Real-Time 3D Markerless Multiple Hand Detection and Tracking for Human Computer Interaction Applications

Soumita Ghosh, Jianmin Zheng, Wenyu Chen, Jane Zhang, Yiyu Cai

Abstract

In this paper we present a purely vision based implementation of markerless hand detection and tracking system which effectively detects and tracks the hand positions irrespective of hand orientation. A shape based detection algorithm using a new line approximation technique followed by an adaptive minimum distance classifier based tracking technique is implemented. The technique is very generic and can be practically used in all types of immersive and semi-immersive environments like simulations, three-dimensional (3D) games, visually assisted medical surgery and other human-computer interactive applications.

The proposed technique overcomes certain limitations of glove based, handheld object based and marker based techniques for a truly immersive experience. The technique also overcomes the main limitation of previous vision based hand tracking systems by providing much larger degrees of freedom essential for effective virtual reality systems. The novelty of the technique proposed is that it separates the detection and tracking parts and therefore allows tracking of multiple hands in a 3D space for real-time interaction.

1 Introduction

With the tremendous growth in computer technologies new challenges have emerged in Human Computer Interaction (HCI) which requires existing approaches to reach beyond isolated technologies and collectively use all modern advancements to achieve a final goal. Virtual Reality (VR) can be perceived as an extension to HCI where the users' sensory impressions of the real world are blocked. As part of auxiliary technologies of VR, tracking devices allow the

virtual reality system to monitor the position and orientation of selected body parts of user. Similarly, a three-dimensional (3D) view also provides depth perception to the user but does not immerse the user in the context; rather the user can interact with the system and can move around and view the 3D representation. In all these applications user interaction is of utmost importance and undoubtedly the users hands are the most effective and natural means for manipulating objects in a 3D space.

Previous attempts at hand tracking have used gloves, markers or handheld objects. Hand gloves are cumbersome for the users and need to be customized for each user. Markers and handheld objects are not natural for video gaming environments when children are involved. Handheld objects are also not effective in fully immersive environments.

The main motivation behind building this system is to explore the VR auxiliary technologies like the tracking devices and incorporate certain tracking methodologies in a 3D world where user can use his or her own body parts to effectively communicate rather than using any external medium to interact with the environment. Among human tracking based interaction systems hand tracking systems have their practical use in almost all science and engineering fields.

2 Relevant Work

The hand glove based techniques can be broadly classified into arm extension and ray casting based techniques. [Bowman and Hodges 1997] discussed these existing techniques for grabbing and manipulating remote objects in immersive virtual environments. The arm-extension technique required the virtual hand of the user to be extended to desired length while ray-casting technique used virtual light ray to grab objects. The rays direction is specified by the users hand. The main drawback of the ray-casting technique is that manipulation is difficult as it is not hand-centered. On the other hand, grabbing is difficult in arm-extension techniques.

Among handheld device based techniques, the Voodoo dolls technique developed by [Pierce et al. 1999] was quite popular. It was a two-handed interaction technique for manipulating objects at a distance in immersive virtual environments. In this technique the user dynamically created dolls which were hand held copies of objects in the scene and the objects in the scene could be moved relative to each other by moving the dolls with respect to one another. This technique allowed both visible and occluded objects to be selected but allowed only relative movement of one object with respect to another. The World in Miniature (WIM) technique developed by [Stoakley et al. 1995] allowed users to manipulate the virtual world objects using the miniature version of the virtual environment which they were able to hold in their hands establishing a direct relationship with real life virtual objects and the iconic version of real objects. However, the WIM technique doesn't work well when the number of objects in the virtual environment becomes large.

In recent times, Nintendo Wii controllers have become popular. The Wiimote is the primary controller for Nintendos Wii console with motion sensing capabilities. [Gallo et al. 2008] presented 3D interaction techniques for manipulating volumetric medical data using the Nintendo Wii controller. However, the Wiimote is not very suitable for fully immersed experience. [Figure 1]

Marker based hand detection techniques for application in VR include techniques developed by [Fredriksson et al. 2008] which uses bracelets and color coded gloves, Vicon motion tracker based techniques developed by [Grossman et al. 2004] and a head mounted display based technique developed by [Manders et al. 2008]. [Fredriksson et al. 2008] used a simple webcam to capture frames for their marker based hand-computer interaction system. They used a color coded bracelet and a color coded glove which had two square palm markers on either side of the

palm and five distinctly shaded finger sheaths to locate the hand region. [Grossman et al. 2004] used multi-finger gestural input to interact with 3D volumetric displays. The volumetric displays provided 360◦ viewing volume allowing manipulation of any viewpoint. Here the user was allowed to directly interact with the display using his fingers, tracked by a Vicon motion tracking system that tracks the positions of markers placed on the users fingers. This technique worked only for a small volume and is not very appropriate for immersive VR systems. [Manders et al. 2008] developed an intuitive gesture recognition system where a marker was placed on a head-mounted display to locate the user's head based on which the exposed facial skin was detected and a skin model was developed. The main drawback of this method is the use of the head-mounted display which makes interaction confined to a single user and also makes the system expensive.

