on using a vision based system. While searching for sensors for the vision based system the literature suggested the use of 3D vision capturing device for fast and efficient implementation of the tracking algorithms required for developing a Natural user interface.
- We chose a commercially available sensor the Microsoft Kinect. The reasons for doing so are primarily three:
  - It provided us with correlated depth and color images at video frame rates.
  - It also provided us with a digital audio data stream with multiple digital signal processing options at source.
  - It provided access to the raw data streams.
  - The Microsoft SDK for the Kinect provides us with access to proprietary algorithms for real time skeleton detection and tracking.
  - It was available within reasonable economic range.
- Once the sensor was acquired a suitable development environment for the application was required. The manufacturer prescribed development environment was Visual Studio and the language preferred was C#. The development environment was very user friendly and allowed for rapid prototyping once the user was accustomed with the language. We studied the prescribed environment and over the course of the project we were able to utilize it efficiently for developing the application.
- Now a suitable actuator interface was required. After considerable research three techniques were shortlisted. The three techniques are listed as follows:
  - Parallel port interfacing with additional electronic components for direct availability of I/O pins.
  - Serial port interfacing with a microcontroller providing I/O pins.
  - USB interfacing with a microcontroller board providing I/O pins and Virtual Serial Port techniques.
  - A market survey led to the selection of the USB interfacing method using the MSP430 LaunchPad toolkit. The toolkit included all the required peripherals for application needed. The LaunchPad allowed the user to write embedded programs in C programming language in Code Composer Studio development environment.
- The communication interface between the PC application and the actuator circuit was developed. The USB to UART emulator present on the LaunchPad allowed for the use of the Virtual Serial Port technique to communicate with the PC application.
- Firmware for the MSP430 was written and tested. Once a robust solution was achieved; the sensor events in virtual world could be used to initiate action in the physical world.
- The task communication between the application and the actuator was developed and successfully implemented. Now the basis for developing the NUI have been established, we move forward to develop a robot platform for testing and demonstration of the NUI applications we will develop.
- The robot was a simple four motor drivable chassis made to demonstrate the effectiveness of NUI in commanding machines.
- We developed tested the following natural user interface based applications :
  - color recognizer based interface; which is able to recognize the three primary colors Red, Green and Blue; present in the scene and use this information to control the direction of rotation of the motors.
  - Hover button based interface; which placed buttons on the screen which the user could select using his hands while facing the Kinect sensor. These buttons were configured to set the motors into the desired rotation state.
  - Gesture recognizer based interface which is able recognize swipe gesture and pose-based gestures to control the direction of rotation of the motors.

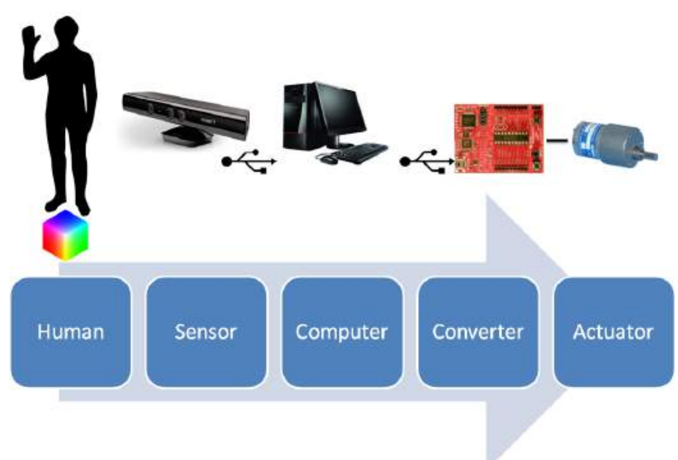


Figure 4. The above figure is a representation of the flow of project implementation

So to summarize the project we captured the gestures made by the user and colors in the scene using the Kinect sensor. The sensor gave us digitized version of the captured information which was further manipulated in the computer application to interpret certain events. On occurrence of these events a message is passed to the actuator(dc-motors) using a communication link provided by the MSP430 Launchpad. The whole process of the project is summarized in the flow diagram given in the figure 4.

