Upper Body Pose Estimation towards the translation of South African Sign Language

Imran Achmed and James Connan
Department of Computer Science
University of the Western Cape, Private Bag X17 Bellville, 7535, South Africa
Telephone: +(27) 21 959-3010, Fax: +(27) 21 959-3006

email: 2507311@uwc.ac.za , jconnan@uwc.ac.za

Abstract—The recognition and estimation of the human body are fundamental to high-level image understanding. It is a challenging problem to infer human body postures in image and video sequences due to the variance in human appearances, the background clutter of natural scenes and the high dimensionality of articulated three dimensional (3D) body models. To illustrate this high dimensionality, the human arm alone has 7 degrees of freedom. This creates a very large search space that makes computation quite difficult. We propose two systems and provide comparative results between a common example-based method and our novel learning-based method. The example-based method implements silhouette and shape context matching, whereas the novel learning-based method utilizes Skin Detection and Support Vector Machines (SVM). We aim to use this research in gesture recognition for Sign Language and therefore restrict our focus to the upper half of the body.

Index Terms—Body Pose Estimation, Chamfer Distance Transformation, Skin Detection, Support Vector Machines

I. Introduction

The study of human pose estimation from visual input is a diligent division of computer vision research in many application domains. Applications ranging from human-computer interaction to intelligent surveillance, analysis in sports, virtual reality and more importantly the focus of this research, Sign Language recognition. Human pose estimation can be defined as the process of finding and estimating the approximate positions of the relevant body parts either in two-dimensional (2D) or three-dimensional (3D) view space. Using computer vision to estimate a pose lightens the burden and stress for users since they do not need to wear additional cumbersome clothing, however, the relation between image observations and poses is arguably one of the most complex problems to solve. The variations in different body dimensions, appearances and environmental scenes give rise to many possible observations for exactly the same pose. Moreover, several distinct poses could result in a similar observation due to camera perspectives, projections, and self occlusions. Apart from these properties, the major difficulty is in the large number of degrees of freedom (DOF) in the human body, requiring computationally intensive searching of the solution space. To address these challenges, many studies have been undertaken on approaches using computer vision, 3D modeling, and machine learning.

In this paper we present a comparison between an example-based and a novel learning-based approach to human pose estimation. We implement the first approach by applying a silhouette feature extraction method, followed by the Chamfer Distance Transformation. The distance transformed images are then stored in a MySQL database. The database is searched when recovering a pose using the distance transformation metrics as a measurement for the most likely pose. In the latter approach, we develop a framework based on a skin feature extraction method and SVMs. Features representing skin, which are most likely to be the hands and face, are extracted and placed into a vector. In the training phase the vectors are labeled according to the class that corresponds to the pose. During the testing phase, a query image is classified into a class predicted by the SVM. We evaluate these approaches and report our findings. We also analyze the comparison between the results.

The paper is furthermore organized as follows: In Section II we discuss related work; Section III discusses silhouette shape context matching and 3D modeling; Skin Detection and SVMs are discussed in Section IV. In section V the experimental design and analysis is presented. We conclude this paper in Section VI and provide some future work in Section VII.

II. Related Work

The approaches to human pose estimation can be broadly classified into three categories: (1) model-based approaches, (2) example-based approaches, and (3) learning-based approaches.

Model-based approaches involve fitting a known parametric model or a partly-based model to an image by optimizing the correspondence between the image and the pose [1]. Parameswaran and Chellappa [2] assume corresponding points are provided between the image and the model. They use geometric constraints to estimate an individual’s pose. Ioffe and Forsyth [3] propose a bottom-up approach. They extract possible body parts and use kinematic constraints to group them into image segment groups. Zhang et. al. [4] propose a hybrid strategy by combining the top-down Markov Chain Monte Carlo method with a bottom-up local search to estimate a 2D pose. Model-based methods can become increasingly complicated and computationally expensive. It also suffers when body part detectors fail.

