Using The Hausdorff Algorithm to Enhance Kinect's Recognition of Arabic Sign Language Gestures

Miada A. Almasre *malmasre@kau.edu.sa*
Faculty of Computing and Information Technology/
Department of Computer Science
King AbdulAziz University
Jeddah, Saudi Arabia

Hana A. Al-Nuaim *hnuaim@kau.edu.sa*
Faculty of Computing and Information Technology/
Department of Computer Science
King AbdulAziz University
Jeddah, Saudi Arabia

Abstract

The objective of this research is to utilize mathematical algorithms to overcome the limitations of the depth sensor Kinect in detecting the movement and details of fingers and joints used for the Arabic alphabet sign language (ArSL). This research proposes a model to accurately recognize and interpret a specific ArSL alphabet using Microsoft's Kinect SDK Version 2 and a supervised machine learning classifier algorithm (Hausdorff distance) with the Candescent Library. The dataset of the model, considered prior knowledge for the algorithm, was collected by allowing volunteers to gesture certain letters of the Arabic alphabet. To accurately classify the gestured letters, the algorithm matches the letters by first filtering each sign based on the number of fingers and their visibility, then by calculating the Euclidean distance between contour points of a gestured sign and a stored sign, while comparing the results with an appropriate threshold.

To be able to evaluate the classifier, participants gesturing different letters with the same Euclidean distance value of the stored gestures. The class name that was closest to the gestured sign appeared directly on the display sign window. Then the results of the classifier were analyzed mathematically.

When comparing unknown incoming gestures signed with the stored gestures from the dataset collected, the model matched the gesture with the correct letter with high accuracy.

Keywords: Pattern Recognition, Hand Gesturing, Arabic Sign Language, Kinect, Candescent.

1. INTRODUCTION

One of the most basic of human needs is communication, yet for the deaf or hearing impaired, communication in their daily activities is challenging, due to the lack of efficient and easy-to-use assistive devices and specialized sign language interpreters. Recently, the need to integrate Arabic hearing-impaired individuals within their communities has received more attention from different public and private organizations. For example, at the college level, there is no general admission to most majors for these individuals due to the small number of interpreters in each field as well as a lack of necessary communication tools.

Sign language is an important system of gestures that many people with hearing impairments use, whereas hand gestures are the basic form of communication in sign language used by people with hearing impairment [1] [2]. A single sign gesture can have different meanings in different cultural contexts.

A gesture is a means of communication with no sound that may be performed by many parts of the body [1]. In general, researchers have classified gestures as being based on a single hand, two hands, a hand and other body parts, the body, and/or the face [1].

Hand gestures use the palms, finger positions, and shapes to create forms that refer to different letters and phrases [3]. Different technologies can be used to detect and interpret hand signs, such as hand-tracking devices based on spatial-temporal features and hand-parsing-based color devices or 3D hand reference models [4].

The use of many software applications has facilitated current hand-gesture recognition research. Such applications use different hardware that recognizes the hand using glove-based, vision-based, or depth-based approaches, not only for recognition but also for hand-gesturing classification [5]. Glove-based input devices sense and transmit data about the hand's motion, camera-based approaches use color space models, and depth-sensor-based approaches use distance measurement to recognize the hand's shape in 3D [6].

Hand posture, which is defined by the pose or configuration of the hand in a single image, whether static or dynamic, has been employed to implement commands such as selection, navigation, and image manipulation with specific movements required by an application [7]. If the hand is to be the direct input for the system, then communication between the human and the computer must pass a hand-detection phase and a hand-recognition or classification phase [7].

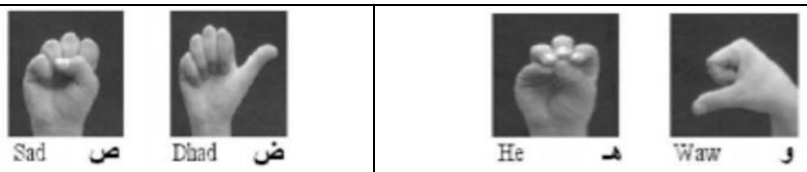
Researchers have used numerous devices to extract hand features in the hand-detection phase that give correct data about the hand's location and certain information about each finger [7]. As an example, researchers have relied on a hand kinematic model and used it to mold the hand's shape, direction, and pose with 3D hand-motion estimation systems [7].

There have been many attempts at 3D modelling of the hand. For example, the Forth 3D hand tracking library models any 3D object with very good performance, but only one hand can be detected, and it is not open source, which makes it hard to use with any sensor devices [8]. In addition, the "Hand Pose Estimation with Depth Camera and Hand Gesture Recognition" project examined an open-source library but only managed to detect one hand, despite its use of a depth camera [8] .
Although sign language recognition with depth sensors such as Microsoft's Kinect is becoming more widespread, it cannot accurately detect signs from the fingers' movements while communicating in Arabic sign language (ArSL); which could be due, according to Kinect's specifications, to its capability of recognizing only 22 joints of the human body.

ArSL recognition systems encounter problems in interpreting and detecting hand parts or gestures, especially in certain cases:

•When signing the letters ض (Dhad) and ص (Sad) (pronounced dhad and sad, respectively), for example, the palm cannot be recognized because the fingers are covered by the palm (Figure 1-A). Therefore, most of the letters in ArSL that use the palm's contour will not be recognizable.

•Using the joints as indicators will cause a problem because it is difficult to count the joints in some cases like و and هـ (pronounced Waw and He) below (Figure 1-B).



A) Palm cannot be recognized | **B)** Difficult to count the joints

FIGURE 1: ArSL Letter Samples.

These examples, whereby gestures tend to involve more general movements than simple pointing, make it evident that recognizing individual fingers in hand signs using depth-sensor devices alone is difficult [9].

However, depth sensors provide essential data about each object near the device, which will help in extracting many of the users' body features, such as their neckline, thumb, and index finger. In real life, hand and finger recognition requires extra features to be extracted and use complex procedures to achieve accurate gesture recognition [10].

Machine learning is critical to interpret or to translate sign language into text. The concept of machine learning is divided into 2 main categories: supervised and unsupervised learning [11]. In supervised learning, the machine would have prior knowledge of the model's characteristics. A sensor, such as a depth sensor device, feeds signal data from users who gesture letters in front of it. Data are then stored for a classification algorithm to recognize all arriving signals. Thus, prior knowledge comes from previously stored data captured by other devices. However, in unsupervised learning, the machine does not have any prior knowledge [11]; the algorithm cannot rely on any prior information, so it works as a descriptor based on certain extracted features such as the intensity of colors, location, and any instantaneous features that may be added. Then, the algorithm will split up the signs into translated objects.

Previously in [12] and [13], the researchers used two depth sensors with supervised machine learning algorithms to recognize ArSL letters. This set up was cumbersome for users to use, costly and each device had compatibility issues when used with the other. To avoid such complications for this research, only one sensor was used with supervised machine learning to examine and classify gestured signs.

