Hand Shape Recognition and Estimation for South African Sign Language

Pei Li[1], Mehrdad Ghaziasgar[1], James Connan[2]
Department of Computer Science
University of the Western Cape[1], Private Bag X17 Bellville, 7535, South Africa
Tel: + (27) 21 959-3010, Fax: + (27) 21 959-3006
and Department of Computer Science
Rhodes University[2], P O Box 94 Grahamstown, 6140
Tel: + (27) 46 603-8291, Fax: + (27) 46 636-1915
email: {2918680, Mehrdad}@uwc.ac.za[1]; j.connan@ru.ac.za[2]

Abstract—In this paper we present a novel system which recognises sign language hand shapes from video data captured in 2D. By mapping the recognised hand shape from 2D to 3D, it is possible to obtain 3D co-ordinates of each of the joints within the hand using the kinematics embedded in a 3D avatar hand and smooth the transformation in 3D space between any given hand shapes. The novelty in our system is that it does not require a hand pose to be recognised at every frame, but rather that hand shapes be detected at a given step size. This architecture allows for a more efficient system with better accuracy than other related systems.

Index Terms — Kinematics, Avatar, Hand shape, 3D co-ordinates, Joints, Step size.

I. Introduction

Hand shape recognition is an important aspect of sign language gesture recognition. It is one of the five fundamental parameters that characterize any sign language gesture, the other four of which are: hand location, hand orientation, hand motion and facial expressions. Consequently, different sign language gestures may have different meanings for the same arm configurations when combined with different hand shapes. Therefore, hand shape recognition is a pivotal part of any system that attempts to translate between sign language and written text such as the SASL Machine Translation system at the University of the Western Cape [1]. In addition to sign language recognition, hand shape analysis forms the pivotal component of other applications such as Human Computer Interaction (HCI), generalized gesture recognition and sign language animation [2] [3].

Currently, the only technology that is capable of carrying out hand shape analysis on hand-based input with a satisfactory accuracy is glove-based sensing [4]. This method requires the user of the system to wear a glove fitted with motion and/or location sensors. Unfortunately, this technology has several drawbacks. It hinders the ease and naturalness with which the user can interact with the system, it may be costly and difficult to acquire by the average user and it may require long calibration setup procedures. Therefore it is not an ideal solution. On the other hand, computer vision has the potential to provide a more natural, non-contact solution. A camera captures hand and body motions of the user. The captured information can then be processed using computer vision to determine the location, motion and shape of the hands. This method has proven to be challenging since the human hand has many joints and has geometrically high degrees of freedom. Additionally, unlike the human face or body, the hand lacks patterns and textures which can be used as reliable recognition features. The most characteristic feature of the hand is its outline. Edges and silhouettes of the human hand are commonly used features in hand pose estimation schemes [2] [5].

This paper investigates the use of a learning-based approach to hand shape recognition under normal lighting conditions. CAMSHIFT and Gaussian Mixture Models (GMMs) are used to track the hand. The contours of the hand are extracted using Connected Component Analysis. Support Vector Machines (SVM) is used to recognize hand shapes from the features extracted. The recognized hand shape is then used to animate a 3D hand. Optimizing the recognition accuracy and processing speed of the recognition process was the focus of this project. Figure 1 is our system High level process flow chart.

Ultimately, this work will be integrated into previous work done at the SASL Group at the University of the Western Cape that focused on upper body pose recognition and estimation by detecting the arm movement and interpreting the meaning thereof [6] [7].

This paper is organized as follows: section II reviews the related work in the field of hand shape recognition and estimation; section III discusses the methodology used to implement the hand shape recognition system and discusses the techniques used; section IV is experimental setup; the result and analysis are presented in section V.

II. Related Work

A. *Shimada et al.'s work*

Shimada *et al.* Combined 2D appearance-based and 3D model-based approaches [5]. They applied a database retrieval approach and constructed a database of 16,000 possible hand silhouette contour images using a 3D hand model. The data was collected using a Data Glove. The data set included every possible hand shape. Each possible hand shape example was labelled along with a set of values for the

3D configuration of the joints in the hand for that hand shape. A transition network searching method was utilized so as to avoid exhaustively searching the database such that only those templates that are likely successors to the previous hand shape are evaluated. By retrieving the hand shape that best matches the input image, the 3D co-ordinates of the input hand shape can be rapidly obtained from the database.

