Journal of Computer Science 10 (3): 382-392, 2014
ISSN: 1549-3636
© 2014 Science Publications
doi:10.3844/jcssp.2014.382.392 Published Online 10 (3) 2014 (http://www.thescipub.com/jcs.toc)

UBIQUITOUS PALM DISPLAY AND FINGERTIP TRACKER SYSTEM USING OPENCV

[1]Sathish Sekar, [2]Shriram K. Vasudevan, [2]Kaushik Velusamy, [2]Mrinal Purohit, Y., [2]Ganesh Alagappan and [2]Nivedita Venkatapathy

[1]Department of CSE, Texas A&M University, College Station, Texas, United States
[2]Department of CSE, Amrita University, Coimbatore, India

Received 2013-10-12, Revised 2013-11-02; Accepted 2013-11-22

ABSTRACT

Ubiquitous Computing is the idea of embedding computation onto everyday objects of the environment we live in. It’s the approach towards making technology and computation available everywhere thereby making people interact with abundant information around them in a more natural and friendly manner. Our project, the palm display system is one such approach of using one’s palm as the graphical user interface enabling the user to interact with it using his/her fingertip. We have implemented this project using computer vision techniques. In this study, we describe the details of the palm display system and explain several issues regarding the challenges in implementation and show the result using a sample photo viewer application. The main idea behind the project is to detect the interaction using shadow based techniques without the use of a fingertip marker.

Keywords: Image, Palm Detection, Ubiquitous Computing

1. INTRODUCTION

The problem domain we have chosen is mainly under ubiquitous computing which is a post desktop model of human computer interaction. We have used computer vision techniques to implement the intended system. To develop the palm display system without marker and run a sample application like photo viewer, where the user slides photos by interacting with the arrow keys displayed on his palm.

2. AVAILABLE TRENDS IN THE SAME AREA OF RESEARCH

Ubiquitous Computing is already known to be the vision of the future. It is a vast area which has immense scope for exploration and innovation. A significant amount of research is being carried out in detecting human palm and fingertip for the purpose of interaction with projected displays. With just a projector, camera and a processing device (mobile) umpteen number of day to day applications can be simulated wherever and whenever one needs, thereby eliminating the need for any other physical object.

When referring to the paper, new interface using palm and fingertip without marker for ubiquitous environment, it is understood that regarding the palm display system, our base paper by Kim *et al*. (2010) uses background segmentation initially to learn the presence of palm on the frame. Background segmentation involves significant amount of processing time. Initially the system has to learn the static objects on the background by analysing a few frames and then build a model which yields us the foreground object.

When coming to the next paper on Colour Based Hand and Finger Detection Technology for User Interaction, Kang *et al*. (2008) have discussed about detecting human hand based on contours. Contours are close curves on the frame and are of several levels.

Corresponding Author: Sathish Sekar, Department of CSE, Texas A&M University, College Station, Texas, United States

The most primitive level has no holes inside it and the human hand is one such contour in the frame.

With this understanding, the problem statement can be clearly defined.

The aim of our project is to develop a system which enables the user to interact with the graphical user interface on the palm using his/her fingertip without a marker. The prototype system developed will be simulated with a sample application like photo viewer where the user slides photos using the arrow keys displayed on his palm.

3. AIM OF THE PROJECT

- Fingertip detection without marker based on finding the local maxima of the various defect end points of the convexity defects in human finger
- Circular palm fitting to eliminate external noises
- Merging Blobs based on extreme point calculation thereby avoiding disrupter of interaction points when the fingertip moves over them
- Frame based Interleaving to differentiate between successive interactions and multiple interaction elimination
- Frame based threshold to avoid false interaction notifications

4. ASSUMPTIONS

- The object should be projected onto the human palm
- The human palm interface and the interaction with fingertip must be made in front of the camera and the distance between camera and the interface should be<= 60 cm. (depending on the resolution of the camera)
- The finger used for interaction should not be perpendicular to the projector
- Prolonged interactions with the same projected object without gap i.e., for single interaction will trigger successive interactions
- The interaction time for a proper event triggering is between 2 and 6 sec
- The interaction object should be bigger than fingertip

5. SYSTEM MODELLING

This section will provide idea on the proposed architecture through modelling.

5.1. Palm Detection Module

Figure 1 represents the model of the Palm detection module which runs in a loop till the palm is detected. It uses canny-edge detection algorithm.

5.2. Fingertip Locator

Figure 2 represents Fingertip locator module which uses contours to detect fingertip.

5.3. Event Tracker

Figure 3 represents event tracker module which is responsible for locating the area of interest and to identify the presence of an event (i.e., Fingertip touching the area of interest).

6. DOMAIN MODEL AND SYSTEM SEQUENCE DIAGRAMS

Figure 4 and 5 represents various individual entities, their inputs and expected outputs which are fed to the subsequent entities.

7. SYSTEM ARCHITECTURE

Architecture diagram shown in **Fig. 6** illustrates the tangible and in-tangible entities.

