

Article

# Recognition of American Sign Language Gestures in a Virtual Reality Using Leap Motion

Aurelijus Vaitkevičius<sup>1</sup>, Mantas Taroza<sup>1</sup>, Tomas Blažauskas<sup>1</sup>, Robertas Damaševičius<sup>1,\*</sup>, Rytis Maskeliūnas<sup>2</sup> and Marcin Woźniak<sup>3</sup>

<sup>1</sup> Department of Software Engineering, Kaunas University of Technology, 50186 Kaunas, Lithuania; aurelijuz@gmail.com (A.V.); mantas.taroza@gmail.com (M.T.); tomas.blazauskas@ktu.lt (T.B.)

<sup>2</sup> Department of Multimedia Engineering, Kaunas University of Technology, 50186 Kaunas, Lithuania; rytis.maskeliunas@ktu.lt

<sup>3</sup> Institute of Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland; marcin.wozniak@polsl.pl

\* Correspondence: robertas.damasevicius@ktu.lt

Received: 27 December 2018; Accepted: 24 January 2019; Published: 28 January 2019



**Featured Application:** We describe a system that uses a Leap Motion device to recognize the gestures performed by users while immersed in a Virtual Reality (VR). The developed system can be applied for the development of the VR applications that require identification of the user's hand gestures for control of virtual objects.

**Abstract:** We perform gesture recognition in a Virtual Reality (VR) environment using data produced by the Leap Motion device. Leap Motion generates a virtual three-dimensional (3D) hand model by recognizing and tracking user's hands. From this model, the Leap Motion application programming interface (API) provides hand and finger locations in the 3D space. We present a system that is capable of learning gestures by using the data from the Leap Motion device and the Hidden Markov classification (HMC) algorithm. We have achieved the gesture recognition accuracy (mean  $\pm$  SD) is  $86.1 \pm 8.2\%$  and gesture typing speed is  $3.09 \pm 0.53$  words per minute (WPM), when recognizing the gestures of the American Sign Language (ASL).

**Keywords:** gesture recognition; machine learning; data mining; pattern recognition; virtual reality; leap motion

## 1. Introduction

Hand gesture recognition is widely researched as it can be applied to different areas such as human-computer interaction [1], robotics [2], computer games [3], education [4], automatic sign-language interpretation [5], decision support for medical diagnosis of motor skills disorders [6], recognition of children with autism [7], home-based rehabilitation [8,9], virtual training [10] and virtual surgery [11]. In industry, gesture recognition can be used in areas requiring very high precision such as to control devices such as robot hands [12] or industrial equipment.

Hand gestures can be employed to control a virtual reality (VR) environment using a Leap Motion Controller. The controller tracks the movements of a VR environment operator's hands and fingers moving over the Leap Motion device in a specific sequence. Then, an operation corresponding to the recognized gesture is executed on the system to which the Leap Motion device is connected. The operating principle of Leap Motion is similar to a computer mouse or a touch screen but its operation is based on video recognition. Using two infrared (IR) cameras, this device can recognize human hands and allows the user to explore the virtual world and interact with the elements of this

world. Although the Leap Motion device is capable of recognizing human hands, it cannot directly recognize the gestures displayed by users. It is able to model the human hand and present its data in a three-dimensional space. There are software libraries that have features capable of recognizing some gestures such as grip but the VR environment requires recognition of many different gestures.

Leap Motion has been used before for the recognition of the Arabic [13,14], Indian [15,16], Turkish [17], Greek [18], Thai [19], Indonesian [20] and American [21–24] sign languages. Ameer et al. [25] used a Support Vector Machine (SVM) trained on spatial feature descriptors representing the coordinates of fingertips and the palm centre to recognize the gestures with an accuracy rate of about 81%, while Chuan et al. [21] achieved an accuracy of 79.83% using SVM trained on average distance between fingertips, spread distance between adjacent fingertips and tri-spread area between two adjacent fingertips. Fok et al. [26] achieved an average recognition rate of 93.14% using data fusion of two Leap Motion sensors and the Hidden Markov Model (HMM) classifier trained on orientation and distance ratio features (relative orientations of distal phalanges to the orientation of the palm; the ratio of the distance between a fingertip to the palm to the sum of distances between finger tips and the palm; the ratio of the distance between finger tips to the total distance among finger tips). Hisham and Hamouda [27] achieved accuracy 97.4% and 96.4% respectively on Arabic signs while using palm and bone feature sets and Dynamic Time Wrapping (DTW) for dynamic gesture recognition. Lu et al. [28] used the Hidden Conditional Neural Field (HCNF) classifier to recognize dynamic hand gestures, achieving 89.5% accuracy on two dynamic hand gesture datasets. Avola et al. [24] trained Recurrent Neural Network (RNN) on features defined as angles formed by the finger bones of the human hands, achieving over 96% of accuracy using the American Sign Language (ASL) dataset.

In VR applications, Leap Motion has been used in educational context to learn the laws of classical mechanics demonstrated by the application of physical forces on bodies and how the forces influence the motion of bodies [29], while McCallum and Boletsis [30] used Leap Motion for gesture recognition in an Augmented Reality (AR) based game for elderly. Sourial et al. [31] used Leap Motion in a virtual therapist system, which focused on helping the patient do perform physical exercises at home in a gamified environment and to provide guidance on exercising, observe the condition of a patient, adjust movement errors and evaluate the exercising achievements of the patient. Valentini and Pezutti [32] evaluated the accuracy of Leap Motion for the use in interactive VR applications such as virtual object manipulation and virtual prototyping. Pathak et al. [33] used the Leap Motion device to interplay with three-dimensional (3D) holograms by recognizing hand, finger movements and gestures. Jimenez and Schulze [34] used the Oculus Rift VR device with a Leap Motion controller for a continuous-motion text input in VR. Komiya and Nakajima [35] used Leap Motion for implementing text input in Japanese while reaching an average input speed of 43.15 CPM (characters per minute) for input of short words. Finally, Jiang et al. [36] used the fusion of signals captured using force myography (FMG) recognizing a muscular activity during hand gestures and the Leap Motion data, for grasping of virtual objects in VR. The use of moving hands for controlling the virtual space in VR games and applications has been confirmed as being important in making VR environments realistic and immersive [37].

This paper presents a system that uses a Leap Motion device to record the positions of user hands as they perform gestures and uses these data to recognize the corresponding gestures in a VR environment. The system uses the HMM classification [38], which is used to recognize gesture sequences in an unsupervised way. The system can be applied for the development of VR projects that require identification of the user's hand gestures for control of virtual objects in VR environments.

## 2. Methods

### 2.1. The Leap Motion Device and Gesture Recognition

Leap Motion has two monochrome IR cameras and three IR light emitting diodes (LEDs). These LEDs generate a 3D dot model, which is registered with the monochrome cameras. From two 2D images obtained with monochrome IR cameras Leap Motion generates a spatial pattern of user's hands.

Unlike Microsoft's Kinect that treats a complete human skeleton, Leap Motion follows only the hands of users and can predict the position of the fingers, palm of your hand or wrists in case these are occluded. Leap Motion can handle a distance of 25 to 600 mm, 150 degrees wide, allowing the user to perform gestures freely in space.