VI. Results

During the course of the project various aspects of natural user interfaces were studied and understood. The study resulted in development of a platform which can be controlled using NUI. The applications developed during the study have the following NUI applications :

- Color recognizer based user interface
- Hover button based user interface
- Gesture recognizer based user interface

The platform developed to test the user interfaces during the project is shown in figure 5. The details about the applications are given in the following subsections. Also the first subsection will explain the communication process developed for the project.

Figure 5. The platform developed to test the user interfaces during the project.

A. The Communication Process

The communication process developed for the application utilizes the serial communication protocol configured with the following specifications:

- 1 Stop Bit
- No parity
- 8 Bit Data Packet
- 9600 baud Data rate

The flowchart of the communication process between the PC side software and the Firmware is given in Figure 6.

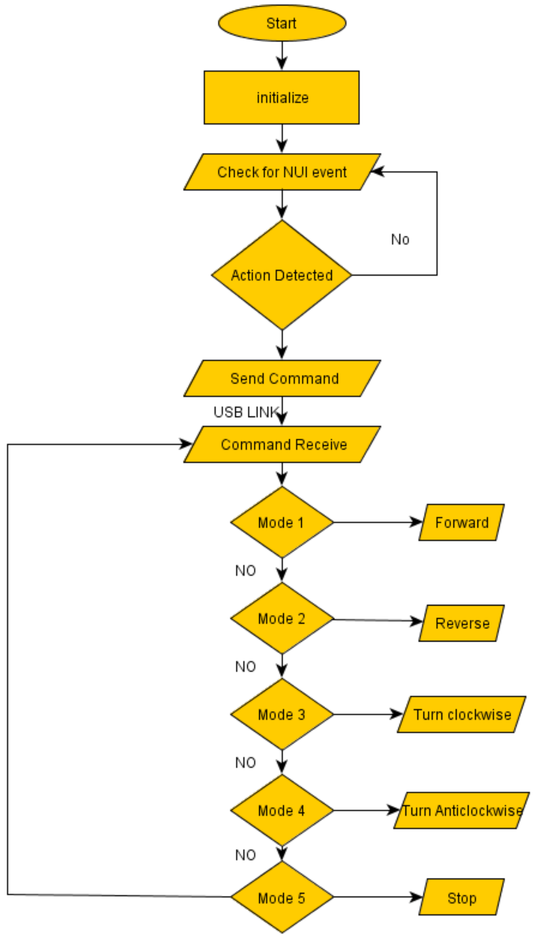
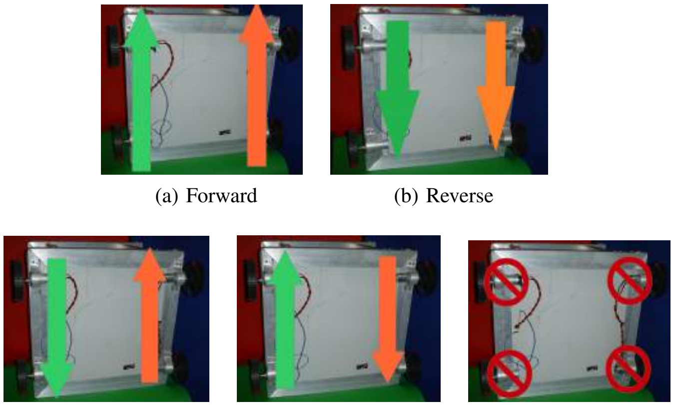


Figure 6. Communication Process Flowchart showing the different motor states and the whole event chain of the program developed.

