# Static gesture recognition using features extracted from skeletal data

Ra'eesah Mangera
Mobile Intelligent Autonomous Systems
Council of Scientific and Industrial Research
Pretoria
Email: rmangera@csir.co.za

***Abstract*—Gesture recognition has become a popular area of research with applications in medical systems, assistive technologies, entertainment, crisis management, disaster relief and human-machine interaction. This paper presents a static gesture recognition system which uses an Asus Xtion Pro Live sensor to obtain the skeletal model of the user. Typically, joint angles and joint positions have been used as features. However these features do not adequately divide the gesture space, resulting in non-optimal classification accuracy. Therefore to improve the classification accuracy, a new feature vector, combining joint angles and the relative position of the arm joints with respect to the head, is proposed. A k-means classifier is used to cluster each gesture. New gestures are classified using a Euclidean distance metric. The new feature vector is evaluated on a 10 static gesture dataset, consisting of 7 participants. The vector containing only joint angles achieves a classification accuracy of 91.98%. In contrast, the new feature vector containing both joint angles and the relative positions of the arm joint with respect to the head achieves a classification accuracy of over 99%.**

***Keywords*—Gesture recognition, depth sensor, Asus Xtion Pro, skeleton model**

## I. Introduction

The study of gestures has been of great interest to researchers since the 18th century as they are thought to be a natural way in which human beings communicate expression and intent. More recently, much work has been focused on imbuing computers and robots with the ability to recognize and interpret gestures to enable more natural Human Computer Interaction (HCI). Typically, this would involve analysing captured images of a person performing a gesture using their bare hands. In static gesture recognition, each image represents a single gesture.

Each image is analysed and features such as skin colour, joint position, orientation and joint angles are extracted from the image using image processing. These are then used as an input to a classifier. In this paper features are extracted from the NiTE skeleton model obtained using depth data from an Asus Xtion Pro Live. Two feature vectors are extracted: one containing the angles of the arm joint and another containing both the joint angles and position of the arm joints relative to the head. A k-means classifier is used to cluster the data and the performance of the feature vectors is evaluated on a collected static dataset.

The rest of this paper is ordered as follows: Section II describes current gesture recognition systems. Section III details the calculations used to extract the joint angle features and relative position features from the skeletal data and describes the algorithms, training and test procedure used in classification of the samples. Section IV provides details on the static dataset collection used for testing. Section V presents the experimental results obtained on both the recorded dataset and a publically available dataset and Section VI concludes.

## II. Related Work

Much research been done in the field of gesture recognition, with two main approaches being employed – vision-based systems and glove-based systems. The glove-based system captures the motion of the joints using sensors worn by the user which may hamper natural movements. On the other hand, vision-based systems do not require the user to wear any additional equipment as they use cameras and other types of vision sensors to capture the gesture.

More recently depth sensors, such as the Kinect, have been used for the purposes of gesture recognition. Suarez and Murphy [1] present an in-depth review of gesture recognition using depth images.

Examples of gesture recognition systems which use depth sensors are those developed by Lai et al. [2], Jaemin [3] and Zafrulla et al. [4].

Lai et al. [2] use a Kinect camera in order to control a computer interface. Similarly to the approach presented in this paper, features are extracted from the skeleton model. However, the Kinect SDK skeleton model is used rather than the NiTE skeleton. Additionally, two types of feature vectors are used and compared - joint positions and a covariance matrix of the joint positions. The gestures are temporal rather than static. Both of the approaches presented in [2] achieve a classification accuracy of 97.25%. Jaemin [3] also uses depth data for the purposes of gesture recognition. Joint angles are extracted from the Kinect skeleton model and these are fed as inputs to a HMM (Hidden Markov Model) classifier. As in [2], the gestures are temporal rather than static and an accuracy of 81.8% is achieved. Zafrulla et al. [4] achieve a classification accuracy of 73.62% using Kinect data and an HMM classifier. They define a gesture set consisting of 19 gestures for the purposes of sign language recognition. The features extracted from the skeleton are the vectors between joints, joint angle and the distance between the hands resulting in a 20-dimensional feature vector.