8. CLASS DIAGRAM

8.1. Algorithm and Explanation

8.1.1. Palm Detector Module

This is the initial module which is carried out to detect a human palming the frame captured by the camera **Fig. 7**. It's done using a series of cascading intermediate stages which are as follows:

- The source frame is converted to HSV color space and a skin based threshold is applied to eliminate all non-skin pixels from the frame
- The result obtained is smoothened to avoid sharp edges on the frame and it is then subjected to Canny edge detection algorithm
- The various edges are then processed in search of contours
- Several Level contours are obtained, from which the most primitive level (without any holes) is detected to give the outline of the human palm

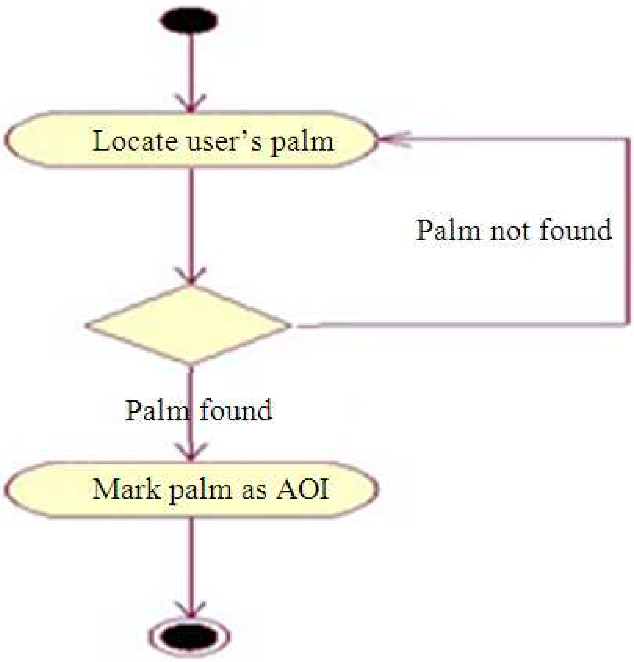

