

Trajectory based Arabic Sign Language Recognition

Ala addin I. Sidig, Sabri A. Mahmoud
Collage of Computer Science and Engineering
King Fahad University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia

Abstract—Deaf and hearing impaired people use their hand as a tongue to convey their thoughts by performing descriptive gestures that form the sign language. A sign language recognition system is a system that translates these gestures into a form of spoken language. Such systems are faced by several challenges, like the high similarities of the different signs, difficulty in determining the start and end of signs, lack of comprehensive and bench marking databases. This paper proposes a system for recognition of Arabic sign language using the 3D trajectory of hands. The proposed system models the trajectory as a polygon and finds features that describes this polygon and feed them to a classifier to recognize the signed word. The system is tested on a database of 100 words collected using Kinect. The work is compared with other published works using publicly available dataset which reflects the superiority of the proposed technique. The system is tested for both signer-dependent and signer-independent recognition.

Keywords—Trajectory processing; sign language recognition; ensemble classifier; polygon description; parameters tuning; signer independent

I. INTRODUCTION

Communicating thoughts and feelings is an essential need for human beings. Hearing disabilities hinder the natural speech based communication. To communicate with each other and with speaking people, deaf has invented nonverbal languages that use descriptive gestures to convey their thoughts. These languages are developed by the deaf communities in different regions of the world. Sign languages are full featured languages with their own vocabularies and grammar. They make use of hands-motion, fingers-configurations, facial-expressions, and body lane in parallel to express different terms. Unfortunately, speaking people find it hard to learn these languages which increases the barrier between them and the deaf community. To communicate with deaf, speaking people need skilled professional translators that knows the spoken and signed languages. These skilled translators are few and can't be available all the time. Sign language recognition systems tries to fill this gap by exploiting the advanced technologies to automatically translate signed language to a form of spoken language such as text or speech. To effectively translate a signed language all its components need to be considered. Of these components, the hands-motion is one of the most important modalities of signed language.

This work proposes to use the 3D trajectory of hands to recognize signs. The 3D trajectory, in contrast to 2D, provides information about the front-back hand motion. We record both of 2D and 3D trajectory using Kinect device. The proposed system is composed of three stages: Preprocessing, Features representation, and Classification. The preprocessing stage removes the noise and compresses the trajectory to form

a polygon. The compression is done by finding N key points that represent the polygon corners. The feature representation stage builds a features vector that describes this polygon. These features are used to train and test different classifiers to recognize the signs in the third stage. Fig. 1 shows the pipeline of the proposed system. The main contributions of this work are:

- Propose a trajectory based sign language recognition system.
- Propose a trajectory compression algorithm.
- Propose a two features representations for 2D and 3D trajectories applied to signer dependent and independent recognition.

The rest of this article is organized as follows: Section II describes some of the related works. Followed by Section III on the trajectory preprocessing. Features representation is described in Section IV. Then Section V on classification. Experimental evaluation is shown in Section VI. Finally, we conclude this article in Section VII.

II. RELATED WORKS

Arabic sign language recognition is addressed by many researchers using different scales and strategies. The work on Arabic sign language recognition in the literature can be classified into three levels. Arabic sign alphabets and numbers recognition level [1]–[4], isolated words recognition level [5]–[9], and sentences recognition level [10]–[12]. This work proposes a system for isolated words recognition based on hands' trajectories. Trajectory processing exists in a wide range of applications. Therefore, a lot of work is done on trajectory processing in on-line character recognition [13], [14], action recognition [15], [16], gesture recognition [17] and more.

Lin and Hsieh in [18] proposed a kernel based trajectory representation using Kernel Principal Component Analysis (KPCA) and Nonparametric Discriminant Analysis (NDA). In their method a 2D/3D trajectory is first min-max normalized then projected to higher dimensional space using KPCA. The dimensionality is reduced using NDA with the hope of maximizing the interclass variability and minimizing the within-class variability. The resulting representation is hoped to be more discriminative. The classification is done using the nearest neighbor rule. The approach is tested on a limited set of 38 words from the Australian sign language and reported accuracy of 69% for 2D trajectory and 78% for 3D.

Naftel and Khalid in [19] encoded the 2D trajectory along x and y dimensions using Discrete Fourier Transform (DFT)