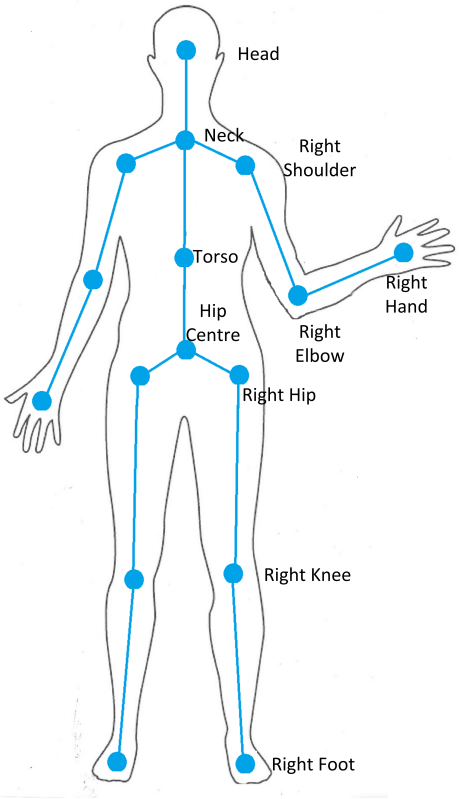


Fig. 1. NiTE Skeleton Model

This paper proposes the use of a new feature – the relative position of the arm joints with respect to the head.

## III. Experimental Design

### A. Feature Extraction

The Asus Xtion Pro Live depth sensor generates depth maps, RGB images and audio streams without requiring a user to wear any additional aids [5]. To interface with the device, the OpenNI [6] and NiTE SDKs [7] are used. These SDKs provide a user control API to the end user by utilising the depth, RGB and audio information received from the depth sensor. In particular, the NiTE library includes a skeleton tracker [7]. The skeleton model generated by NiTE is a tree graph whose nodes correspond to certain joints in the human body as seen in Figure 1. The skeleton tracker tracks the 3-D (x, y, z) coordinates of these 15 joints in real-time, at 30 frames per second (fps). The skeleton model is robust to differences in user size and shape, clothing colour and texture and background clutter making it ideal for feature generation.

As this work considers hand gestures, only the joints in the upper body are of interest. Specifically, the joints of interest are the: left shoulder (ls), right shoulder (rs), left elbow (le), right elbow (re), left hand (lh), right hand (rh) and head (he) joints.

*1) Relative Joint Distances:* For each pose, the 3-D distance between each of the 6 arm joints and the head joint was calculated forming an 18-dimensional feature vector based on joint distances:

$$F_{JD} = [x_{ls} - x_{he}, y_{ls} - y_{he}, z_{ls} - z_{he}, ..., z_{rh} - z_{he}]. \quad (1)$$

In order to compensate for the variation in user height, each distance was divided by the vertical distance between the neck and torso joint as in [2].