Fig. 1. Palm detection module

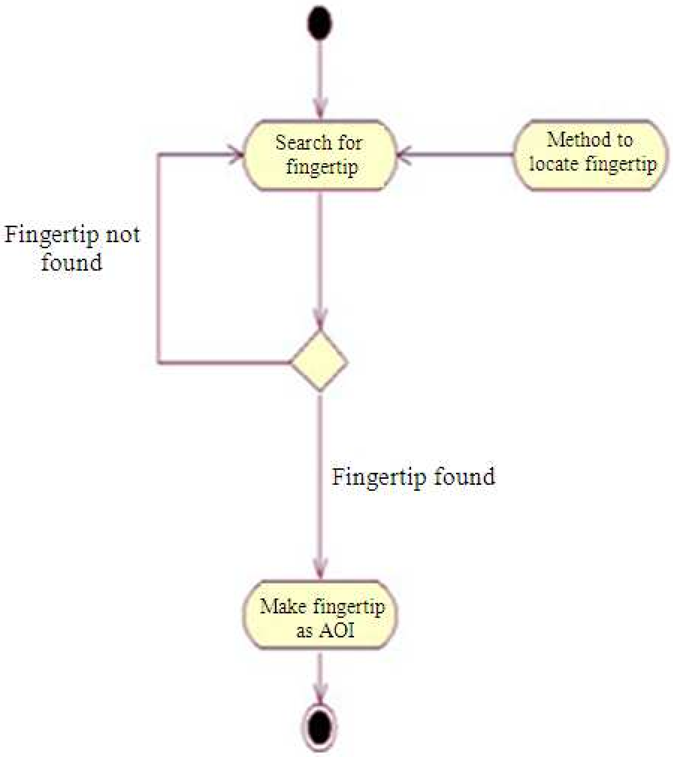

Fig. 2. Fingertip locator

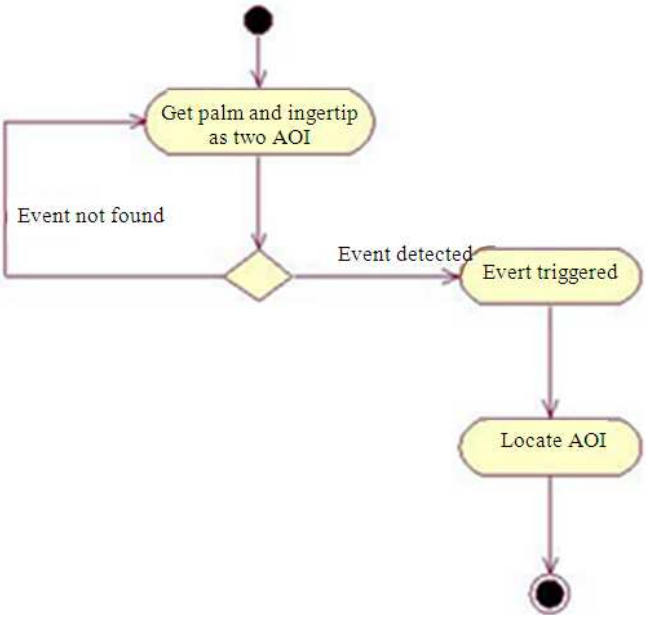

Fig. 3. Event tracker

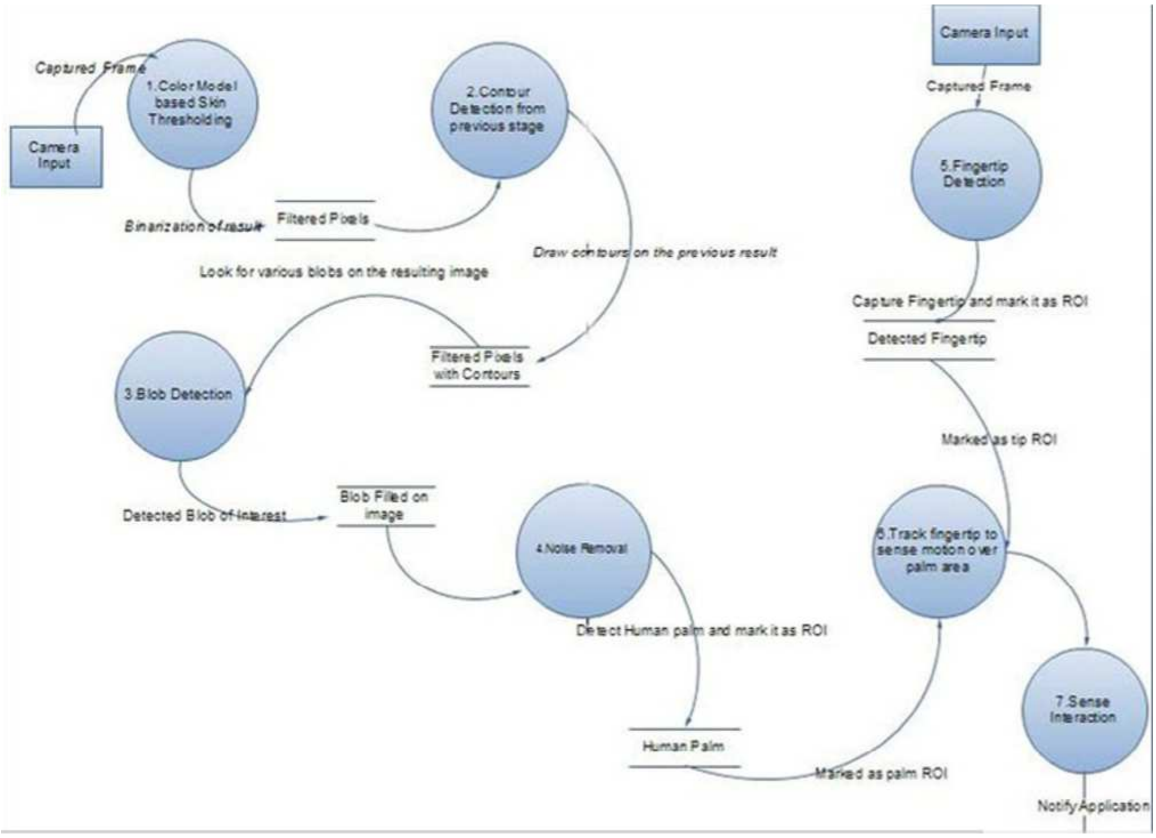
Captured Frame
1.Color Model based Skin Thresholding
Camera Input
Binarization of result
Filtered Pixels
2.Contour Detection from previous stage
Draw contours on the previous result
Look for various blobs on the resulting image
Filtered Pixels with Contours
3.Blob Detection
Detected Blob of Interest
Blob Filled on image
4.Noise Removal
Detect Human palm and mark it as ROI
Human Palm
Marked as palm ROI
Camera Input
Captured Frame
5.Fingertip Detection
Capture Fingertip and mark it as ROI
Detected Fingertip
Marked as tip ROI
6.Track fingertip to sense motion over palm area
7.Sense Interaction
Notify Application