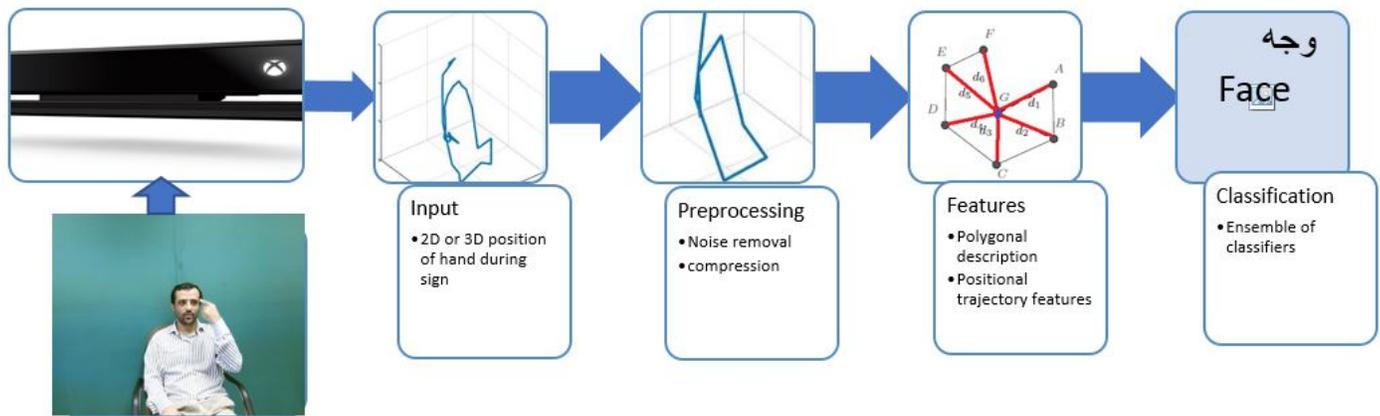


Fig. 1. The stages of the proposed system. The signer performs the sign in front of the Kinect which tracks the hands. The resulting trajectory is then fed to the system.

separately. Then the first four coefficients are used as feature vector that represent the trajectory. The coefficients are then clustered using Self Organized Map (SOM). They tested the approach on 24 words from Australian sign language and reported an accuracy of 70.1%.

Pu et al. [20] modeled the trajectory as a sequence of M sub-motions and used HMM to model the transition between these sub-motions. For each point on the sub-motion trajectory, they find the *shape context* as a histogram of relative coordinates of other points on the sub-motion trajectory. Then a codebook is generated from these shape contexts. The features vector of each sub motion curve is composed as a weighted histogram of the code book centers. The weights are found by soft clustering the shape context of each point. Finally the sign curve feature is a sequence of M sub-motion features. They tested the system on a database of 100 signs from the Chinese sign language and reported an accuracy of 67.3% for signer dependent and 54.4% for signer independent.

Boulares in [21] extracted signatures from 3D hands trajectories and used SVM to classify different signs. To extract trajectory signature, they used non linear regression to fit the trajectory points to a conic section. The trajectory signature along with hand shape and other features is used to train and test SVM classifier. Curve fitting does not accurately represent complex trajectories that include cycles.

Geng et al. in [22] used a combination of trajectory modeling and hand shape representation as a feature to train an Extreme Learning Machine (ELM) classifier. A combination of 3D trajectories of hand, wrist, and elbow are used. They normalized the values of trajectory points to $[0, 1]$ range and smoothed the trajectory by average convolution. To form a feature vector from the smoothed trajectory, they subtract the starting point of the trajectory from all following points. The difference between the hand trajectory and wrist trajectory is represented by spherical coordinates system and similarly for the hand-elbow trajectory difference. The final features vector is concatenation of hand trajectory, hand-wrist spherical difference, hand-elbow spherical difference, and hand shape features from depth image. These features are used to train ELM and 82.8% accuracy is reported on a limited database of 8 words from the Chinese sign language. Normalization

of trajectory points to the range of $[0, 1]$ results in loss of information about where was the hand motion with respect to body when signing the word.

Wang et al. in [23] formed the trajectory of hands as a combination of hands location and orientation. The hand location is defined as the hand location with respect to the face centroid and with respect to the non dominant hand location. Similarly, the orientation is defined as the direction between successive hand locations. For single handed signs the trajectory of non dominant hand is set to zeros. All trajectories are normalized to have the same length. Similarities between trajectories are measured by dynamic time wrapping (DTW). Based on the trajectory matching the top 10 accuracy of the sign search results is about 74% and was improved to 78% when incorporating additional hand shape feature. They slightly modified the trajectory feature in [24] by including the hand velocity and defining separate feature for single handed signs doesn't include the hand location with respect to non dominant hand location. However the information of single or two handed sign need to be given by the user.

Bhuyan et al. in [25] modeled the trajectory as a combination of shape and motion features. The shape features include, the trajectory length, and the number of curves in the trajectory. The motion features include, the average speed, standard deviation of the speed, and the number of minima in the velocity. The classification of gestures is done in two stages. First candidate signs are included based on the trajectory shape similarities using maximum boundary deviation as similarity measure. In the second stage trajectories are aligned using DTW then the trajectory features are classified based on the nearest candidate template.

Mohandes and Deriche proposed a system for Arabic sign language recognition [26]. The trajectory is composed of 3D position and orientation with 12 dimensional vector for both hands. For each dimension the acquired readings are partitioned into 5 equal partitions. From each partition the mean and standard deviation is calculated. That results in 120 dimensional features vector. LDA is used to reduce the dimensionality to 20. The nearest neighbor classifier is used to find the class of a sign. They reported an accuracy of 84.7% on a dataset of 100 words.

III. PREPROCESSING

In this work, Kinect is used to record signs. A synchronized color image, depth image, and 25 body joints locations are recorded. For each joint the 3D locations of joints and the 2D mapping to both color and depth images are recorded. For this work the sequence of hands locations in 3D is used to recognize signs.

