South African Sign Language Recognition using Microsoft Kinect

A Literature Review*

Shaheel Kooverjee
University of Cape Town
skooverjee@gmail.com

ABSTRACT

Various motion sensing devices have been used in research to investigate gesture recognition for sign language. The HANDGR project looks to work towards the creation of a gesture recognition system for South African Sign Language, specifically with the use of the Microsoft Kinect. This review looks at various classification methods, mainly machine learning techniques such as support vector machines, artificial neural networks and hidden Markov models. Methods for feature extraction from the Kinect are also discussed and considered. Based on the material reviewed, it is evident that an effective and useful system for the HANDGR project can be created, with enough experimentation.

KEYWORDS

Classification, feature extraction, gesture recognition, Kinect, machine learning, sign language

1 INTRODUCTION

There is an obvious barrier in communication between deaf and hearing people all over the world. Over the years, technology has been used to help assist the deaf community in this regard. However, there is always more that can be done to make the deaf feel less excluded.

One such area of research has been the use of cameras and motion sensors to read and correctly interpret sign language gestures [9]. This has been done with various techniques, primarily based on machine learning, in order to train the system to recognize and classify gestures correctly. Three of the more frequently used devices on which these algorithms have been implemented include the Myo armband, the Leap Motion controller and the Microsoft Kinect motion sensor.

Sign languages can make use of the different parts and positions of the body, such as the face, arms, hands and body posture. The South African Sign Language (SASL) alphabet, however, is recognised and gestured by only the use of the hand in various positions.

The HANDGR project will be based on evaluating selected algorithms (which have been reported to work efficiently and accurately on research of sign language gesture recognition) for each of the three motion-sensing devices. This will be done in term of the SASL alphabet (or a subset thereof), in order to possibly develop a system to assist in breaking down the communication barrier between the deaf and the hearing. Later phases of the project will look at possible combinations of inputs from these devices to refine results.

This particular review focuses on how the Microsoft Kinect has been used by various researchers to implement sign language recognition tools through various algorithms.

2 MICROSOFT KINECT

The Kinect is a motion sensing device that is mainly used in conjunction with Microsoft Xbox consoles. However, it is also compatible with Microsoft Windows computers, and this, along with the available online software development kit (SDK), has made research possible in other non-gaming contexts, that require sensors for gestures and/or spoken commands.

What makes the Kinect particularly special is the feature of a depth sensor, which is particularly useful when it comes to gesture recognition and the distinguishing of the human body from a busy background. The Kinect effectively provides a cheap (in comparison to other commercially available depth sensing cameras) and easily available depth sensor [12, 16, 19] that projects an infrared light pattern to provide a reliable depth map as output [19]. Additional advantages include the Kinect not requiring background image calibration, or special markers or gloves for tracking [6], and the fact that the depth sensors are not affected by environmental conditions, such as low lighting [1, 6]. These features make it even more suitable for this project.

However, due to the low resolution of the depth sensing camera (only 640x480), it can be a challenge to find and separate specific objects in the image [12]. In our case, the hands of the person using the device are most important, and a hand can occupy a very small portion of the already low-quality image extracted from the Kinect's depth sensor.

In all of the methods discussed in this review, the Kinect is placed some distance away from and directly in front of the user, such that the camera faces the user's body.

3 GESTURE RECOGNITION METHODS

Various methods of both feature extraction and recognition/classification have been experimented with and tested, in the Kinect context. Commonly used machine learning techniques for this kind of classification include neural networks, hidden Markov models [6, 9], and more recently, support vector machines [1, 16]. Other methods that do not necessarily use machine learning are also considered. These are described below according to the technique used, along with the respective feature extraction methods used. The accuracy rates, datasets used, features extracted and flaws with experimentation are the main aspects discussed. Table 1 provides a summary of the experiments done in the reviewed Kinect-based papers.

