American Sign Language Recognition with the Kinect

Zahoor Zafrulla
Sch. of Interactive Computing
Georgia Inst. of Technology
Atlanta, GA 30332-0760
zahoor@gatech.edu

Helene Brashear
Tin Man Labs, LLC
Austin, Texas
helene@tinmanlabs.com

Thad Starner
Sch. of Interactive Computing
Georgia Inst. of Technology
Atlanta, GA 30332-0760
thad@cc.gatech.edu

Harley Hamilton
Sch. of Interactive Computing
Georgia Inst. of Technology
Atlanta, GA 30332-0760
hhamilton@cc.gatech.edu

Peter Presti
Interactive Media Tech. Center
Georgia Inst. of Technology
Atlanta, GA 30332-0760
peter.presti@imtc.gatech.edu

ABSTRACT

We investigate the potential of the Kinect depth-mapping camera for sign language recognition and verification for educational games for deaf children. We compare a prototype Kinect-based system to our current CopyCat system which uses colored gloves and embedded accelerometers to track children's hand movements. If successful, a Kinect-based approach could improve interactivity, user comfort, system robustness, system sustainability, cost, and ease of deployment. We collected a total of 1000 American Sign Language (ASL) phrases across both systems. On adult data, the Kinect system resulted in 51.5% and 76.12% sentence verification rates when the users were seated and standing respectively. These rates are comparable to the 74.82% verification rate when using the current (seated) CopyCat system. While the Kinect computer vision system requires more tuning for seated use, the results suggest that the Kinect may be a viable option for sign verification.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation (e.g., HCI)**]: User Interfaces (D.2.2, H.1.2, I.3.6); K.3.1 [**Computers and Educations**]: Computer Uses in Education

General Terms

Experimentation, Performance

Keywords

Educational Games, Sign language Verification, Microsoft Kinect

1. INTRODUCTION

The CopyCat project was designed to develop an interactive educational adventure game to help deaf children acquire language skills. The main goals of the project are to improve the language and memory abilities of deaf signing children, advance basic research in computer-based sign language recognition, and design an efficient language interaction model in order to assist in the language learning of deaf children.

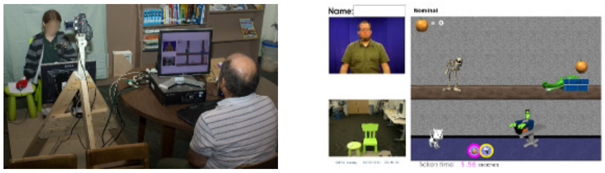
(a) (b)

Figure 1: (a) CopyCat system kiosk deployed at a partner school. (b) CopyCat game screen with tutorial video (top-left) and the raw video (bottom-left).

Table 1: CopyCat game vocabulary

subject	object	prepositions	adjectives
alligator, cat, snake, spider	bed, box, chair, flowers, wagon, wall	behind, in, on, under	black, blue, green, orange, white

As part of the CopyCat project, several computer-assisted language learning games have been designed [6]. In each game, children interact with the hero via sign language. After a child signs a phrase to the hero, the child's signing is classified as correct or incorrect. The phrases consist of a subject, preposition and an object with one or two optional adjectives. See Table 1 for a complete list of signs in the game vocabulary. If the child signs game phrases correctly, he or she will progress through the game and gain points.

The CopyCat systems have included research towards both automatic sign language recognition [4, 3] and sign language phrase verification [33, 34]. Both approaches have used hidden Markov models (HMMs) to model signs used during game play. User studies suggest that the CopyCat system provides educational benefits to deaf children of hearing parents in as little as three weeks [29]. Figure 1a shows the CopyCat system as deployed at a partner school.

The previous CopyCat sensor framework consisted of two types of sensors, a camera for eye and hand tracking and a pair of accelerometers that were strapped on the wrists over colored gloves.

ICMI'11, November 14–18, 2011, Alicante, Spain.

Although the gloves and accelerometers enhance the performance of the system, there are other significant drawbacks with their use. They increase the complexity of the system making it harder for teachers to setup the system in the classroom. They require constant maintenance; the batteries need to be charged frequently; and the gloves need to be washed regularly. Additionally, the wearable sensors add a possible point of failure.

In this paper, we propose a new multimodal system that uses the Microsoft Kinect sensor. The depth information is used for ASL phrase verification, whereas the RGB image stream is used to provide live signing feedback to the signer. The CopyCat game itself remains unchanged [34] (see Figure 1b) . We present experimental results that show that the Kinect system can yield performance comparable to the CopyCat sensor platform but may require a change in user interaction. With the Kinect, we are focusing on improving user comfort, robustness, user interaction, cost, system sustainability and ease of system deployment.

1.1 Related Work

Sign language recognition is a growing research area in the field of gesture recognition. Research on sign language recognition has been performed around the world using many sign languages, including American Sign Language [5, 26], Korean Sign Language [12], Taiwanese Sign Language [13], Chinese Sign Language [9, 10], Japanese Sign Language [17], and German Sign Language [2]. Many sign language recognition systems use HMMs for their ability to train useful models from limited and potentially noisy sensor data [10, 20, 26]. Sensor choices vary from data gloves [13] and other tracker systems to computer vision techniques using a single camera [20], multiple cameras, and motion capture systems [25] to hand-crafted sensor networks [11].