Trajectory preprocessing includes: Noise removal and Compression. The joints' locations obtained by Kinect are noisy and include some outliers. The noise removal stage smooths out these outliers by using median filter. Since the frame-rate for recording is at 30 frames per second, fine details of part of second trajectory is not very useful and results in redundant information. Trajectory compression stage compresses the trajectory into few key points. To find such key points the trajectory is treated as a polygon formed by connecting the locations of the hand while signing. The key points are obtained by reducing the number of vertices of this polygon to a specific number. The reduction is done by recursively calculating the importance of each vertex based on angle and segment length and then removing the least important. The process is repeated until the desired number of vertices is reached. Fig. 2 shows the calculation of vertex importance. The algorithm for trajectory compression is shown in Algorithm 1. Fig. 3 shows the effect of 3D trajectory preprocessing. The preprocessing of a 2D trajectory is shown in Fig. 4.

Algorithm 1 TrajectoryCompression

```

1: procedure COMPRESS(Traj, NumVers)
2:   TrajLength ← LENGTH(Traj)
3:   for all points v in Traj do
4:     IMP(v) ← CALCIMPOTRANCE(Traj, v, NumVers)
5:   end for
6:   while TrajLength > NumVers do
7:     I ← IndexOfMin(IMP)
8:     Traj ← Traj − Traj(I)    ▷ The - sign is set
    difference
9:     IMP ← IMP − IMP(I)
10:    TrajLength ← TrajLength − 1
11:    update IMP by recomputing the importance of the
    removed vertex's neighbors.
12:   end while
13:   function CALCIMPOTRANCE(Traj, v, NumVertices)
14:     imp ←  $D_{vp} \times D_{va} \times \Theta$ 
15:     return imp
16:   end function
17: end procedure

```

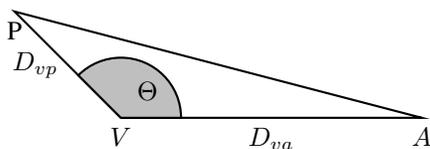


Fig. 2. The importance calculation for vertex V is found by multiplying the distances from v to adjacent vertices P (previous), A (after) and the angle Θ as $IMP_v = D_{vp} \times D_{va} \times \Theta$.

Some of the previous works as stated in Section II include another stage in preprocessing called min-max normalization.

In this stage the trajectory is normalized to be in [0-1] range. In this work, such stage is excluded arguing that it leads to loss of discriminative features. Signs can have similar trajectory pattern but at different locations. Min-max normalization leads to loss of the localization feature of the trajectory.

IV. FEATURES REPRESENTATION

After noise removal and compression, features are extracted from each sign trajectory. Here we describe two types of features.

A. Polygon Description

In this method the 3D hand trajectory is represented as a polygon. The description of this polygon is represented by: its center of gravity and the distances from the perimetric points to the center of gravity point. The center of gravity point is approximated by the mean of perimetric points calculated as $G = (\bar{x}, \bar{y}, \bar{z})$ where $\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i$ and N is the number of perimetric points. The distance from G to perimetric points is calculated using the Euclidean distance formula $d_i = \|G - P_i\|, i = 1, 2, 3, \dots, N$. Fig. 5 illustrates the polygon description procedure.

Then the polygonal description feature is formed by concatenating G and d_i as

$$F = [\bar{x}, \bar{y}, \bar{z}, d_1, d_2, d_3, \dots, d_n]$$

This feature representation captures both of the trajectory shape and more importantly the position of hand motion. The position of hand motion is important as it distinguishes between signs with similar trajectories but different body positions.

B. Positional Trajectory Feature

In this feature representation only perimetric points of the trajectory polygon are included. The feature vector is a concatenation of perimetric points formed as

$$F2 = [x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N]$$

This feature representation although is simple, but has shown very good discrimination and generalization as will be shown in the experimental results section.

V. CLASSIFICATION TECHNIQUES

After preprocessing and features representation of all trajectories at hand, features are used to train and test classifiers. In this work several classifiers are tested and the best accuracy is obtained when using ensemble of classifiers. Specifically, the best performing classifier is Ensemble Subspace KNN. The tested classification algorithms are listed in Table I. We use five folds cross validation.

In subspace ensemble algorithm, a set of N weak learners each is trained on a randomly chosen partition of the features vector of M dimensions less than the D dimensions of the original feature vector. On prediction, the average score from weak learners is calculated and the class with the highest average score is chosen as the true class [27]. This work used KNN as a weak learner to build the ensemble subspace