As an example, for clarification, suppose we need to make the machine learn to translate certain ArSL letters; we have to divide the characters into several groups according to a property that we obtain from the shape of the sign. Assume we have 2 groups of 4 Arabic letters, each of which use similar forms of signs in terms of the direction of the fingers. The 2 groups are the letters: أ, ب, ت,ث and ج,ح,خ,د, respectively. The finger direction in the first group of signs is up, while that of the second is to the left. Hence, the machine can recognize an unknown sign based on the finger direction.

Compared with other work carried out for ArSL, this research may be novel in its use of Microsoft's Kinect V2 to examine the accuracy of ArSL classifications, which have only recently witnessed a surge of interest. Therefore, it is the objective of this research is to utilize mathematical algorithms to overcome the limitations of the depth sensor Kinect in detecting the movement and details of fingers and joints used for the Arabic alphabet sign language (ArSL). This research proposes a model to accurately recognize and interpret a specific ArSL alphabet when users gesture letters from the same distance. The model uses Kinect SDK Version 2 and a supervised machine learning classifier algorithm (Hausdorff distance) with the Candescent Library.

2. LITERATURE REVIEW

In [10], the researchers presented a method using the finger joints to apply a skeleton algorithm directly by storing the possible decisions of gesture classes in a tree data structure.

In [14], the researchers offered a review of image-based gesture-recognition models and discussed applications using different features among a variety of data-collection methods. They concluded that gesture recognition needs more research within the computer vision and machine learning fields. In addition, Arpita et al. [15] tends to agree with Wu and Huang [14] and add that with the expansion in applications, gesture-recognition systems demand more research in diverse directions.

Many projects have attempted to enhance the communication process with the hearing impaired. One such project is SignSpeak, which is an initiative to increase communication between the signer and hearing communities through a vision-based sign-language interpretation approach [16]. However, with the lack of a standard dictionary and even before the establishment of a sign language, “home signs” have been used by hearing-impaired communities as a means of communication with each other [17]. These home signs were not part of a unified language but were still used as familiar motions and expressions employed within families or regions [17]. This is similar to what has been observed in Arabic countries, where no standard sign language is used, even within the same country.

In 2012, Kurakin and Zhang proposed a system to evaluate data for 12 dynamic American Sign Language gestures. Their system achieved 87.7% recognition precision on this challenging dataset under huge variations in style, with variations in rapidity and hand orientation for each gesture. Furthermore, they developed a real-time system with which any user can accomplish hand gestures in front of the Kinect device, which executes gesture recognition automatically [18]. In 2013, Lin et al. from Wuhan University in China [19] proposed a novel real-time 3D hand gesture recognition algorithm. They collected a benchmark dataset using Kinect. When they tested the proposed algorithm, they observed a 97.7% average classification accuracy, which supports the effectiveness of the proposed algorithm [19].

In 2014, a test was conducted on a dynamic real-time hand gesture recognition system [20]. Ten participants produced 330 cases for the system, which recognized 11 hand gestures with three different poses for each gesture; the accurate recognition rate was 95.1% [20].

In 2015, Wang et al. presented a hand gesture recognition system based on the earth mover’s distance metric together with the depth camera in the Kinect. Their proposed system achieved a high mean accuracy and fast recognition speed, in comparison with 2 real-time applications [21].

Tarner, Weaver, and Pentland (1998) proposed two real time hidden Markov model based systems for American Sign Language (ASL) recognition, for continuous sentences. Using a camera mounted on the desk in front of the user achieved an accuracy of 92%, while using a cap mounted camera worn by the user, achieved an accuracy of 98 % [22].

Other researchers worked on real time dynamic systems, such as the finger-knuckle-print recognition model proposed by Ying Xu (2014), where a set of points were extracted from images to create a hand convex hulls. However, the sensitivity to light, clutter and skin color in color cameras dictated the need to use a depth camera with the Kinect sensor, tracking only one point in the hand [23].

Although, there have been advances in recognition systems to allow the hearing impaired to interact with society, limited research has addressed gesture recognition for ArSL, and there have been no serious efforts to develop a recognition system that the hearing impaired can use as a communication tool [2].

The American University of Sharjah presented an enhanced solution for user-dependent recognition of isolated ArSL gestures using disparity images [24]. Moreover, researchers from Helwan University suggested a system with a large set of samples that has been employed to recognize 20 words from ArSL. Their experiments involved real videos of hearing-impaired people in various outfits and with different skin colors. The proposed system achieved an overall recognition rate of up to 82.22% [2].

Mohandes et al. (2014), presented review systems and methods for the automatic recognition of Arabic Sign Language to develop a novel chart that classifies the main classes of ArSL recognition algorithms [25].

Although some Arabic countries, especially the Gulf countries, have made efforts to provide sign language communication assistive tools such as websites, mobile applications, relying on a standard dictionary [26], many researchers have only examined sign language recognition systems from a technical perspective, while disregarding the accuracy of the meaning of the letters to the Arabic hearing impaired.

3. GESTURE RECOGNITION PHASES

3.1 The Data-Collection Phase

Depth sensors provide essential data about each object or human who is close in distance to the sensors, which will help in extracting many of the user's hand and body features, such as the user's neckline, thumb, and index finger. Hand recognition requires extra values (features) to be extracted and the use of complex procedures to achieve accurate gestures [3].

Such sensors exist in many devices used in gaming systems, like Microsoft's Kinect for the Xbox 360. This video game console device is designed to support single-player or multiplayer gaming using handheld controllers. The Kinect incorporates several types of advanced sensing hardware, such as a depth sensor, a color camera, and a 4-microphone array that provides full-body 3D motion capture and facial recognition [27]. The Kinect has a wide sensing range and tracks not only fingers but also the complete skeleton. The Kinect has a number of valuable camera-based sensors available; each tends to concentrate on certain features. It also has multiple sensors that can be used jointly to facilitate the tracking of additional users at the same time or to get a wider view of the space. It can be used for tracking standing skeletons and high-depth fidelity [9].

The recent advances in the 3D depth cameras that Kinect sensors use have created many opportunities for multimedia computing. They were built to extend far beyond the way people play games to help people interact with specific applications with their bodies, such as by using natural face, hand, and body gestures [28].

The Kinect interpreter can be implemented in public and private institutions and by service providers like hospitals, airports, and classrooms to orient, train, guide, and even teach the hearing impaired [29]. This will allow the hearing impaired to interact with others semi-naturally without the need to hold certain positions or sign in a special way, which will make their interactions natural and require fewer stilted poses [29].

The Microsoft Kinect relies on depth technology, which allows users to deal with any system via a web camera without the need to touch the controller, through a natural user interface with which the user utilizes hand gestures or verbal commands to recognize objects [29]. Xilin Chen, a professor at the Chinese Academy of Sciences who is working on a Microsoft Kinect interpreter project, states that machine learning technology can provide the meaning of gestures by segmentation from one posture to another and also combining the trajectory. Chen adds that one can make decisions on the meaning of a gesture by using machine learning technology [29].

3.2 The Classification Phase

The classification phase has two steps; the first step performs a hand feature extraction that uses many parameters for detecting the details of the hand's shape by relying on the application type and goal, such as detecting the hand contour, fingertips, joints, hand center point, etc. [7]. The second step is the gesture classification or recognition process. The algorithms being used in this step will vary depending on whether the gesture is static, involving one frame, or dynamic, involving continuous frames [7].

