Hand Gesture Mouse Interface System

Verghese Koshy Puthukkeril[1], Shyam Sundar E H[2], Nandha Kumar P R[3]
Department of Computer Science and Engineering, Panimalar Engineering College,
Chennai, Tamil Nadu, India
rohangeorge91@yahoo.com[1], shyamsundareh@gmail.com[2], nandhakumar84@gmail.com[3]

***Abstract* --The requirement of simple, yet sophisticated user interface has driven the IT professionals to develop enhanced devices ranging from the basic punch-card system which were used for the earliest computer devices, to the widely used keyboard, mouse and the latest touch screen technology which are emerging on today's computers. These improved gadgets and technology basically reduce the invisible barrier between the everyday user and the system. However, these devices have their limitations and inconvenient for the user to interact with the system. To simplify this, a new system is developed which takes human gestures into consideration to perform the corresponding operation on the computer device. Hand Gesture Mouse Interface is developed to capture the hand gesture from the user through web camera. The primitive hand movement for the respective gesture is used. The relative positioning of the hand is being used in the system to ensure that the gesture recognition is made accurately. The Hand Gesture Mouse Interface simplifies the procedure which outperforms the mouse operation that is in vogue and also provides a low cost and easy to use interfacing for the user.**

***Index Terms* - Hand Gesture, Human Computer Interaction (HCI), Mouse Interface, Relative positioning**

I. INTRODUCTION

Since the genesis of computer, we have found better way to interact with the system, reducing the invisible barrier between man and machine, from the most basic punch-card system, to hardwired device like mouse, keyboard and to the current generation touch-screen technology. The system has become easier to interact and use. Computer is used by many people either at their work or in their spare-time and it proves invaluable to mankind.

The trackball, the joystick and the mouse are extremely successful devices for hand-based computer input. Devices such as controllers for games, security systems and television remotes require the user to either push a button or touch a screen for their functioning. Over the centuries, these devices have always been improved and enhanced by advancements in technology, making them easier to operate from the user's point of view.

The ability to recognize human gestures in a natural way have grown interest on computational vision methods in recent years. The images acquired from a camera or from a stereo pair of cameras as input is used in these methods. The main goal of these algorithms is to measure the hand configuration at each given time instant.

The real time operation using a controlled background makes it possible to locate the hand in an efficient manner. The background complexity which involves a considerable amount of distance from camera can be solved by [7]. However it must not impose restrictions on the user and on the interface setup. Hence a thorough study about the different types of gesture recognition is made using [8].

II. RELATED WORK

The sensors [1] play a vital role in identifying the gesture made by humans. This process is carried out by a number of recognition applications that make use of gloves or markers on hands or fingers [3]. The various movements of the hand's different parts are analyzed and categorized based on its biomechanical motions. Some algorithms even use hand segmentation based on skin colour [4].

The hand motions can be looked as complex combinations of the movements (rotations around different axes) of different bones at various joints [11]. Hence a Vision-based Human Computer Interaction [2] hand-gesture analysis is introduced which includes variety of gestures as input and a virtual hand as output to display the results. A vision-based hand tracking system [5] is employed that does not require any special markers or gloves and can operate in real-time on a commodity PC with low-cost cameras.

In the past decade, a lot of devices that were previously button-operated have been replaced successfully with touch-technology, in which the user generates a command by simply touching a screen. The touch screen system provides much more creative input gestures but with the additional cost, it may not be feasible for more users. Today, we recognize a definite possibility of reducing these human-machine interactions to a simple hand gesture to perform operations based on our desire. Hence we have described a Hand Gesture Mouse Interface system which basically uses only the web-camera.

III. PROPOSED METHODOLOGY

The system which we have proposed consists of software that obtains the inputs from the camera and sends it to the image enhancement subsystem. It improves the quality of the image that is captured from the web-camera. The system must subsequently, detect the hand from all other objects in the user's environment where he/she uses the given computer system. The system uses the HAAR [10] based algorithm to get the metadata which is used to differentiate the human hand from all other objects within the environment. Once the system identifies the hand in the environment, the tracker must use an algorithm which is less time consuming than HAAR based detection which is a very CPU intensive process. Hence, the requirement is to find a less complex and less CPU intensive process to keep track of the detected hand. So, we use CAMSHIFT [6] to track the hand after the system detects the hand via HAAR detection. The CAMSHIFT uses an image which is a combination of the image detected via HSV skin detection and edge detection using canny edge detection. This improves the tracking rate of the CAMSHIFT algorithm.

