REAL-TIME HAND GESTURE RECOGNITION USING RANGE CAMERAS

Hervé Lahamy * and Derek Litchi

Department of Geomatics Engineering, University of Calgary, 2500 University Dr NW, Calgary, Alberta, T2N1N4 Canada - (hdalaham, ddlichti)@ucalgary.ca

Commission I, WG I/3

KEY WORDS: Range camera, point cloud, segmentation, gesture recognition, moving object, real-time tracking

ABSTRACT:

In order to enable a more natural communication with virtual reality systems, automatic hand gesture recognition appears as a suitable means. Hand gesture recognition making use of digital images has been a research topic for many years. However, the use of range cameras for automatic gesture recognition is in its infancy. Range cameras, with their capability of simultaneously capturing a full 3D point cloud with an array sensor at video rates, offer great potential for real-time measurement of dynamic scenes. The aim of this research is to build a human-machine interface using the 3D information provided by a range camera for identifying hand gestures. The first application designed recognizes the number of raised fingers that appear in a hand gesture while the second allows manipulating a moving object in a virtual environment with a hand gesture. The first application is a preliminary step towards dynamic hand gesture recognition, the second is a real-time application intended for oil and gas reservoir modelling and simulation. The proposed methodology involves 3D image acquisition, segmentation of the hand information, tracking of the hand blob, recognition of the hand gestures and determination of the position and orientation of the user's hand. This research demonstrates the capability of a range camera for real-time gesture recognition applications.

1. INTRODUCTION

Interactions between human and computer are currently performed using keyboards, mice or different haptic devices. In addition to being different from our natural way of interacting, these tools do not provide enough flexibility for a number of applications such as manipulating objects in a virtual environment. In order to improve the human-machine interaction, an automatic hand gesture recognition system could be used.

Extensive research has been conducted on hand gesture recognition making use of digital images (Tsuruta et al. 2001, Mahesh et al. 2009). However, it's still ongoing research as most papers do not provide a complete solution to the previously mentioned problems. Range cameras can simultaneously capture a full 3D point cloud with an array sensor at video rates. They offer great potential for real-time measurement of static and dynamic scenes (Malassiotis et al, 2008). Investigation on 3D range cameras for automatic gesture recognition is in its infancy. However, some research has been conducted in this area (Liu et al. 2004; Breuer et al. 2007). The final aim of this research is to design and build a human-machine interface using the 3D information provided by a range camera for automatic and real time identification of hand gesture. In this paper, we focus on two applications. The first one is designed to recognize the number of raised fingers that appear in a hand gesture and the second is intended for moving an object in a virtual environment using only a hand gesture on the acquired images.

For a real-time application, the expectation is to obtain the best possible images of the hand gesture within the lowest possible time. Some experiments have been conducted with the purpose of defining the best configuration for imaging the hand. This configuration includes, among others, the relative position of the hand and the camera, the influence of the integration time of the camera, the amplitude threshold, the lighting conditions of the environment, the surrounding objects and the skin colour. Some of these parameters will be discussed in this paper which is organized in seven parts as follows. In Section 2, the range camera is presented as well as some of the parameters for defining the best configuration for imaging the user's hand. Section 3 discusses the different steps for extracting the hand information from the image. In Section 4, the methodology designed for recognizing the number of raised fingers in a hand gesture is presented. Section 5 highlights the process applied for the manipulation of a moving object in a virtual environment using a hand gesture. Section 6 focuses on the obtained results and their analysis. Conclusions and future work are provided in Section 7.

2. THE RANGE CAMERA AND ITS SENSITIVITY FOR IMAGING THE HAND

The range camera used in this research is the Swiss Ranger SR4000. It is manufactured by MESA Imaging AG, a Zurich-based company of Switzerland. The SR4000 (Figure 1) is a time-of-flight camera that takes both range and intensity images at the same time using an integrated sensor. It has a low resolution of 176 × 144 pixels. Once the image is acquired, the range information is used for generating the x, y, z coordinates in meter for each pixel. The range camera produces images at a rate of up to 54 frames per second.

Figure 1. SR4000

* Corresponding author.

In order to define the best configuration of the camera for hand imaging, the sensitivity of the camera has been tested to a number of parameters.