Fig. 4. Domain model

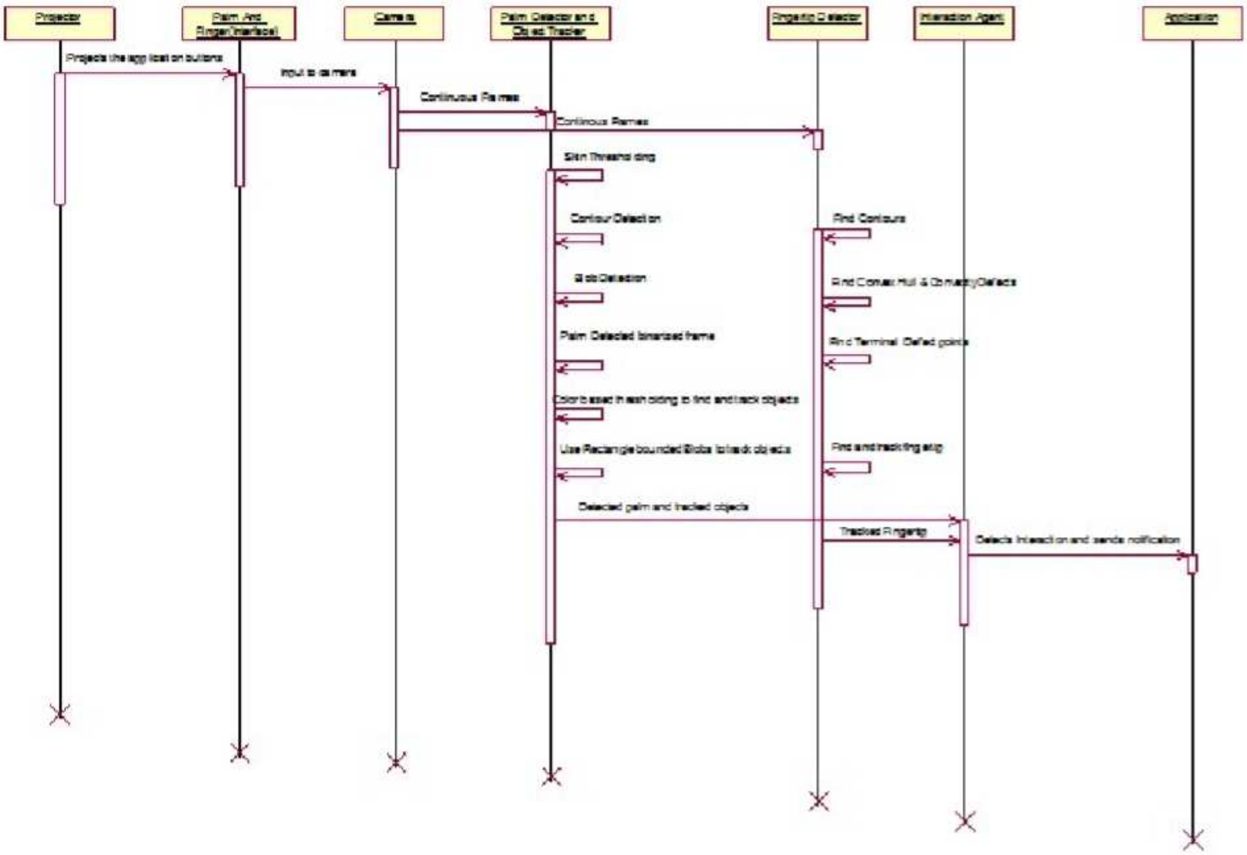
Projector
Palm And Finger(Interface)
Camera
Palm Detector and Object Tracker
Fingertip Detector
Interaction Agent
Application
Projects the app ical on buttons
Input to camera
Continuous Frames
Continous Frames
Skin Thresholding
Contour Detection
Blob Detection
Palm Detected binarized frame
Color based hash coding to find and track objects
Use Rectangle bounded Blobs to track objects
Detected palm and tracked objects
Find Contours
Find Convex Hull & ConvexityDefects
Find Terminal Defect points
Find and tracking fingertip
Tracked Fingertip
Detects Interaction and sends notification

Fig. 5. Sequence diagram

8.2. Fingertip Tracker Module

This is the other module which runs independent of that of the palm detector module. The detection of fingertip again involves a series of cascading intermediate stages as follows:

- Initially the source frame is threshold looking for dark regions. Since the projector makes a dark region on the side of the human finger and also a shadow falls on the palm due to the projection
- The previous result is subjected to contour detection looking again for primitive level contours
- From the detected contours, convexity hull points and hence the convexity defects are determined
- From the various convexity defects detected the ones outside the human palm blob (from previous module) are eliminated
- Again from the resulting subset of defects, the local maxima is picked up which manifests as the fingertip

8.3. Interaction Agent Module

The interaction agent works in parallel with the other modules trying to track the application's interaction points. The work of the agent is as follows:

- Detect the interaction points and fix their region (draw blobs) based on their property (color or position with respect to other points)
- As the interaction region will be broken when the fingertip is moved over it, small blobs of similar interest (same interaction point) are merged together based on the extreme points of the various rectangles (drawn over the broken blobs) taken together
- Once the fingertip detected is located over the interaction point area, it triggers the application notifying the respective interaction made (for example left arrow, right arrow or top arrow)