B. The Motor States

The motors are attached at the base of the platform, as shown in figure 7 . The rotation of these motors can be controlled by the the program by sending commands to the msp430 Launchpad. The commands that can be sent by the application to micro-controller can cause the motor to run in the following five states:

Figure 7. The base panel of the Platform where the motors have been attached



(a) Forward (b) Reverse

(c) Turn Clockwise (d) Turn Anticlockwise (e) Stop

Figure 8. Motor Rotation State diagrams. The arrows represent the direction of rotation of the motor.

- Forward; All four motors run in forward rotation as shown in figure 8(a)
- Reverse; All four motors run in reverse rotation as shown in figure 8(b)
- Turning Clockwise; Two motors on the left run in forward rotation and two motors on the right run in reverse rotation as shown in figure 8(c)
- Turning Anti-Clockwise; Two motors on the left run in reverse rotation and two motors on the right run in forward rotation as shown in figure 8(d)
- Stop; All motors in braking condition as shown in figure 8(e)

C. Color recognizer based interface

Color recognizer based interface; which is able to recognize the three primary colors red, green and blue present in the scene and use this information to control the direction of rotation of the motors. It was observed that the dominant color present in the scene was usually red; hence it was assigned as the color for stop mode. The color green makes the motors run in forward state and the color blue in reverse state. The figure 9 shows the Application interface detecting the colors in the scene.

Figure 9. Color Recognizer GUI showing the image and its filtered version and the colour identified in the scene.

D. Hover button based interface

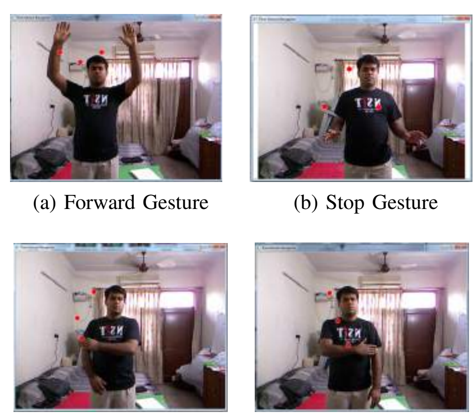
Hover button based interface; is able to recognize when a button created on the screen was pressed by the user. The different buttons have different functions and can be clicked by the user by placing his hand over them. The buttons are configured as follows : The red button as shown in figure10 stops the motor, the green button as shown in figure10 makes the motors rotate in forward direction, the blue button as shown in figure10 makes the motors rotate in clockwise direction and the yellow button as shown in figure10 makes the motors rotate in anticlockwise direction.To avoid false clicks a delay component is used, thus a user has to place his hand on the button and wait for the conformation animation showing the button was clicked. The figure 10 shows the screenshot of the application GUI.

(a) The Interface (b) A button being clicked

Figure 10. Hover button GUI screenshots . The images show the user touching the buttons created on the screen in software. The figure 10(a) shows the screenshot of buttons and the user. The figure 10(b) shows the screenshot user interacting with the red button, which stops the motors from rotating.

E. Gesture recognizer based interface

Gesture recognizer based interface; is able to recognize different poses of the user and also detect swipe gestures. The gestures developed are ; Forward rotation gesture : both Arms up, clockwise rotation gesture : right arm pointing left, anti-clockwise rotation gesture : left arm pointing right, and stop rotation gesture ; both arms down. The gestures are shown in action as being performed by the user in figure 11 with their captions detailing their action on the motor.



(a) Forward Gesture (b) Stop Gesture

(c) Turn Clockwise Gesture (d) Turn Anticlockwise Gesture

Figure 11. Gesture Recognizer Interface GUI snapshots.In this figure four different gestures that can be identified by the program are shown. In fig.11(a) the motor start gesture is shown, in fig.11(b) the motor stop gesture is shown,In fig.11(c) the motor turning clockwise gesture is shown and In fig.11(d) the motor turning anti-clockwise gesture is shown

VII. Conclusions

During the course of the project the concept of NUI is studied, understood and implemented successfully. A platform that could be controlled using a Natural User Interface is built and tested successfully. An application is developed to identify the color present in each image and consequently this result is used to actuate corresponding actions. An application is developed to identify the gestures posed (captured by the sensor) by a user and consequently the result is used to actuate corresponding actions. An application is developed to allow the user to click on buttons (hover buttons) created in virtual space by reaching out to them in physical space (captured by the sensor) and consequently the result is used to actuate corresponding actions.