2.1 Sensitivity to integration time

The integration time is the length of time that the pixels are allowed to collect light. Not setting the appropriate integration time results in a loss of information (saturated pixels) while imaging the hand (Figure 2). Pixels receiving more light than expected are saturated and do not provide any information in the output image. Ideally, the more the light is collected without saturation, the better the hand gesture will be imaged. Saturation occurs due to excessive signal and/or background light. Thus one should look for the highest possible integration time without having any saturated pixels in the hand blob. The appropriate integration time is a function of distance and has been determined empirically during an experiment where hand images are collected at different distances with different integration times.

Appropriate Integration time Inappropriate Integration time

Figure 2. Effect of the integration time on hand images

The total time required to capture a depth image is four times the integration time, plus four times the readout time of 4ms. This is reflected in the achievable frame rate for a given integration time. Figures 3 and 4 provide for a given distance, the appropriate integration time and the corresponding frame rate as determined by saturation-free imaging.

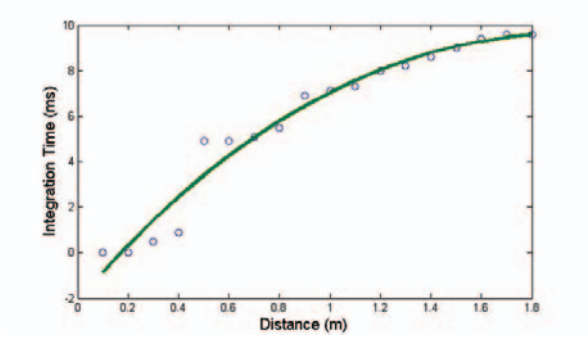


Figure 3. Integration times for imaging the hand

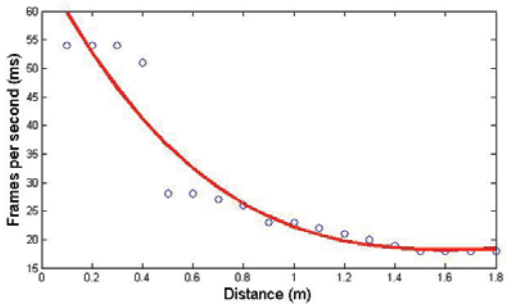


Figure 4. Frame Rates for imaging the hand

2.2 Sensitivity to lighting conditions

The SR4000 is an active sensor that emits infrared light and records the reflected radiation through a filter. In order to check whether any additional desk lamp could disturb our applications, an experiment has been conducted where a spot light has been placed in front of the SR4000 camera. It has been noticed that part of the light coming from the lamp passes through the filter and is recorded by the camera (Figure 5). This creates some saturated pixels but it doesn't prevent imaging the hand if the number of saturated pixels is insignificant with respect to the size of the hand segment. It has also been noticed that the light at the ceiling as well no light in the room have any specific influence on the hand images. As a concluding remark about the lighting conditions for imaging the hand by making use of a range camera, avoiding any additional light directly placed in front of the camera is clearly advisable.

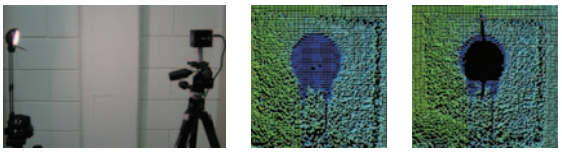


Figure 5. Spot light and camera – Spot light off – Spot light on

2.3 Sensitivity to surrounding objects

The SR4000 produces hand images which are highly dependent on reflexive surrounding objects. These objects may cause multiple reflections and/or multiple paths of some light rays and consequently generate some hanging pixels in the images which do not represent any physical object in the reality. In case some of these pixels appear between the hand and the camera, they should be discarded while extracting the hand information.

3. HAND SEGMENTATION

Segmentation is the process of grouping points that belong to the same object into segments. The idea here is to extract from the point cloud the set of points that describe the user's hand (Figure 6).

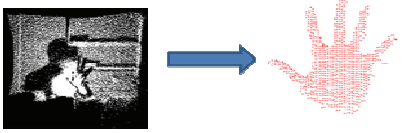


Figure 6. Objective of the Segmentation process

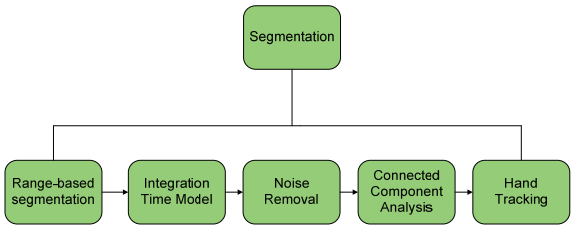


Figure 7. Methodology for Segmentation

The cue mainly used for hand segmentation is the colour (Xiaoming et al. 2001). Another method is based on the difference image between consecutive video frames (Zhang et al. 2008). Fusion of range and intensity images is suggested in (Heisele et al 1999). The SR4000 does provide intensity but not any colour information. In this paper, multiple-step based segmentation has been designed (Figure 7).