In the last few years the interest in real time markerless hand tracking has increased and many exciting techniques have been proposed. A flock of features based technique for fast hand tracking was developed by [Kolsch and Turk 2004]. The tracker selects a set of KLT features within a hand region supplied by a hand detection algorithm and tracks them from frame to frame. The features are forced to maintain a flock of birds like behaviour by setting upper and lower thresholds for minimum and maximum distances between any two features. A probability map is generated using color information to update the positions of features. The technique however is not very robust against illumination changes and also when there are other skin-colored regions in the image. [Pan et al. 2010] improved the technique by adding a velocity weighting factor to the flock of features. The approach used the fact that the hand would be moving faster than any other skin colored object in the scene, to move the flock of features towards the hand. The technique is not suitable for tracking multiple hands. Moreover, in case of tracking failure

the system needs to be reinitialized with an open palm position, which is not suitable for many applications.

A markerless object manipulation technique was developed by [Lee and Chun 2009] for augmented reality applications. The technique identifies fingertips using curvature and tracks them using optical flow. The hand orientation is determined based on the fingertip positions and the virtual object is rendered on it. For object manipulation predefined hand postures that correspond to particular operation on the object are matched with current hand posture. However, since the technique depends heavily on fingertip detection, it may not work well under self occlusion. Also, using predefined hand gestures severely limits the scope of the applications.

A markerless hand and head detection tracking system was developed by [Nickel et al. 2004] which used a Hidden Markov Model based approach to classify and recognize the gestures. They used the orientation of head as an additional feature to recognize the gestures of the hands. They also established experimentally that the gesture recognition performs significantly better using the head as an additional feature. However this might not hold true in our case as we intend to allow multiple users, pairing the heads with their corresponding hands especially for gaming applications where the users move around significantly, is difficult.

In this paper we present a technique that allows completely markerless detection and tracking of hands in the 3D space. Our primary objective in this work is to completely free the user from any kind of environmental constraints and allow as much freedom in movement and gestures as possible. The technique presented in this paper is very generic and can therefore find its use in many types of virtual reality and augmented reality applications.

Further, the novelty of the technique presented in this paper is that it can effectively track multiple hands in the scene, which is essential for multi-user environments. The technique also works well under limited self occlusion as well as when occluded by other objects.

3 Proposed Method

The system uses Point Grey's binocular camera system Bumblebee2. The hand detection and tracking application works by first registering a user's hand and then continuously tracking it. When a new user wishes his or her hand to be tracked by the system, he/she needs to hold the hand in open palm position, with fingers separated in front of the camera for a second. Once the hand is detected, the user can see it displayed on the screen highlighted with some color. The user can then start moving the hand as he/she wishes and the system tracks the hand by using an adaptive cluster updation and classification technique.

The processing of the 3D data obtained from the binocular system can be divided into three stages, the pre-processing stage followed by the hand detection stage and then the hand tracking stage. Figure 1 provides a schematic diagram of the tasks involved in each stage and the flow of control. The pre-processing stage extracts out the skin colored regions in the image and is performed for every frame. The detection stage extracts detailed shape information from the skin regions to identify hands. The detection stage does not execute for every frame, it executes only once every 30 frames (i.e. once every second). The tracking stage on the other hand uses very limited shape information. It combines the limited shape information with the knowledge of the position of hands in previous frames to classify each skin color region as hand or non-hand region. The tracking stage also executes for every frame. In order to use the position of hands in a frame as a predictor for hand position in the following frames the system maintains a Hand Means table. The Hand Means table contains information about the hands being tracked. The

table stores a unique identifier, the center of the hand and an aging factor for each hand being tracked. The unique identifier is used to identify each hand being tracked from the others, the center of the hand gives its position in 3D space while the aging factor is used to remove entries of hands that are no longer in the view of the camera.

Separating the detection and tracking makes the system more robust, as detecting a hand in open palm position with fingers separated, with a high confidence is easy while it still does not pose much constraint on the user. While once the hand has been detected, it can then be tracked very fast and accurately in a 3D space. Compared to continuous detection of hand in each frame this technique performs better because it inherently overcomes the problem of detecting hand in arbitrary positions.

The tracking works by identifying skin colored regions as blobs and then classifying these blobs as belonging to hand using its size information and distance to hand regions identified in the previous frame. As the system does not use any particular shape information other than size, this approach allows the system to give users a very high degree of freedom. Further by tracking the hand in 3D space the problem of the hand being partially occluded or being lost in a skin colored background is overcome.

Since our tracking is independent of the detection, it is not constrained by any shape considerations and therefore can track the hand robustly in many orientations unlike some techniques that depend on the phalanges of the hand being visible.

In the following sections the steps involved in the three stages are described.