Using the IR cameras, one can set the coordinates of each hand point. In order to recognize the hand gestures, it is necessary to process a large amount of data by determining the parts of the forearm, wrist, hand and fingers. The Leap Motion software receives a 3D spatial skeleton from the 3D image, analyses it and aggregates into certain objects that hold the corresponding hand part information. The Leap Motion controller has three main hand objects: full arm, palm and fingers. A full-arm object provides information about the position of a hand in space, its length and width. The hand object holds information about the hand (left or right) position and the finger list of that hand. The most important part of Leap Motion required for gesture handling is the fingertip object. This item holds the basic bone data for each person's fingertip.

Although the Leap Motion device is capable of recognizing human hands, it cannot directly recognize the gestures displayed by users. It is only able to simulate the spatial model of a human hand but this device does not have the functionality which, based on this data, could tell when a user shows a single-pointed finger gesture. The Leap Motion device presents a 3D spatial model of a human hand (see Figure 1). With this model, one can get coordinates, turning angles for each hand, their bones or palm centre and other necessary information. If this device is always located in the same position in front of the user and the user will display the same gesture, the device will provide almost the same data with only a small error.

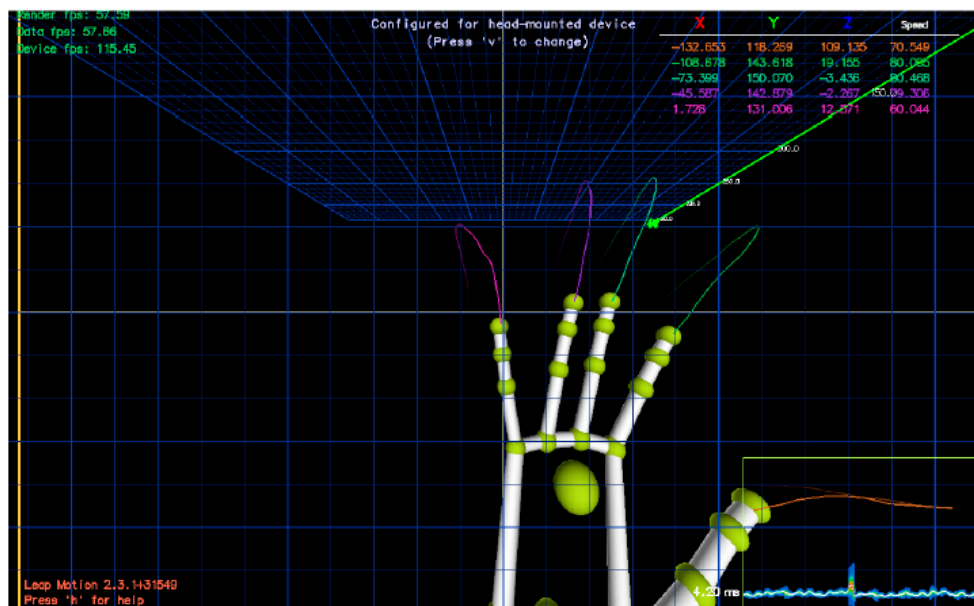


Figure 1. Leap Motion hand model in 3D.

This means that each hand gesture that does not carry any movement that can be seen both in the forearm and in the position of each finger in the space. If we had an indication of a motion gesture and we recorded the data from Leap Motion to the database, then we could use it as a pattern for recognizing that gesture. Only three Degrees of Freedom (DoF) are required for the recognition of static gestures: deviation, inclination and pitching. If the angles between the forearm and fingers are similar, the displayed static gesture is recognized. The recognition of dynamic gestures that leads to a certain movement is similar. The gesture database contains spatial data typical for each gesture. When a user is showing a dynamic gesture, the algorithm checks how the spatial data for each image frame varies. If the change is similar to the data in the database, the dynamic gesture is recognized.

## 2.2. Network Service Gesture Identification System

The amount of data needed to recognize the gesture can grow exponentially depending upon the number of gestures. For ten or more gestures, algorithms of the gesture recognition system require a considerable amount of time, on average about half a minute. However, the alphabet of ASL has thirty two different gestures (signs). As it requires large computation resources for online retraining, the system was implemented using a cloud-based network service. For implementation, we have employed microservices—a case of the service-oriented architecture (SOA) that defines an application as a stack of loosely coupled services [39].

The connection of the gesture recognition system to the network service (Figure 2), has made it possible to easily allocate the resources necessary for the system between several computers. All gesture identification data are stored in the remote gesture database. By launching several services which send requests to the same database, it has been possible to manage algorithmic training. At the same time, it is possible to record new gestures, carry out research and teach algorithms to recognize new gestures.

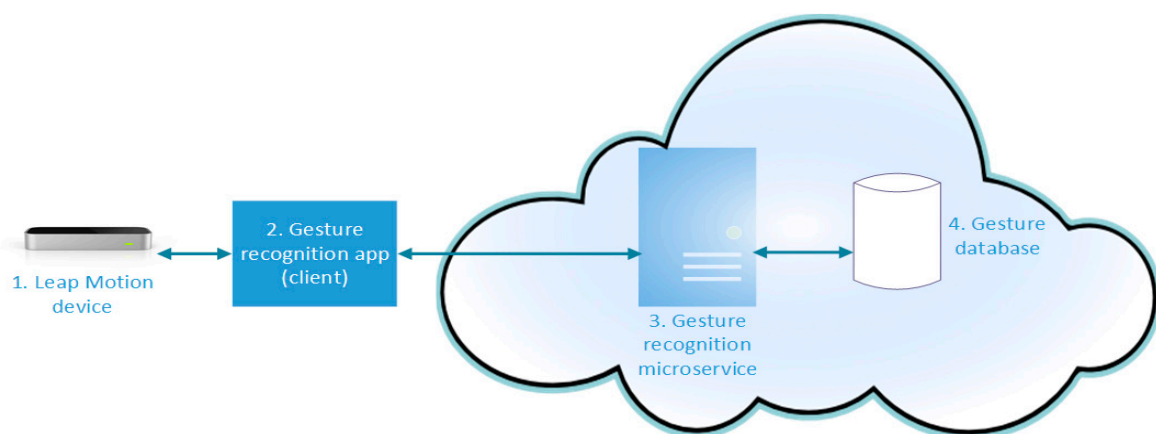


Figure 2. Outline of a system.

The network service provides an easy access to the gesture recognition system from different environments. The Leap Motion device can be used in games created using Unity or Unreal Engine gaming engines and easily integrated into any Windows application or web page. The gesture recognition system launched as a network service allows very many different systems to communicate with it. For the time being, due to the easier installation, the Simple Object Access Protocol (SOAP) was used but the system can easily be expanded to accept the Representational State Transfer (REST) data requests. This functionality would allow the gesture recognition system to be accessed from any environment. Data recorded with the Leap Motion device is stored in the Microsoft Structured Query Language (MS SQL) database, which allows for creation of personalized gesture collections as in Reference [40].