Example-based approaches search a large database containing a set of training examples for the most likely pose. Hayashi et. al. [5] developed a framework that uses silhouettes represented by shape context descriptors. They estimate parameters for an input image by computing the weighted average of the pose that closely corresponds to the shape context descriptors. Poppe and Poel [6] compared the performance of Fourier Descriptors used for pose recovery. Shakhnarovich et. al. [7] proposed an efficient search algorithm that learns a set of hash functions to retrieve approximate nearest neighbours likely to resemble a given query image. Example-based approaches often incur problems where there are not enough examples to cover the entire pose space. If we limit the space to a particular domain such as Sign Language, we limit the pose search space.

Learning-based approaches extract features from an image which is later used to train a model and predict a pose depending on a regression function. Agarwal and Triggs [1] proposed a tracking framework that recovers poses by using Sparse Bayesian nonlinear regression of joint angles against silhouettes encoded by a histogram of shape context descriptors. In Okada and Soatto [8], feature vectors are based on histogram of oriented gradients and discriminated using an SVM to predict the pose. Ronford et. al. [9] trained support vector classifiers for body parts based on Gaussian derivative filters. Learning-based approaches are appealing due to them being computationally inexpensive and fast enough for real-time applications.

III. Silhouette Shape-Context Matching

Shape matching is a common problem in pattern recognition, tracking and image analysis. Various matching methods are based on the type of features used [10] and can be categorized into three method groups: (1) methods that directly use image pixel values, (2) methods that use high-level features and (3) methods that use low-level features. The edges in an image are essentially low-level features and provide a useful solution. When matching a template to an image, edge points are used to directly compute the exact Euclidean distance between the two; however, this approach takes an excessive amount of time and is resource intensive. An alternative approach is to use more efficient algorithms, such as the distance transform to obtain the approximations thereof. Several approximation algorithms have been developed that are adequately efficient to calculate distances using a rectangular coordinate system. Apart from the Euclidean metric, there is city block or Manhattan metric, the chessboard metric and the chamfer metric. We focus on the chamfer metric, as the upper limit for the difference between the Euclidean and the chamfer metric is much better than the upper limits for the city block and chessboard metric [11].

A. Chamfer Distance Transformation

Chamfer Distance Transformation, based on the chamfer metric, is a technique that calculates the best fit from two individual images using edge points, by finding the least approximate distance between them [12]. The technique assigns values to each pixel according to the distance from the respective pixel to the nearest edge. This operation involves using a mask and scanning twice over an image. The commonly used mask is the 3x3 mask but has been extended by [13] to 5x5 and 7x7 masks. Borgefors [13] recommends using (3:4) and (5:7:11) approximations for 3x3 and 5x5 masks respectively. She also finds no significant improvements when using the 7x7 masks. We have therefore chosen to use 3x3 masks with the (3:4) approximation. The (3:4) approximation can be defined in the following expression:

$$\begin{aligned} V_{i,j}^{k} = minimum\big(&V_{i-1,j-1}^{k-1} + 4, V_{i-1,j}^{k-1} + 3, V_{i-1,j+1}^{k-1} \\ &+ 4, V_{i,j-1}^{k-1} + 3, V_{i,j}^{k-1}, V_{i,j+1}^{k-1} + 3, V_{i+1,j-1}^{k-1} \\ &+ 4, V_{i+1,j}^{k-1} + 3, V_{i+1,j+1}^{k-1} + 4\big) \qquad (1) \end{aligned}$$

The scanning is initiated with a forward scan, by starting in the top left corner of the image, proceeding from left to right and from top to bottom. This is followed by the backward scan. The scan begins in the bottom right corner, proceeding from right to left and from bottom to top. The process is illustrated in Figure 1. From the procedure we are able to obtain local distances which represent the pixel values, for example $d1, d2$ and $d3$. During each scan, the pixel values are added and the minimum of the five sums are assigned to the pixel in the center. Following both scans, pixels near to an edge will have a lower pixel value and pixels further from an edge, a higher pixel value [12]. The transformed image will subsequently consist of an approximation of the distance from a pixel to the nearest edge.