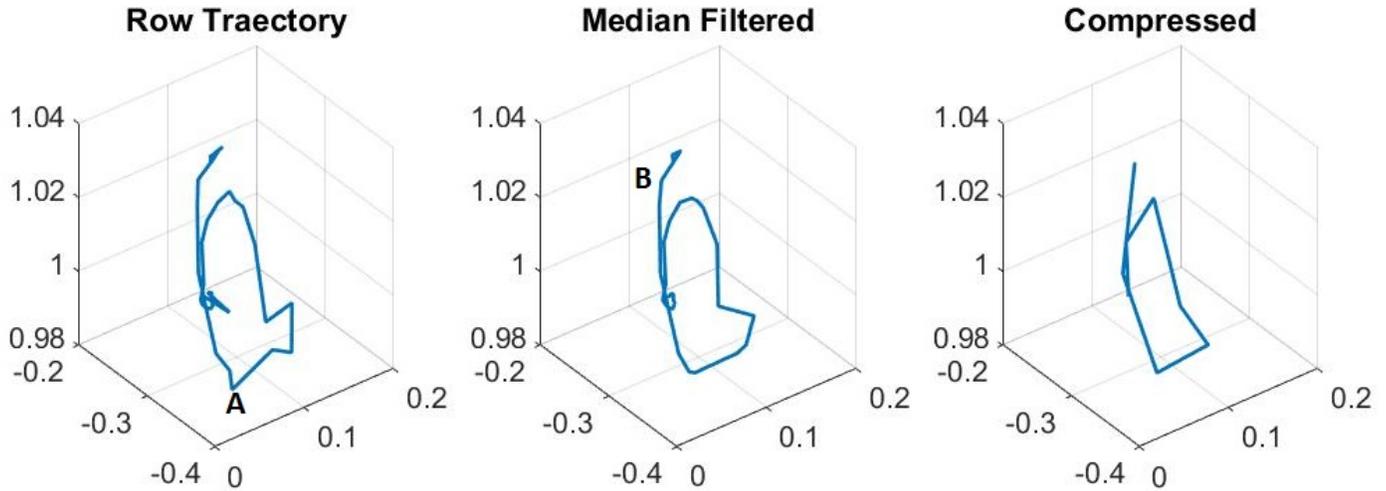


Fig. 3. The preprocessing stage of the trajectory. 'A' is a noisy point smoothed out by the median filter. 'B' is a less important point removed by compression stage.

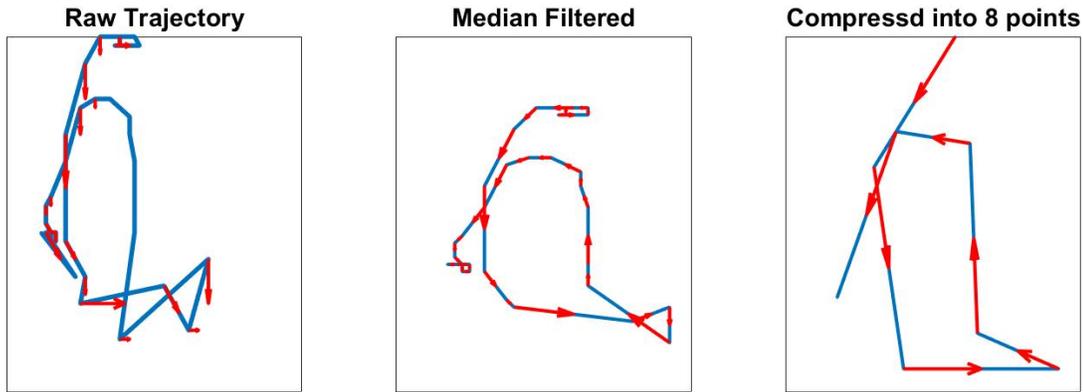


Fig. 4. The preprocessing stage of a 2D trajectory. The median filter reduce the noise of the trajectory resulting in smoother one. The compression stage finds the most important 8 points in the trajectory. The arrows indicate the direction of motion.

TABLE I. LIST OF CLASSIFIERS USED IN THE EXPERIMENTS

Tree	Linear Discriminant	Quadratic Discriminant
SVM Linear	SVM Quadratic	SVM Cubic
SVM Gaussian	KNN Euclidean	KNN Cosine
KNN Cubic	Ensemble Boosted Trees	Ensemble Bagged Trees
Ensemble Subspace Discriminant	Ensemble Subspace KNN	Ensemble RUSBoosted Trees

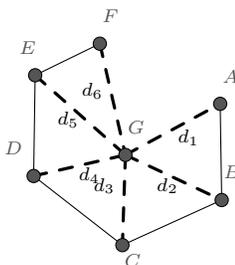


Fig. 5. The polygon description feature is found by the center of gravity G and distances $[d_1, d_2, d_3, d_4, d_5, d_6]$ form G to perimetric points $[A, B, C, D, E, F]$ respectively.

classifier. It is clear that N, M and K (of the KNN) are hyper parameters that need to be chosen for best performance of the classifier. To find the best values for these parameters cross validation is used as shown by Algorithm 2.

The algorithm first runs KNN with different values of K to find the best performing one (BestK). Then it fixes the number of weak classifiers to 100 and K to BestK and searches for the best number of partitions, BestM. With BestK and BestM the algorithm then searches for best number of weak learners BestN.