Once we manage to track the hand from the environment the system must use this information to find the relative displacement of the hand from the original position of the hand in the environment. Using this displacement value we move the mouse pointer on the system. To perform the click operation and other mouse operations we must use the convex-hull algorithm on the hand. The convex-hull detection algorithm and the convex-hull defects detection algorithm together are used to calculate the various ratios in the hand which can be used by the system to determine the operation which it must perform.

In this paper, we present a new approach for implementing a simple click operation which is triggered by the display of the user's palm.

IV. HAND GESTURE RECOGNITION

The logical design which has been made is turned out into a prototype. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

A. *PRIMARY COMPONENTS*

The primary technology or components which are used for the development of the system consists of three main components. The system consists of primitive image processing components which are used to implement the basic primitive operation of the system. It also consists of components which optimize Parallel processing, thereby improving CPU utilization in the case of image processing. The final component which is required for the system consists of a threading library which implements the producer and consumer scenarios which are needed for the hand gesture system. Here we use

- OpenCV libraries (Open Source Computer Vision Library)
- Intel® Threading Building Block
- Boost C++ Libraries

Fig. 1 represents the system architecture of HGMI which gives the complete overview of system functions with the environmental variables.

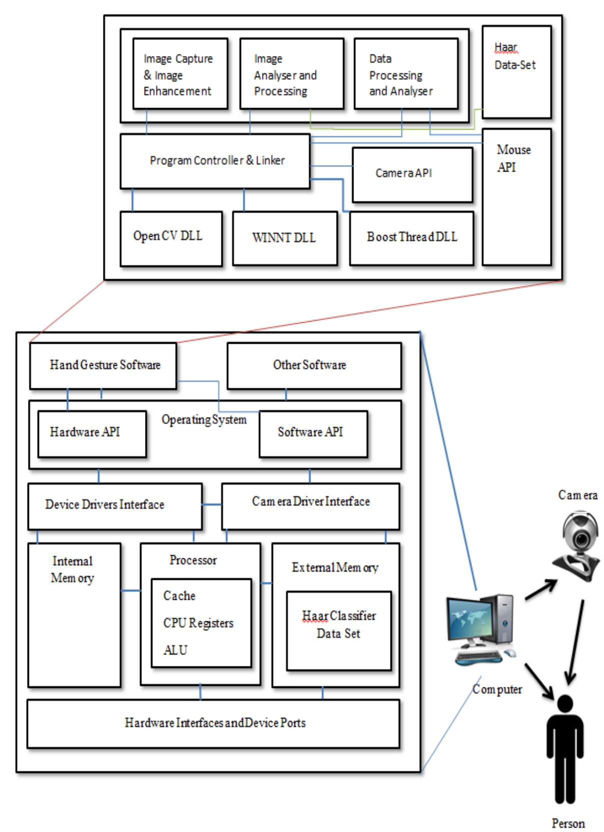

Fig.1. Architectural Representation of HGMI

The system will need the core hardware component of a webcam or any digital camera which takes the image of the environment in order to get the image feed which will be required by the program to ensure the proper connectivity between the webcam and the system. The Operating System will require the primitive low level instruction set which will be provided by the webcam device drivers. The Operating System using the device driver will provide the captured image frame, the program then fetches the frames for next stage of processing.

The system uses the open Computer Vision (openCV) libraries for the functioning of the various system components; the library provides real time camera support as well as predefined image processing support. Additionally, the libraries can be applied to both the x86 and x64 architecture. Also, since the libraries can be imported to C++, we are provided with fast and optimal program without the bottlenecks due to post compilation of the program. The downside is that since the programming is done in C++, the coding must be well coded with little or no room for error, since there is no just-in-time debugger and the playing field for the program is open to the system memory space.

For the Runtime environment the application will use the precompiled dynamic link libraries, the already available services of the operating system and its subcomponents.