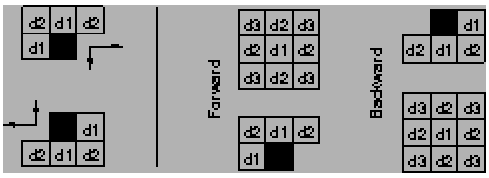

Figure 1: The procedure in which the mask scans the image [12].

We have only provided a brief explanation to the procedure of the Chamfer Distance Transformation. For further information regarding the Chamfer Distance Transformation, see “Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm” by Gunilla Borgefors[13].

B. 3D Human Body Model

Human body models describe the human body in shape, appearance and kinematic properties. The human body is represented as segments linked by joints where every joint has a number of DOF. The DOF specifies the limitations regarding movement of the body joints. Collectively, every DOF in the body model forms a representation of a pose. The body models are described as either 2D or 3D skeletal structures that resemble the articulated nature of a human body. Many pose estimation approaches make use of a common 3D model called POSER; however, this model falls under a commercial license. An alternative 3D model has been developed by Van Wyk, who was part of the SASL group at UWC [14]. He developed an open framework that uses various standards and open technologies to model and animate virtual 3D human models, mainly for the purpose of visualizing Sign Languages. The 3D model employs standards and technologies that include H-Anim, MakeHuman, Blender, and Python. The H-Anim technology in his system has been extended and adapted to perform sign language movements effectively. We have therefore chosen to use his system as it is ideal for our purpose. The 3D model allows us to easily manipulate the joints into any position required and at the same time allows us to retrieve ground truth data for the x, y and z coordinates for the respective joints. In our pose estimation process, we are mainly focused on retrieving the coordinates for the joints at the shoulders, elbows and wrists as these joints are primarily used in performing sign language. In Figure 2 we illustrate a pose of the 3D model.

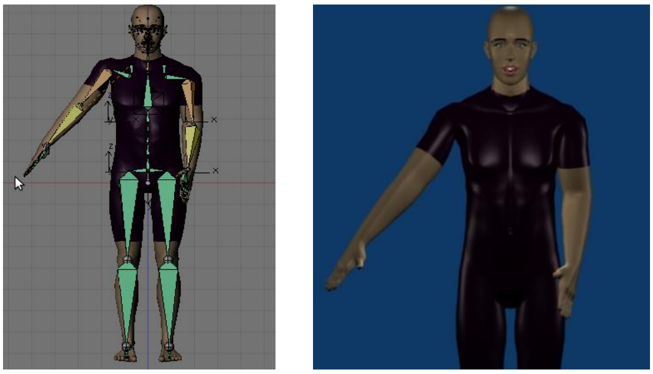

Figure 2: A single pose created by the Blender 3D model.

The 3D model furthermore allows us to animate a Sign Language phrase while effectively estimating the pose throughout the phrase.

IV. Skin Detection and Support Vector Machines

A. *Skin Detection*

Skin detection can be defined as a process of identifying pixels in images and video as possible skin-coloured pixels. This process plays a significant task in a wide range of applications and has a special usefulness in detecting faces or hands in images or even tracking hands in video. It has gained much attention as skin colour information has proven to provide robust information against rotations, scaling and partial occlusions of human body parts. Detecting skin-coloured pixels, however, is not an easy task. The appearance of colour pixels in images varies due to various factors such as illumination changes, viewing geometry and camera characteristics. In order to detect skin in an image using colour pixel information, the pixels need to be represented in a suitable colour space. On the human body, the appearance of skin is formed by a combination of colours relating to blood (red) and melanin (brown and yellow) [15]. Furthermore, skin colour occupies only a small area in a given colour space and differs amongst the colour spaces [15]. Pixels in an image can be classified as either a skin pixel or a non-skin pixel. Choosing an appropriate colour space determines how effectively the skin-colour distribution in an image can be identified. It also affects the sensitivity of skin detection to variations in illumination conditions. Various colour spaces have been proposed with the aim of detecting pixels as either skin pixels or non-skin pixels.