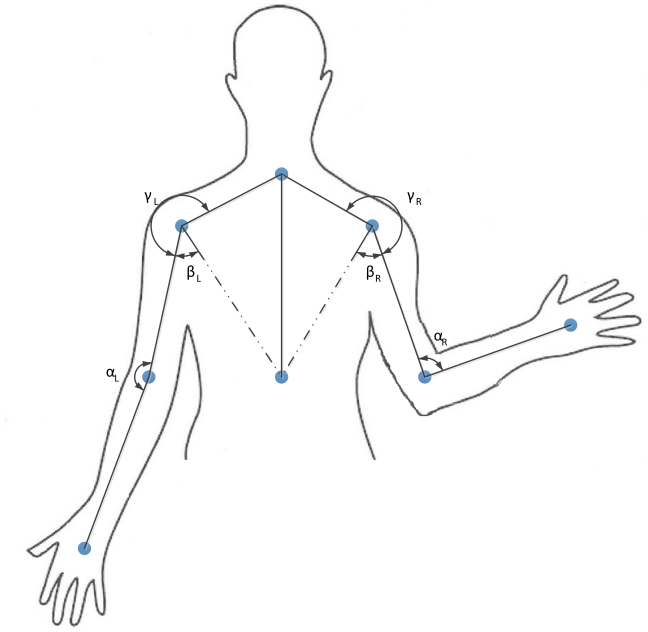


Fig. 2. Joint angles calculated from the skeleton tracker data

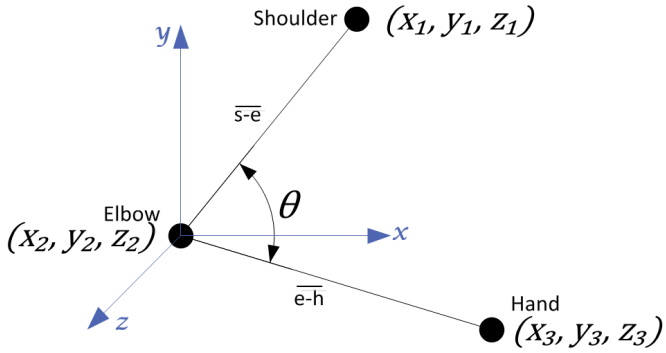


Fig. 3. Calculation of elbow angle

*2) Joint Angles:* As discussed, the distance between joints is affected by the height of the user. Therefore, relative distance is not a scale invariant feature. Joint angles on the other hand, are both scale and rotation invariant, as they are not dependent on the height of the subject, or the distance from the camera, or the orientation of the user relative to the camera plane. Six joint angles were calculated for each pose. These are shown in Figure 2. Figure 3 illustrates the elbow angle. To calculate the joint angle, the vector between joints must be computed. The shoulder-elbow vector ($\overline{s-e}$) and elbow-hand vector are given by Equations 2 and 3 respectively.

$$\overline{s-e} = (x_2 - x_1)\hat{i} + (y_2 - y_1)\hat{j} + (z_2 - z_1)\hat{k} \quad (2)$$

$$\overline{e-h} = (x_2 - x_3)\hat{i} + (y_2 - y_3)\hat{j} + (z_2 - z_3)\hat{k} \quad (3)$$

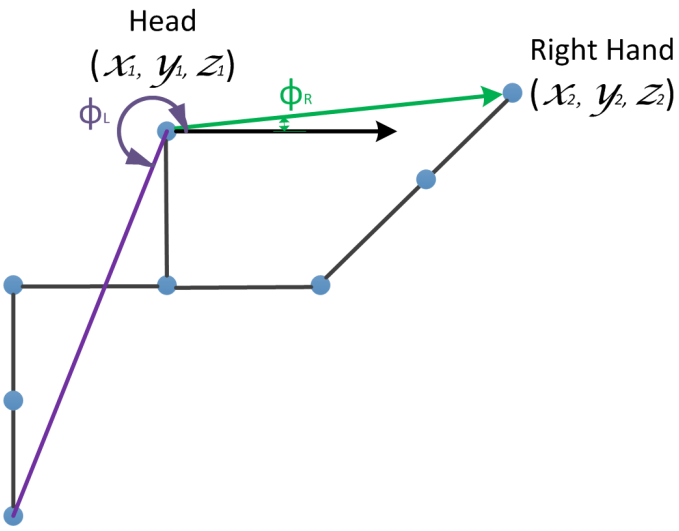


Fig. 4. Depiction of the relative position of the right and left hands with respect to the head.

The elbow angle is then given by Equation 4.

$$\theta = \arccos\left(\frac{\overline{s-e}\cdot\overline{e-h}}{|\overline{s-e}||\overline{e-h}|}\right) \tag{4}$$

where the numerator, $\overline{s-e}\cdot\overline{e-h}$ is the scalar product of the vectors $\overline{s-e}$ and $\overline{e-h}$. The denominator is the product of the magnitudes of the vectors $\overline{s-e}$ and $\overline{e-h}$.