3.1 Range-based segmentation

The underlying principle is that there shouldn't be any object between the camera and the hand. Thus the hand appears in the foreground of the image. Any point between the hand and the camera is considered as a noise. A simple range threshold was used to extract the hand information. The algorithm is designed as follows:

a) Find the closest point to the camera using the range;
b) Select points that are less than a threshold from that point;
c) If total number of points lower than a threshold, delete the closest point to the camera and re-start from a)

3.2 Integration time Model

To avoid saturated pixels, once the approximate distance of the hand to the camera is known, the model described by Figure 3 is used to determine the appropriate integration time for the following image to be acquired; which result in a range of 17 to 54 frames per second.

3.3 Noise Removal

The results obtained contain the appropriate information but it appears noisy due to the presence of hanging points (Figure 8) as well as points describing part of the image background depending on the closeness between the hand and its background. The hanging points appear isolated compared to the ones that belong to the hand. The point density of the hand is much higher than the one of the hanging points. The point cloud obtained from the range-based segmentation is split into voxels (3D cells). Voxels that have a low point density are discarded from the segment.

Figure 8. Noise due to the presence of hanging points

3.4 Connected Component Analysis

Connected component labelling is used in computer vision to detect unconnected regions. It is an iterative process that groups neighboring elements into classes based on a distance threshold. A point belongs to a specific class if and only if it is closer within the distance threshold to another point belonging to that same class. After the noise removal, the hand segment appears to be the biggest one in the dataset. An example of the results from the point density-based noise removal and connected component analysis are provided in Figure 9.

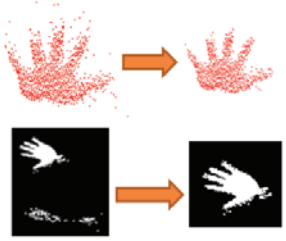


Figure 9. Example of noise removal using the point density and connected component analysis

3.5 Hand tracking

To avoid a time-consuming segmentation on every acquired frame, tracking the hand gesture appears to be an appropriate alternative. The Kalman filter is a suitable tool designed for this purpose (Elmezain et al. 2009; Stenger et al. 2001). It is used to predict the position of the hand in the coming frame and after having measured the actual position of the hand in that frame, this prediction is corrected and the adjusted value is used for the prediction in the following frame. The Kalman filter is thus a recursive estimator for linear processes. To track the hand gesture, we consider the centroid of the hand region. The state vector is represented as $x_t = (x(t), y(t), z(t), v_x(t), v_y(t), v_z(t))^T$ where x(t), y(t), z(t) represent the locations of the centroid in the camera frame and $v_x(t)$, $v_y(t)$, $v_z(t)$) represent the velocity of the hand in the t^{th} image frame. It is assumed that between the $(k-1)^{th}$ and the k^{th} frames, the hand undergoes a constant acceleration of a_k. From the Newton's laws of motion, we can conclude that:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + G_k a_k \qquad (1)$$

where F_k is the dynamic matrix which is applied to the previous state $\mathbf{x}_{k-1}$ and G is the driving matrix:

$$F_k = \begin{pmatrix} I_3 & \Delta t I_3 \\ O_3 & I_3 \end{pmatrix} \qquad (2)$$

$$G_k^T = \begin{pmatrix} \frac{\Delta t^2}{2} & \frac{\Delta t^2}{2} & \frac{\Delta t^2}{2} & \Delta t & \Delta t & \Delta t \end{pmatrix} \qquad (3)$$

Figure 10. Kalman Filter using Newton's law of motion

The first four steps of the segmentation process stand as initialization steps for to the tracking process. They are applied once, at the beginning of the procedure. An example of the result of hand tracking is provided in Figure 11 where 50 consecutive frames have been acquired.

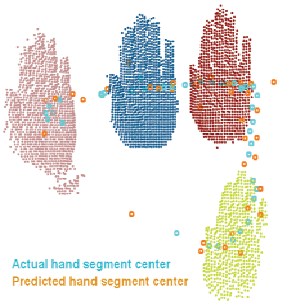


Figure 11. Example of hand tracking

4. METHODOLOGY FOR RECOGNIZING THE NUMBER OF RAISED FINGERS IN A HAND GESTURE

After segmentation of the hand information from the captured scene, points are projected into the palm plane. The following step is the extraction of the hand outline which is smoothed. The number of active fingers is then inferred from the number of U-turns that appear in the refined outline. The methodology scheme is provided in Figure 12 and detailed explanations are provided in the following Sub-Sections.