VI. EXPERIMENTAL RESULTS

A set of experiments are carried out to evaluate each stage of the proposed system. Starting by the preprocessing stage to

Algorithm 2 Fine Tune parameters of Ensemble Subspace KNN

```
1: function FINETUNE(SampleFeatures)
2:    $S \leftarrow$  NumberOfSamples
3:    $D \leftarrow$  NumberOfDimensions
4:    $KCandidates \leftarrow$  set of 10 values between 1 and  $\log S$ 
5:   for all  $k$  in  $KCandidates$  do
6:      $Loss(k) \leftarrow$  CROSSVALIDATEKNN( $SampleFeatures, KCandidates[k]$ )
7:   end for
8:    $BestK \leftarrow KCandidates[MinimalLoss]$  ▷ Find the best K
9:    $MCandidates \leftarrow$  set of 10 values between 1 and  $D$ 
10:   $N \leftarrow 100$  ▷ Fixed Number of weak classifiers
11:  for all  $m$  in  $MCandidates$  do
12:     $Loss(m) \leftarrow$  CROSSVALIDATEENSEMKNN( $SampleFeatures, MCandidates[m], BestK, N$ )
13:  end for
14:   $BestM \leftarrow MCandidates[MinimalLoss]$  ▷ Find the best M
15:   $NCandidates \leftarrow$  set of 100 values between 1 and 100
16:  for all  $n$  in  $NCandidates$  do
17:     $Loss(m) \leftarrow$  CROSSVALIDATEENSEMKNN( $SampleFeatures, BestM, BestK, NCandidates[n]$ )
18:  end for
19:   $BestN \leftarrow NCandidates[MinimalLoss]$  ▷ Find the best N
20:  return  $BestK, BestM, BestN$ 
21: end function
```

the classification stage to fine tune the hyper parameters and then test the sign language recognition.

A. Arabic Sign Language Dataset

To our knowledge, there is no public dataset for Arabic sign language, so we collected a dataset of 100 words from the health chapter of Arabic sign language dictionary [28]. The dataset is recorded using Kinect to record synchronized color video, depth data, and 25 skeletal joints of body. The dataset was recorded by 3 signers repeated each sign 50 times on different sessions. For this work, only the hands joints' trajectories are employed to recognize signs. A list of the words in this database are shown in Table VI.

B. Effect of Trajectory Compression

This section investigates the effect of the number of vertices used to represent the trajectory as a polygon on the accuracy. This experiment used the trajectories of all signs performed by one signer and apply the preprocessing stage by varying the number of vertices from 4 to 18. Fig. 6 shows the classification error rates for different representations of the trajectory features. In this figure, F1 represent the *polygon description* feature representation of trajectory (see Section IV-A) while F2 stands for the *positional trajectory* feature representation. The 1H and 2H encodes the usage of only one hand trajectory or both hands respectively in building the feature vector. In 1H the features encode only the trajectory of the dominant hand while in 2H a concatenation of features that encode both hand trajectories is used. The 2D and 3D for which trajectory points representation being used, X-Y or X-Y-Z respectively. From this figure, many properties can be inferred. First, the best average accuracy can be obtained when using a polygon with 12 vertices. Using small number of vertices does not capture the complex trajectories well, and using very high number of vertices includes noisy details that mix up distinct classes. Second, the usage of 3D trajectory

always performs better than the 2D one. This can be attributed to the fact that the Z dimension captures front-back motion of hands, and there are some signs in the database with only front-back motion pattern. Third, the inclusion of non-dominant hand in the feature representation increases the discrimination power. The state of non-dominant hand in sign language can either be static, mirrors the motion of dominant hand, or moving in different way than the dominant hand. In all cases of non-dominant state, its motion pattern helps in distinguishing similar signs that are of similar dominant hand trajectory. Forth, as a comparison between the two features representation the *positional trajectory* feature representation outperforms the *polygon description* feature representation of the trajectory.

C. Fine Tuning EnsembleSupspaceKNN Classifier

This experiment applied Algorithm 2 on the same set used in Section VI-B to find the best parameters for each feature representation. Table II lists the best parameters' settings for each feature representation. In this table the best value for K is 1 for all features, the best value for M for feature F1 is roughly half D which is similar to the findings in [27]. The values in BestN column are for the value of N after which no significant drop in loss is seen. Based on this table, the parameters settings for following experiments will be: K=1, N=40, M= BestM from the table.

TABLE II. BEST PARAMETERS FOR ENSEMBLESUPSPACEKNN CLASSIFIER

Feature	D	BestK	BestM	BestN
F1-1H-2D	14	1	8	40
F1-1H-3D	15	1	9	25
F1-2H-2D	28	1	13	40
F1-2H-3D	30	1	14	25
F2-1H-2D	24	1	6	40
F2-1H-3D	36	1	9	40
F2-2H-2D	48	1	6	40
F2-2H-3D	72	1	9	40