B. IMAGE CAPTURE AND CORRECTION

Human gestures constitute a space of motion expressed by the body, face, or hands. The web camera is used to capture the gestures or movements made by the user for performing the operation. A number of producer threads are formed and the system gets activated. The gestures are obtained as live feed from the user and it is given as an input to the gesture system. The obtained video is first converted to an image. Since the image obtained as it is from the system is not suitable for performing operations, it is converted into the gray scale format.

Fig. 2 shows that the system converts the RGB image to gray scale HSV. Using the two sets of images the system then tries to isolate the human being from the environment. The image processing component also removes small scaled noise from the image, increases the brightness and contrast of the image to improve the overall detection rate. The modified frame is now stored on the main memory for more processing. Once the basic processed images frames are available, we can use skin detection technique to remove wide amount of noise in the environment as well as some bit of the background, hence, to a certain extent isolate the human being from the environment. The isolation is also aided using HAAR like detection.

The pixel density needs to be calculated in order to increase or decrease the brightness. The brightness and contrast of the image needs to be adjusted with the nearby environment in which it is operated. So the image processed lies in accordance with the requirements of the operation. The brightness and contrast can be adjusted using the below relation

```
new_image.at<Vec3b>(y,x)[c]=saturate_cast<uchar>(contrastValue*(image.at<Vec3b>(y,x)[c])+brightnessValue );
```

Where contrast Value and brightness Value Determines the image contrast and brightness.

The reason for image enhancement is improve the rate of detection of the various objects and their attributes. The developer's console which is used to test the given hand gesture system and check if the system is properly functioning. This will prove useful for the developer to track error, check if the system is properly functioning.

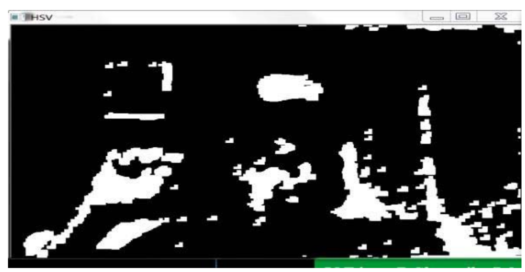

Fig.2. Screenshot of HSV Output of the image captured

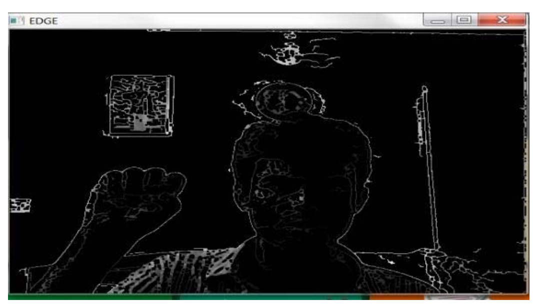

Fig.3. Detecting Edges of the image

C. IMAGE HAND DETECTION

The image is obtained after undergoing various enhancement and correction techniques. Now the hand must be located for recognizing the gesture made by the user. Accurate tracking of the hand is done using Haar-like features, while gesture recognition is done by computing orientation histograms [9] of video frames. Haar-like features are digital image features used in object detection. It recognizes objects in an image based on the value of simple features, instead of using raw pixel values directly. The main purpose of Haar- like features is to meet the real-time requirements, while providing speed, accuracy and robustness against different backgrounds and lighting conditions.

Fig. 3 shows that the image boundaries are being calculated using Edge detection. Each one's hand size varies with one another and hence boundary serves as a better idea for detecting one's hand completely. The camera may focus on all the objects that lie in its view and it is necessary to separate the human part from other objects in order to identify the gesture made by them. Hence masking process is being done with these detection methods and the background environment is subtracted from the human.

The tracking is possible with the help of positive and negative stereotype sets which are used in the Haar based training. The Haar based system uses positive and negative data sets (images) to make the XML file which consists of the threshold and detection values. The values may change from person to person and hence a type of learning needs to be given to the system in order to identify them correctly.

Both the skin detection and Haar object classification is used to improve the accuracy of gesture recognition so that no errors are being made during the process of identification of gesture. Since the time complexity for Haar based systems are really high, hence we use Camshift after the initial detection of the object. Camshift is a simple, computationally efficient face and colored object tracker. The mean shift algorithm operates on probability distributions. The fist and face detection of the user can be shown with the help of **Fig. 4**.