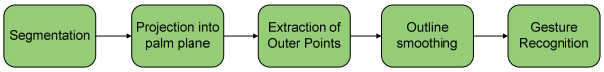


Figure 12. Methodology to identify number of fingers

4.1 Projection into palm plane

In order to extract the outline of the hand, the point cloud describing the hand is projected onto the palm plane. First the palm is detected by computing the centre of gravity of the segmented data. The points within a range threshold from the centre of gravity are assumed to belong to the palm. Using a least square adjustment, a regression plane is fitted within these points. All points in the hand segment are then projected into that plane.

4.2 Extraction of Outer Points

This step takes as input all projected points and produces as output the outer points sorted in the clockwise order starting from one of the corners of the dataset. The objective is to generate a first approximation of the outline with the maximum number of corners.

To achieve this goal, the method of a modified version of the convex hull has been adopted. This technique proposed by Jarvis (1977), was used by Sampath et al. (2007). The convex hull is the smallest convex boundary containing all the points. The use of a modified version of convex hull instead of the original version is justified by the fact that the original convex hull doesn't provide a boundary with all necessary corners. Some concave corners are not part of the outline.

The process starts with the determination of the lowest left corner. The following corners are determined successively. A moving window centred on the current corner is used to collect neighbouring points. The second outline corner is the point that forms with the first corner the least azimuth (Line 1 in Figure 13). For the remaining corners, the exterior topographic angle between the previous corner, the current corner and each of the selected points is computed. The next corner of the outline is chosen in such a way that the computed angle is the least and the current segment line doesn't cross over any previously determined segment (Lines 2 and 3 in Figure 13). The line 4 in Figure 13 shows the result of the computed boundary compared to the raw points and the original convex hull.

4.3 Smoothing outline

This step takes as input data the outer points produced in the previous step and generates line segments. The main difference with the previous step is an outline better regularized with a noticeable reduction of the number of outline points.

The Douglas-Peucker algorithm uses the closeness of a vertex to an edge segment to smooth an outline. This algorithm starts by considering the single edge joining the first and last vertices of the polyline (Stage 1 in Figure 14). Then the remaining vertices are tested for closeness to that edge. If there are vertices further than a specified tolerance, away from the edge, then the vertex furthest from it is added to the simplified polygon. This creates a new guess for the simplified polyline (Stages 2, 3 and 4 in Figure 14).

This procedure is repeated recursively. If at any time, all of the intermediate distances are less than the threshold, then all the intermediate points are eliminated. Successive stages of this process are shown in Figure 14 and an example of result is provided in Figure 15.

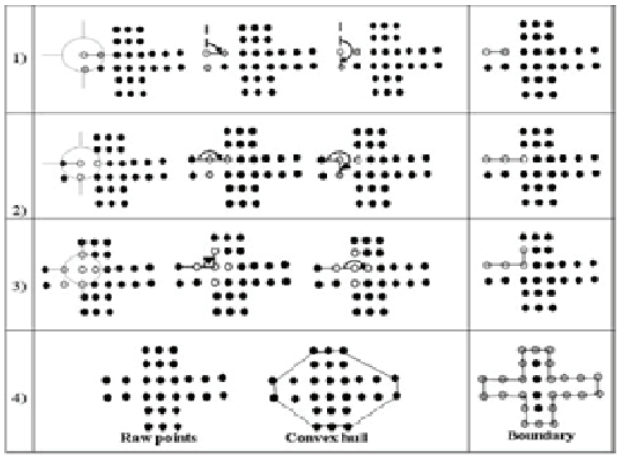
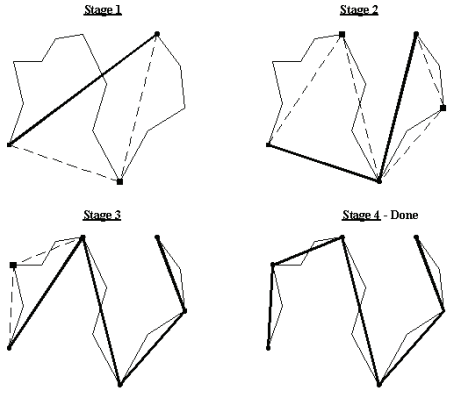