A variety of computer vision features are used in ASLR. Ong lists the following categories in his survey paper [15]: two dimensional segmentation, two dimensional moment-based, motion vectors, three dimensional hand positions and three dimensional hand orientations. In addition to these common features we would add: two dimensional head tracking [8, 33], three dimensional head orientation [24], and motion templates [7, 32].

Researchers differ greatly in their approach to modeling basic units of signed languages. The simultaneous nature of meaningful left hand, right hand, and head gesture in sign languages poses a challenge to many of the sequential techniques used in speech recognition [23]. Many researchers choose to use the sign as a base unit of modeling [4, 20], while others attempt to use a structure similar to phonemes to create models [8, 17].

Vogler and Metaxus have proposed several techniques for handling simultaneous phonemes using the Movement-Hold linguistics model and parallel HMMs [27]. Bowden et. al. use a two-tiered approach which classifies the TAB-SIG-DEZ features from Stokoe's phonology [21] and passes the results to a Markov chain. Bowden's approach is designed to learn from small amounts of data; he achieved results of 84% on a dataset of 49 signs performed in isolation by a single user. When the set was pruned for signs that require non-manual markers or context for meaning, Bowden's technique achieved an accuracy of 97.67% on 43 signs [8].

1.2 Depth Mapping

Depth maps can be generated by a number of algorithms and varying special camera configurations. RADAR, LIDAR, structured light techniques and sonar have all been used to generate depth maps [19]. Time-of-flight cameras are popularly used in computer vision and robotics to generate depth maps at a high frame rate [22]. Until now commercially available depth camera systems were expensive, and only a few researchers have used depth information to determine 3D hand pose [14]. Most previous works either use appearance based methods [1] or use some form of nearest neighbor matching in a database of poses [28] to determine the hand pose.

The release of the Microsoft Kinect sensor has provided a cheap ($150), off-the-shelf option for depth sensors. The Kinect consists of an infra-red (IR) light projector, standard CMOS camera (for the IR), color camera, and two microphones in a single housing. It uses a standard USB interface [16]. The distortion of the IR pattern is used to calculate depth maps, which have a per-pixel depth resolution of 1cm when the camera is 2 meters away [31]. The images are 640x480 and are transferred at 30 frames per second [18]. The Kinect will enable researchers to develop novel algorithms for 3D pose estimation using depth data.

2. DATA COLLECTION

We collected data using the CopyCat sensor framework and the Microsoft Kinect in order to compare the performance of the two sensor platforms. The data contains phrases from the CopyCat task but was collected from adult participants for preliminary testing. The three datasets used for comparison are

- **CopyCat Adult- seated** contains data collected from adult users playing the CopyCat game using colored gloves, wrist-mounted accelerometers and a camcorder.
- **Kinect - seated** contains data which mimics the CopyCat game interactions using the phrase repetition task (discussed in the next section) in a seated position. The seated position matches the configuration used in previous CopyCat studies.
- **Kinect - standing** contains data which mimics the CopyCat game interactions using the phrase repetition task (discussed in the next section) in a standing position. The standing position differs from the original CopyCat task but is more compatible with Kinect skeleton tracking algorithms (which were optimized for standing, not sitting).

2.1 Kinect Data Collection

We recruited seven participants, all of whom were hearing adults with varying levels of ASL expertise, ranging from beginner to expert. All seven users participated in data collection for the seated data, and three users participated in data collection for the standing set. During each session, users had 60 minutes to perform as many phrases from a 60 phrase set as comfortably possible. The dataset contains between one and two sessions per user. One user was unable to complete a standing session due to an interruption, so that user has fewer standing examples. A total of 555 phrase samples were collected in the seated pose and 155 phrase samples were collected in the standing pose.

2.1.1 Phrase Repetition Task

The phrase repetition task required participants to view videos of ASL phrases from the CopyCat game and sign them back to the camera. A front end application was developed in Java which allowed the users to view the ASL videos and view real time information from the Kinect camera consisting of the depth map image with skeleton tracking superimposed and the two-dimensional RGB image. The Java interface is shown in Figure 2.

Once the participant viewed a video, he or she would click the “Start Signing button" to begin signing and then clicked the “Stop Signing" button to indicate the termination of the ASL phrase. Since the purpose of this task was to collect data to examine the feasibility

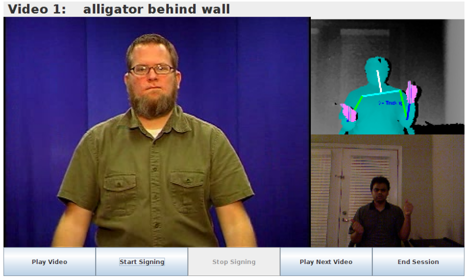

Figure 2: User interface for the phrase repetition task: The left video shows a tutorial for the phrase. The right side shows the raw video (bottom) and the video annotated with the skeleton tracking (top). The bottom buttons allow the user to navigate the system.