## 2.3. Gesture Identification

Leap Motion continuously displays user hand frames. The problem arises when we want to filter out the sequence of these shots when the gesture is started. If we send all the data to the gesture recognition system, the system will recognize these gestures poorly. This phenomenon is due to the fact that certain gestures may consist of several other types of gestures. This is often seen with a motion gesture. The motion gesture consists of a large number of frames, among which there are gestures without motion. To solve this problem, the states of the system are defined as follows:

- **Start.** The system is waiting for the user to start moving. If the hand started moving, the transition to the Waiting state will not start.
- **Waiting until the state changes.** If the system does not see the hand, the system returns to the start state. If the user does not move the hand, the system goes to the Stationary gesture lock state.
- **Stationary gesture lock state.** The user does not move his hand for two seconds and the gesture is fixed. Recorded hand model data is saved and converted to the gesture recognition state. If the user moves the hand in two seconds, the system's state changes to a Motion detection state.
- **Motion detection state.** If the device can not follow the user's hand, the recorded hand model data is saved and the system's state is changed to the Gesture recognition state.
- **Gesture recognition state.** Data captured in this state is sent to the gesture recognition subsystem. When the subsystem returns results, they are presented to the user and the system goes to the Data clearing state.
- **Data clearing state.** Clears unnecessary data and goes to the Start state.

#### 2.4. Feature Extraction and Pre-Processing

The Leap Motion Controller returns the data in units of real-world coordinates (measured in millimetres) that represents positions within the Leap Motion's frame of reference. The data represents the x, y and z coordinates of key hand features (wrist position, palm position, positions of metacarpal, proximal, intermediate and distal bones of fingers and fingertip positions). The 3D positional coordinates from the Leap Motion's Hand Hierarchy are illustrated in Figure 3.

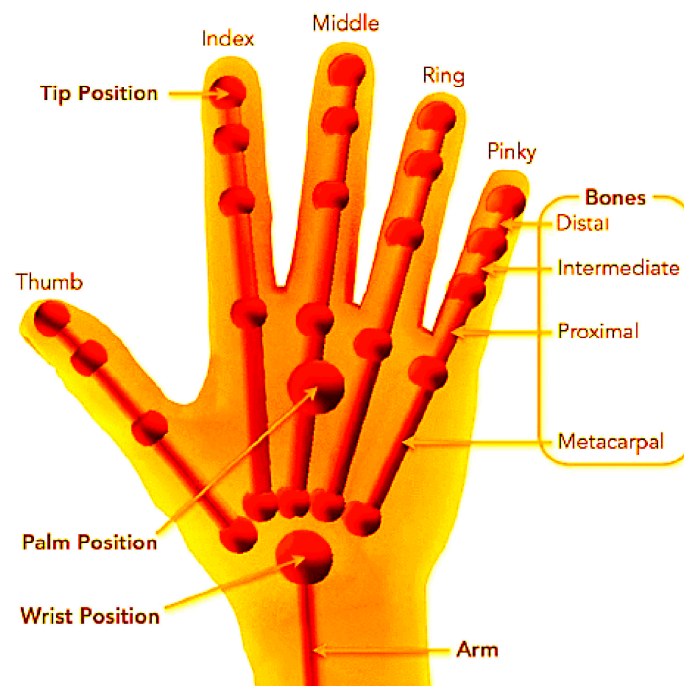


Figure 3. Positional data available from Leap Motion.

The general approach for feature extraction presented here is shown in Figure 4. We extract 4 types of hand features, that is, the 3D positions of the, fingertip distances from the hand centroid, elevations of fingertips above the plane of the palm, angles between the fingertips-to-palm-centre vectors. The fingertip angles (adopted from [41]) are angles representing the fingertip orientation projected on the palm.

The Leap Motion Controller includes the 3D positions of 11 finger joints. For each gesture we calculate the Euclidian distances between the seven main hand vertices, representing the tip positions of thumb, index, middle, ring and pinky fingers, palm position and wrist position. In all

there are 21 distances between the 7 vertices. Additionally, the angular features were generated, representing the angles between any of three different vertices, representing another 35 features. In total, 56 features (21 distance and 35 angular) are extracted. To make all features uniform, the z-score based normalization is applied, which normalizes the data by subtracting the mean and dividing it by standard deviation.

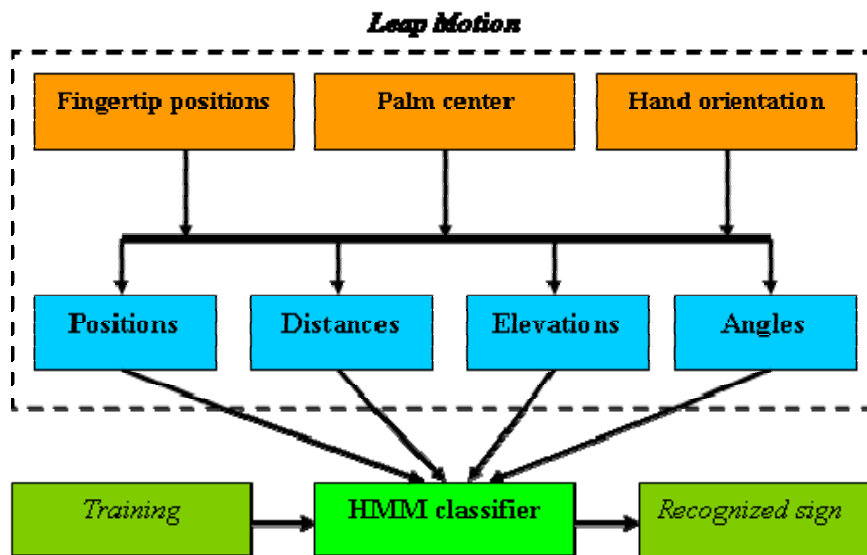


Figure 4. Pipeline of our approach.

Following [42], we describe the kinematic model of the hand movement as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi \\ \cos \theta \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi \\ -\sin \theta & \sin \varphi \cos \theta & \cos \varphi \cos \theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (1)$$

here  $X, Y, Z$  are the 3D coordinate components,  $u, v, w$  are the velocity components,  $\theta$  is the roll angle,  $\psi$  is the pitch angle and  $\varphi$  is the yaw angle of the hand.

The total velocity  $V$  of the hand's centroid is calculated as:

$$V = \sqrt{u^2 + v^2 + w^2} \quad (2)$$

The angles are calculated as follows:

$$\alpha = \arctan \frac{w}{u} \quad (3)$$

$$\beta = \arctan \frac{v}{\sqrt{v^2 + w^2}} \quad (4)$$

$$\gamma = \arccos(\cos \alpha \cos \phi) \quad (5)$$

here  $\alpha$  is the angle of attack,  $\beta$  is the angle of sideslip and  $\gamma$  is the angle of total attack.

The fingertip distances represent the distance of fingertips from the centre of the palm and are defined as:

$$D_i = |F_i - C|, i = 1, \dots, 5 \quad (6)$$

here  $F_i$  are the 3D positions of each fingertip; and  $C$  is the 3D position associated with the centre of the hand palm in the 3D frame of reference.