Although most gesturing systems classify this process in different steps, each system still implements a different process in each step, even if two systems have the same function [30] [31]; this means in each step, there are many methods that researchers can follow even if they have the same aim.

In general, the classification phase usually relies on a mathematical descriptor model, such as the Markova chain, the support vector machine, or the Hausdorff distance classifier [32]. The Hausdorff distance classifier is an algorithm that checks for the maximum distance of a set to the nearest point in the other set [33] [34]. More formally (eq. 1), where a and b are points of sets A and B, respectively, and d(a, b) is a distance between these points [33]. The distance between a and b is a straight line in the Euclidean space, which is called Euclidian distance (Ed). The equation of Ed for a(x,y,z) and b(X1,Y1,Z1) is defined as (eq. 2).

$$h(A,B)=\max_{a\in A}\{\min_{b\in B}\{d(a,b)\}\} \quad \text{(eq. 1)}$$

$$Ed(a,b)=\sqrt{(X_1-x)^2+(Y_1-y)^2+(Z_1-z)^2} \quad \text{(eq. 2)}$$

This algorithm can also be used to measure the similarity between 2 shapes [34]; for example, if we consider the set of contour points of the hand represented as a polygon, it can be used to match simple convex polygons. Then, a model of gesture-based process manipulation can be created by applying all axes of the three-dimensional (3D) space.
FIGURE 2 shows the calculation process of the Hausdorff distance between 1 point a(x,y,z) in polygon A and all points in polygon B. The minimum distance between point (a) and all polygon B points is (a, b), which is the Hausdorff distance from point (a) to polygon B. Thus, the matching process will go through the same process for all points between the two polygons.

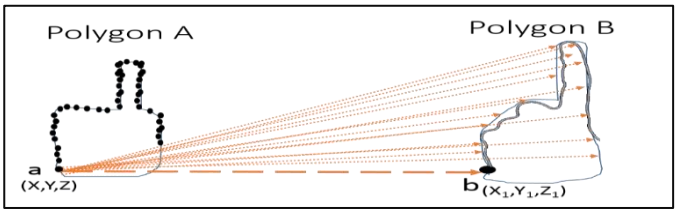


FIGURE 2: Hausdorff distance for 1 point between polygons.

The formal Hausdorff distance between polygons A and B is defined as (eq. 3).

$$H(A,B)=\max\{h(A,B),h(B,A)\}. \quad \text{(eq. 3)}$$

Many technologies and devices used for hand gesturing apply this algorithm, such as depth-sensing technology using a Leap Motion Controller, the time-of-flight camera, and Microsoft Kinect.

4. THE PROPOSED MODEL

The proposed model of this research is to detect and recognize different users' hand gestures of ArSL letters but at the same distance using a Hausdorff classifier as a supervised machine learning approach with a depth sensor.

4.1 The Hardware and Software Used

Kinect Version 2 (V2) has an open-source Software Development Kit (SDK) for developing applications. The minimum SDK requirements for development with Kinect V2 are a 64-bit processor, dual-core 2.66 GHz as minimum processor, and 2-GB RAM with Windows 8 [9]. Kinect V2 can deal with other toolkit libraries, such as OpenCV, OpenNI, and CLNUI. Thus, many developments or simulation environments can be used, such as Microsoft Visual Studio with C++ or C#, Matlab, and some 3D modelling software. Because the Microsoft Kinect™ SDK V2 had just been released around two years prior to this research, the choice between libraries was limited.

In Kinect V1, multiple streams provided frames that the sensor took, but in Kinect V2, instead of the stream layer, there is a source layer that provides multiple readers with frames that the sensor has taken. Therefore, reading data and setting event handlers were the major improvements in V2.

Due to the limitation in the interpreting of gestures ([35], [8]) of software packages used in computer vision in general, such as the Matlab toolbox, Visual Gesture Builder, and OpenCV, the Candescent Natural User Interface (NUI) was used for hand and finger tracking using Kinect depth data (Figures 3 and 4). Candescent NUI contains a set of libraries that are open source, developed with the C# programing language with the OpenNI Natural Interface library and Microsoft Kinect SDK (Candescent NUI, 2016) [37].

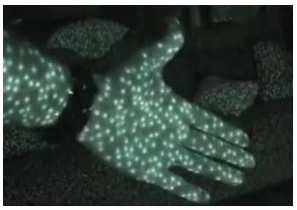


FIGURE 3: Real IR light dots that IR Emitter sends [33].

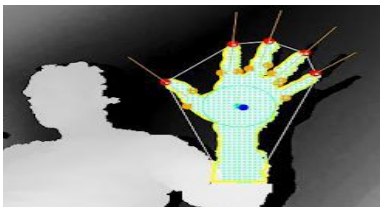


FIGURE 4: Reading dots using Candescent [33]**.**

The Candescent library for hand and finger recognition can detect up to 2 handsets simultaneously and provides useful information for hand and finger tracking. It starts by detecting close objects, if the objects are hands, as well as Candescent extracts, and it stores many of the hands' features, such as:

- The direction of each finger
- The volume of the hand and contour shape
- The palm position: the green circle
- The number of fingers and each finger's base position
- The identification (ID) number for each finger

In addition, Candescent has a convex hull algorithm that gives the fingertip position (x,y,z), which is the smallest polygon around the hand (FIGURE 4). Candescent has fingers numbered from 0 to 4 for each hand, but it does not identify which finger it is [36].

To use the Candescent library with Kinect SDK V2 instead of the openNI it also supports, the data source must be configured properly and must add the 4 necessary dll references of the library's dll, CCT.NUI.Core, CCT.NUI.HandTracking, CCT.NUI.KinectSDK, and CCT.NUI.Visual. FIGURE 5 presents the first perspective structure that explains the main concept of hand detection and recognition for ArSL letters.

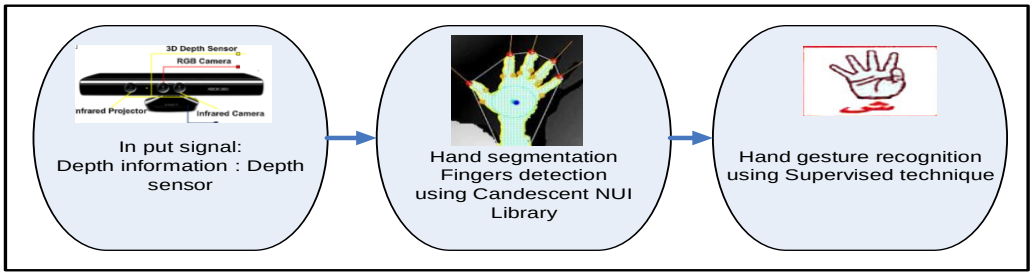


FIGURE 5: The general process of the experiment**.**

4.2 The Experimental Environment for the Proposed Model

The data collection was done in a room with white light. Five deaf participants in white clothing stood in the same position 50 cm away from the front of the Kinect V2 device. The Kinect was mounted on a table 70 cm from the floor. The background behind the participants was white (FIGURE 6).

