Greek sign language vocabulary recognition using Kinect

Nikolaos Gkigkelos
Information and Communication Systems Engineering Dept.
University of the Aegean
Greece
icsd11029@icsd.aegean.gr

Christos Goumopoulos
Information and Communication Systems Engineering Dept.
University of the Aegean
Greece
goumop@aegean.gr

ABSTRACT

Sign language recognition is a challenging problem both when tracking continuous signs (communication mode) or single words (translation mode) [1]. We have developed a system that can recognize Greek sign language vocabulary in translation mode using Kinect technology. The sensor captures 3D hands movement trajectory and then a set of features in the form of body joints are fed to a classifier to recognize the input sign. Normalization is used to align test and stored trajectories using the dynamic time warping algorithm before matching is done using the Nearest-Neighbor approach. The low computational complexity of the involved algorithms allows for building a system with real-time response times. The system was evaluated with a sample of 5 individuals and is capable of recognizing 15 signs of the Greek sign language. Different configurations were tested and the best accuracy achieved was 99.33%.

CCS CONCEPTS

• **Human-centered computing** → Ubiquitous and mobile computing → Ubiquitous and mobile computing design and evaluation methods

KEYWORDS

Sign language recognition; 3D motion trajectory; trajectory alignment; pilot study; Kinect

1 INTRODUCTION

Over 5% of the world's population, approximately 360 million people has disabling hearing loss [1]. Deaf people often use sign language for communication by making hands and face movements to convey meaning [2]. Progress in sensors and devices development as well as research efforts on algorithmic techniques and signal processing empowered with Artificial Intelligence methods have allowed the development of sign language recognition systems. In that respect barriers for deaf people can be lifted and communication gaps with speakers can be covered in everyday life situations.

Microsoft Xbox Kinect is an RGBD camera that was initially developed as a peripheral device for use with the Xbox gaming console. The device itself features an RGB camera, a depth sensor, a multiarray microphone, and is capable of tracking users' body movement. It has not only achieved its initial goal i.e. gaming, but set up the floor to the development of a wide number of useful applications in the field of Computer Vision [3], Gesture Recognition [4] and Robotics [5].

In this paper, we describe the development and evaluation of a sign language recognition system that helps signers and speakers to communicate. The application requires the use of Xbox Kinect sensor and the Kinect PC Adapter. The system, with the appropriate adjustments, can be used in various scenarios. It can be installed at information kiosks of airports, for assisting hearing impaired employees. For example, a passenger walks in, asks her question using spoken language, the signer answers in sign language and the system translates the signs to text or speech. Also, it can be used by individuals who want to learn or practice the sign language. The system displays a sign and asks the user to repeat, the user makes the proper gestures and the system assess whether the sign was performed correctly.

In its current state, the system is able to recognize 15 words of the Greek sign language. Using the spherical coordinate system to track signs, signal filtering, the Nearest-Neighbor classifier and the dynamic time warping algorithm for trajectory alignment, the recognition accuracy reached up to 99.33%.

The rest of the paper is organized as follows. Section 2 discusses related work. The next section presents the key points of the approach followed to develop the sign language recognition system. We describe the main system components, the way data is acquired from the camera, the need for processing with filters and data normalization, the sign classifier used and the implementation environment. Section 4 discusses the evaluation of the system performed with the help of 5 individuals. Results are presented for different testing configurations. Finally, we give our conclusions and discuss our plans for future work.

2 RELATED WORK

A study performed by Chai et al. and supported by Microsoft Research Asia, proposed a sign language recognition and translation system using the Kinect sensor. Their approach uses 3-

PCI 2017, September 28–30, 2017, Larissa, Greece

ACM ISBN 978-1-4503-5355-7/17/09…$15.00
https://doi.org/10.1145/3139367.3139386

D hand trajectory tracking, normalization by linear resampling and trajectory alignment to recognize 239 Chinese sign language words [6]. Final classification is measured by computing the Euclidean distance and results show a matching accuracy between 83.51% and 96.32%. The system can translate sign language into text or speech and help a signer to communicate with a speaking person. The study suggests that the system can be installed in a doctor's office and an airport to improve communication.