The joint angle feature vector is a six-dimensional feature vector defined as follows:

$$F_{JA} = [\gamma_L, \gamma_R, \beta_L, \beta_R, \alpha_L, \alpha_R]. \tag{5}$$

where the symbols are as defined in Figure 2.

*3) Relative Joint Positions:* As joint angles are rotation invariant, a pose with the arms stretched on either side of the torso and arms stretched in front of the torso will have similar feature vectors. Therefore, the relative joint position between the elbow and hand joints and the head joint is calculated for each pose. Figure 4 shows the position of the hand relative to the x-component of the head joint. Similarly to Equation 2, the head-hand vector is given by:

$$\overline{he-h} = (x_2 - x_1)\hat{i} + (y_2 - y_1)\hat{j} + (z_2 - z_1)\hat{k}. \tag{6}$$

The x-component of the head joint is:

$$he_x = x_1\hat{i} + 0\hat{j} + 0\hat{k}. \tag{7}$$

Thus the position of the hand relative to the head is:

$$\varphi = \arccos\left(\frac{\overline{he-h}\cdot\overline{he_x}}{|\overline{he-h}||\overline{he_x}|}\right). \tag{8}$$

If the joint is below the head, the angle is subtracted from 360° to ensure that relative joint positions have unique angular representations.

TABLE I. DESCRIPTION OF FEATURE VECTOR ELEMENTS

| | | | |
|---|---|---|---|
| $\gamma$ | Elbow-Shoulder-Neck Angle | $\varphi$ | Relative position of the elbow relative to the head |
| $\beta$ | Torso-Shoulder-Neck Angle | $\sigma$ | Relative position of the hand relative to the head |
| $\alpha$ | Hand-Elbow-Shoulder Angle | $lh - rh$ | Distance between the left and right hands |

*4) Combined Feature Vector:* The joint angles, relative joint positions and the distance between the left and right hands are combined to form an eleven-dimensional feature vector:

$$F_C = [\gamma_L, \gamma_R, \beta_L, \beta_R, \alpha_L, \alpha_R, \varphi_L, \varphi_R, \sigma_L, \sigma_R, lh - rh] \tag{9}$$

Table I provides a brief description of each element in the feature vector.

### B. Training and Testing

After the features have been collected, a k-means classifier is trained for each of the ten poses (shown in Figure 5) to obtain cluster centres in the form $\{C_1^1...C_1^K, C_2^1...C_2^K, ..., C_{10}^1...C_{10}^K\}$ where K is the number of clusters per gesture. The k-means classifier clusters the data into distinct region such that the distance between points within a cluster is small compared to the distances between points in different regions [8]. The value of K used for these experiments is 14. This value was determined as empirically yielding the best inter-class divion.

For each new query, $F^Q$, the Euclidean distance to each cluster centre is computed by Equation 10.

$$\varepsilon(F^Q, C_{gn}^k) := \sqrt{\sum_{n=1}^{N}(f_n^Q - C_{gn}^k)^2} \tag{10}$$

where N is the number of features. Then, the classification of a new query is given by Equation 11.

$$label(F^Q) = label(F^m),$$
$$m = \underset{k=1..K, g=1..10}{\arg\min} \ \varepsilon(F^Q, C_{gn}^k) \tag{11}$$

The performance of the feature vectors is compared. To evaluate the performance of the system, leave-one-out cross validation (LOOCV) is performed. In LOOCV, the data is partitioned into a $n$ subsets of equal size. The classifier is then trained $n$ times using each of the $n$ subsets in turn as the test set and the remaining data as the training set [9]. In this case, all instances of a particular individual performing all gestures were set aside for testing. The remaining data was used to train the k-means classifier. Then, the gesture label for each gesture in the test set was determined individually. This was repeated for each user. To find the percentage of correctly classified gestures or the classification accuracy rate, the total number of correct classifications is divided by the total number of tests.