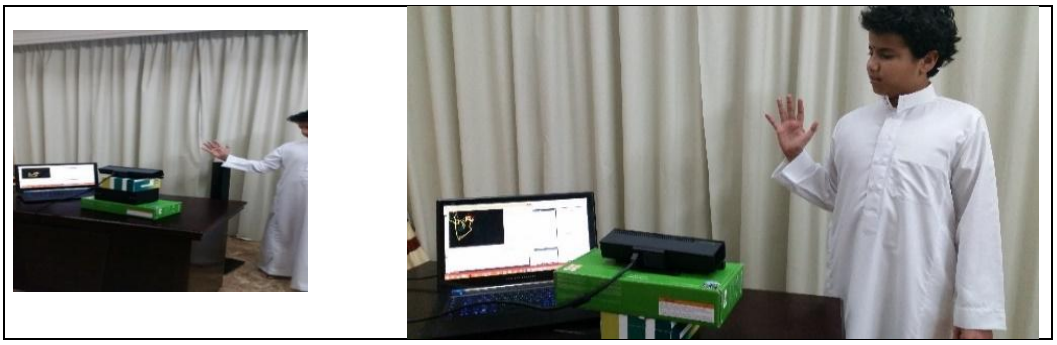


FIGURE 6: Experimental environment with one of the participants.

The capturing procedure used Microsoft Kinect as an input device, and all of the gestures that the device captured and stored were activated. Then, all of the participants were asked to make the same letter gesture continuously in one clip, making 5 clips for each letter. To check and correct the ArSL gestures, we used the "The Arabic Dictionary of Gestures for The Deaf" as a reference that is used in many educational institutes [26], and the institute manager guided the participants to perform the standard gestures according to the camera position.

4.3 The Data Collection Phase in the Proposed Model

For the purpose of this research, nine letters (Alef as A, Lam as L, etc.) (FIGURE 7) were chosen because they are the most difficult to recognize because of the different finger directions. Each letter was considered as a class from the ArSL alphabet.

Alef (أ)	Sad (ص)	Dhad (ض)	He (هـ)	Waw (و)	Lam (ل)	Ya (ي)	Dal (د)	Fa (ف)

FIGURE 7: The nine ArSL letters**.**

Kinect was used to recognize the hand's point distance data and then extracted certain values (features) for finger tips, the hand palm center, and hand contour points to classify each letter using a supervised classifier algorithm. The hand contour is the largest area in the hand image.

In this experiment, to provide prior knowledge for our model for supervised learning, we minimized the experimental dataset with 9 ArSL letters. Each of the participants performed each letter 5 times. Thus, 5 different participants did the representation of the gestures in front of the machine 5 times. Each gesture represents the movement of the desired characters. Thus, there were 9 × 5 × 5 = 225 cases, and we used different age groups to ensure variations in style, speed, and hand orientation for each gesture.

The threshold value is a point or value that the gesture accuracy could affect. Through experimentation, researchers can determine a threshold that is useful in reducing the computation in the search process, as it becomes the point separating the accepted from the

rejected values. For instance, if the value is under the threshold value, the gesture is accepted; otherwise the gesture is rejected if it goes beyond the threshold.

The threshold value used in this research was taken from the Finger-Spelling Kinect project [6]. This project attempted to find out how Microsoft Kinect would recognize and translate German Sign Language from the depth images of the sensor to ordinary speech or text. The researchers in that project conducted many experiments about hand positions and recommended a threshold value in their source code. The XML data file contains many tags that describe the letter, such as:

- Gesture name: the name of the gesture that the user enters in the class name field
- Finger count: the number of fingers that Kinect recognize.
- Palm point: the (x,y,z) coordinates
- Hand volume: the depth, width, and height values for the hand
- Center: the (x,y,z) coordinates of the contour center
- Contour points: the (x,y,z) coordinates for each point that Kinect detects for the contour

Thus, the depth data collected for the 225 cases consisted of 3D coordinate values (x,y,z) for a convex hull and contour points. The error rate or threshold value that we established was 40 Euclidean distance (Ed). In addition, many points that described the gesture were stored. We also used the finger count and contour point as candidate points to classify the gesture.

FIGURE 8 shows a snapshot from the data collection phase to the gestures of the letters ض (Dhad) and ص (Sad). When the first participant clicked on the "start hand tracking" button using a mouse, all 5 participants made the gesture of the same letter one after the other. After closing the training form window, a gesture dataset file was generated as an XML file that could contain up to 220 points for each letter. The classification phase may have enough knowledge to classify or recognize an unknown gesture.

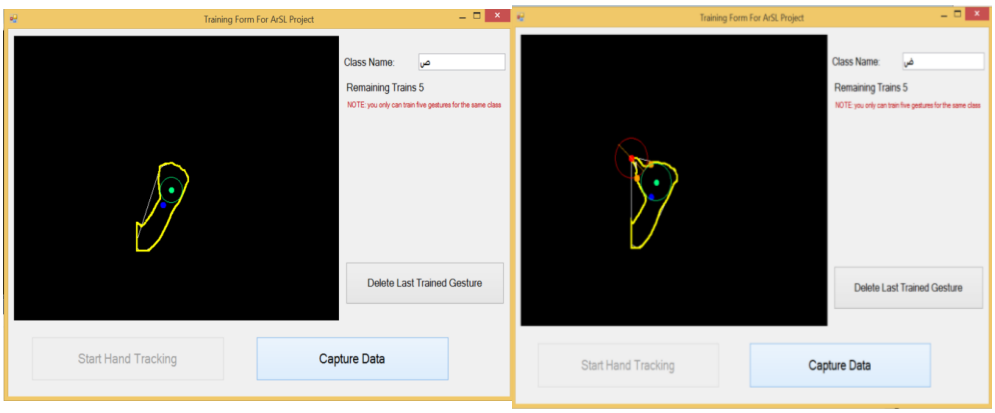


FIGURE 8: A snapshot from the Collecting data.

4.4 The Classification Phase in the Proposed Model

As an enhancement of the Hausdorff distance algorithm, more steps were added to check the number of polygon vertices (fingertips) before the start of the calculation of the Ed. In other words, in Figure 2, if polygons A and B, respectively, have an n and m vertices, they are not a match (where m≠n). This will reduce the search process when we check the match between 2 signs in our model. The enhancement algorithm of the proposed model is as follows:

ALGORITHM: Hausdorff Enhancement

1- Initializing values:
   1. X = Get finger numbers;
   2. I-List [dataset length] = Set of contour points that the dataset has for the first polygon.
   3. Temp-List [gestures] = Empty.
   4. Set A = Points of desired gesture.
2- **Loop** on I-List
   If (finger numbers = X), Temp-List [i] = I-List[i];
   Else, reject gesture.
3- **Loop** on Temp-List
   a) Get the Euclidean Distance (Ed) between Set A points and I-List[i].
   b) Hausdorff-Temp = Get the min value of the previous calculation (shortest).
4- **Repeat** (a) and (b) for the rest of the points in A.
5- Hausdorff-Distance = Max value from Hausdorff-Temp.
6- **If** the Hausdorff-Distance > threshold, ignore the gesture and error message.