In a similar study performed by Hee-Deok Yang et al. a sign language recognition system was developed [7]. Of the 24 signs, 7 were one-handed signs, and 17 were two-handed signs. The system tracked 10 upper body points and the hand shape is normalized with the help of a black wristband. Using a hierarchical conditional random field framework and a BoostMap embedding method, 24 signs of American sign language are recognized at a rate of 90.4%.

In another study, Capilla developed an automatic sign language translator for 14 unofficial signs [8]. In this study, several functions of classification are proposed. The best accuracy achieved was 95.23% by using spherical coordinates and processing the data with the Nearest-Neighbor function. The cost is calculated using dynamic time warping algorithm. Capilla suggests installing the system in hospitals and supermarkets to assist deaf people interaction.

Oszust et al. developed a polish sign language words recognition system using Kinect [9]. Two approaches were evaluated. The best one yielded 98.33% accuracy using nine subsets as the training set and the remaining subset as the test set. The first approach focused on describing hands by using images obtained by the color and depth camera of the Kinect. The second approach used skeletal data obtained by the sensor and delivered 89.33% accuracy.

A study performed by Fernando and Wimalaratne reporting a sign language to Sinhalese language translator applied to a chat application [10]. Using dynamic time warping and Nearest-Neighbor classification, the translator could recognize 15 signs. The system achieved 92.4% accuracy using 225 test samples.

Lang et al. developed an open source framework, called Dragonfly, for general gesture recognition using the Kinect camera. Dragonfly, is written in C++ and makes use of the cross-platform Kinect driver OpenNI [11]. With the use of continuous density hidden Markov models, 25 signs of the German sign language are recognized with an accuracy up to 97%.

In another study performed by Wu et al. and supported by National Science Foundation and Texas Instruments Inc., an American sign language recognition system using wrist-worn motion and surface EMG sensors was developed [12]. The system is capable of recognizing 40 signs. Data are obtained from sensors via Bluetooth and are stored after filtering. Four classification algorithms are used, namely decision tree, support vector machine, Nearest-Neighbor and Naive Bayes. Support vector machine obtains the best accuracy of 95.14%.

A study performed by Starner et al. presented two hidden Markov model-based systems that recognize 40 sentence-level American Sign Language words [13]. The first system uses a desk-based tracking camera and achieves 92% word accuracy by analyzing images at 320x240 resolution and 10 frames per second. The second system uses a hat-mounted camera worn by the signer, pointing to the hands. It achieves 98% accuracy.

Our approach is closely related to the approaches described in [8] and [10] with the following enhancements. We used the newer Xbox's One Kinect sensor instead of the Xbox's 360 sensor and the newer version of the Kinect API (v2). The new sensor provides higher resolution and tracks more skeleton joints. Additional joints that are used with respect to similar approaches are the hand tip and the thumb joints. The use of data filtering techniques found to be useful in improving system accuracy. Finally, our system reaches its best accuracy, 99.33%, when testing signs are compared to training samples that are coming from a single position, which is higher compared to corresponding cases in related work.

3 SYSTEM DEVELOPMENT

A fully functional system recognizing continuous signs is a particularly challenging problem because it is difficult to model the transition between signs. In this paper, a simplified version of the problem is studied i.e. recognizing specific words of the Greek sign language. In terms of implementation each sign is modeled separately and a training phase is required. For training a set of data is needed for each frame acquired and includes information that can be extracted from the available channels i.e. movement and hand position, handshape, facial expression and voice.

The Microsoft Kinect sensor, in addition to RGB video, provides depth footage by using an infrared laser projector combined with a monochrome CMOS sensor and skeletal tracking [14]. Therefore, Kinect's usefulness in gesture recognition applications is evident, since it greatly facilitates the assessment of tracking the position and movement of hands and body. Key elements of the Kinect are the ability to track joints i.e. specific parts of a body and obtaining data of these joints for each frame i.e. an electronically coded still image in real time.

An overview of the sign language vocabulary recognition process is shown in Fig. 1.