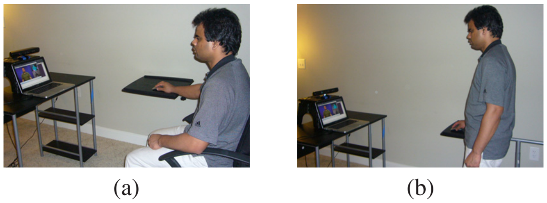

Figure 3: Participant pose: (a) Seated (b) Standing.

of the Kinect camera as a sensor for doing sign language recognition, we encouraged the participants to repeat signing the phrase until they signed it correctly. Each participant was asked to complete signing all the 60 phrases of the CopyCat game at least once. Figure 3a shows the physical setup where the participant is seated facing the Kinect camera and watching ASL videos on a laptop computer. The user controls the interface via a mouse placed on a table next to the chair.

2.1.2 *User Pose: Seated vs. Standing (Kinect data)*

We observed that the skeleton tracker suffered from performance issues in the seated position, losing track of the participant's body pose several times accompanied by jitter in the limb positions. Since we did not anticipate this problem to be a major issue, we did not gather statistics on the frame-level accuracy of the tracker. In many cases the researchers allowed a participant to ignore tracking errors and move forward once the phrase was signed correctly. Two signs that caused most of the failures were BED and ALLIGATOR, with BED being the most unfavorable of the two (see Figure 4).

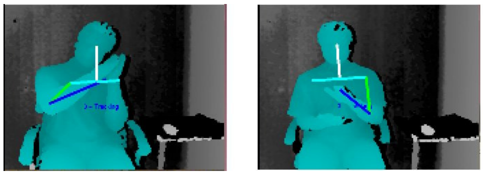

Figure 4: Tracking failure with BED and ALLIGATOR, respectively (Kinect data). Note that the lines should correspond with the user's arm positions but do not.

The experiment in the seated position represents the ideal case for a straightforward comparison since it correctly mimics user interaction used in previous work for the CopyCat project. Once we realized the high occurrence of tracking failures in the seated condition, we collected Kinect phrase repetition data for three users in the standing position. The standing position setup is shown in Figure 3b. The configuration difference for the standing position is that the camera is angled to capture the body knee up, and the mouse is place on a platform that can be easily reached in a standing position. We hypothesized that the skeleton tracking library that we used (OpenNI Framework) may yield better tracking performance in the standing position.

2.1.3 *Data Pruning*

We collected a total of 555 samples in the seated position and 155 in the standing position. The standing position suffers significantly less amount of tracking errors as compared to the seated position. However, we have not confirmed that all the available skeleton tracking implementations for Kinect depth data suffer from the same problem. The resulting dataset, after removing the samples with tracking errors, contained 146 signed phrases in the standing pose and 348 signed phrases in the seated pose.

2.2 CopyCat Adult Data Collection

For the CopyCat adult dataset we recruited eight participants, all of whom were hearing adults with varying levels of ASL expertise, ranging from beginner to expert. Of the eight participants recruited for this data set, three also participated in the Kinect data collection. The dataset contains two sessions per participant. Each session contains 20 phrases, for a total of 320 phrases in the dataset. Users sat at the CopyCat kiosk (Figure 1a) and played the CopyCat game.

2.2.1 *Sensor Configuration*

The CopyCat platform data collection used the CopyCat system as described in previous work [4, 33, 34, 3]. CopyCat uses computer vision and in-house designed three-axis accelerometers to collect data for use in sign language recognition [4, 6]. Video is collected on a single IEEE 1394 DV camcorder that faces the signer. The signer wears colored gloves, which contain small accelerometers mounted on the outside of the wrist (shown in Figure 5). These accelerometers provide information on movement: acceleration, direction, and rotation of the hand. The distinct color of the gloves helps distinguish the hands from the skin color of the face and cluttered backgrounds.

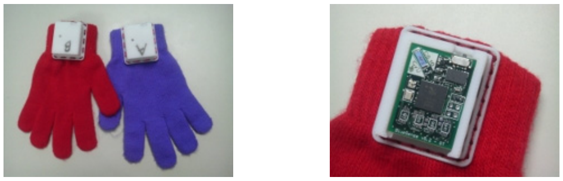

Figure 5: Left: Gloves with accelerometer. Right: Detail of wrist-mounted accelerometers (CopyCat sensor platform).

2.2.2 *Data Pruning*

A total of 320 phrase samples were collected for the CopyCat adult dataset. In order to choose samples compatible with the phrase repetition task, we selected samples that contained only correct signing. Five samples were excluded due to incorrect signing, and 26 samples were excluded because they contained variations in the signing such as sign repetition or other disfluencies. One of the signs excluded for language variation also contained machine errors. The resulting dataset contained 290 signed phrases of which 33 samples were removed due to data corruption (hand tracking errors or loss of accelerometer samples).

2.3 Data Collection Summary

Table 2 summarizes the number of samples collected in each dataset, sans all human anomalies, and the percentage of the dataset that was unusable due to data collection errors (machine errors). A total of 1000 ASL phrases were collected and the least amount of data collection errors were observed in the Kinect standing dataset.

