Gesture Recognition using a Touch less Feeler Machine

M. Shanmugapriya
Asst.Professor, Dept of IT
MP.Nachimuthu
M.Jeganathan Engineering College

R. Lokesh Kumar
Asst.Professor
Dept of IT
Bannari Amman Institute of Technology, Sathy.

V. Vimal Kumar
UG Scholar
Dept of CSE
Sri Ramakrishna Institute of Technology

ABSTRACT
This paper proposes a method to recognize hand gestures from a video and use the same for mouse controls. It uses coloured gloves to detect specific region of hand easily. By using this method, The occlusion problem due to colour can be treated easily. Hand gestures are recognized by tracing the RGB colours from the markers automatically from the proposed method. There are other methods in recognizing the hand gestures like using IPA (Infrared Proximity Array) and TOF (Time of Flight) Cameras. The proposed device is much economical as it doesn't need any additional hardware to implement. By this method both user's hands and markers are used as the input device to control the pointers and can play the games using the same.

Keywords
Touch less; Gesture; sign language; web camera.

1. INTRODUCTION
The Human Machine Interaction is moving in a way which cannot be stopped by any means. The Human Machine Interaction is much essential in day to day life and which needs the ease of use. The interaction earlier was done mechanically and evolved to hardware buttons, touch screens and now in the place of Touch less as the technology evolves. The Touch less performs the interaction by recognizing the gestures. The typical interaction can be carried out by means of the vision sensors. Hand gestures are extracted from the image is pre-processed and converted to specified commands. The commands are then mapped to the control motions of the cursor. The touch less interface is thus carried out by means of several techniques such as using the stereo cameras or by using the time of flight cameras. The time of flight cameras are used to produce the 3-D distance map of the image by which the input can be recognized. But the implementation of the TOF need more cost in commercialization. In the

IPA the drawbacks are like the constraints in the position of the fingers and palms while giving the input i.e., the hand should be placed in the predefined distance as mentioned as d-min and d-max. In this system the proposed model uses no external hardware and with low computational burden. The implementation of this model requires an ordinary web cam as shown in the fig 1.

2. Web Camera
Webcams (short for web cameras) is the designation for real-time camera (meaning the state at this very moment), whose image can be accessed or viewed via the World Wide Web, instant messaging program, or a video call applications. The term refers webcam technology in general, so that said web is sometimes replaced with other words that describe the scene shown on camera, for example Street Cam showing street scene.

2.1 Working of Web camera
A simple web camera consists of a standard lens, mounted on a circuit board to capture the image signal; casing (cover), including the front casing and casing cover next to a standard lens and has a hole in the casing front lens that is useful to insert an image; support cables, made of flexible material, one end connected to the circuit board and another end has a connector, this cable is controlled to adjust the height, direction and angle of view of a web camera. A web camera is usually equipped with software; this software takes the images from digital cameras continuously or in specified time intervals and broadcast through internet connection. There are several methods of broadcasting, the most common method is the software to alter images in JPEG file and upload it to the web server using File Transfer Protocol (FTP).

Frame rate indicates the number of images of software can capture and transfer in one second. For streaming video, it takes a minimum of 15 frames per second (fps) or ideally 30 fps. To obtain high frame rates, it takes high speed internet connection. A web camera is not necessarily connected with the computer; there are web cameras that have the webcam software and built-in web server, so they only need an internet connection. Web cameras like this are called "network cameras". It can also avoid the use of wires using radio links, Ethernet or Wi-Fi connections.

3. Touch less user interface
In front of a PC, the VGA camera is located instead of mouse, and the user controls the cursor by hand motion. While the interface recognizes hand motion and generates command to control cursor, the information detected by the camera is fully utilized to distinguish the hand gestures. The interface provides two different operation modes to control the cursor. First, the finger input mode utilizes the position information of an active index finger. In the finger input mode, the user rest his hand in front of camera and just point out the direction where s/he want to move the cursor, and then the cursor is exactly controlled to move along that direction. The other mode is the marker input mode. The marker input mode measures the colour of the marker while the user points his or her marker before the camera. For the click operation, predefined gestures are used in the finger input mode as shown in Fig. 8. The user folds his or her hand to close the gesture function.