## IV. STATIC DATASET COLLECTION

### A. Gesture Set

In order to validate and compare the performance of the proposed feature vectors, a gesture set consisting of 10 different static gesture classes was created. Figure 5 shows the


Fig. 5. Sample frames from the static gesture set

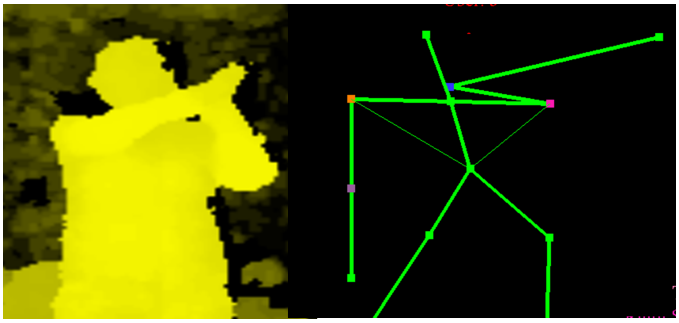

Fig. 6. The depth image in the left depicts the actual pose of the user "Sleep" whilst the skeleton model being tracked is shown on the right. It is evident that the skeleton tracking for the pose is inaccurate.

sample frames of a single person performing the 10 gestures. The gesture classes were given the following labels: "star", "cross", "flow", "my", "sleep", "victory", "hands-up", "left arm extended", "right arm extended", and "both arms extended".

### B. Data Collection

The dataset was collected using an Asus Xtion Pro Live and the NiTE skeleton tracking SDK. In each frame, the skeleton joint positions of the tracked user are received from NiTE. These are used to calculate the joint angles. The joint positions and joint angles are stored in a text file.

The dataset was collected in a laboratory environment. As discussed, the skeleton model is robust to variations in lighting and background therefore these conditions were not controlled. Seven participants were recorded performing each of the ten gestures three times, holding the pose for five seconds. Hence there are 7 participants$\times$10 gestures$\times$3 repetitions$\times$5 seconds $\times$30 frames per second = 31 500 samples.

*1) Data Pruning:* It was observed that the skeleton tracker did not always accurately track the users pose. This was noted particularly for gestures which crossed the chest such as the sleep, cross and my gestures. Examples of the 'lost' tracking can be seen in Figure 6. Therefore, samples with tracking errors were removed from the dataset. The content of the resulting dataset is shown in Table II.

## V. Experimental Results

Tables III and IV show the confusion matrices for the joint angle and combination feature vector methods. A confusion matrix details the actual and predicted classifications performed by a classification sytems. The rows of the confusion matrices indicate the true class label for each gesture and the columns indicate the predicted class label [10]. Cells highlighted in green indicate the percentage of the gesture which are correctly classified or the true positive rate. The cells highlighted in orange depict the proportion of gesture samples which are misclassified. Rows highlighted in red indicate gestures where the correct classsification rate is less than 80%. The joint angle method achieves a classification accuracy of 91.98% and the combined feature vector method a classification accuracy of 99.35%. It is evident that the feature vector which includes the relative position of the joints achieves much higher classification accuracy.

Increasing the number of features often introduces issues related to computational complexity and the amount of training data required[11]. This is illustrated in Table V which shows the training and classification time for the two methods.

TABLE II. Dataset contents after samples with tracking errors were removed

| Gesture | Number of Samples | Gesture | Number of Samples |
|---|---|---|---|
| Bextend | 3150 | Rextend | 3150 |
| Cross | 2979 | Sleep | 2602 |
| Hands Up | 3150 | Star | 3000 |
| Lextend | 3150 | Flow | 2150 |
| My | 3081 | Victory | 3150 |
| | | Total | 29 562 |

TABLE III. Confusion matrix when the joint angle feature vector is used (average accuracy = 91.98%).

| | Bextend | Cross | Hands Up | Lextend | My | Rextend | Sleep | Star | Flow | Victory |
|---|---|---|---|---|---|---|---|---|---|---|
| Bextend | 1.00 | | | | | | | | | |
| Cross | | 0.77 | | | | | 0.25 | | 0.01 | |
| Hands Up | | | 0.96 | | | | | 0.04 | | |
| Lextend | | | | 0.98 | | | 0.02 | | | |
| My | | | | | 1.00 | | | | | |
| Rextend | | | | | | 1.00 | | | | |
| Sleep | | 0.18 | | | | | 0.75 | | 0.07 | |
| Star | | | | | | | | 0.87 | | 0.13 |
| Flow | | 0.02 | | | | | | | 0.98 | |
| Victory | | | | | | 0.06 | | 0.08 | | 0.85 |