*In partial fulfilment of the CSC4000W Project

Table 1: Comparison of highlighted previous research on sign language recognition using Kinect

Reference	Classification technique	Feature extraction	Dataset	Accuracy
Sun et al. [16], American Sign Language	Latent Support Vector Machine	’Ordinary’ features: HOG and motion/appearance; and ’Kinect’ features: body pose, hand shape and hand motion	Part A: 73 unique ASL signs; 1971 phrases by 9 participants	86.0%
			Part B: 63 unique ASL sentences; 1890 sentences by 10 participants	82.9%
Agarwal et al. [1], Chinese Sign Language	Multi-class Support Vector Machine	Depth histograms and motion characteristics between frames	2 datasets of 47 video sequences each, of the Chinese Numbers	81.48% (Linear Kernel); 87.67% (RBF Kernel)
Huang et al. [6], American Sign Language	Support Vector Machine	Hand, wrist, arm, shoulder positions and velocities; Distance separation between left and right body parts	100 pre-segmented examples of 10 different signs, gestured by 2 different participants (50 gestures each)	97%
Pizzolato et al. [11], Brazilian Sign Language	Multi-layer Perceptron (ANN)	Hand image after finding depth threshold for segmentation	5000 samples of 10 different letters, each having 250 images with hand cropping and 250 without	100%
Zafrulla et al. [19], American Sign Language	4-state Hidden Markov Models	Body pose and hand features	555 seated samples, including 207 corrupt samples	95.16% before factoring in corrupt samples; 58.86% after factoring in corrupt samples
			155 standing samples, including 9 corrupt samples	94.49% before factoring in corrupt samples; 88.02% after factoring in corrupt samples
Sarhan et al. [15], Arabic Sign Language	3-state Hidden Markov Models	Location, orientation, axes, shape roundness, convexity/concavity, rectangularity and trajectory of hand	215 samples of 16 difernt words, signed by 4 different participants	73.06%
	4-state Hidden Markov Models			78%
	5-state Hidden Markov Models			80.47%
Jangyodsuk et al. [7], American Sign Language	Dynamic Time Warping	Hand trajectory and hand shape mapped as HOG features	2226 gestures signed by 2 participants	82.09% at top 10 rank; 92.54% at top 30 rank
Santos et al. [14], American Sign Language	Dynamic Time Warping, combined with Hidden Markov Models	Hand contour shapes	12 dynamic hand gestures	97.49%

3.1 Support Vector Machine

A support vector machine (SVM) is a supervised machine learning technique that is mostly used for classification problems. Each data item gets plotted as a point in an n-dimensional space, and a hyperplane (a subspace of the space in n-1 dimensions) is found to differentiate the data. The clear division in the training instances allows new data to then be classified. Basic SVMs are usually binary classifiers, but they can be extended to support multi-class classification.

A relatively recent experiment [16] uses a combination of 'Ordinary' features, along with special 'Kinect' features. Together, these features help find the position and shape of the hands, along with the body pose. More specifically, the histogram of oriented gradients (HOG) of the 'Ordinary' features was extracted using an already successful algorithm, as shown in [5]. Very briefly, this method involves dividing the image into small pixel regions. After the gradient orientation (with respect to surrounding pixels) of each pixel in the region has been approximated according to one of nine orientation bins, 1D histograms of gradients in that image portion are accumulated, in order to record local shape properties. With regard to generating information about the hand, a 48x48 pixel region around the hand point is cropped, after which HOG features are extracted on each patch of the region. For generating hand motion, HOG features are extracted on patches of each frame. The remaining feature extraction focuses on aspects such as body pose and is not relevant in our context.

For the classification of the experiment in [16], a multi-class latent SVM was used. This method is helpful in that it classifies videos of sign language gestures describing words/sentences. Hence, this method would be geared towards the few letters of the SASL that involve dynamic gestures. The SVM has desired state values (discriminative frames in the gesture videos), which are treated as the latent variables (inferred, through a mathematical model, from variables that are directly measured) in the model. Once model learning has taken place, the most discriminative and representative frames can be found and classified correctly.