Table 2: Samples collected versus sensor failures for all data

Configuration	#Samples	Corrupt Samples%
Kinect Seated	555	37.3%
Kinect Standing	155	5.8%
CopyCat Adult	290	14.8%

3. EXPERIMENTAL DESIGN

We follow a three step process. The first step is feature extraction where salient information that will enable us to perform automatic recognition is obtained. The second step is to train Hidden Markov Models (HMMs) for each sign in the game vocabulary. In the final step, the generalization performance of these models is measured by testing on independent data.

3.1 Features

The details of the feature extraction process for the Kinect sensor and the CopyCat sensor platform are given below.

3.1.1 Kinect

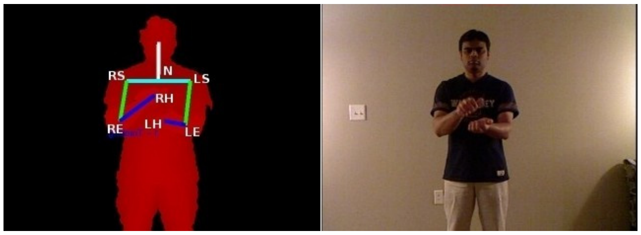

Figure 6: Visualization of upper body skeletal joints generated using OpenNI Framework (Kinect data).

Table 3: The 20 body pose features generated using OpenNI framework (Kinect data)

Unit 3D Vectors	Angles	distances
$RS \rightarrow RE$ (3)	$\angle N - RS - RE$ (1)	$(RH \rightarrow LH)$ (1)
$RE \rightarrow RH$ (3)	$\angle RS - RE - RH$ (1)	
$LS \rightarrow LE$ (3)	$\angle N - LS - LE$ (1)	
$LE \rightarrow LH$ (3)	$\angle LS - LE - LH$ (1)	
$RH \rightarrow LH$ (3)		

The Kinect features are a combination of body pose and hand shape features.The pose features are extracted using the skeletonization capability provided by the the OpenNI framework. OpenNI stands for "Open Natural Interaction" and provides a set of APIs that allows developers to write applications that utilize natural interaction. The OpenNI framework interfaces with NITE middleware provided by PrimeSense, the company that developed the Kinect hardware [16]. The skeleton tracker is configured to return only the joints in the upper body. We obtain the shoulder, elbow and hand positions for the right and the left side of the body. We build the body pose feature from the joint angles of the shoulders and the elbows, the unit vectors of the elbows with respect to the shoulder, the hands with respect to the elbows and finally the left hand with respect to the right hand. See Figure 6 and Table 3 for details of the body pose feature. The body pose feature has 20 dimensions.

For generating the hand shape feature we collect all the 3D points in the neighborhood of each of the hand positions (RH and LH in Figure 6). We cluster the points corresponding to each hand separately to obtain a mixture of six Gaussians (MoG). We use six Gaussians because the human hand with the palm open has 6 distinct regions, five for the fingers and one for the palm. For simplicity we use diagonal covariance matrices while learning the mixture using Expectation Maximization (EM). The parameters of the Gaussians serve as the initial features. Each gaussian has six parameters, three for the mean and three for the diagonal of the covariance matrix. This results in a combined feature of 72 dimensions. We then perform Principal Component Analysis (PCA) on the entire dataset of MoG hand features to reduce the dimensionality of the hand shape feature from 72 to 20.

3.1.2 CopyCat

Features for data collected with the CopyCat kiosk are a combination of vision and accelerometer data. Video frames are captured at 20 fps, and the accelerometers, which have a range of +2g to -2g, are sampled at 40 Hz. Vision features are obtained by tracking the eyes and the colored gloves while playing the game [34]. The corresponding accelerometer features are obtained by matching the accelerometer sample that is nearest in time to the video frame and computing the Fast Fourier Transform (FFT) within a window of samples. For a complete list of features generated for each hand, see Table 4.

Table 4: Feature types generated from the CopyCat sensor platform

Type	Description
Blob (hand Shape)	second moment shape descriptors (length of major and minor axes, eccentricity, orientation of major axis)
2D Image Motion	d_x and d_y of the blob center
Acceleration	x, y & z acceleration values and frequency domain representation of each axis.
Pose (2D geometry)	angle formed between the blob center and the horizontal passing through the midpoint between the eyes

3.2 Training and Testing

Once the features are collected, we train 4-state Hidden Markov Models (HMMs) for each of the 19 signs in the vocabulary shown in Table 1. Training and testing was done using the Georgia Tech Gesture Toolkit GT^2K [30]. To evaluate the performance of the system, we perform two types of tasks: *recognition* and *verification*. We encourage the reader to understand the difference between the two in this context. In **automatic sign language recognition** the most probable transcription, based on models, is chosen that fits the input. We use the part of speech grammar shown below for our automatic recognition:

[adjective] subject preposition [adjective] object

This phrase structure supports three to five sign phrases using the optional adjectives. From a fixed dictionary of signs shown in Table 1 the *Recognizer* fills in one sign for each slot. If the right sign has been filled into all the slots then the phrase is recognized as correct, otherwise it is incorrect.