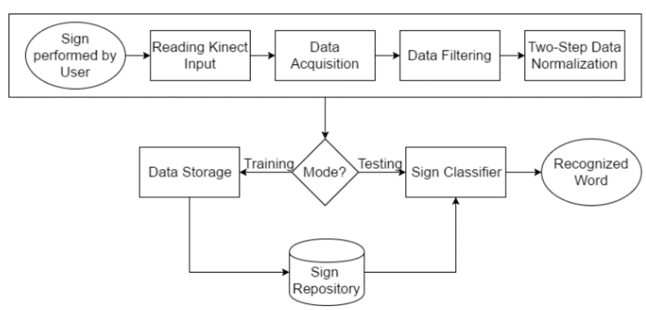

Figure 1: Overview of the sign language vocabulary recognition process.

The user is standing in front of the Kinect camera performing a sign. A new frame is obtained every 0.033sec (30 fps) and the video stream is updated. When a sign is given either in training or translation mode the data of the selected tracked joints are obtained using functions given by the Kinect API. Afterwards, these data are normalized and saved to a data structure, until the recording is stopped. In the training mode, the recorded sign is added to the training set whereas in the testing/translation mode the recorded sign is compared with signs that already exist in the sign repository using a classifier and the matching output is displayed as text (see Fig. 2)

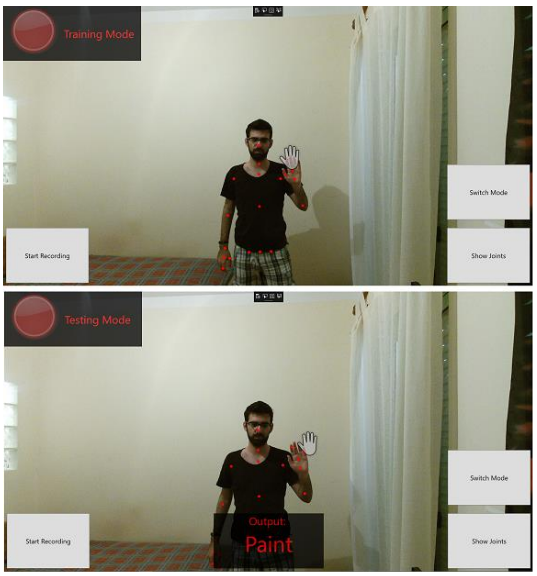

Figure 2: **Training and testing modes of the system.**

3.1 Data Acquisition

With the use of the Kinect API, up to 25 joints of person's body can be tracked. Tracking all of them results in generating redundant data because the signs reenactment requires only hand movement, therefore the program would track only 10 joints. These joints are defined as the set:

$J = \{HTL, TL, WL, HL, EL, HTR, TR, WR, HR, ER\}$,

where *HTL* is the Hand Tip Left, *TL* is the Thumb Left, *WL* is the Wrist Left, *HL* is the Hand Left, *EL* is the Elbow Left, *HTR* is the Hand Tip Right, *TR* is the Thumb Right, *WR* is the Wrist Right, *HR* is the Hand Right and *ER* is the Elbow Right as illustrated in Fig. 3.

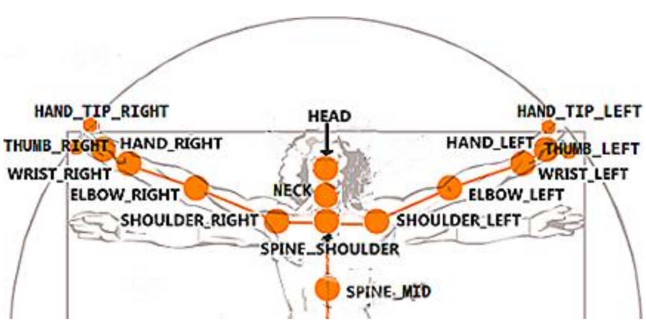

Figure 3: **Upper body joints.**

Even though these joints are enough for describing a sign, two more joints are required for the data normalization steps. These are the Head and the Spine Mid defined as *H* and *O* respectively (see Fig. 3). In this way, the final selected tracked joints are reduced from 25 to 12.