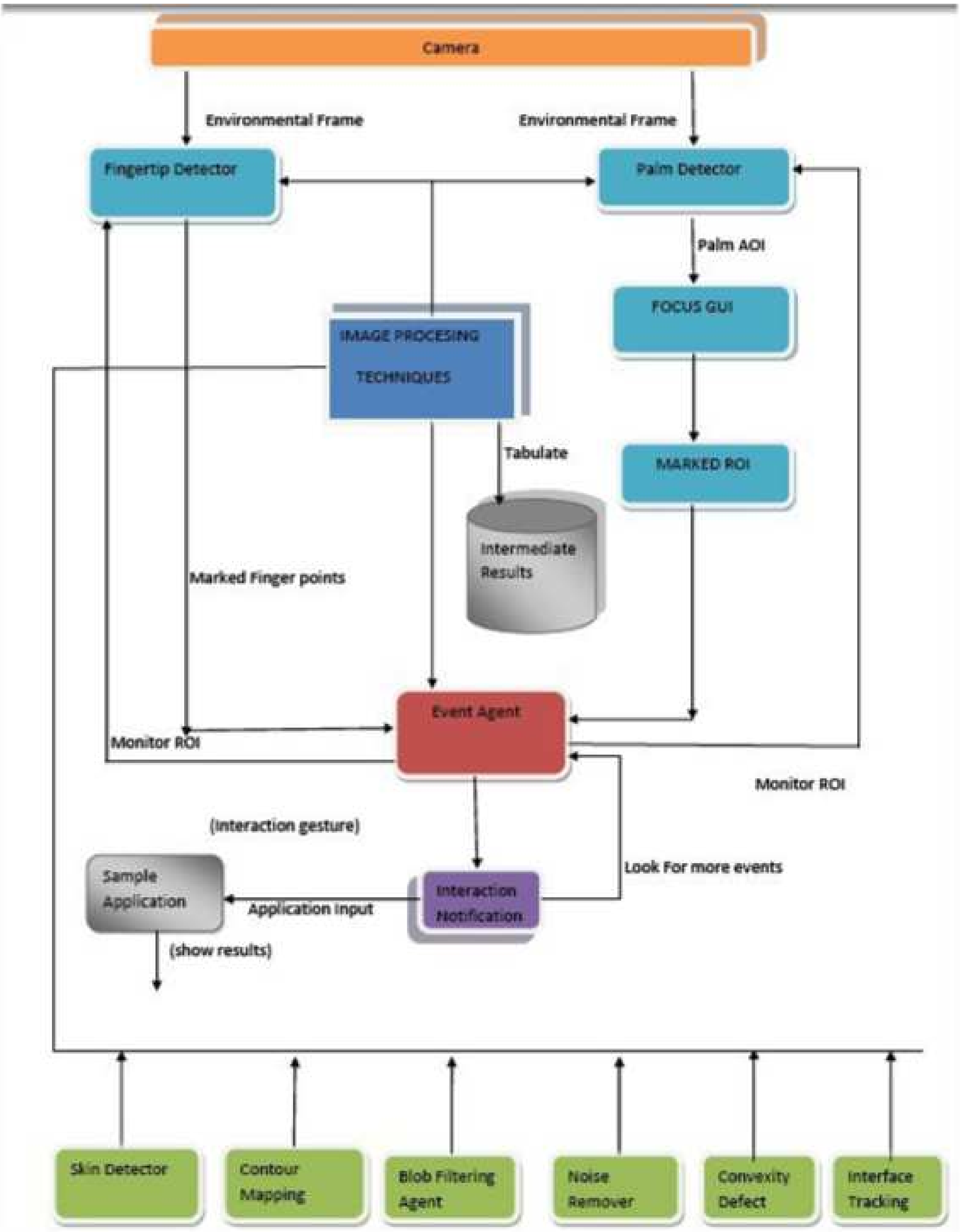

Fig. 6. Architecture

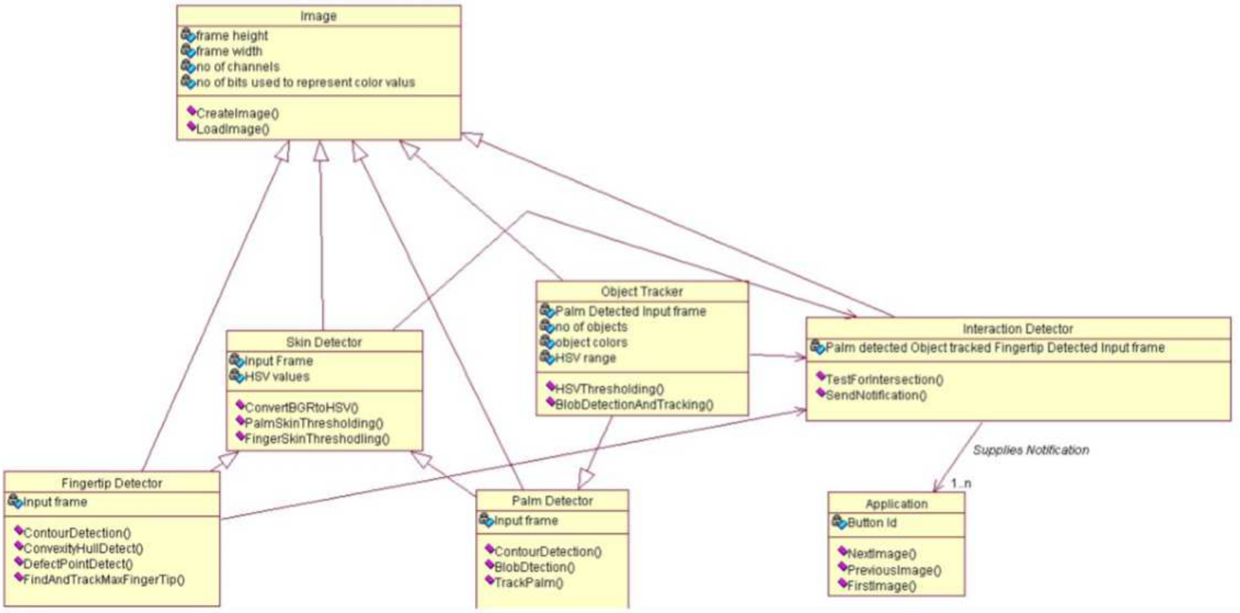

Fig. 7. Class diagram

- It also keeps track of the number of frames of successive detections in order to avoid continuous triggering of the same point touched
- It also differentiates between continuous interactions detected in successive frames and interleaved interaction actually made by the user. This is done by checking with a threshold value of frame count that can map to an interaction between 2 and 6 sec
- The system is frame based rather than being time based. It is more robust because the system actually processes it frame by frame