3.1 Pre-processing

The pre-processing steps involve image capture using the binocular system, stereo image processing and foreground extraction. The pre-processing steps are performed for each incoming frame.

Step 1: Stereo Image Acquisition and Rectification

This step involves processing the stereo image pairs to obtain the 3D depth image and generating the rectified image.

Step 2: Background Subtraction

Frame differencing based background subtraction is then performed on the rectified image by subtracting each incoming frame from the running average of the background.

Step 3: Skin Segmentation

Skin segmentation is performed by converting the image into HSV color space and thresholding in HS space.

Step 4: Morphological Closing

A single iteration of closing operation is performed using a 3x3 rectangular structuring element for fast closing.

Step 5: Connected Component Analysis

A single iteration of connected component analysis and external contour extraction of identified regions is performed.

Since the camera system in most of the applicable scenarios can be safely assumed to be stationary and having a fixed field of view, the frame differencing based background subtraction was found to give sufficiently good results. Effects of illumination changes to the foreground extraction is countered to some degree by the fact that skin-colored region extraction is done in HS space. [Figure 2] [Figure 3]

The skin color segmentation allows further reduction in the search space for the detection algorithm. For color based segmentation, color models that separate the illumination or intensity components from the chrominance components like the YCrCb and HSV are preferred over RGB which stores the color components in terms of the three primary colors red, green and blue making it highly sensitive to slight changes in illumination and brightness. Figure 2 shows the distribution of skin pixels in the RGB color space and Figure 3 shows the distribution of same set of pixels in HS space. The pixels were taken from skin images acquired under different illumination conditions and from varying skin tones of people from different ethnicity. The HSV color model as seen in Figure 3 gives a very compact clustering of the skin color pixels.

Next, morphological closing is done to compensate for the discontinuities produced in the skin segmentation step and smoothen out the edges of the segmented regions by filling in small gaps. Connected component analysis is then done to fill in the gaps within well connected regions in the image. Any holes within a closed region are then ignored i.e. the entire area inside the external contour is considered as filled. This gives more continuous regions, and allows the hand region to be represented by two or three large contours, making the clustering and ultimately tracking easier and more efficient. The contours are generated by finding every pixel having a foreground to background transition. For each region, the level of contours is calculated by studying their relative positions.

3.2 Hand Detection

Fingers are undoubtedly the most unique feature of human hands and we primarily use them to effectively distinguish hand regions from the other regions of the scene primarily the facial regions. At this stage, the scene will comprise of moving skin-colored regions, such as the arms,

neck and the facial regions, which we want to eliminate as candidates for being part of a hand. The hand detection stage involves the following steps.

Step 1: Contour Approximation

This step involves approximating the external contours of the regions found after Connected Component Analysis.

Step 2: Line Approximation

In this step the narrow, stretched regions of the contours are approximated with lines using a simple line approximation technique. The identified lines represent possible fingers.

Step 3: Line Clustering

The lines found in Step 2 are clustered in Euclidean space using a single pass threshold based clustering.

Step 4: Cluster Classification

Each line cluster is then classified as a possible hand based on cluster features using a minimum distance classifier in the feature space.

The region contours extracted in the pre-processing stage have a very large number of points and can lead to inefficient processing. In order to reduce the number of points on the contour the Douglas Peucker Approximation algorithm is applied [Douglas and Peucker 1973]. [Figure 4] [Figure 5]

This is followed with an indigenous line approximation technique that is devised, to approximate the fingerlike regions in the image with lines. The idea is to efficiently find elongated regions with sharp corners in the contour, and approximate them with lines. To identify these regions, a threshold angle and a threshold length were found experimentally using the training data.

The algorithm used to identify finger regions works as follows: START

INPUT: contour point list C, threshold length L and threshold angle A

INITIALIZE: finger line list LINES

1. If contour point list C is of length 1, RETURN LINES.

2. If contour point list C is of length 2, create a line using these two points, add the line in LINES and RETURN LINES.

3. If contour point list C is of length 3, find the smallest angle of the triangle formed by the three points.

(a) If the smallest angle is below the threshold angle value A, go to Step 3.(b) else RETURN LINES

(b) Calculate the length of the line from the point with the smallest angle in the triangle to the midpoint of its opposite side. If the length is greater than the length threshold L, add the line to the finger line list LINES.

RETURN LINES.

4. If contour point list C is of length 4 or more. For each point pn, do the following:

(a) If n =sizeof(C) RETURNLINES

(b) Calculate angle between the lines formed by the points pn−1 and pn, and pn and pn+1

(c) If the angle calculated in Step 4.(a) is less than the threshold angle A go to Step 4.(c) Else set n = n + 1 and go to Step 4.

(d) Calculate the length of the line formed by joining pn to the mid-point of the line joining pn−1 and pn+1

(e) If the length of the line in Step 4.(c) is greater than the threshold length L, add the line found in Step 4.(c), to the finger lines list LINES.