3.2 Data Filtering

Median filtering is widely used to remove noise from an image or signal. In one of the proposed configurations, we apply a median filter to smooth the data obtained in both modes. The algorithm used is shown in Fig. 4.

```
function MedianFiltering (file sign, int WindowSize)
    declare int x <- (WindowSize -1) / 2
    declare file filteredSign

    sign <- TreatEdges(sign, x)

    for all frames in sign
        for all frames in WindowSize
            sortFrames()
            filteredSign <- addMiddleFrame()

    return filteredSign
end function
```

Figure 4: **Median Filtering algorithm.**

3.3 Two-Step Data Normalization

The normalization of the data must take into account the position of the person in the room and the data must be stored in the repository accordingly. A different position of the user in the room causes a variation of the Cartesian coordinates X, Y, Z, resulting in different values for the same sign. These coordinates are obtained using the Kinect API.

In the literature, different normalization rules have been proposed. We have examined two of them, based on the middle of the spine [8] and based on the center of distance between the two shoulders [10]. In our system, all the joint coordinates will be normalized with respect to the position of the middle of the spine (*O*), because this position remains constant while recording a sign and will make the user's position independent. Furthermore, instead of storing the Cartesian coordinates X, Y, Z in the repository, the spherical coordinates considering the above normalization will be stored.

The spherical coordinate system is a coordinate system for representing geometric figures in three dimensions using three coordinates: the radial distance of a point from a fixed origin (*R*), the zenith angle from the positive z-axis (Θ), and the azimuth angle from the positive x-axis (Φ).

In the proposed system, the position of the point results by combining the currently tracked joint set *J* and the middle of the spine origin *O*. The set of radial distances $R = \{r_{HTL}, r_{TL}, r_{WL}, r_{HL}, r_{EL}, r_{HTR}, r_{TR}, r_{WR}, r_{HR}, r_{ER}\}$, the inclinations (polar angles) $\Theta = \{\theta_{HTL}, \theta_{TL}, \theta_{WL}, \theta_{HL}, \theta_{EL}, \theta_{HTR}, \theta_{TR}, \theta_{WR}, \theta_{HR}, \theta_{ER}\}$ and the azimuths (azimuthal angles) $\Phi = \{\varphi_{HTL}, \varphi_{TL}, \varphi_{WL}, \varphi_{HL}, \varphi_{EL}, \varphi_{HTR}, \varphi_{TR}, \varphi_{WR},$

φ_{HR}, $\varphi_{ER}\}$ are defined as follows (n is the number of features in set J):

$$\sum_{i=1}^{n} R(i) = \sqrt{(J(i)_x - O_x)^2 + \left(J(i)_y - O_y\right)^2 + (J(i)_z - O_z)^2} \tag{1}$$

$$\sum_{i=1}^{n} \Theta(i) = atan2\left(\sqrt{(J(i)_x - O_x)^2 + \left(J(i)_y - O_y\right)^2},\ (J(i)_z - O_z)\right) \tag{2}$$

$$\sum_{i=1}^{n} \Phi(i) = atan2\left((J(i)_x - O_x), \left(J(i)_y - O_y\right)\right) \tag{3}$$

Additionally, the data of the recorded sign must be stored in the repository without taking into account the user's body size. The different size of each user causes a significant variation to the distance from one joint to another.

After applying the normalization on the user's position, as described above, every joint in J is expressed by the relative distance to the origin O and the two angles θ and φ. Θ describes the angle between the zenith direction and the line segment OJ and Φ describes the signed angle measured from the azimuth reference direction to the orthogonal projection of the line segment OJ on the reference plane (Fig. 5).

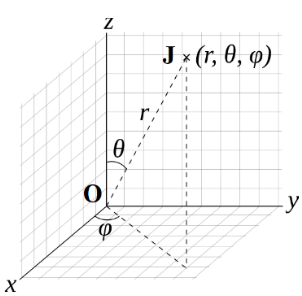

Figure 5: Spherical coordinates of a joint J.

In order to deal with that problem, the radial distance OJ will be normalized by using the distance between the head joint and the middle of the spine (r_{HO}) as proposed in [8]. This factor tells about the size of the user and can be used to solve the problem when taller users have greater distance between their head and the middle of the spine than shorter.