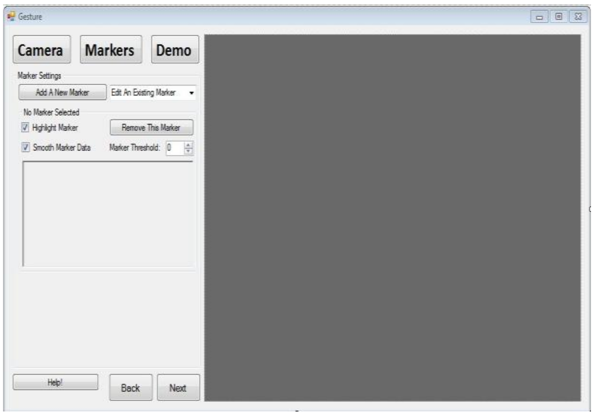

Fig 1. Touch-Less Interface

3.1.Finger input mode

To detect the finger input system uses the edge detection method and silhouettes. In this technique the device extracts the position of the hand from the back ground. Image is then compared with the predefined gesture which then maps the gestures to the commands assigned to each gesture. The drawback of the variation in different lighting conditions is fixed here by automatic adjustment of the brightness of the camera according to the background lighting condition.

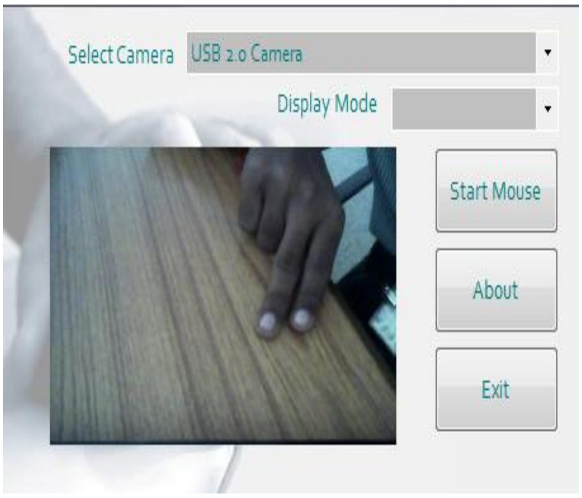

Fig 2. The interface to recognize the hand gestures

Finger input mode has the first step of operation is to find the position of the finger infront of the camera. The finger thus found is extracted for further motions such as click, navigation, pointer control. after selecting proper posture of the finger, the background of the finger will be recognized. Finger input mode operation is started by placing middle finger in front of the camera, thus navigation operation can be performed by means of middle finger. Other operations such as right click, left click, close, minimize can be done by other fingers. Markov models are used for the purpose of predefining postures involved in the finger input model.

Fig 3. System recognizes that no hand is selected, displaying wait for mouse.

The system is designed in a way that if the input(hand or marker) is not shown in front of the camera then the interface will inform the user about the status of the touch less interface and alerts the user about the back ground lighting conditions. From the fig 3. the alert message describes the status of the input before the camera.

3.2. Marker input mode

In the marker input mode colour of the marker will be recognized from the background colour easily as the pre processing converts the colour in to Cyan colour images. Then the detected marker should be stored in the cache for future use as the user can access the interface the marker which was already stored. User can also select different marker when using the touch less interface to make use of object at that time of accessing.

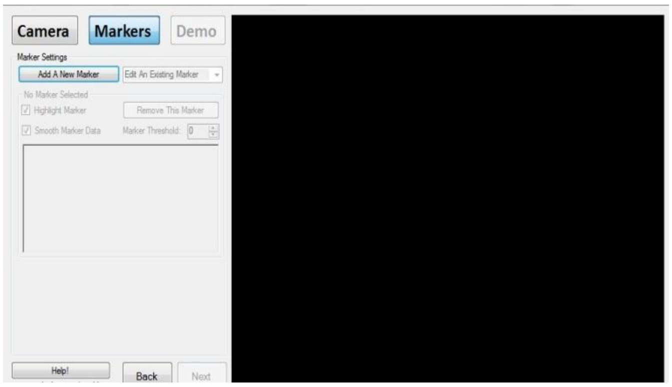

Fig 4.I nterface to use marker as input

4. Markov Models

The simplest Markov model is the Markov chain. It models the state of a system with a random variable that changes through time. In this context, the Markov property suggests that the distribution for this variable depends only on the distribution of the previous state. An example use of a Markov chain is Markov Chain Monte Carlo, which uses the Markov property to prove that a particular method for performing a random walk will sample from the joint distribution of a system.