### 2.5. Markov Classification

A Hidden Markov Model (HMM) is formally defined as 5-tuple representing a given process with a set of states and transition probabilities between the states [43]:

$$Q = \{N, M, A, \beta, \pi\} \tag{7}$$

here  $N$  indicates the number of unique possible states which are not directly observable except through a sequence of distinct observable symbols  $M$ , also called emissions;  $\beta$  represents the discrete/continuous probabilities for these emissions,  $A$  indicates the probability of state transition and  $\pi$  are the starting probabilities.

The state  $q_1^t = \{q_1, \dots, q_t\}$  of the Markov chain is implicitly defined by a sequence  $y_1^t = \{y_1, \dots, y_t\}$  of the observed data. Given the observation sequence  $y_1^t = \{y_1, \dots, y_t\}$ , where  $y_i^t$  represents the feature vector observed at time  $i$  and a separate HMM for each gesture, then the sign language recognition problem can simply be solved by computing:

$$Q = \arg \max_i P(y_i^T, q_i^T), \tag{8}$$

here  $i$  corresponds to the  $i$ -th gesture.

The probability of the observed sequence  $P(y_1^T)$  is found using the joint probability of the observed sequence and the state sequence  $P(y_1^T, q_1^T)$  as follows:

$$\begin{aligned} P(y_1^T, q_1^T) &= P(y_T, q_T | y_1^{T-1}, q_1^{T-1}) P(y_1^{T-1}, q_1^{T-1}) \\ &= P(y_T | q_T, y_1^{T-1}, q_1^{T-1}) P(q_T | y_1^{T-1}, q_1^{T-1}) P(y_1^{T-1}, q_1^{T-1}) \\ &= P(y_T | q_T) P(q_T | q_{T-1}) P(y_1^{T-1}, q_1^{T-1}) \\ &= P(q_1) \prod_{t=2}^T P(q_t | q_{t-1}) \prod_{t=1}^T P(y_t | q_t) \end{aligned} \tag{9}$$

here  $P(q_1)$  is the initial state probability distribution of  $q$  at time 1,  $P(q_t | q_{t-1})$  is the probability of  $q$  at time  $t$  given  $q$  at time  $t - 1$ ,  $P(y_t | q_t)$  is the emission probability.

We calculate the probability  $P(y_1^t, q_t)$  of an observed partial sequence  $y_1^t$  for a given state  $q_t$  using the forward-backward algorithm as a conditional probability in the product form:

$$P(y_1^t, q_t) = P(y_t | y_1^{t-1}, q_t) P(y_1^{t-1}, q_t) \tag{10}$$

Given that

$$P(y_1^t, q_t, q_{t-1}) = P(q_t | q_{t-1}) P(y_1^{t-1}, q_{t-1}) \tag{11}$$

and

$$P(y_1^{t-1}, q_t) = \sum_{q_{t-1}} P(q_t | q_{t-1}) P(y_1^{t-1}, q_{t-1}) \tag{12}$$

we get the following equation:

$$P(y_1^t, q_t) = P(y_t | q_t) \sum_{q_{t-1}} P(q_t | q_{t-1}) P(y_1^{t-1}, q_{t-1}) \tag{13}$$

We define  $\alpha_q(t) = P(y_1^t, q)$  then the above equation is written as:

$$\alpha_q(t) = P(y_t | Q_t = q) \sum_r (Q_t = q | Q_{t-1} = r) \alpha_r(t-1) \tag{14}$$

here  $Q_t$  is the state space at time  $t$ .

We calculate the partial probability from time  $t + 1$  to the end of the sequence, given  $q_t$  as:

$$\beta_q(t) = \sum_r \beta_r(t+1) P(y_{t+1} | Q_{t+1} = r) P(Q_{t+1} = r | Q_t = q) \tag{15}$$

here  $\beta_q(t) = P(y_{t+1}^T | Q_t = q)$  and  $Q_t$  is the state at time  $t$ .

Then the probability of the observed sequence  $P(y_1^T)$  is calculated as:

$$P(y_1^T) = \sum_{q_t} P(q_t, y_1^t, y_{t+1}^T) = \sum_{q_t} \beta_{q_t}(t) \alpha_{q_t}(t) \tag{16}$$

The most likely state sequence  $q_1^T$  corresponding to a given observation sequence  $y_1^T$  is defined by the probability,  $P(q_t | y_1^T)$ , which is the product of forward-backward variables and normalized by the joint distribution of the observation sequences as the follows:

$$P(q_t | y_1^T) = \frac{P(q_t, y_1^T)}{P(y_1^T)} = \frac{P(y_1^t | q_t) P(q_t) P(y_{t+1}^T | q_t)}{P(y_1^T)} = \frac{P(y_1^t, q_t) P(y_{t+1}^T | q_t)}{P(y_1^T)} \tag{17}$$

The most likely state is measured by maximizing  $P(q_t | y_1^T)$  for  $q_t$ .

After the sequence of observations is calculated, the HMM training is performed by applying the Baum-Welch algorithm [44] to calculate the values of the transition matrix  $A$  and the emission matrix  $B$ . Following the HMM training, the gesture  $Q$  with best likelihood corresponding to the feature vector sequence  $P(y_i^T, q_i^T)$  is found using the Viterbi algorithm [44].

### 3. Experiment and Results

#### 3.1. Settings and Data

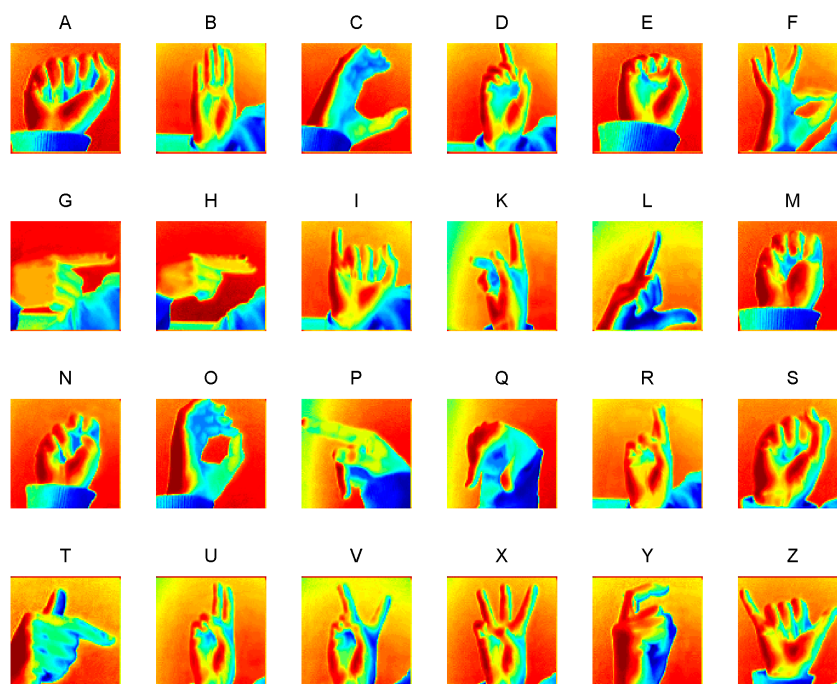
Participants in the pilot study were twelve (12) people, aged 20 to 48 (mean: 27.2), with a differing range of experience in the use of computer equipment. All subjects were healthy (with no known problems of vision or VR sickness) and inexperienced users of ASL, therefore, the subjects were given about 1 h of prior training to learn the signs of ASL as well as to familiarize with the developed system and the VR device used. For experiments, we used a conventional desktop computer with Microsoft Windows 10 and the Leap Motion device placed on the table at a normal room lighting conditions. Before the study, the participants were asked to remove the rings, watches, because they could affect the results. The output of the Leap Motion controller representing a three dimensional spatial model of subjects' hands was displayed to subjects using the Oculus Rift DK2 device.