The default colour space for most image formats is the RGB colour space. The RGB colour space is a combination of red, green and blue pixel values, and is not a preferred colour space because varying illumination affects all the red, green and blue pixel values. The other colour spaces are obtained from a linear or non-linear transformation of the RGB colour space. The transformation decreases the overlap between skin and non-skin pixels, and also presents robust parameters against variations in illumination conditions. From a survey on skin-colour detection methods [16], it has been ascertained that skin-colour pixels differ more in intensity that in chrominance and it has been found that the luminance component can be eliminated when detecting skin. In this paper we make use of the HSV colour space since it is known to be more compatible with human perception and therefore also referred to as a perceptual colour space. The HSV colour space defines colour as Hue, Saturation and Value (also referred to as Intensity or Lightness). The non-linear transformation from RGB to HSV colour space can be formulated as [17]:

$$v = max_{(r,g,b)}$$

$$s = \frac{max_{(r,g,b)} - min_{(r,g,b)}}{v}$$

$$h = \begin{cases} \dfrac{g-b}{6(max_{(r,g,b)} - min_{(r,g,b)})}, if\ v = r \\ \dfrac{2-r+b}{6(max_{(r,g,b)} - min_{(r,g,b)})}, if\ v = g \\ \dfrac{4-g+r}{6(max_{(r,g,b)} - min_{(r,g,b)})}, if\ v = b \end{cases} \qquad (2)$$

where r,g,b represents red, green, and blue pixel values and h,s,v represents Hue, Saturation and Value pixel values respectively.

An advantage of the HSV colour space is that a boundary of skin colour pixel values can be specified in terms of the Hue and Saturation. The Value (Intensity or Lightness) contains brightness information. It is therefore not used and eliminated to reduce illumination on skin colour.

B. *Support Vector Machines*

An SVM is a machine learning tool, based on continuous advances in statistical learning theory that classifies data into two or more classes [18]. SVMs have proved to deliver competitive generalization performance in many real-world applications where data contains a small number of features and where data contains many features [18].

1) *Definition of SVMs*

In principle, an SVM is a mathematical algorithm that maximizes a mathematical function with respect to a given collection of data, either for classification or regression [19]. Their common factor is the use of the “kernel trick” that offers both power and flexibility. This allows the SVM to apply linear classification techniques to non-linear classification problems. It is achieved by replacing the default linear kernel with a radial basis function, sigmoidal, polynomial or other more recent kernels that may separate the data points more clearly for the given classification problem.

Given a set containing two classes of data points, it is possible to create a boundary that separates these two classes. The SVM can be trained to find this boundary, often referred to as the decision boundary. The decision boundary separates the two classes by a straight line, Figure 3(a).

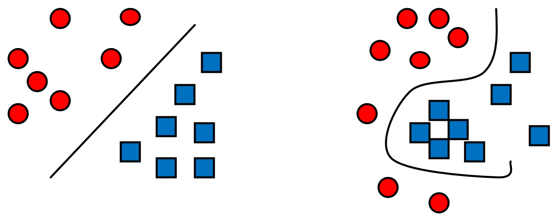

Figure 3: (a) Linear classification (b) Non-linear classification

This straight line in a high-dimensional space is generally referred to as a hyperplane. It is a geometrical conception and a generalization of the concept of a plane (Figure 4). The selection of the hyperplane is one that not only separates the data points clearly but also has the greatest distance to the closest data point from both classes. The distance is referred to as the margin and the data points that lie closest to the hyperplane are referred to as the support vectors. We wish to find the maximum margin since this allows the SVM to more accurately classify an unseen data point when the separation between the two classes is greater. The selected hyperplane is called the maximum-margin hyperplane or the optimal hyperplane. To sum this up, SVMs separates the data points with an optimal hyperplane by learning a decision boundary.

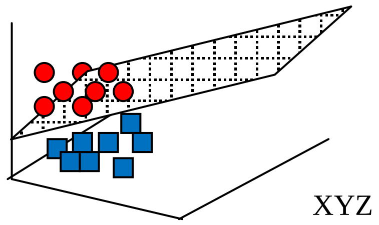

Figure 4: Linear classification of a plane

.

In the classification of linear cases, Figure 3(a), SVMs find a linear hyperplane that seperates the data points with a maximum-margin. However, in the classification of non-linear cases a slightly different approach is taken. The SVM maps the data points into a higher dimensional space and then finds a decision hyperplane that separates the data points linearly [18].