In this system the pre-defined models are used instead of the markov models, here the system extracted the concept of markov images and without storing those images in the data base images and are stored in the temporary bin. The gestures which are used in the pre-defined models are as follows,

1. Moving mouse cursor
   1.1 pointer and middle finger should be kept together and start the program
2. double click action
   2.1 pointer finger should be moved left side; middle finger fixed
3. Right click action
   3.1Ring finger should be shown and then hide it
4. Maximize and restore action
   4.1 To maximize and restore thumb finger should be shown and hidden
5. close
   5.1 Close the hand for five seconds (seconds can be varied manually) and the program will be closed

5. IMPROVEMENT FROM EXISTING SYSTEMS

The system proposed here is implemented to concentrate on the limitations of the following areas and to overcome those limitations in a effective and efficient way. Most of the drawbacks are associated with the recognition of marker data, this problem is solved by the RGB & HSV colour variations.

A. ColourLib

- Improved HSV colour space partitioning model.perceived similar colours better. Potentially replace with a group clustering algorithm. Perhaps just refine the per-dimension bin counts, or replace the hash function.
- Use of lookup table instead of transforming RGB to HSV. It can be just terminated early if it's not in the lookup table.
- Reduced loop overhead of converting ARGB values into RGB values, then into HSV values, then into Binned HSV values, then finally into a hash # for colour lookup during marker update. Potentially use a lookup table for a subset of colours to avoid the math altogether.
- Improved HSV colour grouping, consider refining the per-dimension bin counts or using a different HSV colour-space partitioning model that better suits human perception of similar colours.