Given the set of radial distances $R = \{r_{HTL}, r_{TL}, r_{WL}, r_{HL}, r_{EL}, r_{HTR}, r_{TR}, r_{WR}, r_{HR}, r_{ER}\}$, the normalized set of distances R_{norm} is obtained as follows:

$$\sum_{i=1}^{n} R_{norm}(i) = \frac{R(i)}{r_{HO}} \tag{4}$$

where n is the number of radial distances from R and r_{HO}. There is no need to normalize the angles θ and φ since the user's size does not affect angles.

3.4 Data Storage

Once the data for each sign are obtained and normalized, signs are stored to the repository in a specific unique way. Three files are generated, one for each spherical coordinate R, Θ and Φ. Each file contains information for every frame for each of the ten selected tracked joints.

3.5 Sign Classifier

The sign classifier will classify the test sample (translation mode) with the closest one from the repository and will output the recognized word. The problem here is that the two compared signs do not have the same number of frames.

3.5.1 Nearest-Neighbor with Dynamic Time Warping. The Nearest-Neighbor classifier is used with the dynamic time warping algorithm employed for trajectory alignment (Fig. 6). The recorded test sign is classified with the most similar single sign from the repository. The most similar sign from the training set is the one with the smallest Euclidean distance, which is calculated within the dynamic time warping algorithm.

```
function NN-DTW (file test, dictionary <file> training [1..m] )
    declare double minimum, cost
    for all i in training
        cost <- DTW(test, training[i])
        if cost < minimum then
            minimum <- cost
        end if
    end for

    return minimum
end function
```

Figure 6: Nearest-Neighbor algorithm.

3.5.2 Dynamic Time Warping. Dynamic time warping is a well-known algorithm to find an optimal alignment between two given sequences which don't have the same length, in our case two signs that are described with different number of frames. Originally, DTW has been used to compare different speech patterns in automatic speech recognition but it has also been successfully applied to automatically cope with time deformations and different speeds associated with time-dependent data. The algorithm used is shown in Fig. 7. The sequences are warped in a nonlinear fashion to match each other.

```
function DTW (file test, file training)
    declare array dtw[0..n, 0..m]
    //n,m frames of each sign in both files
    for all n in dtw
        dtw[i,0] <- infinity
    for all m in dtw
        dtw[0,i] <- infinity
    dtw[0,0] <- zero
    for all joints in test
        for all i in n
            for all j in m
                cost <- distance(test[i], training[j])
                dtw[i,j] <- cost + minimum(dtw[i-1, j  ],    // insertion
                                           dtw[i  , j-1],    // deletion
                                           dtw[i-1, j-1])    // match

    return dtw[n,m]
end function
```

Figure 7: Dynamic time warping algorithm.

3.6 Implementation Environment

The application was developed in C# programming language using the Visual Studio 2015 Community Edition, the .NET

Framework, the Kinect API and the Speech Recognition API provided by Microsoft in a machine running Windows 10. The data were saved locally in text files.

4 EVALUATION

For the evaluation of the system a sample of 5 individuals (2 male, 3 female) with different size and characteristics was assembled. All participants did not know the words of the Greek sign language prior to participation. Each user who participated in the evaluation process performed 15 signs, four times. This results in 300 samples that were used to assess the accuracy of the system. The four samples for each sign were obtained as follows: 2 were obtained from the center position, 1 from the right position and 1 from the back position.

Prior to participation in the evaluation study the participants studied and learnt how to perform the 15 signs based on videos provided by the Institute of Educational Policy [15]. They were presented with all the details, such as key points of each sign, and how to start recording a sign through speech or standard graphical interaction in both modes supported by the system.

From the 15 signs, 6 were one-handed signs and 9 were two-handed signs. Two-handed signs can be sub-categorized into 4 signs that require mirrored hand movement and 5 that require normal hand movement as shown in Table 1. Some additional criteria for the selection of the signs were considered. Signs that indicate direction are representative of the way they are performed and have everyday usage. Also, signs that are performed in a similar position e.g. 'book', 'middle' and in a vague way e.g. 'paint' were selected in order to challenge the classification. The participants performed the signs that require one hand with their dominant hand and the two-handed signs on their convenience.