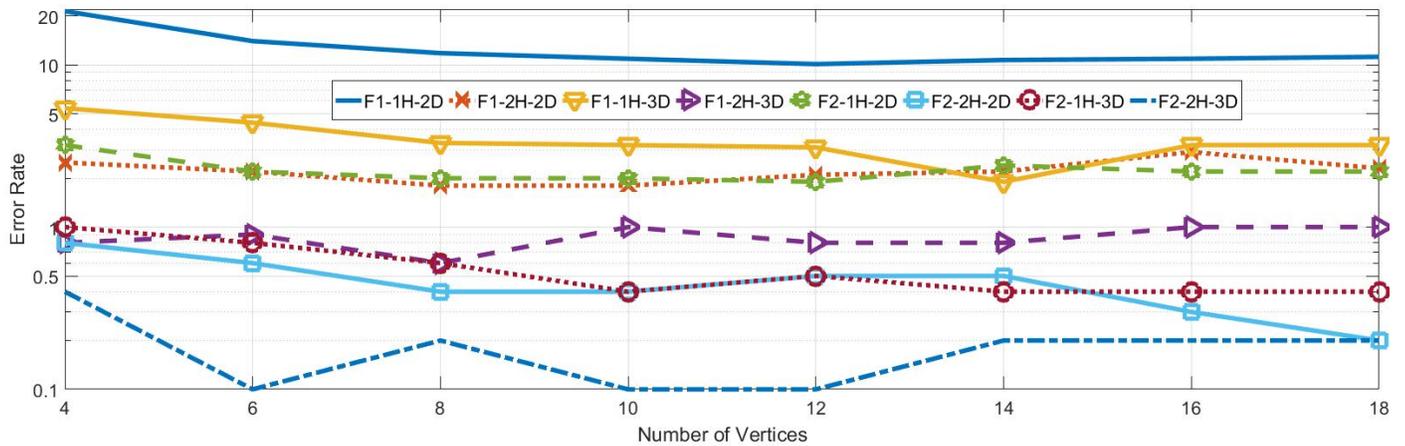


Fig. 6. The compression effect on the classification error rate for different versions on the proposed features representation. The Y axis is log scaled for better visualization.

D. Evaluation of the Proposed Features

After choosing the best trajectory compression ratio and the best parameters settings for the classifier, the system is tested on the collected database. Table III lists the recognition rates obtained when using each feature representation for each signer in the database. The results reflect that the 3D trajectory is more informative and discriminative than the 2D one, and the inclusion of non dominant hand status improves the accuracy for both types of trajectories. The third signer shows better accuracies than the other two which can be attributed to the less variability in his performance of signs, and the samples used for fine tuning the hyper parameters are performed by him. The fifth column lists the accuracies when using mixed samples from all signers for both training and testing. This shows the scalability of the system to larger number of samples and different signers.

TABLE III. SIGNER DEPENDENT CLASSIFICATION RECOGNITION RATE

Feature	Signer1	Signer2	Signer3	All Signers
F1-1H-2D	89.8	88.8	91.5	84.4
F1-1H-3D	96.2	95.0	97.6	94.7
F1-2H-2D	97.6	96.8	97.9	96.4
F1-2H-3D	99.3	98.8	99.4	99.5
F2-1H-2D	97.7	96.0	98.2	95.6
F2-1H-3D	99.2	99.0	99.8	99.2
F2-2H-2D	99.5	98.9	99.7	99.1
F2-2H-3D	99.7	99.6	100	99.7

Although the number of signers is not big enough to evaluate the system for signer independent recognition, experiments are done to get initial intuition about the generalization of the system to unseen signer. Table IV lists the accuracies of the different types of features in signer independent mode. Each column is named by the test signer when the training is done by samples performed by the other two signers. The lower results of the second signer are due to the different signing style, some signs are repeated more than once in the same sample. Overall average performance is around 53% for all features 48%, and 57% for F1, and F2 features, respectively.

TABLE IV. SIGNER INDEPENDENT CLASSIFICATION RECOGNITION RATE

Feature	Signer1	Signer2	Signer3
F1-1H-2D	40.1	27.2	43.1
F1-1H-3D	44.6	30.7	50.9
F1-2H-2D	57.5	48.6	60.1
F1-2H-3D	60.0	51.0	64.8
F2-1H-2D	56.8	41.9	58.7
F2-1H-3D	60.2	43.2	65.7
F2-2H-2D	58.9	47.8	63.9
F2-2H-3D	61.3	49.7	64.4

E. Comparison with Published Work

This experiment tests the proposed features representation and classification algorithm on a publicly available dataset and compares the results of the proposed method with published work on the same dataset. The dataset is composed of 95 Australian sign language words. Each word is performed by 1 signer 27 times. For each sample a vector of 22 measures is recorded per frame. These measures include the 3D position of hands (X,Y,Z), the orientation of hands (Roll, Pitch, Yaw), and the status of fingers. Some previous work used only the (x,y) points to form 2D trajectory while others used 3D. This work, uses the 2D/3D trajectory as well as the hand orientation. The same steps of trajectory preprocessing, features representation, and classification are applied on this database. In this dataset, the signer starts with his hands on the rest position and return them back to the rest position after signing. This makes the center of gravity of some signs to be the same. To avoid that, the compression stage is applied twice. First with 14 vertices which will include the starting and ending rest position. Then it finds the 12 vertices after excluding the first and last points which results in removing the rest position from the calculation of the center of gravity. Table V shows the accuracy reported by different previous works along with our work (the last 4 lines). The first row shows the number of classes out of 95 used. In this table, F1 stands for the polygonal description feature representation and F2 for the positional feature. 3D stands for the only use of 3D hand position to form the feature while 3DO for inclusion of the hand orientation too.