TABLE IV. Confusion matrix when the joint angle and relative joint position feature vector are used (average accuracy = 99.35%).

| | Bextend | Cross | Hands Up | Lextend | My | Rextend | Sleep | Star | Flow | Victory |
|---|---|---|---|---|---|---|---|---|---|---|
| Bextend | 1.00 | | | | | | | | | |
| Cross | | 0.98 | | | | | | | 0.02 | |
| Hands Up | | | 1.00 | | | | | | | |
| Lextend | | | | 1.00 | | | | | | |
| My | | | | | 1.00 | | | | | |
| Rextend | | | | | | 1.00 | | | | |
| Sleep | | 0.01 | | | 0.01 | | 0.98 | | | |
| Star | | | | | | | | 0.99 | | 0.01 |
| Flow | | | | | 0.02 | | | | 0.98 | |
| Victory | | | | | | | | | | 1.00 |

TABLE V. COMPARISON OF THE TRAINING AND CLASSIFICATION TIME FOR THE JOINT ANGLE FEATURE VECTOR AND THE COMBINATION FEATURE VECTOR

| | Joint Angle Feature Vector | Joint Angle and Relative Position Feature Vector |
|---|---|---|
| **Training Time** | 2.496 s | 4.256 s |
| **Classification Time** | 74 $\mu$s | 29 $\mu$s |

TABLE VI. CONFUSION MATRIX FOR THE CORNELL MILITARY GESTURE DATASET (AVERAGE ACCURACY = 99.24%).

| | Abrest | Antigesture | Backup | Enemy | Freeze | Gas | Hide | Injury | Land | Listen | Pistol | Rifle | Stop | Unknown | Watch |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Abreast | 1.00 | | | | | | | | | | | | | | |
| Antigesture | | 1.00 | | | | | | | | | | | | | |
| Backup | | | 0.99 | | | | | | 0.01 | | | | | | |
| Enemy | | | | 1.00 | | | | | | | | | | | |
| Freeze | | | | | 1.00 | | | | | | | | | | |
| Gas | | | | | | 0.92 | | 0.04 | | 0.04 | | | | | |
| Hide | | | | | | | 1.00 | | | | | | | | |
| Injury | | | | | | | | 1.00 | | | | | | | |
| Land | | | 0.01 | | | | | | 0.99 | | | | | | |
| Listen | | | | | | | | | | 1.00 | | | | | |
| Pistol | | | | | | | | | | | 1.00 | | | | |
| Rifle | | | | | | | | | | | | 1.00 | | | |
| Stop | | | | | | | | | | | | | 1.00 | | |
| Unknown | | | | | | | | | | | | | | 1.00 | |
| Watch | | | | | | | | | | | | | | | 1.00 |

It can be seen that the joint angle feature vector is almost two times faster to train than the combination feature vector with both tests been performed on an Intel Core i7 CPU @ 3.20 GHz using software developed in C++. However, the classification time for both methods is significantly less than 45 ms. Thus, the response time would be perceived as instantaneous by humans according to a study done by Sheriden and Ferrell [12].

In order to test the robustness of the proposed feature vector, the performance of the feature vector was also evaluated on a publically available dataset.

The Cornell Military Gesture Dataset [13], consists of 15 static gestures of 3 people performing each gesture ten times. Each recording is taken at 50 fps and lasts five seconds, hence there are 112 500 samples. LOOCV was used. The confusion matrix is shown in Table VI. A classification accuracy of 99.24% is achieved. This is a 0.1% drop in accuracy compared to the accuracy achieved using the collected dataset and may be attributed to the fact that the Cornell dataset contains more gestures. In addition, the proposed feature vector attains a better classification accuracy than that reported in [13]. A comparison of the classification accuracies is shown in Table VII.