In **automatic sign language verification**, prior knowledge of the expected ASL phrase is used to determine if the input data matches the sentence. To perform the matching, the *Verifier* aligns the input data with the signs that constitute the expected phrase and filters based on the likelihood values [34]. If all tokens pass the likelihood threshold test, then the phrase is verified. Since the current real time CopyCat system, with the gloves and accelerometers, uses the thresholding approach, we use the same method which allows a comparison between the CopyCat system and the Kinect.

For both *recognition* and *verification* the generalization performance of trained models is assessed using leave one out cross validation (LOOCV). However, the procedure is different in each case. For *recognition*, the approach is straightforward. We set aside one participant's data for testing and train on data from the remaining participants. In *verification*, since a likelihood threshold is applied, we first need to determine the appropriate threshold to use. The procedure to learn a threshold for each sign is outlined in Figure 7. It is a two-level LOOCV procedure where the inner LOOCV step is used to determine the the log likelihood threshold to be used for each sign and the thresholds thus computed are then used in the outer LOOCV step. The results from each LOOCV run, the outer runs in the case of verification, are combined to obtain the overall recognition and verification accuracy.

```
for all participants (p) do
    $V_p \leftarrow set\ aside\ participant\ \mathbf{p}'s\ data\ for\ validation$

    for all remaining participants (r) do
        1. $V_r \leftarrow set\ aside\ participant\ \mathbf{r}'s\ data\ for\ validation$
        2. $Train\ on\ remaining\ N-2\ participants\ and\ test$
           $on\ V_r$
        3. $For\ each\ instance\ of\ a\ sign\ in\ V_r\ note\ the\ log$
           $likelihood\ value\ obtained\ via\ Viterbi\ alignment$
    end for

    $For\ each\ sign\ calculate\ the\ mean(\mu)\ and\ standard$
    $deviation(\sigma)\ of\ the\ log\ likelihood\ values\ collected$
    $Set\ a\ log\ likelihood\ threshold\ for\ each\ sign$
    $(\mu - \kappa * \sigma, \kappa > 0)$
    $Train\ on\ remaining\ N-1\ participants\ and\ test\ on\ V_p$
    $using\ the\ computed\ thresholds$
end for
```

Figure 7: Procedure for computing log likelihood thresholds to perform verification.

4. RESULTS

We evaluate performance based on both recognition and verification accuracies.

4.1 Baseline from Previous Work

Previous work, which compared phrase recognition and verification rates for data from deaf children playing the game, resulted in a maximum phrase recognition accuracy of 67.0% and a maximum phrase level verification rate of 82.0% (see Table 5) [34]. Training data for the experiments was collected using a Wizard of Oz version of the game in which an external observer played the role of the ASL verifier. A total of 1204 phrase examples were collected from 11 deaf children playing the game during the phase two deployment of CopyCat. This set included 894 correctly signed phrases and 310 phrases that were incorrectly signed.

Table 5: Phrase-level cross validation accuracies from previous work that show a maximum phrase recognition accuracy of 67.0% and a maximum phrase level verification rate of 82.0% [34].

	Verification (%)			Recognition (%)		
	TP	**FP**	**Acc.**	**TP**	**FP**	**Acc.**
M=1	86.0	38.0	79.8	54.0	3.0	64.9
M=2	88.6	37.2	**82.0**	56.8	3.3	**67.0**

Hidden Markov Models were trained with the correctly signed phrases using the Georgia Tech Gesture Recognition Toolkit GT^2K [30]. Cross validation was performed by using 90% of the data for training and 10% for testing. The trained models were then evaluated on the incorrectly signed data to determine the false positive rate. This process was repeated 50 times and the results were averaged. Additionally, up to two Gaussian mixtures components were used for the training process (M=1 and M=2). For verification, a common rejection threshold was applied to the normalized log likelihood score, obtained using forced alignment, for all classes.

4.2 Kinect Data

Table 6 and Table 7 show the recognition and verification results of leave-one-out cross validation in the seated and standing poses, respectively. For the seated dataset the threshold selection for verification was done using the procedure shown in Figure 7.

Table 6: Recognition and Verification results - Kinect seated

Kinect Seated				
	Recognition		Verification	
Subj	Word Acc	SENT Acc	Word Acc	SENT Acc
1	51.79	26.67	89.29	60.00
2	77.19	43.90	96.56	87.80
3	73.1	35.00	95.17	82.50
4	69.11	25.00	92.68	71.88
5	75.53	32.2	96.38	85.59
6	80.00	43.48	92.78	76.09
7	70.00	40.00	96.00	86.67
Overall	74.48	36.2	95.16	82.18

Table 7: Recognition and Verification results - Kinect standing

Kinect Standing				
	Recognition		Verification	
Subj	Word Acc	SENT Acc	Word Acc	SENT Acc
1	65.04	26.79	87.61	58.93
2	75.00	38.33	98.33	93.33
3	85.71	50	99.25	96.67
Overall	73.62	36.3	94.49	80.82