This method was implemented with two different datasets (one with words and one with sentences) and a total of 1971 phrase videos and 1890 sentence videos were collected. The test results were favorable, and for words, the model using both 'Ordinary' and 'Kinect' features together reported the highest accuracy, at 86% (in comparison to the model using just 'Ordinary' features, which was 82.3% accurate). Two other baseline algorithms (SVMs combined with either hard-assignment coding or soft-assignment coding) were used for comparison, and it was found that with or without the 'Kinect' features, the SVM outperformed the other algorithms. The results for the sentence gestures were similar (with slightly lower percentages due to transitions between words causing some decrease in the accuracy).

A very similar study was undertaken [1], involving both the HOG feature extraction, and an SVM classifier model. Very good results were also obtained (with around 80% accuracy rates), however, this experiment used simpler gestures - only those for the numbers 0-9 - and the dataset was much smaller than in the 2015 [16] study. It was, perhaps unwisely, assumed that the same algorithms could be extended to larger datasets to produce similar results. It was also 'safely concluded' (more likely to be an unsafe conclusion, considering that the paper does not contain any direct comparisons to, or even mentions of, other experiments) that the recognition system created is faster than other techniques in hand tracking or hand shape analysis.

Another research paper [6] also highlighted the training of an SVM for a single-sign classifier, which, too, returned high accuracy results. This study, however, biases the dataset in choosing only gestures that involve arm movements, and not only individual finger movements, so as to make feature extraction easier. This defeats the purpose of working towards a system that would be able to cater for an entire language, even though the accuracy results are very impressive. Hence, this paper shows what to be aware of in this regard of the project.

3.2 Artificial Neural Networks

An approach for classification using artificial neural networks (ANNs) is explored in [11], focusing on static gesture recognition. What is interesting about this study is that it obtained 100% accuracy in testing.

The process for segmentation entails using middleware, NITE, by Primesense, and this enables the system to segment the image to find all user pixels, as well as the user's center of mass (CoM). The tested Brazilian sign language, Libras, involves hands held in front of the torso, while possibly making contact with the face or torso. To segment the user's hands, a depth threshold is calculated, to work like an invisible wall in front of the user. This is called the 'Virtual Wall' and is equal to the difference between the depth of the CoM, and α (some offset to avoid the wall being transposed by the face or torso). Blobs that appear in front of this virtual wall can then be extracted, and noise removed. To obtain the relevant blobs, a linear time-component analysis must be applied [4]. A similar approach is also described in [2], whereby the hand shape contour is extracted.

After obtaining the relevant blobs, these are sorted in terms of area, and classified as hands: being right, left, or dominant (if only one hand is present). The region of interest (ROI) is then defined as the minimum rectangle enclosing the blobs. A heuristic method defined in the paper as the Aspect Ratio Hand Cropping Algorithm (ARHCA) is then applied to check and adjust ROIs, which involves the aspect ratio of the blob. The ARHCA method is meant to improve accuracy by reducing, or ideally eliminating, the visible arm from the blob.

The static gesture classifier involves a multi-layer perceptron (MLP, an ANN that is an evolution of the standard perceptron), which helps learn to distinguish between non-linearly separable data. Input, hidden and output layer neurons structure this MLP, and this model tries to map inputs to a subset of possible outputs. Weights of the neurons store the knowledge of the network, and these weights are determined through training of the system, with the use of a backpropogation algorithm. The basic structure of the classifier used had 625 different inputs (which were all cropped beforehand to equivalent resolution), 100 hidden neurons, and 5 differing outputs. Two groups of five different letters were tested (in total, the dataset contained 5000 samples), both with and without ARHCA. The tests run without ARHCA on the inputs produced accuracy results of between 60% and 75%, whereas the improved inputs after using ARHCA produced perfect (100%) accuracy. Something worth noting in this result is that the hand cropping algorithm only accounts for the hand pointing upwards, and so it would need to be extended to work for gestures that require the hand to be in another position. Aside from this, and that there is no mention of how many users recorded the data in the dataset, the experiment seems to be relatively successful.