The participants were asked to perform 24 gestures analogous to the letters of the ASL [45] (see Figure 5). The gesture of each letter was performed ten times resulting in a total of 2880 data samples. We recorded the gestures of the participants' hands in the Leap Motion environment and took pictures of real gestures shown by hand. Subsequently, the data from this study were analysed.

To evaluate the accuracy of the results, we divide the collected dataset in a train and a test sets by using the LOPO (leave-one-person-out) subject-independent cross-validation strategy. The results are averaged to obtain the resulting accuracy.

Processing and visualization of results presented in this paper was done using MATLAB R2013a (The Mathworks, Inc., Natick, MA, USA).





**Figure 5.** Alphabet Gestures for American Sign Language Alphabet.

For the text input experiments using the ASL, we used 18 pangrams (i.e., sentences using every letter of an alphabet at least once) as follows:

*A quick movement of the enemy will jeopardize six gunboats.  
 All questions asked by five watched experts amaze the judge.  
 Amazingly few discotheques provide jukeboxes.  
 Back in June we delivered oxygen equipment of the same size.  
 Few black taxis drive up major roads on quiet hazy nights.  
 Five quacking zephyrs jolt my wax bed.  
 Fix problem quickly with galvanized jets.  
 Glib jocks quiz nymph to vex dwarf.  
 How vexingly quick daft zebras jump.  
 Jackdaws love my big sphinx of quartz.  
 Jived fox nymph grabs quick waltz.  
 My girl wove six dozen plaid jackets before she quit.  
 Pack my box with five dozen liquor jugs.  
 Sphinx of black quartz, judge my vow.  
 The five boxing wizards jump quickly.  
 The quick brown fox jumps over a lazy dog.  
 The wizard quickly jinxed the gnomes before they vaporized.  
 Woven silk pyjamas exchanged for blue quartz.*

The pangrams have been used for typing using gesture recognition by Leap Motion before [46].

### 3.2. Results

An example of gestures and their representation by Leap motion is shown in Figure 6.

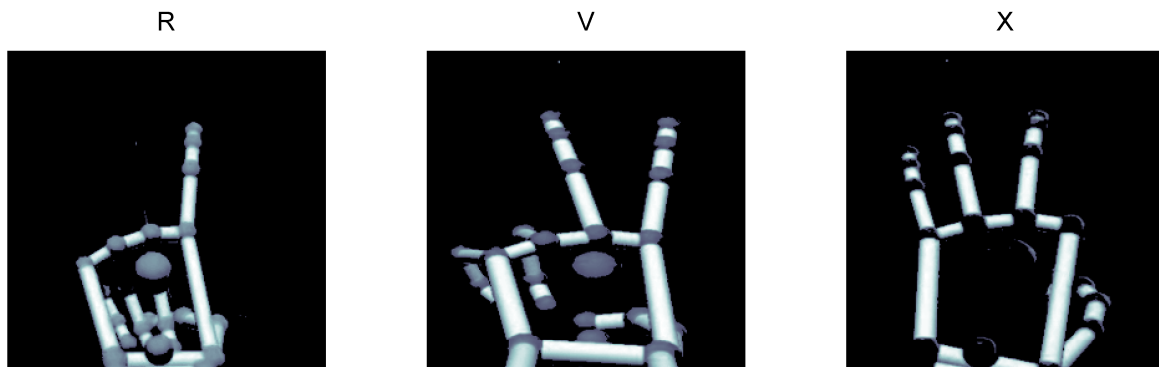


Figure 6. Example of signs recognized.

The experiments were implemented using stratified 10-fold cross validation and assessed using the macro-accuracy (averaged over classes and folds) performance measure [47]. The results of the gesture recognition are presented in Figure 7. The averaged recognition accuracy (mean  $\pm$  SD) achieved is  $86.1 \pm 8.2\%$ .

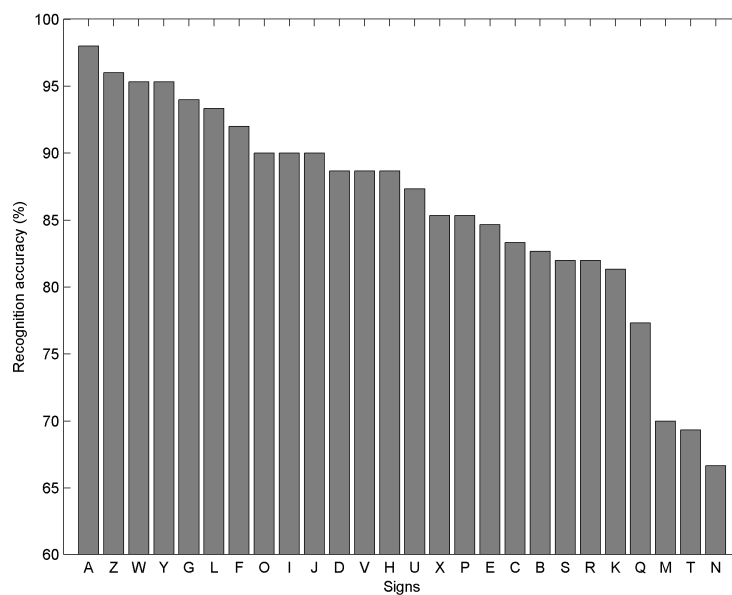


Figure 7. Accuracy of ASL Gesture Recognition.

The confusion plot of classification is presented in Figure 8. We have obtained a true positive rate (TPR) of 0.854, an F-measure of 0.854 and a Cohen’s kappa [48] value of 0.987.

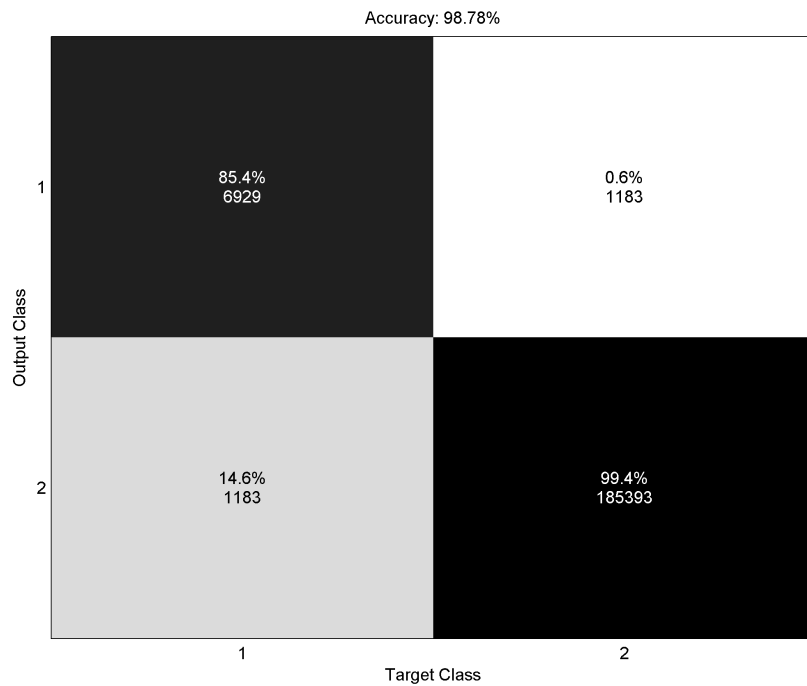


Figure 8. Confusion matrix of gesture classification (1-vs-all).