TABLE VII. COMPARISON OF THE RESULTS OBTAINED USING THE PROPOSED FEATURE VECTOR AND THOSE PRESENTED IN [13]

| Gesture | Cornell Accuracy | Proposed Feature Vector |
|---|---|---|
| **Abreast** | 1.00 | 1.00 |
| **Antigesture** | 0.99 | 1.00 |
| **Backup** | 0.95 | 0.99 |
| **Enemy** | 1.00 | 1.00 |
| **Freeze** | 1.00 | 1.00 |
| **Gas** | 0.90 | 0.92 |
| **Hide** | 1.00 | 1.00 |
| **Injury** | 1.00 | 1.00 |
| **Land** | 0.93 | 0.99 |
| **Listen** | 0.74 | 1.00 |
| **Pistol** | 1.00 | 1.00 |
| **Rifle** | 1.00 | 1.00 |
| **Stop** | 1.00 | 1.00 |
| **Unknown** | 1.00 | 1.00 |
| **Watch** | 0.85 | 1.00 |

## VI. CONCLUSION

A real-time static gesture recognition system using an Asus depth sensor has been developed. Using a feature vector consisting of both joint angles and relative joint positions achieves a much higher classification rate compared to using a feature vector consisting of just joint angles. In addition the robustness of the feature vector has been proven using a publically available dataset.

## ACKNOWLEDGMENT

The author would like to thank Fred Senekal and Dr. Fred Nicolls for their valuable comments, ideas and assistance in the undertaking of the research summarised here. This work is supported by a CSIR studentship.

## REFERENCES

[1] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in *RO-MAN, 2012 IEEE*, 2012, pp. 411–417.

[2] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using Kinect," *2012 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 185–188, Apr. 2012.

[3] L. Jaemin, H. Takimoto, H. Yamauchi, A. Kanazawa, and Y. Mitsukura, "A robust gesture recognition based on depth data," in *Frontiers of Computer Vision, (FCV), 2013 19th Korea-Japan Joint Workshop on*, 2013, pp. 127–132.

[4] Z. Zafrulla, H. Brashear, H. Hamilton, T. Starner, and P. Presti, "American sign language recognition with the kinect," in *Proceedings of the 13th international conference on multimodal interfaces*, ser. ICMI '11, no. September, Sch. of Interactive Computing, Georgia Inst. of Technology, Atlanta. New York, NY, USA: ACM, 2011, pp. 279–286.

[5] Asus, "Asus Xtion PRO Live," 2012. [Online]. Available: http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

[6] OpenNI, "OpenNI Programmer's Guide," 2013. [Online]. Available: http://www.openni.org/openni-programmers-guide

[7] PrimeSense, "NiTE 2 API Programmer Tutorial Guide," 2013. [Online]. Available: http://www.primesense.com/wp-content/uploads/2013/04/PrimeSense_NiTE2API_ProgTutorialGuide_C++Samples_docver0.2.pdf

[8] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[9] T. M. Mitchell, "Machine learning. 1997," *Burr Ridge, IL: McGraw Hill*, vol. 45, 1997.

[10] R. Kohavi and F. Provost, "Glossary of Terms Special Issue on Applications of Machine Learning and the Knowledge Discovery Process," *Machine Learning*, vol. 30, no. 2/3, pp. 271—-274, 1998. [Online]. Available: http://robotics.stanford.edu/~ronnyk/glossary.html

[11] S. Theodoridis and K. KOutroumbas, *Pattern Recognition*, 4th ed. Elsevier, 2009.

[12] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, p. 60, Feb. 2011. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1897816.1897838

[13] G. Bernstein, N. Lotocky, and D. Gallagher, "Robot Recognition of Military Gestures CS 4758 Term Project," Robot Learning Lab, Cornell University, Tech. Rep., 2012. [Online]. Available: http://pr.cs.cornell.edu/humanactivities/handgesture/index.php