Note that the work in [33] uses the 22 features while ours use three - in case of 3D feature - or six - in case of 3DO -

TABLE V. COMPARISON WITH PUBLISHED WORK ON AUSLAN

Reference	2 Words	4 Words	8 Words	16 Words	29 Words	38 Words	All Words
Khalid et al. [29] 2D	98%	92%	88%	83%	-	-	-
Khalid et al. [30] 2D	99%	95%	92%	88%	-	-	-
Bashir et al. [31] 2D	96%	92%	86%	78%	69%	66%	-
Wu et al. [32] 3D	93%	89%	83%	-	-	-	-
Naftel et al. [19] 2D	96%	90%	82%	76%	-	-	-
Wei et al. [18] 2D	98%	93%	86%	78%	72%	69%	-
Wei et al. [18] 3D	99%	96%	92%	89%	82%	78%	-
Simao et al. [33]	-	-	-	-	-	-	86.7%
F1-2D	100%	100%	95.4%	76.7%	63.7%	58.8%	46.8%
F1-3D	100%	100%	98.1%	90.4%	76.4%	70%	58.3%
F1-3DO	100%	100%	99.1%	95.4%	89.5%	86.5%	82.8%
F2-2D	100%	100%	96.3%	85.2%	74.3%	68.8%	61.7%
F2-3D	100%	100%	99.1%	94.8%	86.2%	79.7%	74.5%
F2-3DO	100%	100%	98.1%	95.9%	92.8%	88.7%	88.4%

of them. It is included to compare with a work that examined the whole database. Although it shows better performance than some of the proposed features, yet it uses more measures that are not related to the hand trajectory. The proposed system features lower dimensionality and simplicity.

VII. CONCLUSIONS

This work proposes a system for Arabic sign language recognition based on the trajectories of hands. It models the trajectory as a polygon and proposes two polygonal description features. The system shown good performance for both signer dependent and signer independent recognition. The accuracy of the system reaches 99% for signer dependent and 64% for signer independent recognition. The proposed system is tested on two different datasets and is compared with published works that use the same dataset and shown better performance than most of them. The proposed system features simplicity, scalability, and generalization to unseen signer. The work in database collection is still in progress to extend the vocabulary size and number of signers.

ACKNOWLEDGMENT

The authors like to acknowledge the support provided by King Fahd University of Petroleum and Minerals (KFUPM) for funding this work through project number IN151008.

REFERENCES

- [1] A. Tharwat, T. Gaber, A. E. Hassanien, M. Shahin, and B. Refaat, "Sift-based arabic sign language recognition system," in *Afro-european conference for industrial advancement*. Springer, 2015, pp. 359–370.
- [2] N. El-Bendary, H. M. Zawbaa, M. S. Daoud, A. E. Hassanien, and K. Nakamatsu, "Arslat: Arabic sign language alphabets translator," pp. 590–595, 2010.
- [3] K. Assaleh and M. Al-Rousan, "Recognition of arabic sign language alphabet using polynomial classifiers," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 2136–2145, 2005.
- [4] M. F. Tolba, M. Abdellwahab, M. Aboul-Ela, and A. Samir, "Image signature improving by pcnn for arabic sign language recognition," *Can. J. Artif. Intell. Mach. Learn. Pattern Recognit.*, vol. 1, no. 1, pp. 1–6, 2010.
- [5] A. A. Ahmed and S. Aly, "Appearance-based arabic sign language recognition using hidden markov models," in *Engineering and Technology (ICET), 2014 International Conference on*. IEEE, 2014, pp. 1–6.
- [6] A. S. Elons, M. Abull-Ela, and M. F. Tolba, "A proposed pcnn features quality optimization technique for pose-invariant 3d arabic sign language recognition," *Applied Soft Computing*, vol. 13, no. 4, pp. 1646–1660, 2013.
- [7] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, "A signer-independent arabic sign language recognition system using face detection, geometric features, and a hidden markov model," *Computers & Electrical Engineering*, vol. 38, no. 2, pp. 422–433, 2012.
- [8] T. Shanableh and K. Assaleh, "Arabic sign language recognition in user-independent mode," pp. 597–600, 2007.
- [9] M. Al-Rousan, K. Assaleh, and A. Talaa, "Video-based signer-independent arabic sign language recognition using hidden markov models," *Applied Soft Computing*, vol. 9, no. 3, pp. 990–999, 2009.
- [10] M. F. Tolba, A. Samir, and M. Aboul-Ela, "Arabic sign language continuous sentences recognition using pcnn and graph matching," *Neural Computing and Applications*, vol. 23, no. 3–4, pp. 999–1010, 2013.
- [11] K. Assaleh, T. Shanableh, M. Fanaswala, H. Bajaj, and F. Amin, "Vision-based system for continuous arabic sign language recognition in user dependent mode," in *Mechatronics and Its Applications, 2008. ISMA 2008. 5th International Symposium on*. IEEE, 2008, pp. 1–5.
- [12] K. Assaleh, T. Shanableh, M. Fanaswala, F. Amin, H. Bajaj *et al.*, "Continuous arabic sign language recognition in user dependent mode," *Journal of Intelligent learning systems and applications*, vol. 2, no. 01, p. 19, 2010.
- [13] C.-L. Liu, I.-J. Kim, and J. H. Kim, "Model-based stroke extraction and matching for handwritten chinese character recognition," *Pattern Recognition*, vol. 34, no. 12, pp. 2339–2352, 2001.
- [14] M. Kherallah, L. Haddad, A. M. Alimi, and A. Mitiche, "On-line handwritten digit recognition based on trajectory and velocity modeling," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580–594, 2008.
- [15] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.
- [16] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314.
- [17] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A hidden markov model-based continuous gesture recognition system for hand motion trajectory," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.
- [18] W.-Y. Lin and C.-Y. Hsieh, "Kernel-based representation for 2d/3d motion trajectory retrieval and classification," *Pattern Recognition*, vol. 46, no. 3, pp. 662 – 670, 2013.
- [19] A. Naftel and S. Khalid, "Motion trajectory learning in the dft-coefficient feature space," in *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, Jan 2006, pp. 47–47.
- [20] J. Pu, W. Zhou, J. Zhang, and H. Li, "Sign language recognition based on trajectory modeling with hmms," in *International Conference on Multimedia Modeling*. Springer, 2016, pp. 686–697.
- [21] M. Boulares and M. Jemni, "3d motion trajectory analysis approach to improve sign language 3d-based content recognition," *Procedia Computer Science*, vol. 13, pp. 133–143, 2012.