In the typing experiment, during the training pre-stage the subjects learned how to use the research system consisting of a software application and the Leap Motion Controller. Then their task was to type three times each of the pangrams. The pangrams were presented in a random order. In case of error, the subjects were instructed to ignore errors and keep typing the phrases.

We used the words per minute (WPM) as a performance measure and the minimum string distance (MSD) as an error rate as suggested in Reference [49]. The obtained results are presented in Figures 9 and 10 and are summarized as follows (mean ± SD): 3.09 ± 0.53 WPM and 16.58 ± 5.52 MSD.

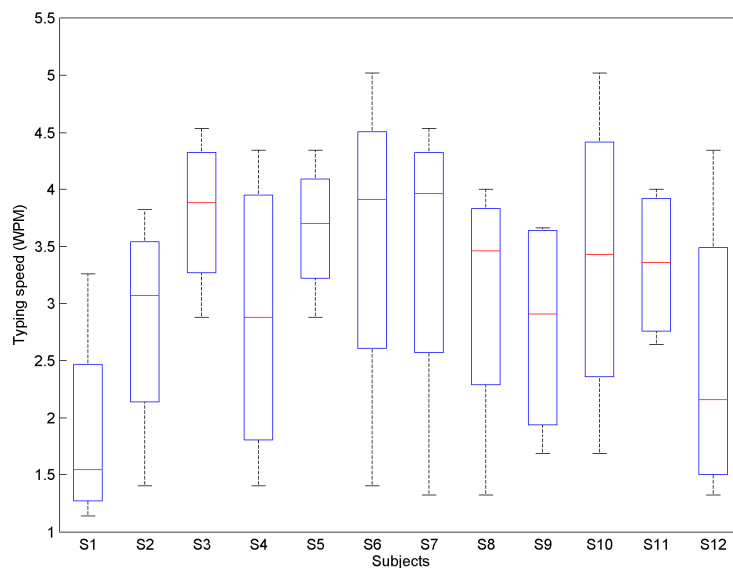


Figure 9. Gesture typing performance of all subjects.

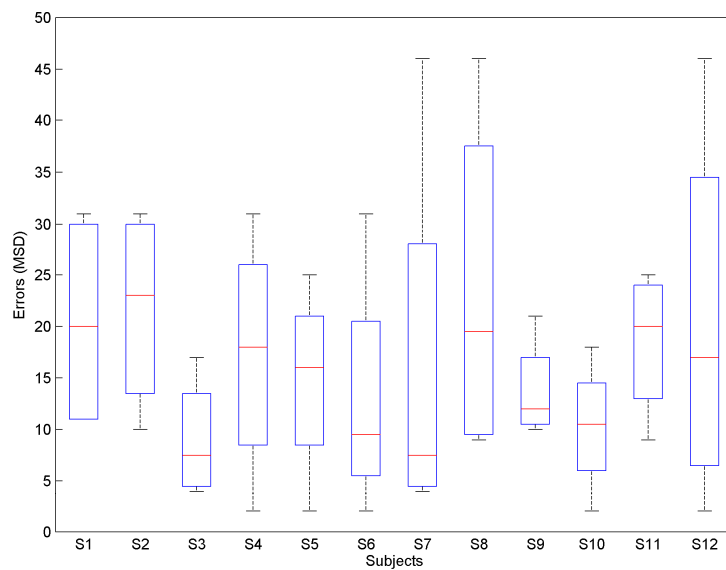


Figure 10. Gesture typing error rate of all subjects.

We performed the linear regression analysis of relationship between gesture typing speed and error rate and found the following linear relationship, which is within 95% of confidence:

$$msd = -5.2 * wpm + 32.7 \tag{18}$$

here *msd* is the minimum string distance and *wpm* is the words per minute.

The results, demonstrated in Figure 11, show that more proficient users demonstrate both higher performance and lower error rate and vice versa.

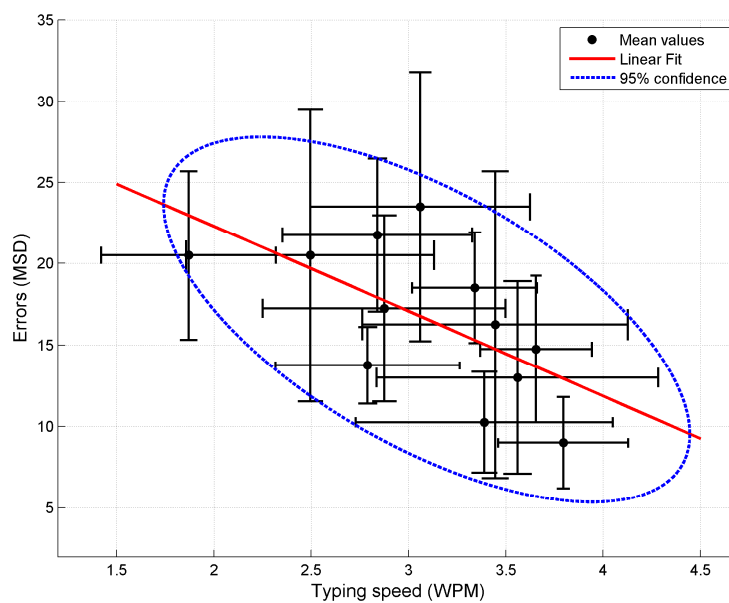


Figure 11. Linear regression analysis of relationship between gesture typing performance and error rate with linear trend and 95% confidence ellipse shown.

### 3.3. Evaluation

We have achieved 86.1% accuracy of recognition of ASL signs. These results are in the range of accuracy achieved by other authors as indicated by a survey in Reference [50]. Note that the subjects participating our study were not experienced users of sign languages, therefore, the quality of sign

gesturing could have adversely affected the accuracy of recognition. Other authors used stand-alone letters or words in ASL for training, whereas we used complete sentences (pangrams), which has been a more difficult task for subjects. Moreover, the view of gestures was presented to subjects as 3D models of hands using a head-mounted Oculus Rift DK display, so the subjects were not able to view their physical hands during the experiment and this could have made the gesturing task more difficult as well.