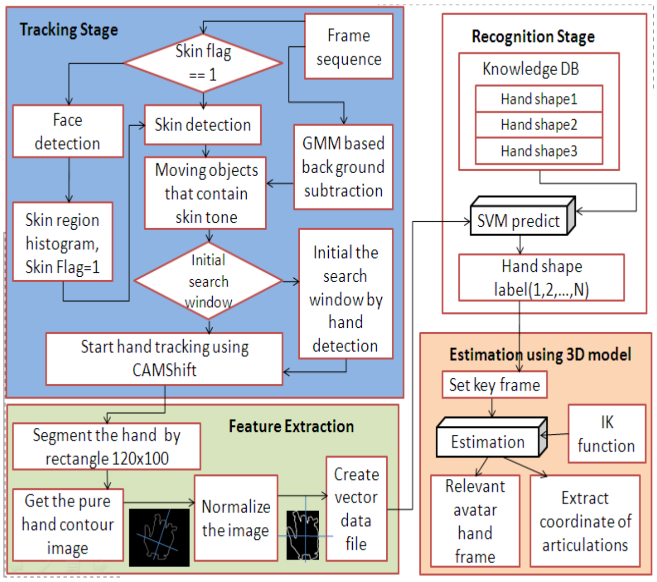

Tracking Stage
Skin flag == 1
Frame sequence
Face detection
Skin detection
GMM based background subtraction
Moving objects that contain skin tone
Skin region histogram, Skin Flag=1
Initial search window
Initial the search window by hand detection
Start hand tracking using CAMShift
Recognition Stage
Knowledge DB
Hand shape1
Hand shape2
Hand shape3
SVM predict
Hand shape label(1,2,...,N)
Estimation using 3D model
Set key frame
Estimation
IK function
Relevant avatar hand frame
Extract coordinate of articulations
Feature Extraction
Segment the hand by rectangle 120x100
Get the pure hand contour image
Normalize the image
Create vector data file

Figure 1: *High level system process flow chart*

They tested their system on 6 computer cluster framework, with each pc comprising of a single Pentium III 600 MHz CPU and 256 MB memory. They achieved a processing time of 33 milliseconds per frame. The system accuracy rate that they achieved is not clear from the literature. The main drawback of their system is that the transition network can easily lose track in the event of self-occlusions or fast deformation of fingers. In this case, it is difficult for the system to recover even if a beam search is performed. Consequently, the system has to be manually re-initialized with a given hand shape. This makes the implementation unattractive in real-life applications such as sign language detection systems [2].

B. *Schreer and Ngongang's work*

Schreer and Ngongang constructed an avatar animation system for sign language video communication [2]. Their system recognizes a limited number of hand shapes in 2D space as opposed to recognizing every possible hand shape as in the method used by Shimada *et al.* [5]. In the hand tracking phase, uses a prior-knowledge-based skin detection method to segment skin pixels from non-skin pixels in the image. Only skin pixels are considered resulting in the appearance of skin blobs in the image, mainly the face and two hands. This is seen in Figure 2. Thereafter, a combination of horizontal-vertical projection and neighbourhood analysis is used to search the biggest blobs to identify the position of the hands and head. A bounding box is then drawn around the hand blobs where the search window begins. Similar to the MeanShift algorithm, the hand is detected in each frame by centering the search window from the last frame to the new centre of the same blob in the current frame. The centre of the blob is considered to be the centre of gravity of the blob. The hand contours are segmented by using a region growing approach on the centres of gravity of the respective blobs. For each contour point, the normal of its tangent along the contour is calculated. Thereafter, the distance from the contour point to the opposite side of the contour is measured. Based on the orientation of the fingers and a distance function, the hands are classified into 13 different gestures. This is a completely data-driven approach which does not need any training process or data set of hand shapes. The recognized hand shape was then used to animate a 3D avatar. The avatar automatically produced natural motions by interpolating between two hand shape key frames.