Set n = n + 2 and go to Step 4.

END

Two important things to be noted in this technique are that, for a triangle i.e. contour with three points, the general technique of n points cannot be used as this can result in the region being approximated by more than one line as depicted in Figure 7 below. While for contours with more than three points, to overcome this problem whenever a finger line is detected at a point the next point in the list is not processed and it skips to the next to next point.

The detection technique is designed such that it is possible to detect a large number of hand postures. Human hand is very flexible and is able to perform variety of gestures. The aim is to be able to capture as many postures possible so that users are provided with more degrees of freedom while interacting with virtual objects or in HCI applications. The detection technique is capable of recognizing hand regions irrespective of rotation or scaling in most cases except the extreme cases of hand posture deformation or in cases where no finger regions are visible. The system performs robustly because even though hands can have a large number of orientations and deformations, at least a few fingers are visible in most cases.

The lines found using the line approximation technique are then clustered using a single pass threshold based clustering. The threshold was found empirically from the training dataset. The clustering divides the list of possible finger lines into sets of finger lines in which the lines are relatively close to each other. [Figure 6] [Figure 7]

Finally, in order to identify the hand regions in the 2D rectified image, each of these clusters are classified as representing a hand or a non-hand object. The classification is done using a minimum distance classifier in which the mean vectors are found using k-means clustering of the true finger lines found in the training datasets.

The following features are used to classify each cluster:

1. The number of lines in the cluster
2. The maximum length of a line
3. The average length of all lines
4. The maximum angle between each neighboring line pair
5. The maximum angle between any two lines in the cluster
6. Euclidean distance of the cluster mean from the nearest mean in the Hand Means table.

Further it is important that the cluster lines display similar relationship as the fingers of an actual hand. [Figure 8]

Therefore for a cluster to be classified as a hand cluster in addition to the minimum distance classification, the following constraints are also imposed:

1. The angle between any two neighboring lines should not exceed 90°.
2. The angle between any two lines in the cluster cannot be more than 180°.
3. The ratio of the average length of the lines to the maximum length of a line must be greater than 0.6.

3.3 Hand Tracking

The tracking stage involves identifying possible hand blobs and classifying these blobs again using a minimum distance classifier. However, this time the classification is done in the 3D Euclidean space and the mean vectors are the hand mean positions stored in the Hand Means table. The Hand Means table is also updated after classification is completed for each frame. Therefore, the cluster means change after every iteration making the clustering adaptive. The main advantage of clustering in 3D space is that occlusion between hands or hand and skin colored background can be resolved effectively. This stage involves the following five steps:

Step 1: 3D hand region extraction

The segmented 2D hand region is back projected onto the 3D data obtained in Step 1 of Pre-processing stage to segment out the hand region in 3D space.

Step 2: Region Clustering and Mean Calculation

The hand regions segmented out in Step 1 are once again clustered, this time in the 3D Euclidean space using a single pass threshold based classifier. The 3D mean vector of each identified hand is then calculated.

Step 3: Blob Elimination

Some of the 3D blobs obtained in Step 2 are eliminated based on their size and distance from the nearest mean vector in the Hand Means table using empirically calculated thresholds.

Step 4: Blob Classification

The remaining 3D blobs (represented by the mean vectors calculated in Step 2) are then classified as being part of one of the hands being tracked using a mimimum distance classifier with the entries in the Hand Means table as the mean vectors.

Step 5: Hand Means Table Updation

The Hand Means table entries are then updated using the newly assigned hand regions. If the detection stage was run for the current frame and new hands were detected, new entries are added in the Hand Means table. If no blobs were assigned to any particular entry, its age is increased. Any entry which is assigned a blob is set to an age of zero and entries over a certain age are deleted from the table.

In this stage each hand being tracked, identified by their unique ids are assigned a new set of skin colored regions. Recalculating the means of each hand using the newly assigned region makes the hand centers follow the respective hands they represent in each frame.

Whenever a hand is detected during the detection stage, if the mean of the detected hand is farther away from any of the means in the Hand Means table than the distance threshold used in Step 3 it is considered as a new entry. This way the means continuously follow the hands. The clustering in 3D space and usage of depth information in the mean calculation also ensures that hands that are occluding each other partially can still be tracked effectively.

In many applications it is quite likely that two hands of players may overlap very closely in 3D space or even collide. In such cases, the tracker would lose the hands, however due to the repetitive detection of hands, the hands can be tracked back automatically, provided that the hands have not moved significantly far away from the position where the tracker lost it, within a second.

3.4 Experimental Results

The 3D data obtained from the stereo camera system in Step 1 of the pre-processing stage is masked with the detected hand regions to get a dense set of points on the hand surface. This can then be used to fit any predefined hand model using simple techniques like least mean square error distance. In this project since the goal is to investigate only hand detection and tracking and not its applications, the data itself is rendered into the three-dimensional (3D) virtual space showing the hand surface point cloud in OpenGL. Figure 8 shows the output of this process. The results of hand detection and tracking presented in Figure 9 shows that the system is capable of robustly tracking multiple hands in any orientation in a complex background.