After analysing the recorded gesture recognition data, we have observed that there are problems in detecting the gap between the fingers. Small (one centimetre or less) gap is poorly understood and, for example, it is difficult to discriminate between the gestures of C and O signs. Gestures that require a precise thumb position are also more difficult to determine. Thumbs are often covered with other fingers, which decreases the accuracy of the recognition of the E, M, N, T, H, K, S, V and X signs. The recognition of some gestures requires a very precise 3D image of the hand. This is evident in the gesture of the P sign, when the fingers are only partially folded (not scrambled in the fist) but the device from the palm of the hand has identified the fingers as completely curved. This problem also occurred with the letter R gesture, that is, the fingers have to be crossed but they are presented as a concave and such a gesture corresponds to the letter U in the sign language. In some cases, partially folded fingers are treated as completely folded. Our study revealed the gaps in the algorithm of the Leap Motion device used for gesture analysis. Problems occur when the Leap Motion device does not see some fingers. Then the fingertip positions cannot be captured and the gestures are identified incorrectly.

The gesture recognition was implemented as a microservice over the Internet. The sending of data from the Leap Motion device over the network to the microservice does not significantly increase the duration of the gesture recognition. On average, the size of Leap Motion gesture data batch ranges from 500 to 1500 bytes. Transmission of this amount to a network service does not require a lot of resources or speed. The greatest slowdown occurs at the network service itself by filtering these data and performing the gesture recognition functions. The entire process takes no more than 200 ms.

#### 4. Conclusions

Gesture recognition can be applied to various areas that are not suitable for typical data entry such as VR environments. In this paper, we have presented a system that can learn gestures by using the data from the Leap Motion device and the Hidden Markov classification (HMC) algorithm. We have achieved the gesture recognition accuracy (mean  $\pm$  SD) is  $86.1 \pm 8.2\%$  and the gesture typing speed of  $3.09 \pm 0.53$  words per minute, when recognizing the gestures of the American Sign Language (ASL).

We have identified several problems of using the Leap Motion technology for gesture recognition. First of all, if the user's own fingers are invisible to IR cameras, Leap Motion makes mistakes in predicting their position, for example, the hand is depicted with folded fingers when they are stretched. Similarly, the position of the thumb when it is pressed against the palm or between the other fingers is poorly defined and cannot be reliably used to identify the gesture.

**Author Contributions:** Formal analysis, M.W.; Investigation, T.B.; Methodology, T.B.; Software, A.V. and M.T.; Supervision, T.B.; Validation, R.M.; Visualization, R.D.; Writing—original draft, R.D.; Writing—review & editing, R.M.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Bachmann, D.; Weichert, F.; Rinkeauer, G. Review of three-dimensional human-computer interaction with focus on the leap motion controller. *Sensors* **2018**, *18*. [[CrossRef](#)] [[PubMed](#)]
2. Dawes, F.; Penders, J.; Carbone, G. Remote Control of a Robotic Hand Using a Leap Sensor. In *The International Conference of IFToMM ITALY*; Springer International Publishing: Cham, Switzerland, 2019; pp. 332–341. [[CrossRef](#)]

3. Rocchetti, M.; Marfia, G.; Semeraro, A. Playing into the wild: A gesture-based interface for gaming in public spaces. *J. Vis. Commun. Image Represent.* **2012**, *23*, 426–440. [[CrossRef](#)]
4. Darabkh, K.A.; Alturk, F.H.; Sweidan, S.Z. VRCDEA-TCS: 3D virtual reality cooperative drawing educational application with textual chatting system. *Comput. Appl. Eng. Educ.* **2018**, *26*, 1677–1698. [[CrossRef](#)]
5. Yang, H.-D. Sign Language Recognition with the Kinect Sensor Based on Conditional Random Fields. *Sensors* **2015**, *15*, 135–147. [[CrossRef](#)] [[PubMed](#)]
6. Butt, A.H.; Rovini, E.; Dolciotti, C.; De Petris, G.; Bongioanni, P.; Carboncini, M.C.; Cavallo, F. Objective and automatic classification of parkinson disease with leap motion controller. *Biomed. Eng. Online* **2018**, *17*, 168. [[CrossRef](#)] [[PubMed](#)]
7. Cai, S.; Zhu, G.; Wu, Y.; Liu, E.; Hu, X. A case study of gesture-based games in enhancing the fine motor skills and recognition of children with autism. *Interact. Learn. Environ.* **2018**, *26*, 1039–1052. [[CrossRef](#)]
8. Cohen, M.W.; Voldman, I.; Regazzoni, D.; Vitali, A. Hand rehabilitation via gesture recognition using leap motion controller. In Proceedings of the 11th International Conference on Human System Interaction, HIS, Gdansk, Poland, 4–6 July 2018; pp. 404–410. [[CrossRef](#)]
9. Morando, M.; Ponte, S.; Ferrara, E.; Dellepiane, S. Definition of motion and biophysical indicators for home-based rehabilitation through serious games. *Information* **2018**, *9*, 105. [[CrossRef](#)]
10. Qingchao, X.; Jiangang, C. The application of leap motion in astronaut virtual training. *Iop Conf. Ser. Mater. Sci. Eng.* **2017**, *187*. [[CrossRef](#)]
11. Pulijala, Y.; Ma, M.; Ayoub, A. VR surgery: Interactive virtual reality application for training oral and maxillofacial surgeons using oculus rift and leap motion. *Serious Games Edu. Appl.* **2017**, *11*, 187–202. [[CrossRef](#)]
12. Gleeson, B.; MacLean, K.; Haddadi, A.; Croft, E.; Alcazar, J. Gestures for industry: Intuitive human-robot communication from human observation. In Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI '13), Tokyo, Japan, 3–6 March 2013; pp. 349–356. [[CrossRef](#)]
13. Mohandes, M.; Aliyu, S.; Deriche, M. Arabic sign language recognition using the leap motion controller. *IEEE Int. Symp. Ind. Electron.* **2014**, 960–965. [[CrossRef](#)]
14. Alfonse, M.; Ali, A.; Elons, A.S.; Badr, N.L.; Aboul-Ela, M. Arabic sign language benchmark database for different heterogeneous sensors. In Proceedings of the 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), Marrakech, Morocco, 21–23 December 2015; pp. 1–9. [[CrossRef](#)]
15. Chavan, P.; Ghorpade, T.; Padiya, P. Indian sign language to forecast text using leap motion sensor and RF classifier. In Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, 18–19 March 2016; pp. 1–5. [[CrossRef](#)]
16. Naglot, D.; Kulkarni, M. ANN based indian sign language numerals recognition using the leap motion controller. In Proceedings of the International Conference on Inventive Computation Technologies, ICICT 2016, Coimbatore, India, 26–27 August 2016; Volume 2, pp. 1–6. [[CrossRef](#)]
17. Demircioğlu, B.; Bülbül, G.; Köse, H. Turkish sign language recognition with leap motion. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Turkey, 16–19 May 2016; pp. 589–592. [[CrossRef](#)]
18. Simos, M.; Nikolaidis, N. Greek sign language alphabet recognition using the leap motion device. In Proceedings of the 9th Hellenic Conference on Artificial Intelligence, Thessaloniki, Greece, 18–20 May 2016. [[CrossRef](#)]
19. Tumsri, J.; Kimpan, W. Thai sign language translation using leap motion controller. In Proceedings of the International Multi Conference of Engineers and Computer Scientists, Hong Kong, China, 15–17 March 2017; Volume 2227, pp. 46–51.
20. Anwar, A.; Basuki, A.; Sigit, R.; Rahagiyanto, A.; Zikky, M. Feature extraction for indonesian sign language (SIBI) using leap motion controller. In Proceedings of the 2017 21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand, 15–18 November 2017; pp. 196–200. [[CrossRef](#)]
21. Chuan, C.; Regina, E.; Guardino, C. American sign language recognition using leap motion sensor. In Proceedings of the 2014 13th International Conference on Machine Learning and Applications, Detroit, MI, USA, 3–6 December 2014; pp. 541–544. [[CrossRef](#)]