The value of κ was varied from 1 to 10. At κ = 6 the increase in verification accuracy begins to flatten. The results for κ = 6 are shown in Table 6. See the discussion section for the average verifcation accuracies for all values of κ. Since there were only three participants in the standing dataset the threshold selection procedure could not be applied. Instead we choose a common threshold for all signs, which was obtained by averaging the threshold values from the seated dataset, with $\kappa = 6$, across all signs (s) and all the participants (p) (see Equation 1).

$$\frac{1}{|p| * |s|} \sum_{p} \sum_{s} \mu - \kappa * \sigma \tag{1}$$

Based on the overall recognition and verification accuracies, there appears to be no significant difference in performance between seated and standing positions. This result demonstrates that the recognition/verification framework is robust to the jitter that occurs in the seated position. When tracking errors are ignored, the seated pose does slightly better in terms of verification acheiving 82.18% accuracy compared to 80.82% in the case of the standing pose.

4.3 CopyCat Adult Data

Table 8 shows the recognition and verification results of leave-one-out cross validation on the CopyCat Adult dataset. The threshold selection to perform verification was done using the procedure outlined in Figure 7, which is the same approach used for the Kinect seated dataset. The value of κ was varied from 1 to 10 but only results for $\kappa = 6$, the point at which the verification accuracy begins to flatten, are shown in Table 8.

Table 8: Recognition and Verification results - CopyCat Adult

CopyCat Adult dataset				
	Recognition		Verification	
Subj	**Word Acc**	**SENT Acc**	**Word Acc**	**SENT Acc**
1	80.43	42.42	99.34	97.22
2	63.64	21.05	92.05	63.16
3	79.01	43.24	97.10	87.88
4	91.23	61.54	94.44	78.33
5	86.18	52.78	91.30	70.00
6	93.21	72.97	100.00	100.00
7	80.43	40.00	95.80	84.62
8	84.03	53.85	100.00	100.00
Overall	85.42	51.01	96.86	87.85

4.4 Discussion

Table 9 provides a comparison between the three datasets based on the following factors: Recognition/Verification accuracy, Robustness, Aesthetics & Comfort and User Interaction.

Table 9: Comparison chart between the three datasets (ratings: *good*, *fair*, *poor*, X - further investigation required)

	CopyCat Adult dataset	**Kinect Seated**	**Kinect Standing**
Recognition / Verification Acc.	good	fair	fair
Robustness	fair	X	good
Aesthetics & Confort	fair	X	X
User Interaction	good	X	X

4.4.1 *Recognition/Verification accuracy*

Our experiments showed that the exisiting CopyCat sensor platform performs significantly better than the Kinect in terms of recognition rates and also yields higher verification rates. Table 10 gives a summary of the overall recognition and verification rates for the three datasets. The baseline results on the CopyCat children dataset from previous work is also included for reference.

Table 10: Summary of recognition and verification results

	Recognition		Verification	
	Word Acc	**SENT Acc**	**Word Acc**	**SENT Acc**
Kinect Seated	74.48	36.2	95.16	82.18
Kinect Standing	73.62	36.3	94.49	80.82
CopyCat Adult	85.42	51.01	96.86	87.85
CopyCat children	-	67.0	-	82.0

Based on previous work, the features obtained from the CopyCat sensor platform have gone through several improvements. The Kinect features have been introduced for the first time for the CopyCat task. There definitely is scope for improving the Kinect features, especially with respect to the hand shape. Our current hand shape feature for the Kinect is still primitive and was selected keeping in mind that real-time performance is desired. In the future we will explore more sophisticated hand shape features derived from depth information.

4.4.2 *Robustness*

The CopyCat sensor platform involves more than one sensor resulting in more failure points. Moreover, since conventional computer vision algorithms are used for eye tracking and hand tracking, environmental factors such as lighting conditions can have a negative impact on the robustness of tracking. Accelerometer sampling errors can also occur. If a video frame could not be paired with a matching accelerometer sample, and this problem occurred consecutively for two or more frames, then it was considered as a accelerometer sampling error.

Since the Kinect estimates depth by projecting an infrared light pattern, it is unaffected by environmental factors, resulting in a very reliable depth map output. However, the skeleton tracking facility provided by the OpenNI framework is robust only in the standing pose and fails more than one third of the time when the user is seated. Once we factor in the tracking errors, considering them as recognition and verification failures, the standing pose emerges as the more favorable of the two configurations. A detailed comparison between the CopyCat sensor platform dataset and the Kinect seated dataset is shown in Figure 8. There is a significant drop in the overall verification accuracy for the seated dataset due to the large amount of corrupt data as identified in Table 2.

Further investigation is clearly warranted on the standing vs seated problem, and other frameworks for skeleton tracking also need to be thoroughly explored in order to determine if they may be able to overcome the tracking problem in the seated position.

Table 11 lists the updated recognition and verification rates once we factor in the errors that occur in data collection for all three datasets. The CopyCat sensor platform still performs better at recognition, but the effective verification accuracy for the Kinect standing dataset is better than the CopyCat Adult dataset once all sources of error are included.