Fig.4. Fist and Face detection of the user

D. EVENT TRIGGERING

There are two parts to the event trigger operation. One is the analysis of the user and the other is the event trigger. The analysis provide the additional calculation which is needed like contour detection and defect in contour for complex gesture which may be incorporated in the system. The application now using the isolated hand calculates the ratio between the various fingers from the fixed point and calculates the change thereby determining the operations which have to be performed by the system.

The second part of the system triggers the corresponding event onto the mouse system or touch-driver based on the gesture. The application then uses the knowledge of the operation and determines the system application interface to trigger the event onto the corresponding device. It loads the event onto the device buffer; the Operating System in turn reads the buffer and triggers the device event as if it was the device itself. Now the hand can be used as a mouse and wherever our hand moves, the pointer also moves accordingly which is reflected in the monitor.

The simple click operation is recognized by HGMI and can be seen in **Fig. 5**.

Fig.5. Gesture Recognition of the user by HGMI

V. EXPERIMENTAL SYSTEM AND RESULTS

The HGMI is implemented using Visual Studio with the help of Open CV libraries. The experimental results are illustrated in Table I, II and III respectively. We have tested the gesture recognition on humans with different skin tones. Each person has their own skin color and hence the system must perform the operations accordingly.

Table I: Comparison Results of Skin Detection with and without enhancement

Operation	System Tracking Rate	
	Original System	Proposed System
Dark Skin Human	33.2%	56.3%
Mild Tone Skin Human	55.3%	60.5%
Fair Skin Human	50.1%	47.5%

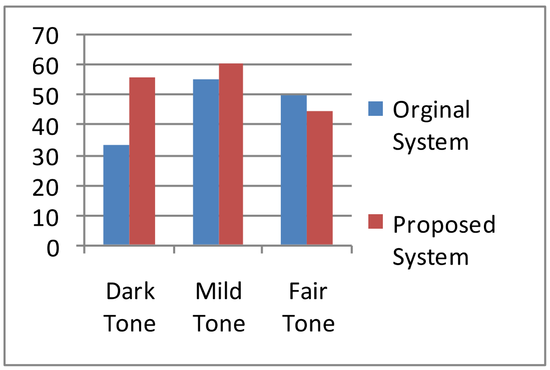

Fig.6. Hand detection based on skin tone

Fig. 6 can be used to infer that HGMI responds quite well to the people with dark and mild skin tone whereas the tracking decreases for people with fair skin tone.

Table II: Comparison between various Object Detection Techniques

Operation	Time (at 2.4GHz)	Accuracy (at 2.4GHz)
Background Subtraction	Less than 7 seconds (Primary delay) Less than 1 second for secondary detection.	34%
Skin Detection	Less than 1 seconds	30%
Haar Based Detection	4 - 20 seconds based on environment	64.2%

The mostly used object detection techniques are background subtraction; skin detection and Haar based detection. The testing was done based on accuracy and time.

From the table II it can be inferred that Haar Based object tracker detects or tracks the hand with better accuracy than skin detection and background subtraction method.

CAMSHIFT Tracking

Table III: Comparison of various input images based on CAMSHIFT Tracking

Operation	Color	Grayscale	Skin detected	Edge detected	Skin & edge detected
Tracking (static environment)	50.2%	40.3%	54.6%	42.2%	63.2%
Tracking (dynamic environment)	24.5%	20.2%	45.1%	35.2%	60.2%