[Figure 9] Experimental results obtained gave near real time rates, which was another area of focus. Table 1 shows the results of average running time of detection and tracking for 5 runs of sets of 200 different images. The detection and tracking steps were run separately for each image along with the pre-processing stage. The pre-processing stage was found to take

90% of the processing time. Different scenarios like multiple hands being tracked were considered. No particular performance degradation was found with the increase in number of hands being tracked. The test was run on a PC with 2.6 GHz Intel Core 2 Duo processor and 4GB RAM. The results show that the application can be run at 20-25 fps. Much higher rates of upto 60 fps can easily be achieved when using a Graphics card and with hardware implementation, of particularly the standard techniques used in the pre-processing stage. [Table 1]

Real time acquisition, detection and tracking of multiple hands enable the system to be used for all real time 3D interaction scenarios like in virtual games or in HCI applications where we replace mouse as an input device with users' hands itself.

The results show that the system developed, is very robustly able to
track the users hands in many orientations as well as scale. Figures 9(a) and 9(b) demonstrates the tracking capability of the system as the hand orientation changes. While Figure 9(c) shows that the system works well under different skin color shades, as it is able to detect the top of the palm as effectively as the inside of it. Figures 9(d) and 9(e) shows that even though the detection is primarily based on finger identification, the tracking still works when only one or no finger is present. Figures 9(f), 9(g) and 9(h) show the robustness of the algorithm for various skin tones. While Figure 9(i) shows the technique is capable of tracking the hand even at low illumination conditions proving the robustness of the technique under varying lighting conditions. [Table 2]

Table 2 gives the accuracy of the detection and tracking technique developed for open palm hand pose and other arbitrary hand poses. The accuracies were calculated by manually comparing the results of running the technique for three different datasets of 100 images each. As expected the

open palm position is detected with very high accuracy and the tracking also performs very robustly for all orientations.

4 Conclusion

The technique presented above has several distinct advantages over existing techniques. First of all, it frees the user completely from any external devices and allows the user full freedom of movement in any direction and orientation. This makes it possible for the system to be used in industrial augmented reality applications, where the users need to interact with the system for long hours on a daily basis.

Furthermore, because of its simple nature the system at its current configuration can track hands within a large distance of up to 7 meters. Since the techniques are all generic and scale invariant, the only factor to be considered is that to the processing power should be proportional to the image resolution.

A distinct advantage of this technique over others is that it puts very little constraint over the initial hand pose of the user. The users overhead to set up the interaction is minimal. This is often very important in case of video games, where it is possible to have children using the system. In such a scenario complicated start up techniques can seem frustrating to the user making the whole game application unpopular. In this system however, all the user has to do is place the hand in the open palm position for about 1 second at the start, which is something intuitive and even children can follow.

The system can effectively track and maintains information of each hand movement over a period of time. This information can be used in many applications, like video games where the users path can be generated based on the series of hand movements over time.

A major salient feature of the developed technique is that it can track multiple hands simultaneously. Whereas most existing techniques can either only track a limited number of hands at a time or can track many hands but need much more user input to initialize each hand.

5 Future Work

One of the limitations of the system however is that although tracking works in large range of distance, detection is limited to a smaller distance of about 2-3 meters. That is, the detection technique is not very robust under scaling. This problem can be solved by performing the detection using a multi-resolution strategy; that is the matching can be done using an image pyramid in the scale space; however this area remains to be further investigated.

Another limitation of the technique is that when the number of hands to be tracked is very high and most of the screen space is covered by hands, clustering in 2D space can become erroneous. This limitation can however be easily avoided by limiting the number of hands to 4 or 5, which usually is sufficient for most applications.

The performance of the technique although acceptable at current output rates, it can be further improved so as to be able to process as high as 60 frames per second (FPS). This is possible with hardware implementation of the image processing algorithms.

Acknowledgements

This work is supported by the ARC 9/09 Grant (MOE2008-T2-1 075) of Singapore.

References

DOUGLAS D. H., PEUCKER T. K. 1973. Algorithms for the reduction of the number ofpoints required to represent a line or its caricature The Canadian Cartographer, vol. 10 issue 2, December 1973, pp 112-122.

MANDERS COREY, FARBIZ FARZAM, YIN KA TING, MIAOLONG YUAN, CHONG BRYAN, GUAN GIM CHUA 2008. Interacting with 3D Objects in a Virtual Environment Using an Intuitive Gesture System, VRCAI 2008, ACM.

KOLSCH M., TURK M. 2004. Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration, In IEEE Workshop on Real-Time Vision for Human-Computer Interaction CVPR, pp 158.