TABLE VI. LIST OF THE WORDS IN THE DATABASE

Sign	ID	Sign	ID	Sign	ID	Sign	ID
mongoloid	146	diabetes	121	anesthetist	96	Skeleton	71
bacterium	147	heart attack	122	surgery	97	Skull	72
microbe	148	cancer	123	dressing	98	Back bone	73
virus	149	aids	124	tape	99	Chest	74
spread	150	hair loss	125	adhesive			
hindrance	151	heart failure	126	pharmacy	100	system	75
hindrance	152	hemiplegia	127	blood analysis	101	Respiratory	
mentality		paralysis	128	physical examination	102	Trachea	76
acاعة جسدية	153	pressure	129	sight test	103	Lungs	77
acاعة بصرية	154	allergy	130	thermometer	104	Inspiration -	78
acاعة سمعية	155	itch	131	stethoscope	105	exhalation -	
epidemic	156	medicine	132	heart rate	106	system	79
immunity	157	دواء شهرية	133	sphygmometroscope	107	digestive	
nerve	158	menstruation	134	heartbeat	108	Face	80
healthy	159	sick	135	analysis	109	pharynx	81
eat	160	capsule	136	medical	110	liver	82
drink	161	liquid	137	analysis laboratory	111	Pancreas	83
sleep	162	ointment	138	X-ray	112	intestine	84
up wake	163	قطارة	139	inflammation	113	Small	
hear	164	أخذ إبرة	140	swelling	114	intestine	85
silence	165	تلقيح	141	cold	115	Large	
inhale	166	تطعيم	142	infection	116	appendix	86
rise	167	أشعة ليزر	143	headache	117	Vermiform	
descend	168	مخدرات	144	pain	118	system	87
open	169	إدمان	145	fever	119	Nervous	
close	170	oneness		diarrhea	120	heart	88
				constipation		senses	89
				colic		Five	
						muscle	90
						tissue	91
						hospital	92
						first aid	93
						wound	94
						burning	95

- [22] L. Geng, X. Ma, H. Wang, J. Gu, and Y. Li, "Chinese sign language recognition with 3d hand motion trajectories and depth images," in *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*. IEEE, 2014, pp. 1457–1461.
- [23] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle, and F. Kamangar, "A system for large vocabulary sign search," in *European Conference on Computer Vision*. Springer, 2010, pp. 342–353.
- [24] P. Jangyodsuk, C. Conly, and V. Athitsos, "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, 2014, p. 50.
- [25] M. Bhuyan, P. Bora, and D. Ghosh, "Trajectory guided recognition of hand gestures having only global motions," *World Academy of Science, Engineering and Technology*, vol. 21, pp. 753–764, 2008.
- [26] M. Mohandes and M. Deriche, "Arabic sign language recognition by decisions fusion using dempster-shafer theory of evidence," in *Computing, Communications and IT Applications Conference (ComComAp), 2013*. IEEE, 2013, pp. 90–94.
- [27] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
- [28] *LAS: Second part of the Unified Arabic Sign Dictionary*, 2006.
- [29] S. Khalid, "Motion-based behaviour learning, profiling and classification in the presence of anomalies," *Pattern Recognition*, vol. 43, no. 1, pp. 173–186, 2010.
- [30] S. Khalid and S. Razzaq, "Frameworks for multivariate m-mediods based modeling and classification in euclidean and general feature spaces," *Pattern Recognition*, vol. 45, no. 3, pp. 1092–1103, 2012.
- [31] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE transactions on Image Processing*, vol. 16, no. 7, pp. 1912–1919, 2007.
- [32] S. Wu and Y. F. Li, "Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition," *Pattern Recognition*, vol. 42, no. 1, pp. 194–214, 2009.
- [33] M. Simão, P. Neto, and O. Gibaru, "Using data dimensionality reduction for recognition of incomplete dynamic gestures," *Pattern Recognition Letters*, 2017.