Table 1: The 15 GSL signs used

Type	Signs
One-handed	key, white, paint, straight, phone, hot
Two-Handed (Normal)	left, right, middle, guitar, hammer
Two-Handed (Mirror)	camera, book, run, table

Six different configurations were evaluated. All configurations compared the spherical coordinates after being normalized as suggested, with every combination of them possible. The configurations were:

(1) Plain classification: Spherical coordinates of the signs are compared only.

(2) Weighted classification: Spherical coordinates of the signs are compared focusing on the hand joints (90% weight on hand joints and 10% weight on elbow joints).

(3) Median filtering classification: Spherical coordinates of the signs are compared after applying a median filter. After testing different window sizes, best results were achieved with a window size of 5. Mean filtering with different window sizes was also tested but it didn't improve the accuracy of the system.

(4) Classification considering less joints: Six joints (*WL*, *HL*, *EL*, *WR*, *HR* and *ER*) instead of ten spherical coordinates of the signs are compared.

(5) Classification considering two individuals instead of five.

(6) Classification considering only the center position i.e. only 150 samples of the center position are used.

The results achieved are shown in Table 2. From the analysis, several conclusions can be obtained. In general, the sensor itself provides accurate data and combined with the processing performed high recognition accuracies are obtained confirming the relevant literature. In particular in the plain classification the best accuracy is 93.00%. A better accuracy is achieved when considering the radial distance and the azimuthal angle, in combination with the median filter applied on the stored data (94.67% accuracy).

From the quantitative analysis of the accuracies obtained and regarding the spherical coordinate system, a conclusion drawn is that the radial distance and the azimuthal angle are the two spherical coordinates that change the most when performing a sign thus they describe the sign better. On the other hand, the polar angle when considered alone provides the worst results and also if combined with the other two spherical coordinates degrades the results. It is considered that the reason is because this angle has a similar behavior when performing a sign. Another observation that could be made is that using all three spherical coordinates doesn't improve the accuracy of the system.

From the qualitative analysis of the accuracies obtained across the six different configurations we can conclude to the following results. When applying weighted classification the best accuracy achieved (92.67%) is lower than the plain configuration, evidencing that elbows play a significant role when performing a sign and must be treated the same way as the hand joints. Reducing tracked joints from ten to six results in losing the detail from the movement of the thumbs and hand tips when performing a sign and lowers accuracy to 92.00%. Median filter smooths the data obtained from the Kinect sensor and raises accuracy to 94.67%. The best result when considering two individuals instead of five, is the same when comparing with raw data (93.00%). That result raised the question of using median filtering and comparing the signs of two individuals instead of five. In that case, accuracy was 94.33%.

One extra configuration was tested because it was observed that considering the radial distance alone, had higher accuracy without applying the median filter (see first row in Table 2). The additional configuration (not shown in Table 2) represented a classification which considered both radial distance without filtering and the azimuthal angle with median filtering. Results showed an accuracy of 94.00%.

The default approach used by the system is the one providing the best results i.e. radial distance and azimuthal angle after applying median filter with a window size of 5 on both spherical coordinates. From the total of 300 samples 16 are misclassified, 9 of them are performed at the 'right' position and 7 at the 'back' position. For 10 of the signs we had 100% accuracy. The 5 signs that aren't classified with absolute accuracy can be seen in Table 3.

Table 2: Accuracies for the different configurations

Spherical Coordinates	Plain	Weighted	Median Filter	Reduced Joints	Reduced Samples	Center Only
Radial Distance (RD)	89.00%	86.00%	87.33%	89.00%	91.33%	94.67%
Zenith Angle (ZA)	81.67%	81.00%	80.00%	75.00%	81.33%	92.67%
Azimuth Angle (AA)	87.67%	87.67%	90.00%	88.33%	88.00%	98.00%
RD + ZA	89.00%	89.00%	89.00%	87.33%	88.33%	98.67%
RD + AA	**93.00%**	**92.67%**	**94.67%**	92.00%	**93.00%**	97.33%
ZA + AA	90.67%	89.67%	91.33%	89.00%	89.33%	98.00%
RD + ZA + AA	92.00%	92.33%	92.67%	**92.67%**	91.00%	**99.33%**

From the observations we made, we found out that the signs that are performed in similar position relatively to the *O* with another sign are more likely to be wrongly classified.