BYUNGSUNG LEE, JUNCHUL CHUN 2009. Manipulation of Virtual Objects in Marker- less AR System by Fingertip Tracking and Hand Gesture Recognition, ACM International Conference Proceeding Series; Vol. 403, Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp 1110-1115.

PIERCE S. JEFFERY, STEAMS C. BRIAN, PAUSCH RANDY 1999. Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments, Symposium on Interactive 3D Graphics, Proceedings of the 1999 symposium on Interactive 3D graphics, pp 141-145.

ZHIGENG PAN, YANG LI, MINGMIN ZHANG, CHAO SUN, KANGDE GUO, XING TANG, STEVEN ZHIYING ZHOU 2010. A Real-time Multi-cue Hand Tracking Algorithm Based on Computer Vision, Virtual Reality Conference (VR), 2010 IEEE, pp 219-222.

STOAKLEY RICHARD, CONWAY J. MATTHEW, PAUSH RANDY 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature, Conference on Human Factors on Computing Systems, Proceedings of the SIGCHI conference on Human factors in computing systems, 1995, pp 265 272.

BOWMAN A. DOUG, HODGES F. LARRY 1997. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments, Symposium on

Interactive 3D Graphics, Proceedings of the 1997 symposium on Interactive 3D graphics, pp 35 ff.

GALLO LUIGI, PIETRO DE GIUSEPPE, MARRA IVANA 2008. 3D Interaction with Volumetric Medical Data: experiencing the Wiimote, Proceedings of the 1st international conference on Ambient media and systems, SIGCHI (2008).

NICKEL KAI, SEEMANN EDGAR, STEIFELHAGEN RAINER 2004. 3D-Tracking of Heads and Hands for Pointing Gesture Recognition in a Human-Robot Interaction Scenario, Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FG 04).

FREDIKSSON JONAS, RYEN BERG SVEN, FJELD MORTEN 2008. Real-Time 3D Hand-Computer Interaction: Optimization and Complexity Reduction, ACM International Conference Proceeding Series, Vol. 358, Proceedings of the 5th Nordic conference on Human- computer interaction: building bridges, 2008, pp 133 141.

GROSSMAN TOVI, WIGDOR DANIEL, BALAKRISHNAN RAVIN 2004. Multi-Finger Gestural Interaction with 3D Volumetric Displays, Symposium on User Interface Software and Technology, Proceedings of the 17th annual ACM symposium on User interface software and technology, 2004, pp 61 70.

PEREZ-QUNINONES A. MANUEL A. PEREZ-QUNINONES, SIBERT L. JOHN 1996. A Collaborative Model of Feedback in Human-Computer Interaction, CHI 96, (1996), ACM Press.