It is not clear as to the exact specifications of their testing hardware. Their system achieved a real-time performance of 25 frames per second using a common off-the-shelf web camera operating at a resolution of 640x480 pixels. Their system was tested using 13 test subjects. The details of the experiment are unclear but it is clear that each test subject was required to display each of the 13 recognized hand shapes. Each subject was asked to rate the recognition of the system based on the hand shape presented on a scale of 1 to 5. A score of 5 indicated an excellent result while a score 1 implied very poor recognition. Figure 2 is a summary of the average scores given to each of the hand shapes over all 13 test subjects.

A major drawback of this system is that it is unable to recognize hand shapes in which the fingers are not distinctly separated and visible.

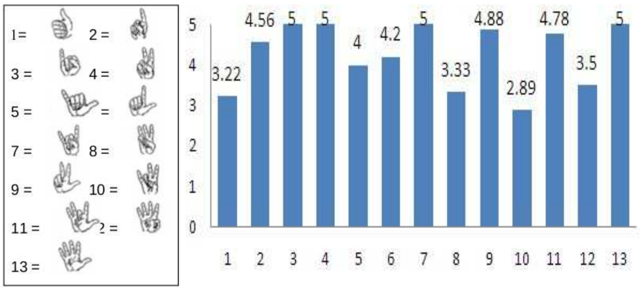


Figure 2: *The overall accuracy from Scheer's work*

C. *Sato et al.'s work*

Sato *et al.* used a learning-based approach to hand shape recognition [8]. They focused on hand shape recognition and did not focus on or specify the hand detection or tracking mechanism that they used. They used a neural network for hand gesture classification. Their system was trained to recognize 6 hand shapes, each hand shape labelled and trained on a number of normalized hand gesture images with the same meaning. When the off-line training phase is complete, the online hand shape recognition can be performed and is produces encouraging results. They tested their system on a Pentium II 450MHz, 384MB memory, WindowsNT4.0 Operating System. The system was found to be able to process images at 20 to 25 frames per second. The 6 hand shape patterns were recognized with more than 85% accuracy overall in this experiment. The typical mode of failure was misclassification occurring during transitions from one hand shape to another. This result is, however, reasonable since the neural network was not trained to recognize the intermediate hand shapes that caused error.

III. APPROACH OVERVIEW

This section explains the method used to detect and track the hands, extract features from and normalize the hands and recognize hand shapes in this research.

A. Hand detection and tracking

Hand tracking is an essential step for gesture recognition. The hand must be located in the image sequences and segmented from the background before recognition can take place.

Considering the fact that the human hand is a highly deformable object as well as the fast and discontinuous motion that is capable of carrying out, a template-based matching method would not be suitable for hand tracking. A skin-based tracking approach would be more suitable for this task. Other approaches that are popularly used to track the hands include the use of particle filters, KLT optical flow and CAMShift.[23] [24] [9]. These strategies are often used for usual object tracking and its effectiveness in this area will thus be evaluated. Each of these methods, however, have their respective strengths and weaknesses.

In this paper, we use CAMShift [9] that is based on the mean-shift algorithm as our primary tracking method. It is an effective colour-based tracking technique and has a low computational cost. It essentially climbs the density gradients to find the peak of probability distributions [10]. It works with a search window that is positioned over a section of the distribution. Within this search window, the maximum can be determined by a simple average computation. The search window is then moved to the position of this maximum and an average computation is repeated. This procedure is repeated until the CAMShift finds a local maximum and converges.

CAMShift tends to be robust against transient occlusion because the search window tends to first absorb the occlusion and then stick with the dominant distribution model when the occlusion passes. It is able to effectively track objects when there are no additional objects in the scene that are similar to the target object, as well as when the colour of the background differs from the target object. This technique, however, is often affected by noise in the background that has a similar or higher color probability distribution than the tracked object. To aid this weakness, background subtraction using Gaussian Mixture Models (GMM) and motion detection can be used as a pre-processing step in order to eliminate tracking of false objects.