Figure 13. Principle of the modified version of the convex hull



At each stage:
Dashed line: Next Approximation
Gray Line: Original Polyline
Black line: Initial Approximation

Figure 14. Douglas-Peucker Principle

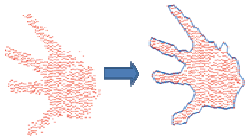


Figure 15. Douglas-Peucker Principle

4.4 Gesture Recognition

The objective of this first application is to identify the number of fingers active in a hand gesture. The strategy applied is to count the number of U turns appearing in the refined outline. From the example in Figure 16, the posture contains five active fingers and ten U-turns. The scale factor of two is the same for all other cases expect for zero that was not considered in this project. The average length of a human finger has been considered as threshold, the reason being to avoid counting

ragged corners that still appear in the outline after the smoothing step.

To detect the U-turns, four consecutive segment lines are considered at a time. By computing the azimuth, the algorithm looks for two segments with opposite directions within a threshold. The process is repeated throughout the outline.

Figure 16. U-turns in a hand posture

5. METHODOLOGY USED FOR THE MANIPULATION OF THE MOVING OBJECT IN A VIRTUAL ENVIRONMENT USING A HAND GESTURE

The methodology applied for this second application is described in Figure 17.

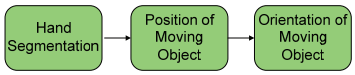


Figure 17. Methodology for moving virtual object

The movement of the virtual object is supposed to be the same as the movement of the hand. Thus the 3D translation and rotation applied to the moving object are obtained from the hand segment. Regarding the translation, the coordinates of the centre of the moving object are the same as the ones of the centre of gravity of the hand segment. The rotation angles ω, ϕ and Κ are taken from three orthogonal vectors computed after fitting a plane to the hand points. The first vector joins the centre of gravity of the hand segment to the farthest point in the computed plane. The second vector is the normal to the plane and the third one is the cross product of the first two. After making them unit, the new coordinates m_{ij} are used to define a rotation matrix M (4) from which, the angles ω (5), ϕ (6) and Κ (7) are derived.This rotation angles are applied to the moving object.

$$M = R_3(\kappa)R_2(\phi)R_1(\omega) = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \quad (4)$$

$$\omega = \arctan\left(\frac{-m_{32}}{m_{33}}\right) \quad (5)$$

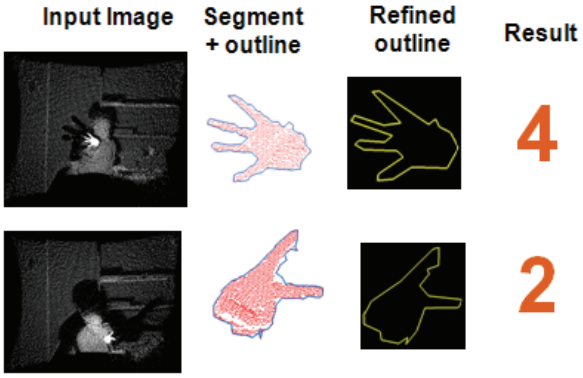
$$\phi = \arcsin\left(m_{31}\right) \quad (6)$$

$$\kappa = \arctan\left(\frac{-m_{21}}{m_{11}}\right) \quad (7)$$

6. RESULTS AND ANALYSIS

6.1 Hand Gesture Recognition

From the results, it can be concluded that the gesture recognition algorithm is independent of the hand type (left or right) and the distance between the hand and the camera (Figure 18).



18. Confusion Matrix of First Application

The results are summarized in the confusion matrix (Figure 19) showing the occurrences of predicted outcomes against actual values. The proportion of correctly classified gestures is estimated as (3+6+5+2+3)/50=38%. Though promising, obtained results are not as good as expected, it can be noticed though a trend showing that a high number of results are close to the true values.

		Reference Values (Number of fingers)					Total
Obtained Results (Number of fingers)		1	2	3	4	5	Total
	1	3	1	0	0	0	4
	2	4	6	5	0	0	15
	3	1	1	5	5	1	13
	4	0	1	0	2	4	7
	5	1	0	0	1	3	5
others	others	1	1	0	2	2	6
Total	Total	10	10	10	10	10	50

Figure 19. Confusion Matrix of First Application

Indeed, most of the obtained segments contain information related to the hand gesture. But in several cases, some noisy pixels could not be removed. Because of the presence of hanging points in the segment, the outline obtained does not properly delineate all fingers appearing in the gesture.