3.3 Hidden Markov Models

An investigation [19] done on recognition and verification of American Sign Language phrases was conducted using the Kinect, with users in both standing and sitting positions while performing the gestures. For feature extraction of this process, both body pose and hand features are extracted. Body pose features are extracted using the OpenNI framework (which interfaces with NITE middleware). More specifically, a hand feature is extracted by collecting all 3D points in the neighbourhood of each hand and its position. For each hand, these points are then clustered to obtain a mixture of Gaussians. Principal Component Analysis is then performed to reduce the dimensionality to that of the body pose feature. This dimensionality reduction aspect may only be necessary for sign language that involves body poses, and so this aspect would not apply to SASL.

A hidden Markov model (HMM) is a stochastic model of a system containing hidden or unobservable states, and in which future states depend only on the current state. For training and testing of the investigated system in [19], 4-state HMMs were trained for each of the 19 signs used in the investigation. This was done using the Georgia Tech Gesture Toolkit [18]. Recognition and verification of the trained models were then recorded using leave-one-out-cross-validation (LOOCV), as described in [19].

The recognition results, before accounting for tracking errors of the data, are 95.16% and 94.49% for seated and standing data respectively. There is a significant amount of accuracy with little difference between the two scenarios, however, this changes when errors during data collection are factored in: accuracy for the seated data drops to 58.86%, while the standing accuracy is less affected, at 88.02%. This indicates that when users are seated, many more tracking errors are made than when users are standing, and obviously results in a loss of accuracy and makes the standing position more favorable. The tracking errors are a result of the skeletal tracking provided by OpenNI. Should users be in a seated position for the SASL gesture recognition system, this framework may not be the most ideal one to use for feature extraction, especially considering that the body pose would not be required for recognition in SASL.

A more recent paper [15] explores the use of HMMs to recognise Arabic Sign Language. Although this language uses the face, eyes and body to enforce the signs and their meanings, the study still focuses on the main component of the signing: the hands.

To extract features of the hands, the signer is first segmented from the image, using the Kinect's segmentation mask. The hands are then segmented from this according to depth and skeletal information, after which, various scale, rotation and translation features of the hand are measured. These include the location, orientation, major and minor axes, shape roundness, convexity or concavity, rectangularity, and gesture trajectory of the hand.

The proposed system was tested using 3-, 4- and 5-state HMMs. An increase of the number of states shows an increase in accuracy: the 5-state HMM achieved an accuracy rate of 80.47%, the highest of the three, while the 3-state HMM had the lowest recognition, at 73.06%. The tested dataset contains 215 instances of 16 different word gestures by 4 different individuals, who each performed each gesture at least 3 times. Varying the user's distance and position from the camera, background and the lighting condition through each gesture was done to strengthen the recognition of the system.

3.4 Other methods

An algorithm for classification that is not a machine learning technique, but is still popularly used for gesture recognition, is Dynamic Time Warping (DTW). This algorithm is used to find an optimal alignment between two given time series (time-dependent) sequences under certain restrictions.

In a study regarding American Sign Language recognition [7], Histogram of Oriented Gradient features and DTW are used. Due to the fact that DTW is a distance measure, no training is required, which helps in systems where the number of training examples is not big.

The method of sign recognition involves two kind of features: hand trajectory (which incorporates hand positions relative to the face, and their velocity vectors) and hand shape (in this case, HOG features, similar to the case in Section 3.1). The HOG features are matched with signs using the Euclidean metric as the distance metric in the DTW algorithm (used as in [17], which deals with a vocabulary sign search, but not with the use of a depth sensor like Kinect).

The testing done with the Kinect camera was done with a dataset of 2226 signs by 2 signers. An accuracy of around 82% was recorded at top 10 rank (10 signs need to be looked up before finding correct matches at 82% accuracy) for the Kinect data, and this improved to 92% for the top 30 rank.

An interesting approach, named HAGR-D, that combines the DTW with HMM using depth maps is explored in [14]. The hybrid approach is motivated by the fact that DTW is not very sensitive to patterns that are very close, and thus could result in some mistakes. DTW first classifies the gesture, using an algorithm for feature extraction involving hand contour shapes. The closest gestures to the input sequence are then returned and the HMM decides on the classification of the input sequence between the candidate gestures returned by the DTW process.