VIII. Future Work

The study can be taken forward as follows:

- The application developed so far can be improved and optimized for better performance.
- The NUI developed so far use only a single technique for input. A Combined interface employing all the techniques can be developed.
- The application can be expanded in the future be able to control a humanoid robots.
- Different algorithms and libraries can be used to expand the applications into various fields.

Acknowlegment

Mentor : Dr. Prerna Gaur

I wish to express my sincere gratitude and indebtedness to my respected mentor, Dr. Prerna Gaur, Netaji Subhas Institute of Technology for providing me with much needed support and guidance to undertake this project and pursue it to completion with utmost sincerity. Her invaluable help and inputs paved the way for the project to be of respectable quality and insight.

I would like to thank my respected head of the department Prof. A.P. Mittal, for his continuous support and encouragement throughout the course. I would like to thank the department for providing me with the requisite resources, essential for the completion of the project. I would also like to thank Kunal Malhotra, Lav Lamba and Hemant Malik for their help and support during the development of this project.

References

[1] M. Fisher. (2012) Matt's webcorner- kinect sensor programming. Stanford.

[2] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*, 2007, pp. 159–168.

[3] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, 1991, pp. 237–242.

[4] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007.

[5] Open kinect research material. Open Kinect.

[6] K. Dotson, "The anatomy of the kinect algorithms explained," 2011.

[7] H. Fairhead, "Practical windows kinect in c-sharp," 2012.

[8] Coding4fun. (2011) Kinect blog.

[9] L. Gallo, A. P. Placitelli, and M. Ciampi, "Controller-free exploration of medical image data: Experiencing the kinect," in *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, 2011, pp. 1–Ş6.

[10] D. Fernandez. (2012) Kinect quickstart series. Microsoft Developers Network.

[11] *MSP430 LaunchPad (MSP-EXP430G2) Manual*, Texas Instruments, 2012.

[12] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "Simultaneous localization and recognition of dynamic hand gestures," in *Motion and Video Computing, 2005. WACV/MOTIONS'05 Volume 2. IEEE Workshop on*, vol. 2, 2005, pp. 254–260.

[13] S. Nagaya, S. Seki, and R. Oka, "A proposal of pattern space trajectory for gesture spotting recognition," *Proceedings of MIRu*, vol. 96, pp. 157–162, 1996.

[14] M. Oshita and T. Matsunaga, "Automatic learning of gesture recognition model using SOM and SVM," *Advances in Visual Computing*, pp. 751–759, 2010.

[15] T. S. H. Pengyu Hong, Matthew Turk, "Gesture modeling and recognition using finite state machines," in *IEEE Conference on Face and Gesture Recognition*, 2000.

[16] M. Reyes, G. Dominguez, and S. Escalera, "Featureweighting in dynamic timewarping for gesture recognition in depth data," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 1182–1188.

[17] G. Rigoll, A. Kosmala, and S. Eickeler, "High performance real-time gesture recognition using hidden markov models," *Gesture and sign language in human-computer interaction*, pp. 69–80, 1998.

[18] H. I. Suk, B. K. Sin, and S. W. Lee, "Hand gesture recognition based on dynamic bayesian network framework," *Pattern Recognition*, vol. 43, no. 9, pp. 3059–3072, 2010.

[19] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, 2011, p. 20.

[20] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, 2002, pp. 423–428.

Akhil Guliani is an Instrumentation and Control Engineer from Netaji Subhas Institute of Technology, Delhi; with a keen interest in embedded systems, robotics and, computer and mobile applications.