1) Adaptive skin detection using face detection

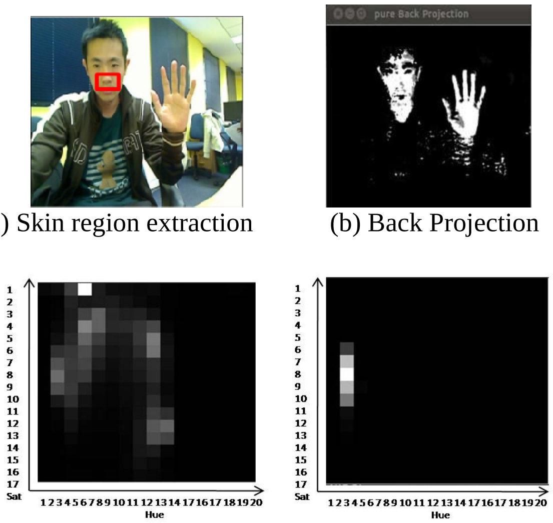
In order to achieve versatile skin detection which can adapt to different skin colours, face detection is used [11] to locate the face within a frame and extract the skin colour distribution of the region around the nose [12]. This colour distribution is used to identify all the skin pixels of an individual in a frame for any skin colour type.

2) Back projection and GMM background subtraction

Using the extracted skin colour distribution, back-projection is applied to the frames in order to highlight the hands and eliminate all objects that do not fall within the skin colour distribution [13]. This method describes a probability distribution of skin pixels in an image depending on an H-S histogram of a pre-selected skin region. In an H-S histogram, if *C* is the colour of a pixel, *F* is the probability that a pixel is skin, *P(C|F)* is the probability of drawing that colour if the pixel actually is skin, then *P(F|C)* is the probability that the pixel is skin given its colour.

$$p(F \mid C) = \frac{p(F)}{p(C)} p(C \mid F) \qquad (1)$$

This probability of skin pixels is back-projected to a gray scale image. An example of a back-projected image is shown if figure 3b. An intensity value of 255 indicates a high likelihood ratio of skin colour while a value of 0 indicates that the pixel has no skin colour.



(a) Skin region extraction (b) Back Projection

(c) Region Image H-S Histogram (d) Skin Region H-S Histogram

Figure 3: *Back projection and H-S Histogram*

However, the image may contain objects which have a similar colour to that of a skin pixel. This problem is solved using Gaussian Mixture Model (GMM) background subtraction [14].

GMMs can efficiently separate the background and foreground. The desired foreground is a scene where only the hands are present. The only requirement of this background subtraction technique is that the user sits or stands still for approximately 2-3 seconds. In this period, the background model is estimated by Expectation Maximization (EM) algorithms [15]. The background model continuously updates until the system obtains the first skin probability image after back projection has been executed. Thus, when the following frame is retrieved, the foreground pixels for the hands can automatically be separated from the image using the probability distribution function combined with the background model. Since it is only moving skin coloured objects that are required, a logical “AND” operation is performed between the motion cue technique and the colour cue technique to produce a final image where the hands have been successfully extracted. To complete the silhouette of the hands, morphology algorithms, namely opening and closing. Noise is smoothed out and holes are filled in the silhouette area of the hand.

3) Initialization of the CAMShift algorithm using Chamfer Distance matching

Using the image produced by combining the motion cue and skin cue techniques, Chamfer distance matching is used initially locate the hand in the scene. Hierarchical Chamfer matching [16] of the hand operates by detecting different scale sizes of the hand, ascending from a small scale to a large scale. In this system, the face size is use to determine the optimal scale size of the hand in the scene at initialization. The theory is based on the proportions of the human body according to the theory of Da Vinci [17]. This method vastly improves the speed of the system initially. It should be noted that the location of the hand is determined only once using chamfer matching so that a region-of-interest (ROI) of the hand is found. Subsequently, this ROI is continuously tracked using CAMShift. The ROI was given a specific size of 120x100 in the experimentation carried out.

On the other hand, face detection is used to eliminate the face skin pixels from the new skin cue image since the face may affect CAMShift tracking when the hand approaches the face.

B. *Feature extraction and normalization*

In order to gain a distinct feature from different hand shapes, we take the hand contour as the feature vector. Considering that noise might still exist in the binary images of our new skin-based image, a chain code based algorithm [18] is used to extract the contours of all blobs in the scene. The contour which has the largest number of contour pixels is considered to be the hand. In addition, a computational geometry algorithm [19] can be used to obtain an oriented minimum bounding box to segment the hand contours from the ROI of CAMShift.