Else return the class gesture name that has less value or is the threshold.

7- **If** there is more than 1 value, then return the minimum.

FIGURE 9 is a flowchart that clarifies the algorithm steps.

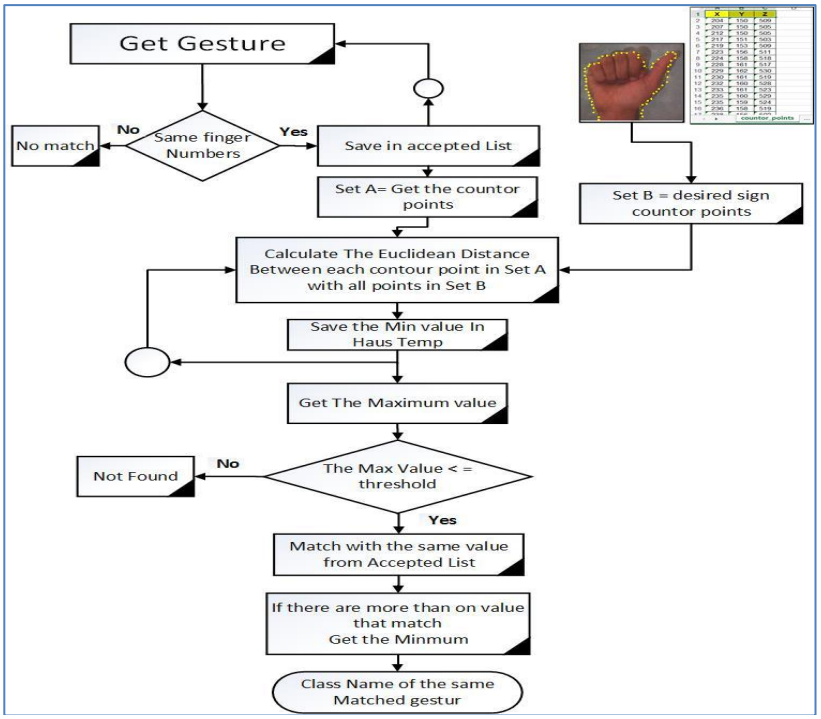


FIGURE 9: Algorithm flowchart.

4.5 Testing the Proposed Model

To evaluate the ability of the classifier to recognize 1 of the 9 gestured signs in our dataset, 10 different participants, who were not deaf, stood in front of the device and gestured a few of the 26 ArSL letters from varying distances after they learned how to sign the letters from the dictionary. The device recognized the sign when it is 1 of the 9 classes only if it was gestured at the same distance that the 5 participants originally gestured from the data collection phase. This means the classifier recognized the letter by producing Ed values, which are close to or the same Ed values of the stored class. The letter (Found Class: class name) was displayed on the classification

window (FIGURE 10). However, when the gesturing distance for any of the 9 stored classes was different from the stored values, the sign was not recognized: “Not Found” was displayed.

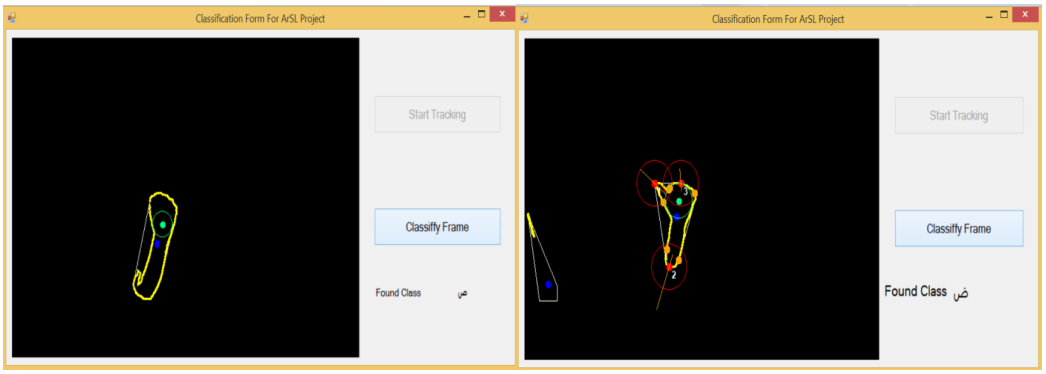


FIGURE 10: A snapshot from the classification phase.

Because the classification phase of the proposed model gives the classification result directly on the window (class name) without any statistical values that give any indication of the accuracy of the Ed stored values, a mathematical approach was followed to assess the accuracy of the stored values from the data collection phase to ensure that the failure of the classifier to recognize the gestured sign during testing was due to the distance of the participants from the device.

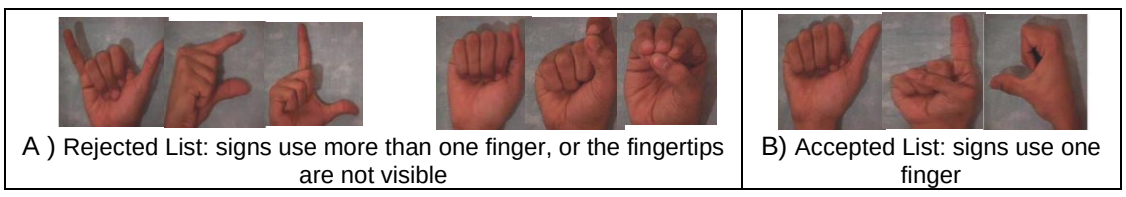
Because the XML file, generated in the data collection phase, was very large (around 220 points for each gestured letter), a function (Ed) was created using Matlab to examine the minimum of the Ed values computed during the user’s gesturing of any letter. The data file was converted from an XML structure to a Microsoft Excel sheet using the online converter tool “xmlgrid” [38]; then, the Excel sheet was uploaded as input for the Ed function.

As an example, and for the sake of simplicity, the details of mathematical operations were applied on only the first 6 of the total points stored in the XML file. In addition, because the algorithm has many nested loops, only the letter ص (Sad) was examined. The following demonstrates the mathematical approach step by step.

4.6 Classification

When a user made a gesture (an unknown letter at this stage) in front of the device, to classify this gesture into the class to which it belongs, the classification phase has to pass into two main steps:

In step 1, the dataset was filtered by comparing the unknown gesture values to the number of polygon vertices (fingers) of the 9 stored gestures. Because the hand shape of the unknown gesture has a single finger, the filtering generated the accepted list, which included gestures with a single finger (FIGURE 11-B). Thus, the gestures that matched the number of fingers were accepted; in this case, ص (Sad), أ (Alef), و (Waw), while the other letters were rejected as clarified in FIGURE 11-A. Thus, the accepted list contains the 3 possible matched letters that could be a match for the unknown letter captured.

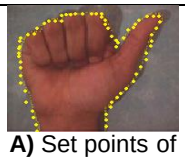


A) Rejected List: signs use more than one finger, or the fingertips are not visible

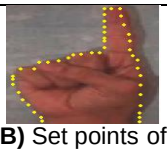
B) Accepted List: signs use one finger

FIGURE 11: Sign distribution based on fingers number.