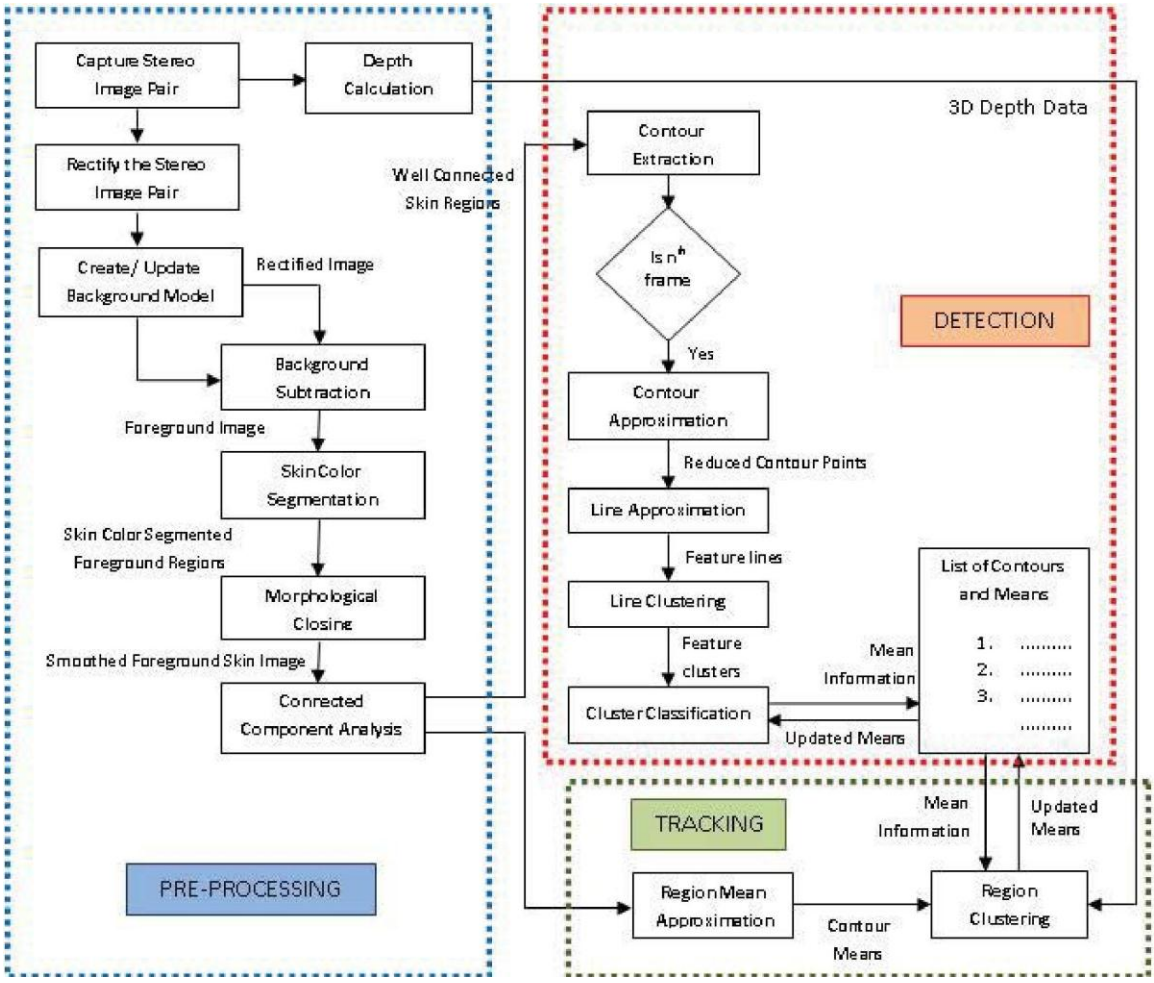

Figure 1: System Diagram Shows the tasks involved in each stage and the flow of control.

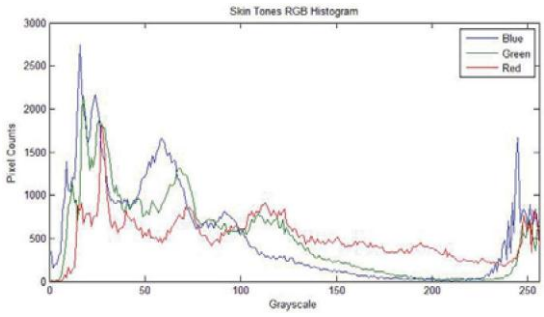

Figure 2: Histogram of Red, Green and Blue color channels for skin pixels.

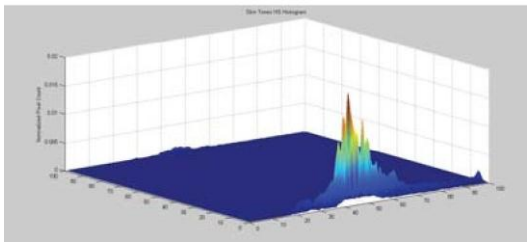

Figure 3: Histogram of HS of the HSV color space for skin pixels.

Figure 4: Result of Contour Extraction - Shows the extracted contour in red.

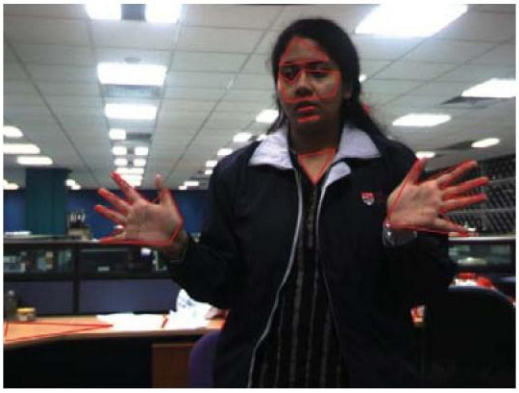

Figure 5: Douglas-Peucker Approximation - Shows the results of Douglas-Peucker approximation of the contours of the skin colored region.

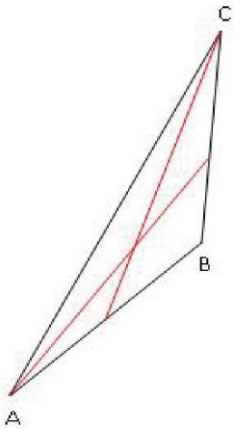

Figure 6: Finger Line Approximation - Angle CAB and ACB both being smaller than the threshold angle, the triangle gets approximated by two lines which is incorrect.

Figure 7: Results of using Finger Line Approximation Algorithm - Shows the result of running the line approximation algorithm on the approximated contour shown in previous figure. The lines found are drawn in blue.