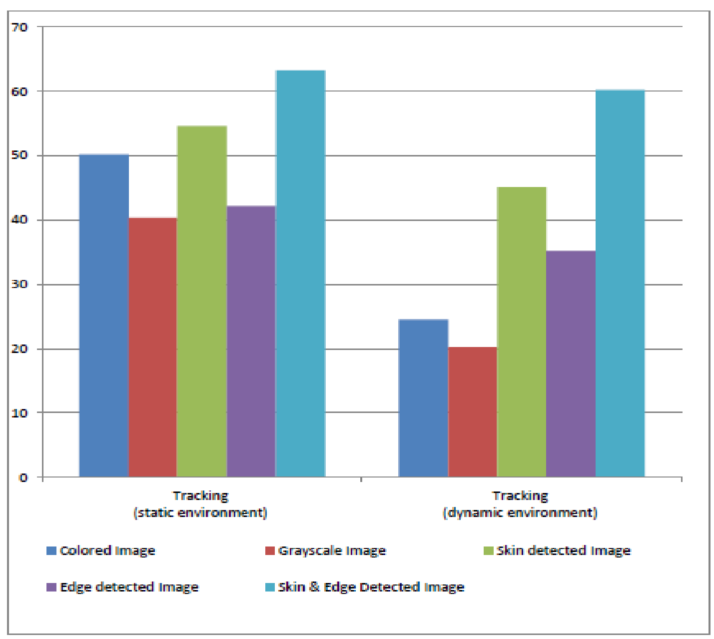

Fig.7. CAMSHIFT Tracking based on input image

From **Fig. 7** it can be inferred that the Edge detected image gets tracked easily under static environment. However it gets slightly reduced under dynamic conditions.

VI. CONCLUSION

In HGMI system, we have found way to interact with the computer using a web-camera to record the gesture and perform the hand gesture analysis and processing to mimic the basic operation of the mouse. The system is capable of tracking the fist and palm of a human in most scenarios. The HGMI system doesn't require any additional expensive hardware. Hence the cost of reproducing the system is simpler and much cheaper. It proves to be primitive and is capable of tracking and performing single left click operation on any object which is available on the system UI.

For our application we will trigger simple mouse operation on the system. The basic operations like click, double click, select can be made possible with the help of this system at the moment. To perform advanced operations, the system must be made to learn complex gestures, for the current system this is not possible because of the complexity of the job. Hence for the current working model of the project only one mouse operation is mimicked by this system.

VII. FUTURE TREND

The Haar dataset which is used for the detection is very primitive and hence the detection rate is little bit on the slower side. The work is under progress to detect and analyse more complex gestures like variation types of click operation, pinch zoom etc. So the future system will be having a better dataset for the Haar detection of fist and palm with much more accuracy and speed.

REFERENCES

[1] Joo-Ho Lee, Kazuyuki Morioka, Noriaki Ando, Hideki Hashimoto (2004) 'Cooperation of Distributed Intelligent Sensors in Intelligent Environment', Proceedings of IEEE/ASME Transactions on Mechatronics, Vol. 9, No. 3, pp 535 – 543.

[2]Beifang Yi, Frederick C. Harris Jr., Ling Wang, Yusong Yan (2005) 'Real-Time Natural Hand Gestures', Proceedings of IEEE CS and the AIP, pp. 92-97.

[3] Kenji Oka, Yoichi Sato, Hideki Koike (2002) 'Real-Time Fingertip Tracking and Gesture Recognition', Proceeding of IEEE Computer Graphics and Applications, pp. 64-71.

[4] Cristina Manresa, Javier Varona, Ramon Mas, Francisco J. Perales (2005) 'Hand Tracking and Gesture Recognition for Human-Computer Interaction', Journal of Electronic Letters on Computer Vision and Image Analysis, pp. 96-104.

[5] Shahzad Malik (2003) 'Real-time Hand Tracking and Finger Tracking for Interaction', project report submitted at Toronto university.

[6] Gary R. Bradski, Santa Clara (1998) 'Computer Vision Face Tracking For Use in a Perceptual User Interface', Published in Journal of Intel Technology, pp. 1-15.

[7] Mohamed Alsheakhali, Ahmed Skaik, Mohammed Aldahdouh, Mahmoud Alhelou (2011) 'Hand Gesture Recognition System', submitted at The Islamic University of Gaza.

[8] RafiqulZaman Khan and Noor Adnan Ibraheem (2012) 'HAND GESTURE RECOGNITION: A LITERATURE REVIEW', Proceeding of International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.4, pp. 161- 174.

[9] William T. Freeman, Michal Roth (1994) 'Orientation Histograms for Hand Gesture Recognition', Proceeding of IEEE International workshop on Automatic Face and Gesture Recognition.

[10] PavithraKannan, RohanPradip Shah, ThulasiSrinivasan, UzmaParveen S (2009) 'Hand Gesture Recognition Using Computer Vision', submitted at Visvesaraya Technological University.

[11] Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai, Tsukasa Ogasawara *(2003)* 'A Hand-Pose Estimation for Vision-Based Human Interfaces', Proceeding at IEEE Transactions on Industrial Electronics, Vol. 50, No. 4, pp. 676-684.