Most real world problems involve non-linear cases where more complex structures are required to find an optimal hyperplane. In these cases, Figure 3(b), the data points are non-separable or unevenly distributed compared to those in Figure 3(a). We also find that a curve does not provide a suitable hyperplane between the classes. By using a kernel function, the SVM maps the data points into a higher dimensional space known as the feature space. When mapping the data to the feature space, the SVM finds a suitable optimal hyperplane that separates the data clearly. With an appropriate kernel function, any data can become linearly separable in the resulting higher-dimensional space. As an alternative to drawing a complex curve to separate the data, we are able to find an optimal line in the feature space that allows the SVM to make accurate classifications given new test data. For a mathematical derivation of SVMs, see [20].

V. Experimental Design and Analysis

In our experimental setup and analysis, we do a comparison between the example-based and the novel learning-based approach. We are therefore required to design two independent systems. The systems are evaluated on their performance and estimation accuracy for given postures in Sign Language phrases.

A. Example-Based System Design

In the following two sections we describe the configuration of the database of poses and the testing procedure of the system.

1) Configuration of the Database

In this approach we require a large database of human postures relating to Sign Language. We therefore make use of the Blender 3D model to generate these poses since a large number of these can be generated automatically along with the corresponding positions for the shoulders, elbows and wrists. The image poses and joint positions are stored in the MySQL database. This is followed by silhouette feature extraction on the image poses using the Canny edge detection algorithm. The Chamfer Distance Transformation is applied to the edge detected image and generates a Chamfer distance transformed image. The transformed image is stored in the database alongside the original image.

2) Testing Procedure

When testing this approach, a video of the signer (person performing the Sign Language) is recorded. The images that represent the frames are extracted from the video. For each image the following procedure is carried out. A human face is searched for in the image using OpenCV's face detection algorithm that uses Haar features and AdaBoost. Using this face detection algorithm, the coordinates of the center of the face can be retrieved, which is intelligently used to move the image in any direction such that the signer would overlap the 3D model in the transformed image. This is an imperative step as the corresponding points between the image and the 3D model's transformed image are used to calculate their distance. The silhouette feature extraction is performed on the image using the Canny edge detection to identify the silhouette of the human body. Since the silhouette now overlaps the 3D model, the edge detected silhouette and each of the transformed images in the database are used to compute a distance value. The transformed image with the lowest distance value represents the most likely pose. Using the MySQL database structure, the pose and joint positions for the chosen transformed image are retrieved. Thus, for an image we are able to retrieve the most likely pose using this procedure.

B. Learning-Based System Design

In the following system design, we describe the procedure required for the training and testing of the system.

1) Training Procedure

Videos of signers performing signs are recorded, from which image frames are extracted. The images extracted represent individual postures of the signs performed. We adopt the same face detection algorithm as the previous approach. Using this algorithm, the coordinates of the center of the face is retrieved and used to move the image such that the signer occupies a centered location at all times. This insures the position of each signer will be constant no matter where he/she might be standing in the image. When applying skin detection to the image, skin pixels are given the pixel value of 255 and non-skin pixels the pixel value of 0, as seen in Figure 5.

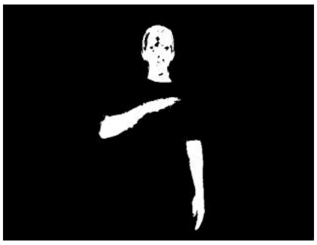

Figure 5: Skin detected image of a signer

Using the face detection algorithm, the size of the face can be determined and a 6x7 grid can be drawn on the image. The size of the grid is determined based on body proportions, often used by artists when sketching a human body [21]. It should be noted that the grid is only used for labeling purposes and does not form part of the feature vector. The blocks in the grid are labeled in a range from 1 to 42. For each image the shoulders, elbows and wrists are manually labeled according to their positions in the relative blocks (Figure 6). The labels and feature vectors consisting of pixel values from the images are used to form a data file. The multi-label data file is transformed to a multi-class data file before training it using the SVM.