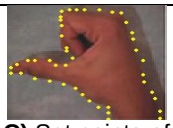
In step 2, the Hausdorff distance algorithm was applied to classify the remaining 3 possible matched letters, which were still considered unknown at this point. A measurement of the Ed between the unknown gesture's contour points and each accepted letter's contour points was applied; then, the minimum value from each calculation was stored in the Hausdorff temp list. For instance, the measurement of the Ed between the unknown gesture contour points (Figure 12-A) and the first accepted letter contour points (Figure 12-B) was calculated. Thus, the values in the first row in Table 1 (220,287,797) were used to calculate the Ed between them and all of the values in Table 2. Then, the minimum value was stored in the Hausdorff temp list. This process was repeated for all values in the accepted list. For more clarification, Figure 12-B, Figure 12-C, and Figure 12-D shows the set of that hand's points for the class, أ (Alef), و (Waw), and ص (Sad), while Table 2, Table 3, and Table 4 present the numeric values of the points, respectively.



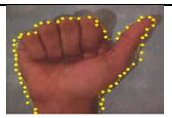
A) Set points of (Unknown Class)



B) Set points of (Class name Alef أ)



C) Set points of (Class name Waw **و**)



D) Set points of (Class name Sad **ص**)

FIGURE 12: Sets of some of contour points**.**

X	Y	Z
220	287	797
225	290	796
226	291	791
227	291	797
229	293	789
230	293	794
.	.	.

TABLE 1: Six of the counter points for the unknown gesture

X	Y	Z
410	338	792
411	337	784
411	336	796
415	332	796
416	332	782
417	331	787
.	.	.

TABLE 2: Six of the counter points for the (class name Alef أ)

X	Y	Z
175	218	635
176	217	616
177	215	644
178	215	612
179	214	606
180	214	579
.	.	.

TABLE 3: Six of the counter points for the (class name Waw و)

X	Y	Z
220	287	797
224	290	794
227	291	800
229	293	791
230	293	796
232	295	789
.	.	.

TABLE 4: Six of the counter points for the (class name Sad ص)

The following example shows the calculation of Ed between the first 6 contour points from the accepted list with the values in Table 2:

- Table 5 presents the calculation of Ed between the first point in the unknown gesture (220,287,797) and each point in Table 2 for the letter أ (Alef).
- Table 6 presents the calculation of Ed between the second point in the unknown gesture (225,290,796) and each point in Table 2 for the letter أ (Alef).
- Table 7 presents the calculation of Ed between the third point in the unknown gesture (226,291,797) and each point in Table 2 for the letter أ (Alef).
- Table 8 presents the calculation of Ed between the fourth point in the unknown gesture (227,291,797) and each point in Table 2 for the letter أ (Alef).
- Table 9 presents the calculation of Ed between the fifth point in the unknown gesture (229,293,789) and each point in Table 2 for the letter أ (Alef).
- Table 10 presents the calculation of Ed between the sixth point in the unknown gesture (230,293,794) and each point in Table 2 for the letter أ (Alef).

X	Y	Z	Ed
410	338	792	196.79
411	337	784	197.86
411	336	796	197.19
415	332	796	200.13
416	332	782	201.66
417	331	787	202.10
418	330	798	202.62

X	Y	Z	Ed
410	338	792	191.17
411	337	784	192.22
411	336	796	191.60
415	332	796	194.59
416	332	782	196.06
417	331	787	196.53
418	330	798	197.11

Min Value	196.79

TABLE 5: Calculation of Ed of the 1st point in the unknown gesture

Min Value	191.17

TABLE 6: Calculation of Ed of the 2nd point in the unknown gesture

X	Y	Z	Ed
410	338	792	189.91
411	337	784	190.76
411	336	796	190.46
415	332	796	193.46
416	332	782	194.58
417	331	787	195.18
418	330	798	196.05
Min Value			189.91

TABLE 7: Calculation of Ed of the 3rd point in the Unknown gesture

X	Y	Z	Ed
410	338	792	189.01
411	337	784	190.11
411	336	796	189.43
415	332	796	192.42
416	332	782	193.98
417	331	787	194.42
418	330	798	194.94
Min Value			189.01

TABLE 8: Calculation of Ed of the 4th point in the Unknown gesture

X	Y	Z	Ed
410	338	792	186.53
411	337	784	187.31
411	336	796	187.14
415	332	796	190.17
416	332	782	191.15
417	331	787	191.81
418	330	798	192.80
Min Value			186.53

TABLE 9: Calculation of Ed of the 5th point in the unknown gesture

X	Y	Z	Ed
410	338	792	186.53
411	337	784	187.31
411	336	796	187.14
415	332	796	190.17
416	332	782	191.15
417	331	787	191.81
418	330	798	192.80
Min Value			186.53

TABLE 10: Calculation of Ed of the 6th point in the unknown gesture

Tables 5-10 shows the measurement of Ed between the points in Table 1 for the unknown gesture and the set point of class أ (Alef) in Table 2. When calculating the Hausdorff distance, the values are stored in the temporary array (Temp array = [196.79, 191.17, 189.91, 189.01, 186.53, 186.53]), which are the values in the shaded cells in the tables above. The maximum (Max) value from the Temp array, 196.79, was compared with the threshold. If this maximum value is equal to or greater than the threshold, which is 40 in our experiment, then the value is ignored; otherwise, it is accepted.

The same previous mathematical process for the letter أ (Alef) was also applied between unknown gesture contour point values in Table 1 and Table 3 with the set points of class و (Waw), and between Table 1 and Table 4 with the set points of class ص (Sad). Table 11 shows the result of the complete process.

When comparing the max Hausdorff distance values with the threshold value such that, if (Max =< threshold (40)), the value is accepted, and otherwise, it will be ignored, the class name ص (Sad) was the matched value. If there was more than one Max value less than or equal to the threshold, then the minimum one would be accepted.

Min Value \ Class name	ص (Sad)	أ (Alef)	و (Waw)
For Point 1 in unknown gesture	1.00	196.79	174.48
For Point 2 in unknown gesture	1.00	191.17	176.16
For Point 3 in unknown gesture	4.24	189.91	172.59
For Point 4 in unknown gesture	3.00	189.01	178.00
For Point 5 in unknown gesture	5.10	186.53	172.66
For Point 6 in unknown gesture	0.00	185.55	177.18
Maximum (Max)	5.10	196.79	178.00

TABLE 11: The minimum values for all gesture classes.

The following chart shows the results of the unknown gesture matched with the first 6 points that were applied (FIGURE 13). As we can see, comparing the values of the points of the unknown

gesture with those of the class name ص (Sad) are not only less than the threshold but also close to zero, which means a high accuracy in matching.

Results from the mathematical approach found the correct letter with high accuracy because the matched letter value was 5.10, ص (Sad), which is less than the threshold. Meanwhile, the other letter values, و (Waw) and أ (Alef), (196.79, 178.00), were not only more than the threshold but also were very far away from it, as shown in Table 11 and FIGURE 13.