The publicly available MSRGesture3D dataset, of 12 dynamic hand gestures in American Sign Language, was used and tested with LOOCV on the hybrid system. Overall, the classification process achieved an accuracy of 97.49%. This is a very high level of accuracy, although very few gestures were used in this dataset, which may have biased the result.

Another effective way to make gesture recognition more robust, in terms of feature extraction, is to track the hand through data gloves, but these are then required to be worn by the users, and could hinder gesture movements. Additionally, they need to be calibrated before use, and are generally expensive, making them a less popular way of hand gesture recognition [13]. Thus, methods that involve data gloves together with the Kinect are not considered in this review.

4 COMBINING KINECT WITH LEAP

Combining the output of Kinect with that of another sensor could increase the accuracy of gesture recognition. The Kinect has been combined with the Leap Motion device [8] to show this.

In this process, the hand is extracted from depth and colour data from the Kinect, after which correlation and curvature features are extracted (the process is described in detail in [8]). From the Leap, positions of fingertips (3D positions), the palm center (in terms of 3D space), and hand orientation details (which are based on the unit vectors parallel and perpendicular to the hand) are acquired. Using this information, the fingertip angles (angles of fingertips with respect to hand orientation), fingertip distances (3D distances of fingertips from hand center) and fingertip elevations (distances of fingertips from the plane corresponding to the palm) are introduced. The five features from the Leap and Kinect are then used, together with the dataset, to train a multi-class SVM.

1400 different data samples (of 10 different gestures performed by 14 different people) formed the fairly-sized dataset. Various combinations of the features pulled from the different devices were then tested using this data. The accuracy of the three Leap features together is 80.86%, and the accuracy of the two Kinect features is 89.71%. Combining all five features gives an improved (slightly from the Kinect, but significantly from the Leap) result of 91.28% accuracy. Clearly, there are properties from each device which contribute to making recognition more accurate.

Another paper [10] also presents a way to fuse data from the Leap and Kinect for the tracking of hand motions. It incorporates a calibration method (using the Corresponding Point Set Registration algorithm for rigid transformation, as presented in [3]) to align the two different frames of reference (3D coordinates of the hand) outputted by the Leap and Kinect. The operating range gets extended through this, as fingertips locations are still provided when the hand is not completely visible to the Kinect. Although it does not particularly investigate the usage of this fusion for sign language purposes, it could serve as a potential feature extraction method.

5 CONCLUSIONS

Many different methods of data extraction from the Kinect, as well as methods of classification of the data extracted, exist. Each of these come with their own strengths and weaknesses, as is clear in this review.

Overall, one of the most ideal methods for the SASL gesture recognition project seems to be the one used in [11], both in terms of the classification and extraction of features. Even though this is the only study referenced that focuses on ANNs, it is also the only study with perfect accuracy due to its ARHCA algorithm in the data extraction process. The experiment involved static gestures, and so this method would be particularly useful for the majority of the SASL alphabet signs are static. Hence, this algorithm can be combined with, or altered according to, other effective methods presented that involve dynamic gestures (such as the feature extraction method in [1]).

The support vector machine seems to be another commonly used and effective method of classification for gestures. Along with HOG features, the experiment in [16] proved to be fairly successful with one of the much larger datasets and more than a few users.

Hidden Markov model classification is another commonly used method, and produced high accuracy, especially in [19]. The results were only let down by the OpenNI framework, which indicates that this is not ideal to use for extraction of features.

A combination of classification methods may also prove to be useful, as seen in [14], which combines a machine learning technique with a time series analysis algorithm to improve classification accuracy. Additionally, as will be explored in later phases of the project, the Kinect can also be combined with other motion sensors, such as the Leap, so as to extract and concurrently use information unique to each device, in order to improve gesture recognition.

There are most definitely many options and variables to consider when creating a system for SASL gesture recognition using the Kinect. However, the material discussed in this review clearly guides the way forward on HANDGR, which should prove to be an exciting, and potentially very useful, project.

REFERENCES

[1] Anant Agarwal and Manish K Thakur. 2013. Sign Language Recognition using Microsoft Kinect. In *Sixth International Conference on Contemporary Computing (IC3)*. IEEE, Noida, India, 181–185.