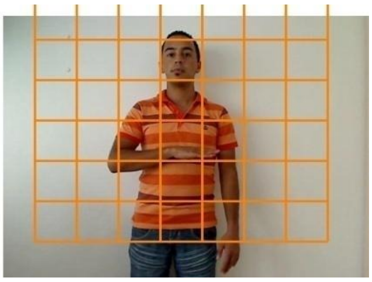

Figure 6: The grid superimposed on the image

2) Testing Procedure

In this procedure the signer performs a sign and every fifth frame is extracted from the video. The center of the face is retrieved using the face detection algorithm and used to move the image such that the signer would be on the same centered location as in the training phase. The image is skin detected, and skin pixels are given a value of 255 whereas non-skin pixels a value of 0. The pixels in the skin detected image are used to form a feature vector. In this phase no grids are superimposed on the images and every image is given a default class of 0. The feature vector is used to form a test data file and used by the SVM to predict the class corresponding to the most likely pose. After predicting the class, the labels for the particular class can be retrieved. Using the labels which correspond to the grid in the training procedure allows us to automatically place the Blender 3D model in the exact same pose as the image. We are therefore able to find ground truth data and estimate a pose for the given image. Furthermore, for every fifth frame predicted, the Blender 3D model is able to fill in key frames in between. For example, the 5th frame and 10th frame are predicted. The frames 6, 7, 8 and 9 are automatically predicted based upon the difference in movement of the limbs from the 5th to 10th frame. Therefore, we are able to estimate an entire sign and find ground truth data using the Blender 3D model.

C. Result to the Approaches

To evaluate the performance and estimation accuracy of both systems, we do a comparison between the two, based on distinct postures taken from 20 signs. Our test set consists of 6 different people, 3 males and 3 females. The same test set is used for both systems. The poses are estimated on its visual similarity within the margin of human perception. The model only provides an estimation of the joints; a comparison can therefore not be made on the positions of these joints as a pose may be deemed correct if the pose is visually similar to the target pose, even though their positions are not exactly the same.

Sign	**Example-Based**	**Learning-Based**
Taxi	100%	100%
Dress	50%	100%
Left	100%	100%
Water	83%	83%
Wizard	100%	100%
To wee	83%	100%
Please	0%	67%
Shorts	33%	83%
Right	100%	83%
Wide	100%	67%
Young	17%	50%
Handkerchief	17%	67%
Hot	83%	83%
Hello	0%	50%
Short	100%	100%
A plank	0%	100%
Smoke	100%	83%
Fat	0%	50%
Me/You	50%	67%
Light	100%	83%
Average	**61%**	**81%**

Table 1: Results to systems based on Sign Language signs.

Our results show the example-based system obtained better results in only 4 of the 20 signs compared to the learning-based system; however, the results obtained by the learning-based system in these 4 signs are satisfactory. Overall the learning-based system achieved an average of 81% compared to the example-based system with an average of 61%. This shows our novel learning-based system outperformed the example-based system and is a more suitable system for estimating human body poses in Sign Language recognition.

VI. Conclusion

In this paper we presented a comparison between an example-based and a novel learning-based approach. The example-based system uses Sign Language postures generated using the Blender 3D model. The system implemented silhouette feature extraction, followed by the Chamfer Distance Transformation and storing it in a database. Human postures are then estimated using a distance metric and joints' positions retrieved from the database. In our novel learning-based system, we developed a framework based on a skin feature extraction method and SVMs. Human data are used whereby features are extracted and trained using the SVM. Given a human signer, we are able to predict Sign Language postures using the SVM and estimate its position using the Blender 3D model. We tested our systems on 20 postures as these postures are common postures not only found in the signs presented but also found in majority of the other signs in the Sign Language dictionary. Based on these postures, we compare the two systems. Our results show that our learning based system achieves better estimation results and performs better than the example-based system.

VII. Future Work

We believe we can achieve better estimation results by further improving the learning-based system by implementing adaptive background subtraction as well as erosion and dilation techniques. The adaptive background subtraction involves continuously updating the background such that areas containing recent movement can be identified. The erosion and dilation techniques decrease and increase pixel regions respectively. This helps to eliminate noise when detecting skin in the images.