The application module actually shows the working of the entire system taking a sample application. It has the following functionality:

- It keeps track of the current state of the application
- Once it is triggered by the Interaction Agent, based on the interaction made, it changes the state of the application. For example, a photo viewer application slides the photo either to the left or right

9. COMPLEXITY ANALYSIS

The following points describe the complexity of the project:

- When the finger is moved over the palm, the curve or edge property doesn't exist anymore making it complex to track the fingertip. The reason is that the finger color merges with the human palm color
- Once the user interacts with the application point, the location of that particular application point is lost in that particular frame making it difficult to detect if the user has made an interaction. The reason is that, the processing is done frame by frame without retaining any information from the previous frames. Retaining information wouldn't help again because the user would have moved his palm to a different location
- The entire system would be in motion because the user will be moving both his palm and fingertip as keeping either of them stationed would be a constraint too costly
- There will be various convexity defects on the frame and the fingertip is just an element of the bigger set.
- There may be various noises which have properties similar to skin color or interface point color and hence have to be removed
- Since processing is done frame by frame, whenever a user interacts, the interaction will be present in a set of frames and hence the system must not trigger multiple interactions. On the contrary, if the user actually makes successive interactions, the system should be able to differentiate this

10. TEST REPORT AND ANALYSIS

The various modules of the system were tested independently and then 1a complete integration testing was performed. From the analysis of the test report, it is found that the system has an accuracy of around 80% and the user would make an interaction again in case of a failure. So the system can be used with many day to day applications.

Using a marker instead of fingertip makes things simpler and accurate but demands the necessity of carrying the marker along where ever the user goes. The analysis also tells us that the system has to be adjusted in case of a projector change or color mode (within the projector) change. Regarding the GUI to be projected on the users palm we learn that the interface points should be slightly bigger than the fingertip or else the information of the GUI is lost in the frame where the interaction is made. Also the GUI points should be separated by significant distance for comfortable interaction. The analysis also tells us that the palm should not be very far from the camera (dependent on camera resolution) and the user is supposed to use the system the way he would use a touch phone held in his palm.

11. PERFORMANCE STUDIES

11.1. Application for Test

The sample application we took for testing the working of the system is a simple picture viewer application with 2 interface points or colored arrows. The colors we used for testing were red and yellow. Interaction basically involves moving your fingertip over

One click mode and Multi Click mode. One click mode is a mode in which the system recognizes the user’s interaction as a single click or interaction even though the user continues his interaction or sustains his interaction. I.e., in a situation where the user may want to click an object twice he’d have to click the interface point once and should remove his fingertip and then must perform the next interaction.

In multi click mode the user can sustain the interaction for multi touch that is the user can perform 2 interactions on a single interface point by resting his fingertip over the interface in a sustained manner.

12. TEST ENVIRONMENT

The testing environment plays a major role in the process of identifying the efficiency of the system. Lighting conditions and the user’s interacting style contributes maximum towards the efficient working of the system. The test process consisted of 10 test cases and were conducted at each of the 2 different lighting conditions namely laboratory light and dark room conditions. The test process consisted of 10 test cases with 6 users having varying interaction speeds and varying skin tones in 3 varying projector conditions- Dynamic, Standard and Natural projector modes (Sony projector). We found that the HSV values that we use for identifying the colored arrows vary widely based on the projector modes. So it’s of high importance to select a particular mode and to stick with it. Dynamism can be added with the cost of prompting the user for the mode each time the user operates the system. Lighting conditions apart from the projector light also play a critical role in determining the accuracy of the system. With only the projector light, i.e., with no room or natural lighting, the system was highly responsive and gave good results in the range 85-90%, since the darkness or the sharpness of the shadow is high. In case of natural light, the system showed medium results i.e. in the range of 80-85% hit rate since the shadows created due to noise factors are present. However we can improve the efficiency of the system by trying to focus the camera on the palm region alone. Test cases also included the use of more than 2 arrows for incorporating few more functions. We found that, on increasing the number of arrows, the users tend to accidentally make a wrong interaction since the user has to move the finger over one of the other arrows in order to interact with the desired arrow. The size of the arrow should at least be a little bigger than the size of the fingertip or else the camera could lose track of the arrow in the frame where makes the interaction.

13. CAMERA BASED ANGLE TESTING

We evaluated the performance by placing the camera at three different angles acute, obtuse and perpendicular. The acute and obtuse angles had more amount of shadow formed due to the fingertip but there were mild fluctuations in HSV values of the interface colors. The tabulated results of the performance evaluation based on camera angle are shown below.

14. RESULTS

The following are the snapshots of the results obtained: **Fig. 8** is the final output of the prototype wherein the arrows are used to slide through the photo viewer application. **Figure 9** is the complete snapshot of the fingertip detection and its visual display on the computer. **Figure 10** is the output of the canny edge detection algorithm which helps to determine the boundary of an object. **Figure 11** is a depiction of object depth being protrayed perfectly, in this case a human hand. **Figure 12** is a portrayal of the human hand which is the main GUI of the application. **Figure 13** is the filling of the object based on object depth determined in **Fig. 11**. **Figure 14** is the final output of palm detector module. **Figure 15 and 16** are live working modules of the palm detector.