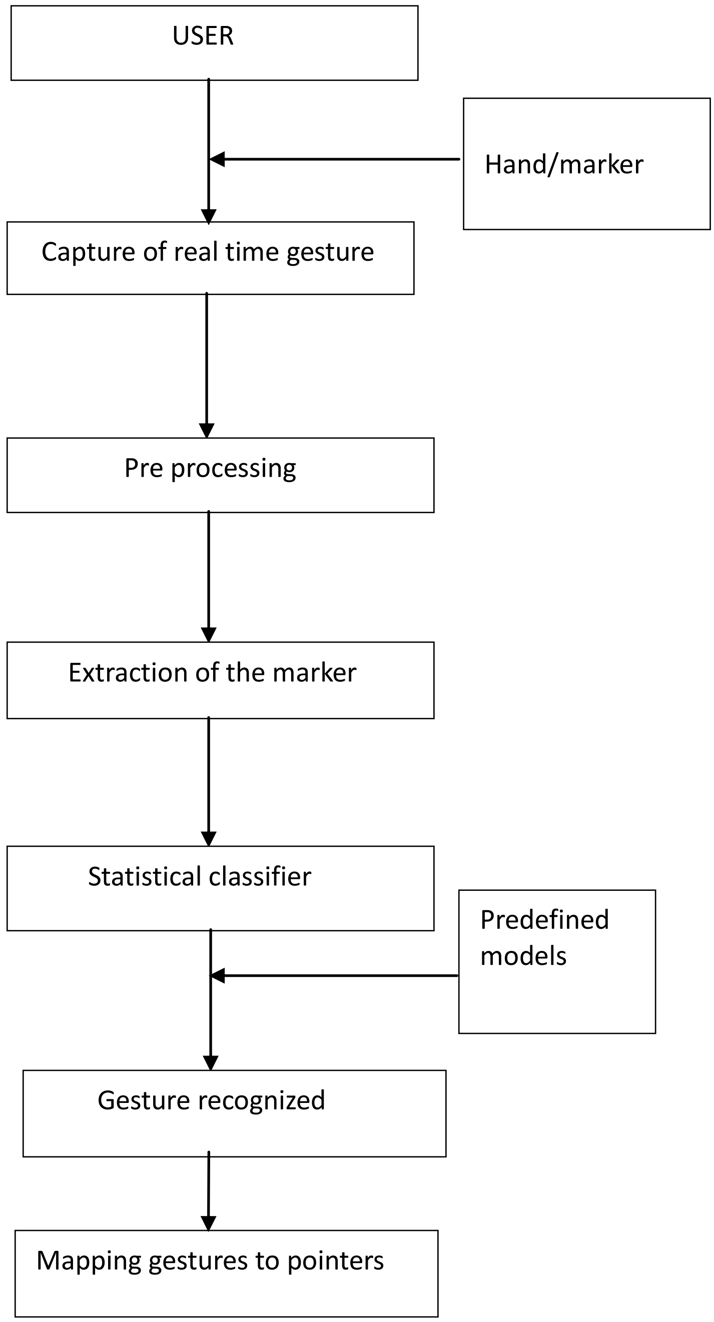

Fig 5. SYSTEM FLOW DIAGRAM

B. Marker

- Implemented a way of getting higher degree moments of inertia. Mostly, concentrated in the axis of least rotational momentum and the roundness factor.
- Allow the user to send a mask image with the add marker bitmap for arbitrary marker region selections."
- Extends or replaces alpha smoothing with exponential decay to provide smoothed marker data and reduce the marker jumpiness.
- Optimized threshold concept with a partial matching. Also, step threshold by numbers that actually make a difference, or just have sensitivity +/- buttons and increment functions

Expose smoothing factor as a public marker property.

Fix and improve the automated marker tests Standardize some marker colours, create an "auto-find makers" Improve the search bounds of a previously absent marker Improve the meta-tracking (cases where small numbers of pixels are missing from the middle of a marker, or are outliers of the concentration of pixels) Periodically/continuously adopt surrounding pixels of confirmed marker pixels Improved Marker highlighting Improved upon the raster scan algorithm used for marker updating.

Optimized the method for getting the marker appearance from a circular area of a bitmap; it could use hierarchical bounds intersection or something smarter than the current scan algorithm.

Optimize the values used to increment/decrement colour frequencies for marker appearance detection. This should be somehow based on signal/noise ratios.

- Improved the expected marker regions used for scanning on update. The marker's acceleration is considered, rather than just the velocity.

C. Touchless Manager

- Add functionality to save and load marker configuration files (reduce repeat training of the same marker, possibly provide auto config files for standard markers... will variant lighting allow for this?)
- Implement additional marker data such as Colour Average, Colour Space, Axis and Roundness.
- Add flood fill algorithm so it can add a marker with a few points in the Bitmap.
- Refine the marker tracked colours as it find colours around the marker.
- The representative colour doesn't always match the perceived colour of the marker.
- Provide subsequent examples of a marker appearance
- Have Touch less Mgr actually expose a way to get a list of the current markers
- Make a better exception for camera start failure
- Validated the Pixel Format of incoming images.
- Created a public interface for demo classes to implement, then allow the user to just invoke start and stop of a demo class on the library
- Standardized error handling and exception generation across the project.

6. APPLICATIONS

- This can be used in operation theatres
- War scenarios
- Sanitary places and operations
- High sensitive equipments
- Places where physical touches are difficult and impossible

7. CONCLUSION

The touch less interface thus proposed has highly effective throughput and was verified by using the group of common people. The interface can employ in many types of equipment in future as the technology grows. By making it as the platform independent model can able to use this all the devices where HMI is needed. The proposed system is highly economical and can be commercialized at low cost.

8. REFERENCES

[1] Niesink, L. Adding A Third Dimension To Multi-Touch Table Top Display Interaction, 2010

[2] Ryu, D. , Um, D., Tanofsky, P. , Koh, D. H., Ryu, Y. S., & Kang, S. (2010). T-less: A novel touch less Human-Machine Interface based on Infrared Proximity Sensing. In *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp.5220-5225). Taipei, Taiwan.

[3] Pickering, Carl A. Burnham, Keith J. Richardson, Michael J. Jaguar ,"A research Study of Hand Gesture Recognition Technologies and Applications for Human Vehicle Interaction", *3rd Conference on Automotive Electronics*, 2007

[4] Juan P. Wachs **,** Helman I. Stern, Yael Edan, Michael Gillam, JonHandler, Craig Feied, Mark Smith, "A Gesture-based Tool for SterileBrowsing of Radiology Images", Journal of the American Medical Informatics Association (2008; 15:321-323, DOI 10.1197/jamia.M24)

[5] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real–time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19 of *7*, page 780–785, 1997.

[6] Stenger, B. (2006). Model-based hand tracking using a hierarchical Bayesian filter, *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 28 (9), pp. 1372– 1384.

[7] **T**-less : A novel touchless human-machine interface based on infrared proximity sensing, Dongseok Ryu; Dugan Um; Tanofsky, P.; Do Hyong Koh; Young Sam Ryu; Sungchul Kang;Texas A&M Univ., Corpus Christi, TX, USA Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on 2010