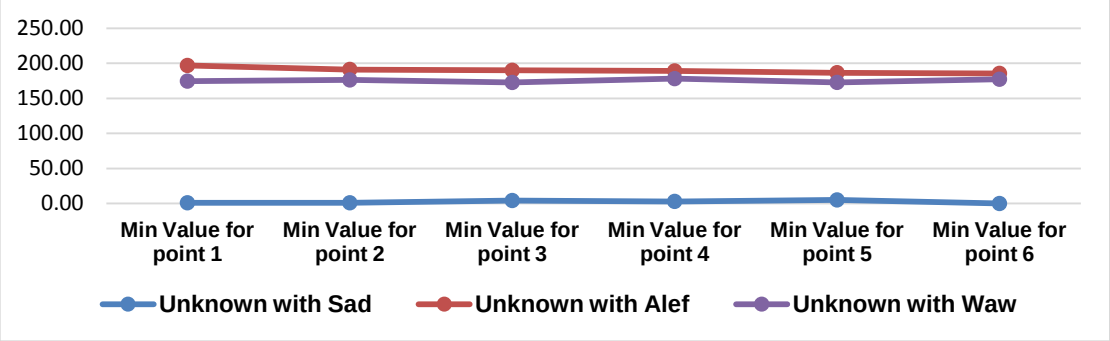


FIGURE 13: Values for the 3 accepted gestures with unknown one for the first 6 points only.

The accepted list in FIGURE 14 relates all of the values for the accepted gestures (Alef, Waw, and Sad) after applying the Ed function on all points as opposed to just the 6 points discussed in the example.

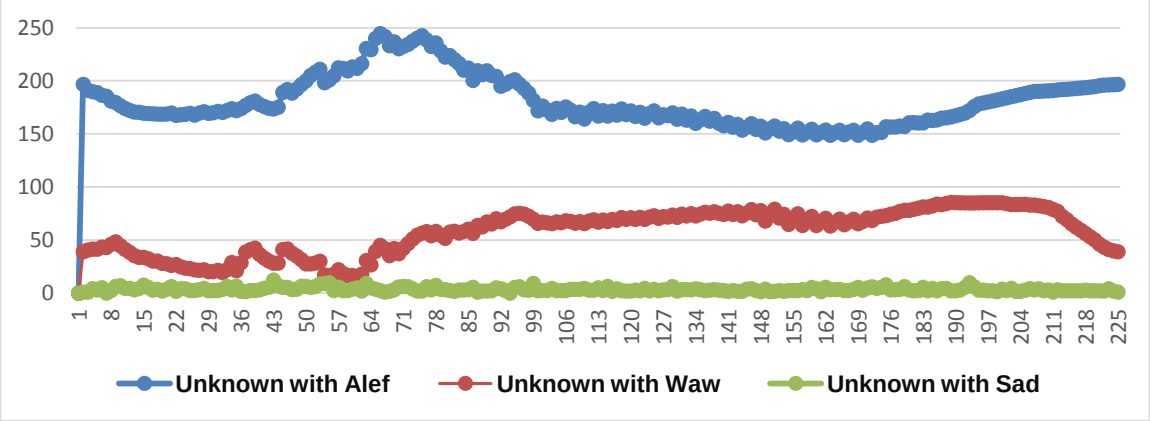


FIGURE 14: Values of the 3 accepted gestures with the unknown gesture for all points.

Note from Figures 13-14 that the unknown gesture has a close match with the gesture that has a class ص (Sad) only. In addition, the Ed function gave the same result that the unknown gesture matches the letter ص (Sad) as shown in FIGURE 14.

5. DISCUSSION AND CONCLUSION

The Candescent and Microsoft Kinect SDK V2 were used to develop an ArSL gesture recognition model, which interprets gestural interactions into letters. A supervised machine learning approach was used with a data collection and classification phases. In the data collection phase, signals of 9 gestured letters that Kinect had captured were stored as a dataset; then, in the classification phase, the Hausdorff distance algorithm matched any unknown new gesture letter with the class to which it belonged.

To provide prior knowledge to the model, we collected 225 matching cases of 9 letters from 5 participants. To test the proposed model, 2 methods were used. Initially, 10 participants stood in front of Kinect individually and 1 after the other gestured letters that included 1 or more of the 9 stored letters. The captured values from each hand's gesture were then stored to be processed using the Hausdorff distance algorithm. The results were compared to the stored classes, and a label (letter) appeared, indicating the closest value to 1 of the 9 stored classes. The results of applying this method were successful, as all gestures matched with the correct letter class as long as the user gestured from the same stored distance.

The second method was based on a multi-step mathematical process implemented to calculate the (Ed-Euclidean distance) values captured via the device for each sign with the purpose of deciding its proper class. An Ed function was created using Matlab to make the mathematical calculations due to the large number of values in the XML file. In addition, for ease of demonstration, only 6 points were used on an unknown gesture. Results were judged either above or below a fixed threshold. This multi-step process matched ص (Sad) as the unknown gesture because it was less than the designated threshold.

Kinect can be used as an interactive interpreter for the hearing impaired if the device is fed with enough data about sign language gestures. However, this research found that the processing speed of the Kinect was slow because the depth and image sensor works only with a frequency of 30 FPS and the algorithms for recognizing gestures requires considerable computation power. Also, choosing a suitable library for the Kinect was challenging. OpenCV is a general machine-learning and image-processing library that is not specific to hand tracking, and therefore, it takes a long time to create robust and efficient performance.

The Hausdorff distance algorithm was a successful supplementary approach that could be utilized to overcome Kinect's limitations in detecting the movement and details of fingers and joints used for the ArSL.

Having concluded that, another limitation of the Hausdorff distance algorithm arises, the limitation is its dependability on geometric values, as the participants must stand in the same position, and the hand must be in almost the same position as the hand's position during the data collection phase. In addition, to compute the Hausdorff distance between 2 polygons, 2 conditions must be met: There is no intersection between polygons, and neither one contains the other. For the accurate recognition of hand gestures, only 1 person should stay in the visible range, which is only 57° in front of the device, to restrict participants' movement. Moreover, the sensor captures the best image at a distance of only about 1 meter and in normal ambient light.

Since this research focused on static gestures – a single frame as one pose, as future work, the researchers recommend applying the Hausdorff algorithm to recognize more dynamic gestures - multiple frames for more than one pose.

5. REFERENCES

[1] L. Chen, F. Wang, H. Deng, and K. Ji, "A Survey on Hand Gesture Recognition," in *2013 International Conference on Computer Sciences and Applications (CSA)*, 2013, pp. 313–316.

[2] A. A. Youssif, A. E. Aboutabl, and H. H. Ali, "Arabic sign language (arsl) recognition system using hmm," *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 2, no. 11, 2011.

[3] H. Liang and J. Yuan, "Hand Parsing and Gesture Recognition with a Commodity Depth Camera," in *Computer Vision and Machine Learning with RGB-D Sensors*, L. Shao, J. Han, P. Kohli, and Z. Zhang, Eds. Cham: Springer International Publishing, 2014, pp. 239–265.

[4] H. Liang, J. Yuan, D. Thalmann, and Z. Zhang, “Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization,” *Vis. Comput.*, vol. 29, no. 6–8, pp. 837–848, Jun. 2013.