[2] Jos Barata, Joo Delgado, and Tiago Cardoso. 2015. Hand Gesture Recognition towards Enhancing Accessibility. *Procedia Computer Science* (2015), 419–429.

[3] Paul J. Besl and Neil D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence - Special issue on interpretation of 3-D scenes - part II* (1992), 239–256.

[4] Fu Chang, Chun-Jen Chen, and Chi-Jen Lu. 2004. A linear-time component-labeling algorithm using Contour Tracing Technique. *Computer Vision and Image Understanding* (2004), 206–220.

[5] Navneet Dalal and Bill Triggs. 2005. Histograms of Oriented Gradients for Human Detection. In *International Conference on Computer Vision & Pattern Recognition*. IEEE, San Diego, United States, 886–893.

[6] Frank Huang and Sandy Huang. 2011. Interpreting American Sign Language with Kinect. *Stanford University term paper for CS 299* (2011).

[7] Pat Jangyodsuk, Christopher Conly, and Vassilis Athitsos. 2014. Sign Language Recognition using Dynamic Time Warping and Hand Shape Distance Based on Histogram of Oriented Gradient Features. In *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, Island of Rhodes, Greece, Article No. 50.

[8] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. 2014. Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, Paris, France, 1565–1569.

[9] Becky Sue Parton. 2006. Sign Language Recognition and Translation: A Multidisciplined Approach From the Field of Artificial Intelligence. *Journal of Deaf Studies and Deaf Education* (2006), Vol. 11, No. 1, pp. 94–101.

[10] Benot Penelle and Olivier Debeir. 2014. Multi-sensor data fusion for hand tracking using Kinect and Leap Motion. In *Proceedings of the 2014 Virtual Reality International Conference*. ACM, Laval, France, Article No. 22.

[11] Ednaldo Brigante Pizzolato, Mauro dos Santos Anjo, and Sebastian Feuerstack. 2012. A Real-Time System to Recognize Static Gestures of Brazilian Sign Language (Libras) alphabet using Kinect. In *IHC 1́2 Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*. SBC, Cuiab, MT, Brazil, 259–268.

[12] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. 2011. Robust hand gesture recognition with kinect sensor. In *19th ACM international conference on Multimedia*. ACM, Scottsdale, Arizona, USA, 759–760.

[13] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. 2013. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. *IEEE Transactions on Multimedia* (2013), 1110 – 1120.

[14] Diego G. Santos, Bruno J. T. Fernandes, and Byron L. D. Bezerra. 2015. HAGR-D: A Novel Approach for Gesture Recognition with Depth Maps. *Sensors 15(11)* (2015), 28646–28664.

[15] Noha A. Sarhan, Yasser El-Sonbaty, and Sherine M. Youssef. 2015. HMM-based Arabic Sign Language Recognition using Kinect. In *Tenth International Conference on Digital Information Management (ICDIM)*. IEEE, Jeju, 169 – 174.

[16] Chao Sun, Tianzhu Zhang, and Changsheng Xu. 2015. Latent Support Vector Machine Modeling for Sign Language Recognition with Kinect. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2015), Special Section on Visual Understanding with RGB–D Sensors.

[17] Haijing Wang, Alexandra Stefan, Sajjad Moradi, Vassilis Athitsos, Carol Neidle Neidle, and Farhad Kamangar. 2012. A System for Large Vocabulary Sign Search. In *Proceedings of the 11th European conference on Trends and Topics in Computer Vision Part I, ECCVfi10*. Springer-Verlag, Berlin, Heidelberg, 342–353.

[18] Tracey Westeyn, Helene Brashear, Atrash Amin, and Thad Starner. 2003. Georgia tech gesture toolkit: supporting experiments in gesture recognition. *ICMI fi03: Proceedings of the 5th International Conference on Multimodal Interfaces* (2003), 85–92.

[19] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. 2011. American sign language recognition with the Kinect. In *13th international conference on multimodal interfaces*. ACM, Alicante, Spain, 279–286.