Table 1 discusses the statistics of the accuracy rate as determined using the palm detector for different angles as inputs for test cases.

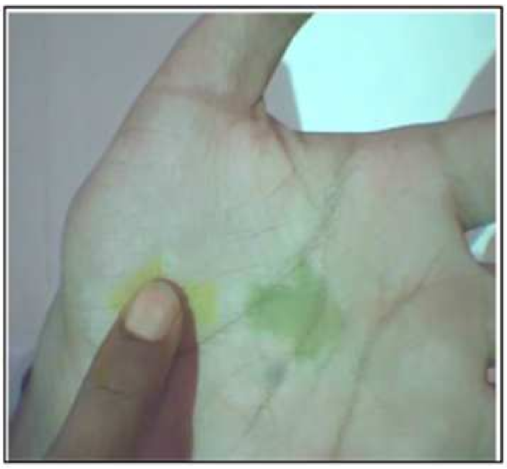

Fig. 8. GUI of photo viewer application

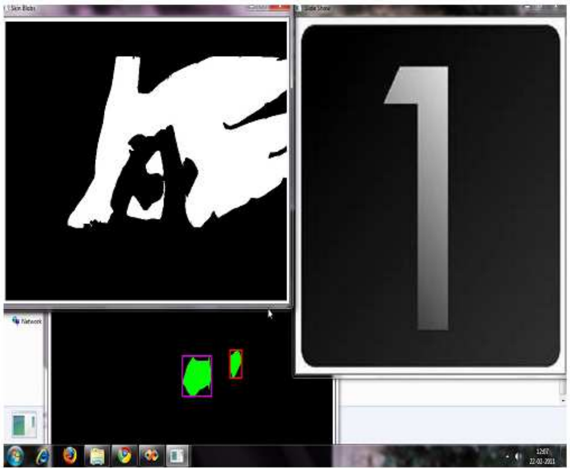

Fig. 9. Complete snapshot

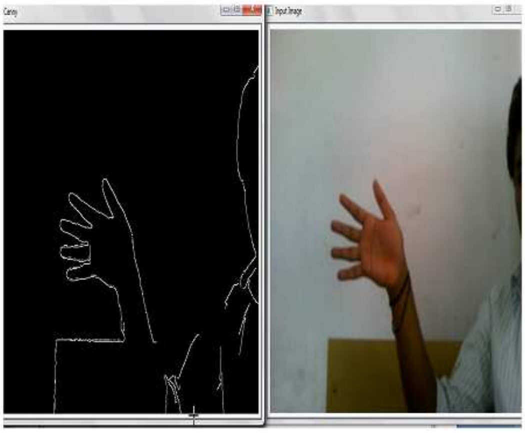

Fig. 10. Output of canny edge detection algorithm

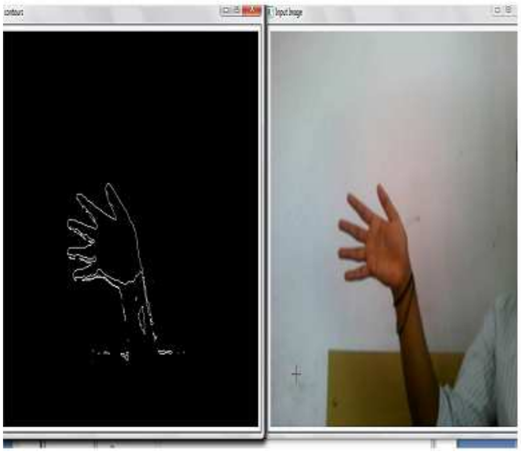

Fig. 11. Output after Drawing Contours

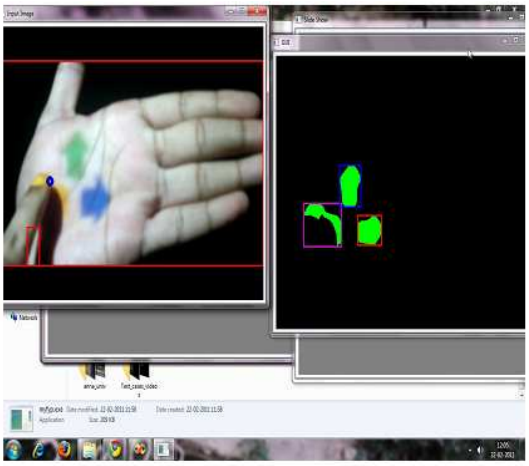

Fig. 12. Rectangle merging to track GUI

Table 1. Results

	Number of test case inputs	Number of interactions	Number of failures	System accuracy rate
Acute angle	10	65	11	83.07%
Perperdicular angle	10	80	14	82.50%
Obbuse angle	10	70	12	82.85%