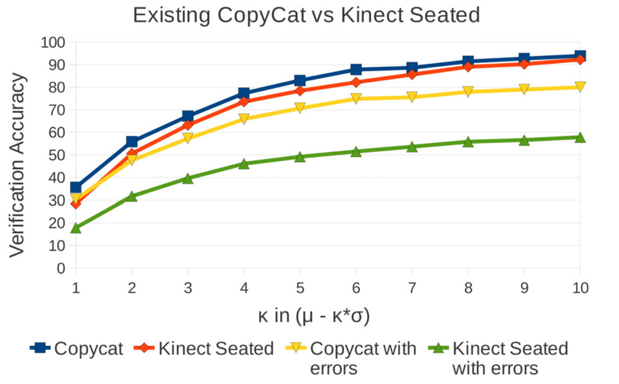

Figure 8: Comparison of verification accuracy between CopyCat Adult dataset and Kinect seated dataset, with and without data collection errors, for different values of the threshold.

Table 11: Results after factoring in tracking errors

	Recognition		Verification	
	Word Acc	**SENT Acc**	**Word Acc**	**SENT Acc**
Kinect Seated	47.74	22.7	58.86	51.5
Kinect Standing	70.45	34.19	88.02	76.12
CopyCat Adult	73.72	43.44	83.59	74.82

4.4.3 *Aesthetics & Comfort*

The Kinect is a clean and neat out-of-the-box solution for obtaining depth information. With minimal hardware, it provides a clutter-free environment to setup the CopyCat game.

The CopyCat sensor platform requires a kiosk-like setup. Additionally, accelerometer batteries may need to be regularly recharged, and circumstances may arise when controlled lighting is required. A researcher has to be present to run the CopyCat system during deployment, whereas the Kinect provides a plug-n-play interface that even individuals without a technical background will be able to operate with ease.

One of the major drawbacks of the CopyCat sensor platform is that it requires the user to wear gloves with accelerometers strapped to the wrist to enable the capture of gestural information. Although previous studies have shown that children did not have problems wearing the gloves, the comfort factor with Kinect and the facility to interact naturally without aids may have a greater appeal to the children for whom the game is designed.

4.4.4 *User Interaction*

The existing CopyCat setup, where the user is seated in a chair and interacts with the game using a mouse placed at a convenient location next to the chair (see Figure 1a), has been deployed several times in schools with deaf children. This interaction model was designed iteratively over several usability studies conducted in partnership with local deaf schools with the involvement of deaf children and their teachers. One of the concerns when deciding upon the seated pose for the user/child was that fatigue that may occur if the children have to stand continuously for a period of 30-45 minutes to play the game (approximately the time it takes to finish one session). That the Kinect skeleton tracking performs poorly in the seated position, but very well in the standing position, suggests an open area for research. We have yet to ascertain the effect that standing may have on user experience and to what extent fatigue may factor.

One critical aspect of CopyCat involves set-up and tear down of the system. With the current CopyCat sensor platform, the teacher would be required to manage the hardware extensively (e.g., charge the batteries regularly and maintain the gloves in good usable condition). In contrast, the Kinect provides a convenient, out-of-the-box solution. Moreover, the Kinect is easy and inexpensive to replace.

5. CONCLUSION

We have presented an alternative approach to the automatic sign language verification task suitable for the CopyCat game using the Kinect camera. This approach mitigates the need for users to wear gloves and strap sensors on their wrists, which is the most significant drawback of the current CopyCat system. The experimental results show that verification results with the Kinect are comparable to the existing system. However, we discovered that when the users are seated the skeleton tracking system tends to make a significant number of tracking errors, which could have a negative impact on the overall user experience while playing the CopyCat game. One alternative is to change the way users interact with system, having them stand rather than being seated. Experiments have shown that standing results in fewer tracking errors and may have suitable verification accuracies. Changing the user's pose from a seated to standing position involves a shift in the interaction model for the game. We are only beginning to explore the effects that this may have on the user experience.

Our future plan is to develop a real time version of the CopyCat game that uses the Kinect sensor and is suitable for deaf children. Our past experience has shown that recognizing children's signing is an extremely challenging task. Our goal is to exploit the potential of the Kinect depth information to its fullest, especially with regards to spotting hand shapes.

6. REFERENCES

[1] V. Athitsos and S. Sclaroff. 3d hand pose estimation by finding appearance-based matches in a large database of training views. In *In IEEE Workshop on Cues in Communication*, 2001.

[2] B. Bauer, H. Hienz, and K. Kraiss. Video-based continuous sign language recognition using statistical methods. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 463–466, September 2000.

[3] H. Brashear. *Improving the Efficacy of Automated Sign Language Practice Tools*. PhD thesis, Georgia Institute of Technology, College of Computing, 2010.

[4] H. Brashear, K.-H. Park, S. Lee, V. Henderson, H. Hamilton, and T. Starner. American Sign Language Recognition in Game Development for Deaf Children. In *Assets '06: Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, Portland, Oregon, 2006. ACM Press.

[5] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. Using Multiple Sensors for Mobile Sign Language Recognition. In *Proceedings of the Seventh IEEE International Symposium on Wearable Computers*, pages 45–52, 2003.