Some incorrect results are also obtained from the gesture recognition algorithm. The strategy designed to identify u-turns fails due to the threshold used to compare two segment lines azimuth or due to a highly smoothed outline. As a consequence, the number of counted u-turns is not correct and so is the number of fingers.

6.2 Moving object Application

An interface has been designed to visualize simultaneously the range image, the segmented hand and the moving object (Figure 20). Though this research is intended for manipulation of oil and gas reservoirs in a virtual environment, it has been considered in this paper as moving object, a 3D cube with different colours on its six faces.

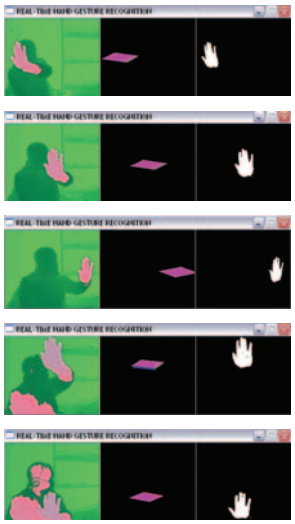


Figure 20. Translation of a moving object

The simultaneous translation and rotation determined from the hand blob and applied to the moving object appear realistic (Figures 20 and 21).

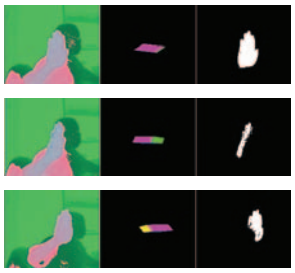


Figure 21. Rotation of a moving object

7. CONCLUSION AND FUTURE WORK

The designed application demonstrated the capability of a range camera for real-time applications. Though the process seems to be promising, further work is required to improve the segmentation speed and the tracking process. Future work includes not only improvement of the designed strategy but also taking into account more challenges such as dynamic gestures involving both hands and/or multiple cameras. Our final objective involves gestures with a high degree of freedom; which may require detection of fingers and articulated hands (Stenger et al. 2001, Sung et al. 2008).

ACKNOWLEDGEMENT

This work is supported by the Computer Modelling Group LTD and the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Breuer, P., Eckes, C. and Muller, S., 2007. Hand gesture recognition with a novel IR time-of-flight range camera - a pilot study, *Proceedings*, 28-30 March 2007 2007, Springer-Verlag pp247-60.

Elmezain, M., Al-Hamadi, A., Niese, R. and Michaelis, B., 2009. A robust method for hand tracking using mean-shift algorithm and Kalman filter in stereo color image sequences. *Proceedings of World Academy of Science, Engineering and Technology*, **59**, 283-287.

Heisele, B. and Ritter, W., 1999. Segmentation of range and intensity image sequences by clustering, *Proceedings 1999 International Conference on Information Intelligence and Systems*, 31 Oct.-3 Nov. 1999 1999, IEEE Comput. Soc pp223-5.

Liu, X. and Fujimura, K., 2004. Hand gesture recognition using depth data, *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 17-19 May 2004 2004, IEEE Comput. Soc pp529-34.

Mahesh, R.J.K., Mahishi, S., Dheeraj, R., Sudheender, S. and Pujari, N.V., 2009. Finger detection for sign language recognition, *IMECS 2009*, 18-20 March 2009 2009, International Association of Engineers pp489-93.

Malassiotis, S. and Strintzis, M.G., 2008. Real-time hand posture recognition using range data. *Image and Vision Computing*, **26**(7), 1027-37.

Stenger, B., Mendonca, P.R.S. and Cipolla, R., 2001. Model-based 3D tracking of an articulated hand, *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, ppII-310-II-315 vol.2.

Sung, K.K., Mi, Y.N. and Phill, K.R., 2008. Color based hand and finger detection technology for user interaction, *2008 International Conference on Convergence and Hybrid Information Technology (ICHIT)*, 28-29 Aug. 2008 2008, IEEE pp229-36.

Tsuruta, N., Yoshiki, Y. and Tobely, T.E., 2001. A randomized hypercolumn model and gesture recognition, *6th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2001*, 13-15 June 2001 2001, Springer-Verlag pp235-42.

Xiaoming Yin and Ming Xie, 2001. Hand gesture segmentation, recognition and application, *Computational Intelligence in Robotics and Automation, 2001. Proceedings 2001 IEEE International Symposium on*, 2001, pp438-443.

Zhang, Q., Chen, F. and Liu, X., 2008. Hand gesture detection and segmentation based on difference background image with complex background, *2008 Intern*