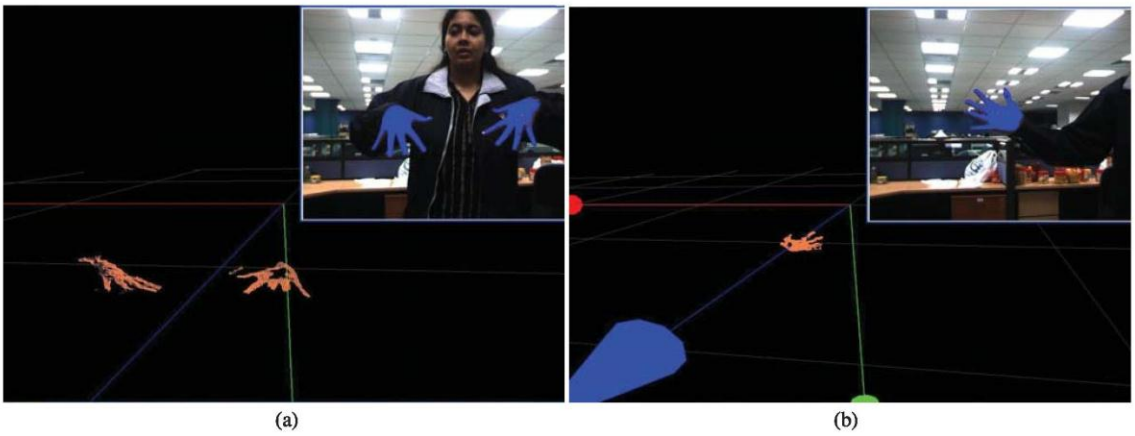

Figure 8: Hand Tracking in 3D - Results of hand point cloud generated in 3D space.

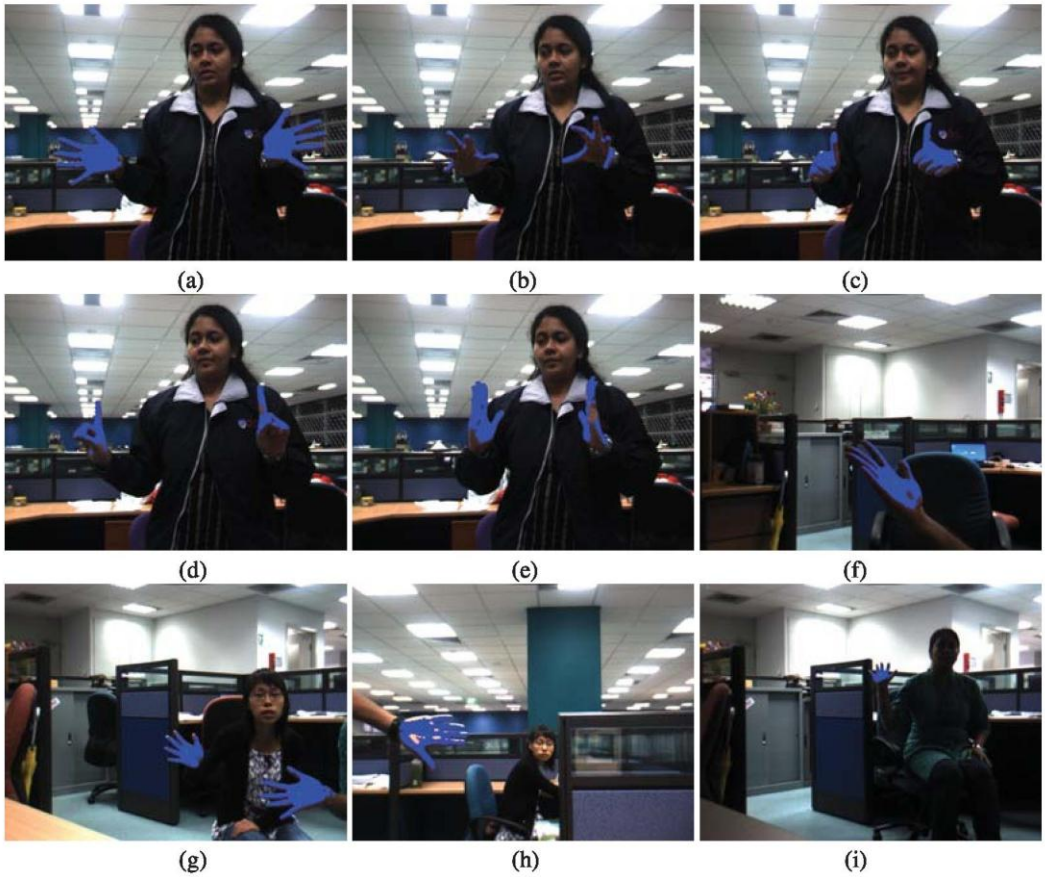

Figure 9: Hand Tracking in 2D - Results of continuous hand tracking in 2D image space.

another area of focus.

Runs	Average Running Time (in milliseconds)	
	Detection	Tracking
1	55.58991	40.01096
2	54.57456	39.45614
3	65.06235	47.57314
4	63.13429	45.91847
5	54.96708	41.78198

Table 1: Detection and Tracking Time Performances.

Hand Pose	Accuracy (% detected correctly)	
	Detection	Tracking
Open Palm	92.26%	96.55%
Others	66.37%	89.93%

Table 2: Hand Detection and Tracking Accuracy for different poses and orientations.