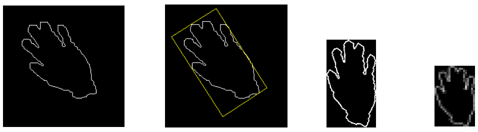


Figure 5: *Hand shape normalization*

In order to solve misalignment invariance, normalization is carried out by rotating and scaling the extracted hand region. The region is rotated based on the orientation of the principal axis of the region, so that the axis is aligned to one of the image axes. Thereafter, the image is scaled to a resolution of 20x30 pixels. This step is essential when aiming to reduce the computational cost as well as the effects of variation in the size of the hand. This pre-processing step allows for better generalization of features when training and testing using SVMs.

C. *Training and recognition using SVMs*

A Support Vector Machine (SVM) [20] is a supervised classification technique for creating a classifier given a set of features. Projecting data onto higher dimensions makes the data more likely to be linearly separable. The algorithm learns to separate hyperplanes that maximally separate the classes in the higher dimension. It is claimed to be among the best classification techniques given limited data [21].

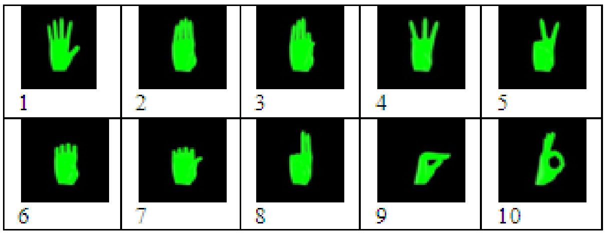


Figure 5: *Hand gestures from South African sign language*

The system was trained on the 10 hand shapes shown in Figure 5. 10 videos, one for each of the 10 hand shapes, were recorded, with each video consisting of no less than 1000 frames. In each video, the performer was asked to hold up the hand shape from start to stop. Using the methods explained in the previous section, the hand contours were automatically extracted from the video. Of the 1000 or so frames, 40 examples were manually selected as training data for each hand shape. This was followed by the feature normalization stage. The resolution of the hand contour image was scaled a 20x30 image. Finally, all the hand contour images were written to a feature data file. Each contour image was represented as a vector with a size of 600x1. In each element of a vector, a “1” was inserted to represent a contour pixel and a “0” for other pixels. The vectors of same hand shape were given a common label. The SVM was trained on the data of all 10 hand shapes. The output of the training is a set of optimized weights which can also be understood as a knowledge-based data model.

Given input from the detection and tracking phases, the SVM attempts to recognize the hand shape based on the data model produced. Practically, this proved to be problematic in cases where the user moved between two known hand shapes. The system was unable to correctly classify intermediate hand shapes between two hand shapes that were known and the correctness of the recognition was affected. It was resolved to carry out recognition after every 3 frames. With this modification, in the deformation between two known hand shapes, most intermediate hand shapes are skipped. The remaining intermediate hand shapes are very similar to the next gesture and do not significantly affect the recognition accuracy.

D. *Estimation using 3D animation*

Using a 3D avatar hand developed by van Wyk [22], two detected hand shapes are set as keyframes, with Blender interpolating between the two keyframes. Figure 6 is an example of the hand shape estimation that Blender carries out between two detected hand shapes G(t) and G(t+1) . The transformation in 3D space between the two hand shapes can be generated by means of interpolation. It should be noted that the animation system continues to correctly update the estimation process so long as the next detected hand shape from the recognition phase G(t+1) is correct.

The fact that the SVM is not 100% accurate implies that, in many cases, an incorrect classification is carried out. Consider the case where the user holds up a stable open hand. The SVM is found to recognize various different hand shapes based on the same stable hand shape from the input. A modification was made to the estimation such that only if three consecutive frames of the same classification were recognized would the estimation process consider the frame as a key frame. All other recognized frames are ignored. This significantly stabilized the estimation process. The system is able to operate at a rate of 18 frames per second.