Table 3: Probabilities of misclassified GSL signs

	book	key	white	paint	straight	middle	phone	hot	run
book	0,8					0,15			0,05
key		0,8	0,05				0,15		
white		0,05	0,9				0,05		
paint				0,8	0,05			0,15	
straight				0,05	0,9			0,05	

Finally, if we consider only the center position during the classification, the accuracy raises to 99.33% using all three coordinates. From the 150 samples only one is misclassified. Given the fact that this is the configuration that is the most tested in the related literature underlines the usefulness of our implementation. However it should be noted that this configuration may cancel the normalization performed with respect to user's position.

5 CONCLUSIONS

By combining the Kinect technology with the Nearest-Neighbor classifier enhanced with dynamic time warping for trajectory alignment and median filtering, the presented system is able to recognize 15 words of the Greek sign language performed by different users. Different configurations were evaluated. A recognition rate of 94.67% is achieved when the position of the subject can vary and 99.33% accuracy is achieved when the subject is standing at a standard position.

Future work will concern expanding the number of signs that can be recognized by the system and testing the system with more individuals without lowering the accuracy. Facial expressions play vital role in sign language and Kinect v2 introduced face tracking. An upgrade would be to obtain data regarding facial expressions so that each sign is modeled more precisely and hopefully raising the accuracy of the system in more challenging environments.

ACKNOWLEDGMENTS

We would like to thank the 5 participants of the pilot study for their contribution to the evaluation process.

REFERENCES

[1] World Health Organization. Deafness and hearing loss. Fact sheet, Updated February 2017. Available on-line: http://www.who.int/mediacentre/factsheets/fs300/en/

[2] Stokoe, W. C. (2005). Sign language structure: An outline of the visual communication systems of the American deaf. Journal of deaf studies and deaf education, 10(1), 3-37.

[3] Han, J., Shao, L., Xu, D., & Shotton, J. (2013). Enhanced computer vision with microsoft kinect sensor: A review. IEEE transactions on cybernetics, 43(5), 1318-1334.

[4] Biswas, K. K., & Basu, S. K. (2011, December). Gesture recognition using microsoft kinect®. In Automation, Robotics and Applications (ICARA), 2011 5th International Conference on (pp. 100-103). IEEE.

[5] Oliver, A., Kang, S., Wünsche, B. C., & MacDonald, B. (2012, November). Using the Kinect as a navigation sensor for mobile robotics. In Proceedings of the 27th Conference on Image and Vision Computing New Zealand (pp. 509-514). ACM.

[6] Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., & Zhou, M. (2013, April). Sign language recognition and translation with kinect. In IEEE Conf. on AFGR.

[7] Yang, H. D. (2014). Sign language recognition with the kinect sensor based on conditional random fields. Sensors, 15(1), 135-147.

[8] Capilla, D. M. (2012). Sign Language Translator using Microsoft Kinect XBOX 360 TM. Department of Electrical Engineering and Computer Science, University of Tennessee.

[9] Oszust, M., & Wysocki, M. (2013, June). Polish sign language words recognition with kinect. In Human System Interaction (HSI), 2013 The 6th International Conference on (pp. 219-226). IEEE.

[10] Fernando, P., & Wimalaratne, P. (2016). Sign Language Translation Approach to Sinhalese Language. GSTF Journal on Computing (JoC), 5(1), 52.

[11] Lang, S., Block, M., & Rojas, R. (2012). Sign language recognition using kinect. In Artificial intelligence and soft computing (pp. 394-402). Springer Berlin/Heidelberg.

[12] Wu, J., Tian, Z., Sun, L., Estevez, L., & Jafari, R. (2015, June). Real-time American sign language recognition using wrist-worn motion and surface EMG sensors. In Wearable and Implantable Body Sensor Networks (BSN), 2015 IEEE 12th International Conference on (pp. 1-6). IEEE.

[13] Starner, T., Weaver, J., & Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(12), 1371-1375.

[14] Tashev, I. (2013). Kinect development kit: A toolkit for gesture-and speech-based human-machine interaction [best of the web]. IEEE Signal Processing Magazine, 30(5), 129-131.

[15] Institute of Educational Policy. Available on-line: http://prosvasimo.gr/el/onlne-lexiko-ennoiwn