22. Mapari, R.B.; Kharat, G. American static signs recognition using leap motion sensor. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, Udaipur, India, 4–5 March 2016. [\[CrossRef\]](#)
23. Chong, T.; Lee, B. American sign language recognition using leap motion controller with machine learning approach. *Sensors* **2018**, *18*, 3554. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Avola, D.; Bernardi, M.; Cinque, L.; Foresti, G.L.; Massaroni, C. Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Trans. Multimed.* **2018**. [\[CrossRef\]](#)
25. Ameer, S.; Khalifa, A.B.; Bouhlel, M.S. A comprehensive leap motion database for hand gesture recognition. In Proceedings of the 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet, Tunisia, 18–20 December 2016; pp. 514–519. [\[CrossRef\]](#)
26. Fok, K.; Ganganath, N.; Cheng, C.; Tse, C.K. A real-time ASL recognition system using leap motion sensors. In Proceedings of the 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Xi'an, China, 17–19 September 2015; pp. 411–414. [\[CrossRef\]](#)
27. Hisham, B.; Hamouda, A. Arabic static and dynamic gestures recognition using leap motion. *J. Comput. Sci.* **2017**, *13*, 337–354. [\[CrossRef\]](#)
28. Lu, W.; Tong, Z.; Chu, J. Dynamic hand gesture recognition with leap motion controller. *IEEE Signal Process. Lett.* **2016**, *23*, 1188–1192. [\[CrossRef\]](#)
29. Bhardwaj, A.; Grover, A.; Saini, P.; Singh, M. Contact dynamics emulation using leap motion controller. In *International Conference on Advances in Computing and Data Sciences*; Springer: Singapore, 2017; Volume 721, pp. 262–271. [\[CrossRef\]](#)
30. McCallum, S.; Boletsis, C. Augmented Reality & Gesture-based Architecture in Games for the Elderly. *Stud. Health Technol. Inform.* **2013**, *189*, 139–144. [\[PubMed\]](#)
31. Sourial, M.; Elnaggar, A.; Reichardt, D. Development of a virtual coach scenario for hand therapy using LEAP motion. In Proceedings of the 2016 Future Technologies Conference (FTC), San Francisco, CA, USA, 6–7 December 2017; pp. 1071–1078. [\[CrossRef\]](#)
32. Valentini, P.P.; Pezzuti, E. Accuracy in fingertip tracking using leap motion controller for interactive virtual applications. *Int. J. Interact. Des. Manuf.* **2017**, *11*, 641–650. [\[CrossRef\]](#)
33. Pathak, V.; Jahan, F.; Fruitwala, P. Proposed system on gesture controlled holographic projection using leap motion. In *International Conference on Information and Communication Technology for Intelligent Systems (ICTIS 2017)*; Springer International Publishing: Cham, Switzerland, 2018; Volume 1, pp. 524–530. [\[CrossRef\]](#)
34. Jimenez, J.G.; Schulze, J.P. Continuous-Motion Text Input in Virtual Reality. *Electron. Imaging* **2018**, 450-1–450-6. [\[CrossRef\]](#)
35. Komiya, K.; Nakajima, T. A Japanese input method using leap motion in virtual reality. In Proceedings of the Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU), Toyama, Japan, 3–5 October 2017. [\[CrossRef\]](#)
36. Jiang, X.; Xiao, Z.G.; Menon, C. Virtual grasps recognition using fusion of Leap Motion and force myography. *Virtual Real.* **2018**, *22*, 297–308. [\[CrossRef\]](#)
37. Lee, Y.S.; Bong-Soo Sohn, B.S. Immersive Gesture Interfaces for Navigation of 3D Maps in HMD-Based Mobile Virtual Environments. *Mob. Inf. Syst.* **2018**, 2585797. [\[CrossRef\]](#)
38. Min, B.; Yoon, H.; Soh, J.; Yang, Y.; Ejima, T. Hand gesture recognition using hidden markov models. *IEEE Int. Conf. Syst. Man Cybern.* **1997**, *5*, 4232–4235. [\[CrossRef\]](#)
39. Pautasso, C. Microservices in Practice, Part 1: Reality Check and Service Design. *IEEE Softw.* **2017**, *34*, 91–98. [\[CrossRef\]](#)
40. Preventis, A.; Stravoskoufos, K.; Sotiriadis, S.; Petrakis, E.G.M. Interact: Gesture Recognition in the Cloud. In Proceedings of the IEEE/ACM 7th International Conference on Utility and Cloud Computing, London, UK, 8–11 December 2014; pp. 501–502. [\[CrossRef\]](#)
41. Marin, G.; Dominio, F.; Zanuttigh, P. Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimed. Tools Appl.* **2016**, *75*, 14991–15015. [\[CrossRef\]](#)
42. Ma'touq, J.; Hu, T.; Haddadin, S. Sub-millimetre accurate human hand kinematics: From surface to skeleton. *Comput. Methods Biomech. Biomed. Eng.* **2018**, *21*, 113–128. [\[CrossRef\]](#) [\[PubMed\]](#)
43. Fink, G.-A. *Markov Models for Pattern Recognition*, 2nd ed.; Springer: London, UK, 2008; pp. 71–106.

44. Rabiner, L.R. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. Eee* **1989**, *77*, 257–286. [[CrossRef](#)]
45. Liddell, S.K. *Grammar, Gesture, and Meaning in American Sign Language*; Cambridge University Press: Cambridge, UK, 2003; pp. 1–384. [[CrossRef](#)]
46. Dobosz, K.; Buchczyk, K. One-Handed Braille in the Air. In *International Conference on Computers Helping People with Special Needs ICCHP 2018, Lecture Notes in Computer Science*; Springer International Publishing: New York, NY, USA, 2018; pp. 322–325. [[CrossRef](#)]
47. Sokolova, M.; Lapalme, G. A Systematic Analysis of Performance Measures for Classification Tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [[CrossRef](#)]
48. Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [[CrossRef](#)]
49. Soukoreff, R.W.; MacKenzie, I.S. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Ft. Lauderdale, FL, USA, 5–10 April 2003; ACM: New York, NY, USA, 2003; pp. 113–120. [[CrossRef](#)]
50. Walde, A.S.; Shiurkar, U.D. Sign Language Recognition Systems: A Review. *Int. J. Recent Res. Asp.* **2017**, *4*, 451–456.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).