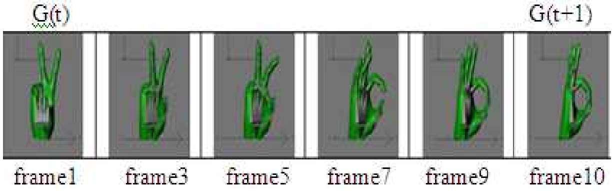


Figure 6: *Example of hand shape estimation by 3D animation* [25]

IV. Experimental setup

The experimentation carried out aimed to determine the recognition and estimation accuracy of the system as well as obtaining an indication of its robustness to variations in skin tone. The system was tested on a MacBook Pro with Inter Core 2 Duo, 2.53 GHz CPU and 4GB RAM, running an Ubuntu Linux 10.10 Operating System. A Logitech note book Quick Cam web camera was used at a resolution of 640x480 pixels. Figure 7 depicts the location at which the experimentation was carried out as well as the equipment used.

Figure 7: *The scene for testing*

5 persons with different skin tones participated in this experiment. Figure 8 depicts the skin tones of each of these subjects. Each subject was required to sit on the chair in Figure 7 in front of the camera. Referring to Figure 5, each subject was asked to perform and transition between hand shapes 1 to 10, in that order, while continuously being recorded. Each subject was also asked to hold each hand shape for 3 seconds – no less than 75 frames – before transitioning.

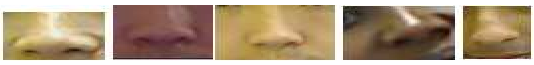


Figure 8: *The skin ton of each subject*

The system carried out recognition and estimation as explained in previous sections. Two sets of results were obtained. The accuracy of recognition was obtained by manually comparing the input for each hand shape to the recognized result. 15 comparisons were performed per hand shape per test subject. The accuracy of estimation was determined by comparing every frame of the input with every frame of the estimated result generated by the 3D model. The results were grouped according to the transitions between every two hand shapes.

V. Results and analysis

Tables I and II summarize the results of the recognition accuracy and estimation accuracy experimentation.

Table I
The Accuracy of hand shape recognition

Hand shape	**Recognition Accuracy (%)**
1	55%
2	80%
3	85%
4	90%
5	94%
6	82%
7	79%
8	85%
9	83%
10	79%

Table II
The Transition between two hand shapes

Hand shapes	Estimation Accuracy (%)
1-2	75%
2-3	90%
3-4	80%
4-5	93%
5-6	94%
6-7	82%
7-8	90%
8-9	85%
9-10	89%

Overall, the system achieved a recognition accuracy of 81% and an estimation accuracy of 86%. These results are very encouraging. Only one hand shape scored recognition below 75%, hand shape 1. This result is attributed to the fact that different users performed this hand shape differently. Figure 9 depicts the contours of this hand shape as detected from different test subjects. Although the underlying hand shape was generally the same, small variations led to very different recognition results. One solution to this problem is to train the SVM on more samples for this hand shape.

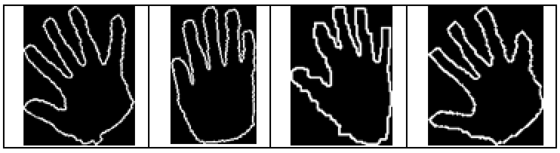


Figure 9: *Different recognition results for hand shape 1 for different test subjects.*

It is seen that the process of estimation minimizes the number of errors made by the recognition using the heuristic mentioned in a previous section. Overall, the final estimated result matched the input with a higher accuracy than the result of the recognition phase. It was also observed that skin tone did not affect the performance of the system.

VI. Conclusion

This paper presented an approach to recognizing and estimating hand shapes. Hand tracking was carried out using the CAMShift algorithm. The hand contours were extracted in each frame. Normalization was carried out to ensure that the system is able to operate at different hand sizes and orientations. An SVM was trained and used to recognize 10 pre-selected hand shapes. Finally, a 3D Blender hand model was used to estimate transformations between hand shapes using key framing. The system was found to recognize and estimate hand shapes with a very high accuracy.

References

[1] J. Connan, Integration of Signed and Verbal Communication: South African sign language Recognition and Animation. [Online] http://www.coe.uwc.ac.za/index.php/SASL.html, (Accessed: April 2011).

[2] O. Schreer, S. Ngongang, "Real-time gesture recognition in advanced videocommunication services", in *Proceedings of the 2007 IEEE International Conference on Image Analysis and Processing, IEEE Computer Society*, 2007.

[3] M. Ghaziasgar, "The use of mobile phones as service-delivery devices in a sign language machine translation system," Master's Thesis, University of the Western Cape, Computer Science, June 2010.