[6] H. Brashear, Z. Zafrulla, T. Starner, H. Hamilton, P. Presti, and S. Lee. CopyCat: A Corpus for Verifying American Sign Language During Game Play by Deaf Children. In *4th Workshop on the Representation and Processing of Sign*

Languages: Corpora and Sign Language Technologies, Proceedings of the 7th Annual International Language Resources and Evaluation Conference, Valetta, Malta, 2010.

[7] Cooper and Bowden. Sign Language Recognition: Working with Limited Corpora. In *Universal Access in Human-Computer Interaction. Applications and Services 5th International Conference, UAHCI 2009, Held as Part of HCI International 2009*, pages 472–481, 2009.

[8] O. Eng-Jon and R. Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 889–894, May 2004.

[9] G. Fang, W. Gao, and D. Zhao. Large Vocabulary Sign Language Recognition Based on Hierarchical Decision Trees. In *International Conference on Multimodal Interfaces*, pages 125–131, 2003.

[10] W. Gao, G. Fang, D. Zhao, and Y. Chen. Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition (CSL). In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 553–558, 2004.

[11] J. L. Hernandez-Rebollar, N. Kyriakopoulos, and R. W. Lindeman. A New Instrumented Approach for Translating American Sign Language into Sound and Text. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 547–552, 2004.

[12] J. S. Kim, W. Jang, and Z. Bien. A Dynamic Gesture Recognition System for the Korean Sign Language KSL. *IEEE Transactions on Systems, Man and Cybernetics*, 26(2):354–359, 1996.

[13] R. Liang and M. Ouhyoung. A Real-Time Continuous Gesture Recognition System for Sign Language. In *Third International Conference on Automatic Face and Gesture Recognition*, pages 558–565, 1998.

[14] Z. Mo and U. Neumann. Real-time hand pose recognition using low-resolution depth images. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1499–1505, Washington, DC, USA, 2006. IEEE Computer Society.

[15] S. Ong and S. Ranganath. Automatic sign language analysis: a survey and the future beyond lexical meaning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):873 – 891, jun 2005.

[16] Prime Sense. *The PrimeSensor^{TM}Reference Design 1.08*. April 2011.

[17] H. Sagawa and M. Takeuchi. A method for recognizing a sequence of sign language words represented in a Japanese Sign Language sentence. In *Proceedings of the 4th IEEE International Conf. on Automatic Face and Gesture Recognition*, pages 434–439, Grenoble, France, Mar 2000.

[18] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[19] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomspon Learning, Toronto, Ontario, Canada, third edition, 2008.

[20] T. Starner and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 1995.

[21] W. Stokoe. Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *Studies in Linguistics, Occasional Papers 8*, 1960.

[22] J. Stuckler and S. Behnke. Combining depth and color cues for scale- and viewpoint-invariant object segmentation and recognition using random forests. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4566 –4571, October 2010.

[23] G. Ten Holt, P. Hendriks, and T. Andringa. Why Don't You See What I Mean? Prospects and Limitations of Current Automatic Sign Recognition Research. *Sign Language Studies*, 6(4), Summer 2006.

[24] C. Vogler and D. Metaxas. Adapting Hidden Markov Models for ASL recognition by using three-dimensional computer vision methods. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 156–161, Orlando, FL, October 1997.

[25] C. Vogler and D. Metaxas. ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 363–369, 1998.

[26] C. Vogler and D. Metaxas. Handshapes and Movements: Multiple-Channel American Sign Language Recognition. In *Gesture-Based Communication in Human-Computer Interaction*, volume 2915, pages 247–258. Springer-Verlag, January 2004. Lecture notes in Artificial Intelligence.

[27] C. Vogler, H. Sun, and D. Metaxas. A framework for motion recognition with applications to American sign language and gait recognition. In *Proceedings of Workshop on Human Motion*, pages 33–38, 2000.

[28] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Trans. Graph.*, 28:63:1–63:8, July 2009.

[29] K. A. Weaver, H. Hamilton, Z. Zafrulla, H. Brashear, T. Starner, P. Presti, , and A. Bruckman. Improving the Language Ability of Deaf Signing Children through an Interactive American Sign Language-Based Video Game. In *Proceedings of 9th International Conference of the Learning Sciences*, June 2010.

[30] T. Westeyn, H. Brashear, A. Atrash, and T. Starner. Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition. In *ICMI '03: Proceedings of the 5th International Conference on Multimodal Interfaces*, New York, NY, USA, 2003. ACM Press.

[31] A. Wilson. Using a Depth Camera as a Touch Sensor. *The ACM International Conference on Interactive Tabletops and Surfaces*, November 2010.

[32] R. Yang, S. Sarkar, and B. Loeding. Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):462–477, March 2010.

[33] Z. Zafrulla, H. Brashear, H. Hamilton, and T. Starner. A novel approach to American Sign Language (ASL) Phrase Verification using Reversed Signing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[34] Z. Zafrulla, H. Brashear, H. Hamilton, and T. Starner. Towards an American Sign Langauge Verifier for Educational Game for Deaf Children. In *Proceedings of International Conference on Pattern Recognition*, 2010.