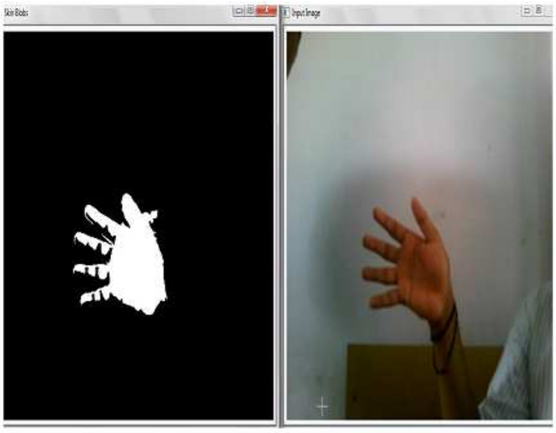

Fig. 13. Blob detection and filling

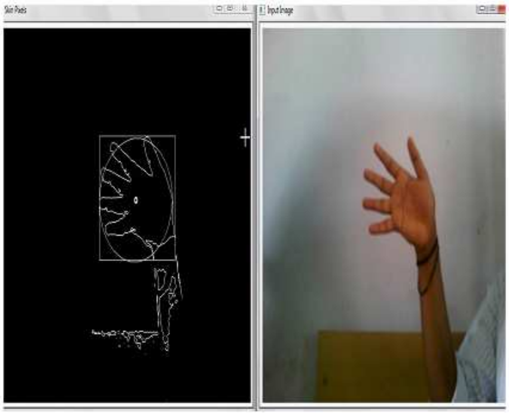

Fig. 14. Final Output of palm detector module

Thus the system was designed with a GUI projected on the user's palm and the user interacting with his fingertip. The working and usage of the system was tested using a sample application running on top of it. We have used a photo viewer application where the user slides the photos depending on the arrow key he has interacted with.

Thus the prototype system developed is a very effective one in a ubiquitous environment where the user can simulate whatever application he wants on his palm thereby eliminating the need for the purpose of the real world object.

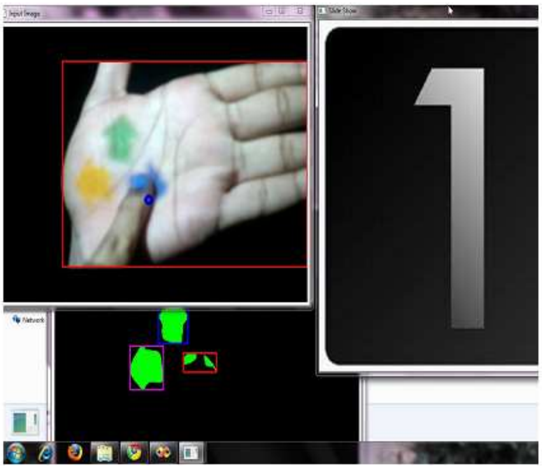

Fig. 15. Fingertip detection (blue circle)

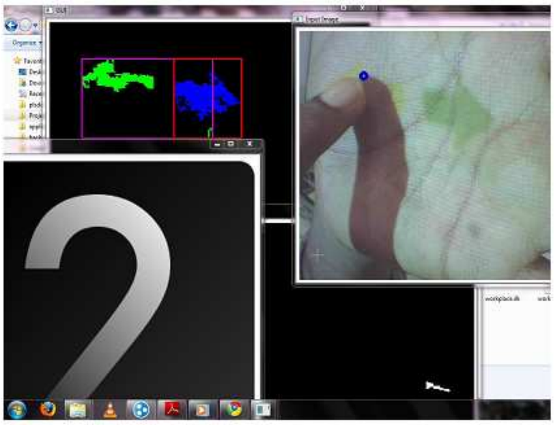

Fig. 16. Working of system with live input from projector and cam inside lab

For example, if one can build a calculator application he won't have to carry a calculator along with him and since we have detected the fingertip without the marker he/she can interact with any projected display.

15. PROBLEMS FACED

- The GUI color may vary depending on the color wheel used in the projector and hence the application has to be written taking into account the color wheel used in projector

- There will be no dark regions and shadows when the angle between the camera and projector is close to zero and hence no fingertip will be detected
- The input to the system is just a 2D frame and hence it is very tough to detect a touch sense. Thus it is compensated by frame count and thresholding
- The user interface points disappear when the fingertip covers the entire area of the interface points and hence they are supposed to be slightly larger than that of the fingertip area

16. CONCLUSION

The main purpose of ubiquitous computing is to carry technology everywhere, integrate them onto real world objects and try to eliminate the need for the real world object in order to use its functionality. A lot of research is being carried out in the area of Wear Ur World (WUW) by which the user can carry his world with him. This WUW comprises of a palm sized projector, camera, Bluetooth device and a mobile computing device. With just this WUW the user can simulate and use whatever application he wishes to and wherever he wants. Our system helps him to simply use his palm as the interface and his fingertip as the interaction tool to perform his actions and thus makes things look more friendly and natural to him. This system can be integrated to work with the popular android mobile device/IPhone and a palm sized projector so that the user can carry this system wherever he/she goes.

17. REFERENCES

Kang, S.K., M.Y. Nam and P.K. Rhee, 2008. Color based hand and finger detection technology for user interaction. Proceedings of the International Conference on Convergence and Hybrid Information Technology, Aug. 28-30. IEEE Xplore Press, Daejeon, pp: 229-236. DOI: 10.1109/ICHIT.2008.292

Kim, S., S. Takahashi and J. Tanaka, 2010. New interface using palm and fingertip without marker for ubiquitous environment. Proceedings of the 9th International Conference on Computer and Information Science, Aug. 18-20, IEEE Xplore Press, Yamagata, pp: 819-824. DOI: 10.1109/ICIS.2010.110