[4] Ali Erol, George Bebis, 2005, "Vision-based hand pose estimation: A review", *ScienceDirect* 108 (2007) 52-73.

[5] N. Shimada, K. Kimura, Y. Shirai, Real-time 3D hand posture estimation based on 2D appearance retrieval using monocular camera, in: *ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, *IEEE*, Vancouver, *BC*, Canada, 2001, pp. 23–30.

[6] N. L. Naidoo, J. Connan, "Gesture recognition Using Feature vectors ," in Proceedings of the Southern Africa Telecommunication Networks and Applications Conference.SATNAC, 2008.

[7] I. Achmed and J. Connan, "Upper body pose estimation towards the translation of South African sign language," in Proceedings of the Southern Africa Telecommunication Networks and Applications Conference. SATNAC, 2010, pp. 427–432.

[8] Y. Sato, M. Saito, H. Koik, Real-time input of 3D pose and gestures of a user's hand and its applications for HCI, in: *IEEE Virtual Reality Conference, IEEE Computer Society*, 2001, pp. 79.

[9] G. R. Bradski. "Computer visions face tracking for use in a perceptual user interface", In *Proc Workshop on Application of Computer Vision*, pp. 214-219, 1998.

[10] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time Tracking of Non-rigid Objects using Mean Shift", *CVPR.*, South Carolina, 2000, pp. 142–149.

[11] P. Viola and M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Proceedings of IEEE Computer Society's Computer Vision and Pattern Recognition (CVPR 2001)*, Vol. 1, pp. 511-518, 2001.

[12] I. Achmed, "Upper body pose recognition and estimation towards the translation of South African sign language", Master's Thesis, University of the Western Cape, Computer Science, December 2010.

[13] M. J. Swain, D. H. Ballard,"colour index", *International Journal of Computer Vision,* 7:1, 11-32(1991)

[14] Stauffer C, Grimson W. E. L. Adaptive background mixture models for real-time tracking. in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Cat. No PR00149). *IEEE Comput. Soc*. Part Vol. 2, 1999.

[15] P. KaewTraKulPong and R. Bowden, "an improved adaptive background mixture model for real-time tracking with shadow detection", in *Proc. 2nd European workshop on advance video based surveillance systems*, 2001.

[16] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849–865, 1988.

[17] L. DA VINCI, A Treatise of Painting, trans. from the original Italian, London, 1721.

[18] H. Freeman, "On the classification of line-drawing data," Models for the Perception of Speech and Visual Form (pp. 408 – 412), 1967.

[19] H. Freeman and R. Shapira, "Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve", Comm. ACM, 1975, pp.409-413.

[20] C. W. Hsu, C. Chang, C. J. Lin, "A practical guide to support vector classification" [Online] Available at http://www.csie.ntu.edu.tw/~cjlin, [Accessed: April 2011]

[21] V. Vapnik, The Nature of Statistical Learning Theory, New York: Springer Verlag, 1995.

[22] D. Van Wyk, "Virtual human modeling and animation for sign language visualization," Master's Thesis, University of the Western Cape, Computer Science, December 2008.

[23] M. Kolsch and M. Turk. Fast 2D hand tracking with flocks of features and multi-cue integration. In *IEEE Workshop on Real-Time Vision for Human-Computer Interaction, in association with CVPR*, Washington DC, USA, July 2004.

[24] V. Spruyt, A. Ledda, S. Geerts, "Real time multi-colour space for real-time hand tracking and segmentation", *Proceedings of 2010 IEEE 17th International Conference on Image Processing*. 2010. pp.3117-3120.

[25] P. Li, South African Sign Language Recognition and Estimation. [Online] http://www.cs.uwc.ac.za/~lpei/index_sub_demo.htm, (Accessed: June 2011).

Pei Li is currently an M.Sc student at the University of the Western Cape. He is currently doing research on Sign Language recognition and estimation for South African sign language recognition.

Mehrdad Ghaziasgar is the project manager of the South African Sign Language (SASL) research group. He has a wide range of interests that include internet programming, mobile computing, computer vision and machine learning.

James Connan heads up the South African Sign Language (SASL) research group. His interests computer vision and machine learning.