[5] R. Z. Khan and N. A. Ibraheem, “Hand Gesture Recognition: A Literature Review,” *ResearchGate*, vol. 3, no. 4, pp. 161–174, Aug. 2012.

[6] Y. Li, “Hand gesture recognition using Kinect,” in *2012 IEEE International Conference on Computer Science and Automation Engineering*, 2012, pp. 196–199.

[7] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, “Vision-based hand pose estimation: A review,” *Comput. Vis. Image Underst.*, vol. 108, no. 1–2, pp. 52–73, Oct. 2007.

[8] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, “Tracking the articulated motion of human hands in 3D,” *ERCIM NEWS*, p. 23, 2013.

[9] A. Jana, *Kinect for Windows SDK programming guide: build motion-sensing applications with Microsoft’s Kinect for Windows SDK quickly and easily*. Birmingham: Packt Publ, 2012.

[10] H. Liang, J. Yuan, and D. Thalmann, “Resolving Ambiguous Hand Pose Predictions by Exploiting Part Correlations,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1125–1139, Jul. 2015.

[11] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[12] M. A. Almasre and H. Al-Nuaim, “Recognizing Arabic Sign Language gestures using depth sensors and a KSVM classifier,” in *2016 8th Computer Science and Electronic Engineering (CEEC)*, 2016, pp. 146–151.

[13] M. A. Almasre and H. Al-Nuaim, “A Real-Time Letter Recognition Model for Arabic Sign Language Using Kinect and Leap Motion Controller v2,” *IJAEMS Open Access Int. J. Infogain Publ.*, vol. Vol-2, no. Issue-5, 2016.

[14] Y. Wu and T. S. Huang, “Vision-Based Gesture Recognition: A Review,” in *Gesture-Based Communication in Human-Computer Interaction*, A. Braffort, R. Gherbi, S. Gibet, D. Teil, and J. Richardson, Eds. Springer Berlin Heidelberg, 1999, pp. 103–115.

[15] A. R. Sarkar, G. Sanyal, and S. Majumder, “Hand gesture recognition systems: a survey,” *Int. J. Comput. Appl.*, vol. 71, no. 15, 2013.

[16] “SignSpeak,” *SignSpeak*, 2016. [Online]. Available: http://www.signspeak.eu/en/publications.html. [Accessed: 19-Nov-2016].

[17] R. J. Senghas and L. Monaghan, “Signs of Their Times: Deaf Communities and the Culture of Language,” *Annu. Rev. Anthropol.*, vol. 31, pp. 69–97, 2002.

[18] A. Kurakin, Z. Zhang, and Z. Liu, “A real time system for dynamic hand gesture recognition with a depth sensor,” in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, 2012, pp. 1975–1979.

[19] L. Song, R. M. Hu, Y. L. Xiao, and L. Y. Gong, “Real-Time 3D Hand Tracking from Depth Images,” *Adv. Mater. Res.*, vol. 765–767, pp. 2822–2825, 2013.

[20] H. Y. Lai and H. J. Lai, “Real-Time Dynamic Hand Gesture Recognition,” in *2014 International Symposium on Computer, Consumer and Control (IS3C)*, 2014, pp. 658–661.

[21] C. Wang, Z. Liu, and S. C. Chan, “Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera,” *IEEE Trans. Multimed.*, vol. 17, no. 1, pp. 29–39, Jan. 2015.

[22] T. Starner, J. Weaver, and A. Pentland, “Real-time American sign language recognition using desk and wearable computer based video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.

[23] Y. Xu, Y. Zhai, J. Gan, and J. Zeng, “A novel finger-knuckle-print recognition based on convex optimization,” in *2014 12th International Conference on Signal Processing (ICSP)*, 2014, pp. 1785–1789.

[24] M. El-Jaber, K. Assaleh, and T. Shanableh, “Enhanced user-dependent recognition of Arabic Sign language via disparity images,” in *2010 7th International Symposium on Mechatronics and its Applications (ISMA)*, 2010, pp. 1–4.

[25] M. Mohandes, M. Deriche, and J. Liu, “Image-Based and Sensor-Based Approaches to Arabic Sign Language Recognition,” *IEEE Trans. Hum.-Mach. Syst.*, vol. 44, no. 4, pp. 551–557, Aug. 2014.

[26] S. M. ElQahtani, *The Arabic Dictionary of Gestures for the Deaf - Issue No. 1*, 2nd edition. Sayeed M. Al-Qahtani / H.R.H. Al Jowhara Bint Faisal Bin Turki Al Saud, 2008.

[27] J. Han, L. Shao, D. Xu, and Jamie Shotton, “Enhanced Computer Vision With Microsoft Kinect Sensor: A Review,” *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.

[28] Z. Zhang, “Microsoft Kinect Sensor and Its Effect,” *IEEE Multimed.*, vol. 19, no. 2, pp. 4–10, Feb. 2012.

[29] X. Chen, H. Li, T. Pan, S. Tansley, and M. Zhou, “Kinect Sign Language Translator expands communication possibilities,” *Microsoft Res. Outubro2013 Disponível Em Httpresearch Microsoft Comenuscollaborationstorieskinect-Sign-Lang.-Transl. Aspx*, 2013.

[30] G. R. S. Murthy and R. S. Jadon, “A review of vision based hand gestures recognition,” *Int. J. Inf. Technol. Knowl. Manag.*, vol. 2, no. 2, pp. 405–410, 2009.

[31] J. Suarez and R. R. Murphy, “Hand gesture recognition with depth images: A review,” in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012, pp. 411–417.

[32] M. K. Bhuyan, K. F. MacDorman, M. K. Kar, D. R. Neog, B. C. Lovell, and P. Gadde, “Hand pose recognition from monocular images by geometrical and texture analysis,” *J. Vis. Lang. Comput.*, vol. 28, pp. 39–55, Jun. 2015.

[33] “Hausdorff distance,” 2016. [Online]. Available: http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html. [Accessed: 19-Nov-2016].

[34] M. Tang, M. Lee, and Y. J. Kim, “Interactive Hausdorff Distance Computation for General Polygonal Models,” in *ACM SIGGRAPH 2009 Papers*, New York, NY, USA, 2009, p. 74:1–74:9.

[35] H. Liang, “Hand Pose Estimation with Depth Camera,” *Projects*, 2016. [Online]. Available: https://sites.google.com/site/seraphlh/projects. [Accessed: 18-Jan-2017].

[36] “Candescent NUI,” *CodePlex*, 2016. [Online]. Available: https://candescentnui.codeplex.com/documentation?ProjectName=candescentnui. [Accessed: 19-Nov-2016].

[37] S. Brunner and D. Lalanne, “Using Microsoft Kinect Sensor to Perform Commands on Virtual Objects,” *Publ. Oct*, vol. 2, 2012.

[38] Online XML, “Convert XML To Excel Spreadsheet xls/xlsx File Online,” 2016. [Online]. Available: http://xmlgrid.net/xmlToExcel.html. [Accessed: 19-Nov-2016].

[39] M. Hu, F. Shen, and J. Zhao, “Hidden Markov models based dynamic hand gesture recognition with incremental learning method,” in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 3108–3115.