VIII. References

[1] A. Agarwal and B. Triggs, “Recovering 3D Human Pose from Monocular Images”, IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 28, No. 1, pp. 44–58, January 2006.

[2] V. Parameswaran and R. Chellappa, “View Independent Human Body Pose Estimation from a Single Perspective Image”, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2004.

[3] S. Ioffe and D. A. Forsyth, “Probabilistic Methods for Finding People”, International Journal of Computer Vision, Vol. 43, No. 1, pp. 45–68, June 2001.

[4] X. Zhang, C. Li, X. Tong, W. Hu, S. Maybank & Y. Zhang, “Efficient Human Pose Estimation via Parsing a Tree Structure Based Human Model”, IEEE 12th International Conference on Computer Vision, pp. 1349–1356, 2009.

[5] K. Hayashi, L. Heng & V. Strivastava, “Pose Estimation from Occluded Images”, 2006.

[6] R. Poppe and M. Poel, “Example-Based Pose Estimation in Monocular Images using Compact Fourier Descriptors”, In 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, 2005.

[7] G. Shakhnarovich, P. Viola & T. Darrel, “Fast Pose Estimation with Parameter Sensitive Hashing”, Proceedings of the Ninth IEEE International Conference on Computer Vision, Vol. 2, pp. 750–758, 2003.

[8] R. Okada and A. Soatto, “Relevant Feature Selection for Human Pose Estimation and Localization in Cluttered Images”, Proceedings of the 10th European Conference on Computer Vision: Part II, pp. 434–445, 2008.

[9] R. Ronfard, C. Schmid & B. Triggs, “Learning to Parse Pictures of People”, Proceedings of the 7th European Conference on Computer Vision-Part IV, pp. 700–714, 2002.

[10] P. Suetens, P. Fua, and A. Hanson, “Computational Strategies for Object Recognition”, ACM Computing Surveys, Vol. 24, No. 1, pp. 5–62, 1992.

[11] O. Cuisenaire, “Distance Transformations: Fast Algorithms and Applications to Medical Image Processing”, PhD Thesis, Universite Catholique de Louvain, 1999.

[12] I. Achmed and J. Connan, “A Panoramic Video System”, Proceedings in South African Telecommunication Networks and Applications Conference, pp. 279- 284, September 2009.

[13] 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, November 1988.

[14] D. Van Wyk, "Virtual Human Modelling and Animation for Sign Language Visualisation", MSc Thesis, University of the Western Cape, 2008.

[15] M. M. Fleck, D. A. Forsyth & C. Bregler, “Finding Naked People”, 4th European Conference on Computer Vision Cambridge, Vol. 2, pp. 593–602, April 1996.

[16] P. Kakumanu, S Makrogiannis and N Bourbakis, “A Survey of Skin-Color Modeling and Detection Methods”, Pattern Recognition, Vol. 40, No. 3, Pages 1106–1122, March 2007.

[17] Y. Lin, H. Tseng & C. Fuh, “Pornography Detection Using Support Vector Machine”, 16th IPPR Conference on Computer Vision, Graphics and Image Processing, pp. 123–130, 2003.

[18] L. Yi and J. Connan, “KernTune: Self-tuning Linux kernel performance using support vector machines ”, Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, pp. 189–196, 2007.

[19] WS. Noble, “What is a Support Vector Machine?” Nature Biotechnology, pp. 1565–1567, 2007.

[20] A. Tzotsos, “A Support Vector Machine Approach for Object Based Image Analysis” In Proceedings of International Conference on Object-Based Image Analysis, 2006.

[21] Trena, “Body Proportions in art”, available online at, http://www.worsleyschool.net/socialarts/body/proportions.html, viewed on 23 April 2009.

Imran Achmed is currently a Telkom Centre of Excellence M.Sc. student at the University of the Western Cape. He is currently doing research on sign language synthesis and novel communication applications for the Deaf and hearing impaired.

James Connan heads the South African Sign Language (SASL) research group. He has a wide range of interests that include: databases, computer vision and machine learning.