Kybernetes

Posture labeling based gesture classification for Turkish sign language using depth values

Ediz Saykol Halit Talha Türe Ahmet Mert Sirvanci Mert Turan

Posture labeling based gesture classification for Turkish sign language using depth values

Ediz Saykol, Halit Talha Türe, Ahmet Mert Sirvanci and Mert Turan
Department of Computer Engineering, Beykent University, Istanbul, Turkey

Abstract

Purpose – The purpose of this paper to classify a set of Turkish sign language (TSL) gestures by posture labeling based finite-state automata (FSA) that utilize depth values in location-based features. Gesture classification/recognition is crucial not only in communicating visually impaired people but also for educational purposes. The paper also demonstrates the practical use of the techniques for TSL.
Design/methodology/approach – Gesture classification is based on the sequence of posture labels that are assigned by location-based features, which are invariant under rotation and scale. Grid-based signing space clustering scheme is proposed to guide the feature extraction step. Gestures are then recognized by FSA that process temporally ordered posture labels.
Findings – Gesture classification accuracies and posture labeling performance are compared to k-nearest neighbor to show that the technique provides a reasonable framework for recognition of TSL gestures. A challenging set of gestures is tested, however the technique is extendible, and extending the training set will increase the performance.
Practical implications – The outcomes can be utilized as a system for educational purposes especially for visually impaired children. Besides, a communication system would be designed based on this framework.
Originality/value – The posture labeling scheme, which is inspired from keyframe labeling concept of video processing, is the original part of the proposed gesture classification framework. The search space is reduced to single dimension instead of 3D signing space, which also facilitates design of recognition schemes. Grid-based clustering scheme and location-based features are also new and depth values are received from Kinect. The paper is of interest for researchers in pattern recognition and computer vision.

Keywords Classification, Image processing, Automata theory, Kinect sensor

Paper type Research paper

1. Introduction

A gesture is defined as a form of visual communication in which the actions and relative positions of body parts correspond to particular messages possibly in a temporal sequence of postures. Due to the possibility of detecting visual communication primitives, gesture recognition has become one of the trendy topics in the recent years. The output of automated gesture recognition/classification can be used in training the persons having hearing disabilities as well as it helps them communicate with persons unfamiliar to sign languages. Since the video data has become ubiquitous, many systems have been proposed to automate gesture recognition process.

Most of the existing techniques utilize low-level features of the human body to train machine learning algorithms, and later use this trained set for classification and recognition purposes. For example, a recognition scheme for Chinese sign language based on Hidden Markov Models (HMM) is presented in Gao *et al.* (2004). In Shanableh *et al.* (2007), the authors present a gesture recognition technique that uses spatio-temporal features for Arabic sign language. There are also techniques to recognize American sign language via utilizing the low-level features that are obtained from sensory gloves

Kybernetes
Vol. 45 No. 4, 2016
pp. 604-621

0368-492X
DOI 10.1108/K-04-2015-0107

(Öz and Leu, 2011). A gesture can also be modeled as a temporally ordered set of states representing spatial information in the scene. In Davis and Shah (1999), a finite state machine is used to model 4 distinct phases of a generic human-hand gesture. There are also similar techniques using finite state machines for gesture recognition purposes (e.g. Bobick and Wilson, 1997; Hong *et al.*, 2000a, b; Yeasin and Chaudhuri, 2000). A method with the lack of sensory gloves for tracking hand movements is also presented in Davis and Shah (1994) utilizing a set of 3D cylindrical models. An extensive survey can be found in Mitra and Acharya (2007), which also mention that the significant challenge remaining is to alleviate certain restrictions on lighting conditions and design of a specialized hardware. Moreover, gesture segmentation also remained as an open problem with multiple approaches in action recognition research (Weinland *et al.*, 2011).

Along with the increase in the accessibility of 3D sensors, this topic has become more interesting via utilizing the depth values to contribute the gesture recognition process. As argued in Khoshelham and Elberink (2012), Kinect provides a platform for action recognition problems with high accuracy rates. In Agarwal and Thakur (2013), the feature matrix based on depth values is trained using a multi-class Support Vector Machine (SVM) classifier to demonstrate a typical use of these ideas. There are also various techniques aiming at different sign languages to alleviate most of the limitations of 2D. In Chai *et al.* (2013), a system to recognize Chinese sign language is presented to demonstrate the possibility of sign-language recognition with low-cost 3D sensors using the body tracking features of Kinect. Recognition of American sign language with Kinect is presented in Zafrulla *et al.* (2011), Keskin *et al.* (2011) using 3D hand model representing the hand with 21 parts. In Akram *et al.* (2012), authors present a method using Kinect for recognition of isolated Swedish sign language signs. Discriminative Exemplar Coding (DEC) approach is proposed in Sun *et al.* (2013) to model various signs by utilizing Kinect. In a very recent study (Lee *et al.*, 2016), a Kinect-based system is proposed for Taiwanese sign language using HMM on trajectory of the hand movement, and a trained SVM to recognize the hand shapes. There is also a Turkish sign language (TSL) recognition system that uses spatio-temporal features with Kinect (Memiş and Albayrak, 2013). A recent study reviews the literature for automatic recognition of Arabic sign language (Mohandes *et al.*, 2014). Techniques based on pure images, and techniques utilizing various sensors are discussed in this survey, along with the presentation of the remaining challanges.

Based on the above observations, we focus on TSL and employ a set of Kinect-based features. A grid-based signing space clustering scheme is employed during the extraction of cluster numbers as features. A posture labeling algorithm is proposed to recognize a predefined set of gestures in TSL using finite-state automata (FSA). The labels given to the postures are used to classify the gestures with respect to a known vocabulary. The overall gesture classification process is given in Figure 1. The phonologic properties and linguistic nature of TSL is investigated (e.g. Arik, 2012), and a set of challenging gestures is chosen to evaluate our technique. It is quite obvious that our techniques can be tailored to other sign languages by providing an appropriate posture analysis and FSA design for each gesture. Our scheme can also be extended by adding a new gesture to the known vocabulary in a similar fashion.

The contributions of our posture labeling based gesture classification scheme can be summarized as follows:

- Posture labeling technique is proposed to represent the gestures as a spatio-temporal sequence of distinct poses. Once the posture detection algorithm is executed at a frame, a similarity-based metric is used to assign a label for the posture.

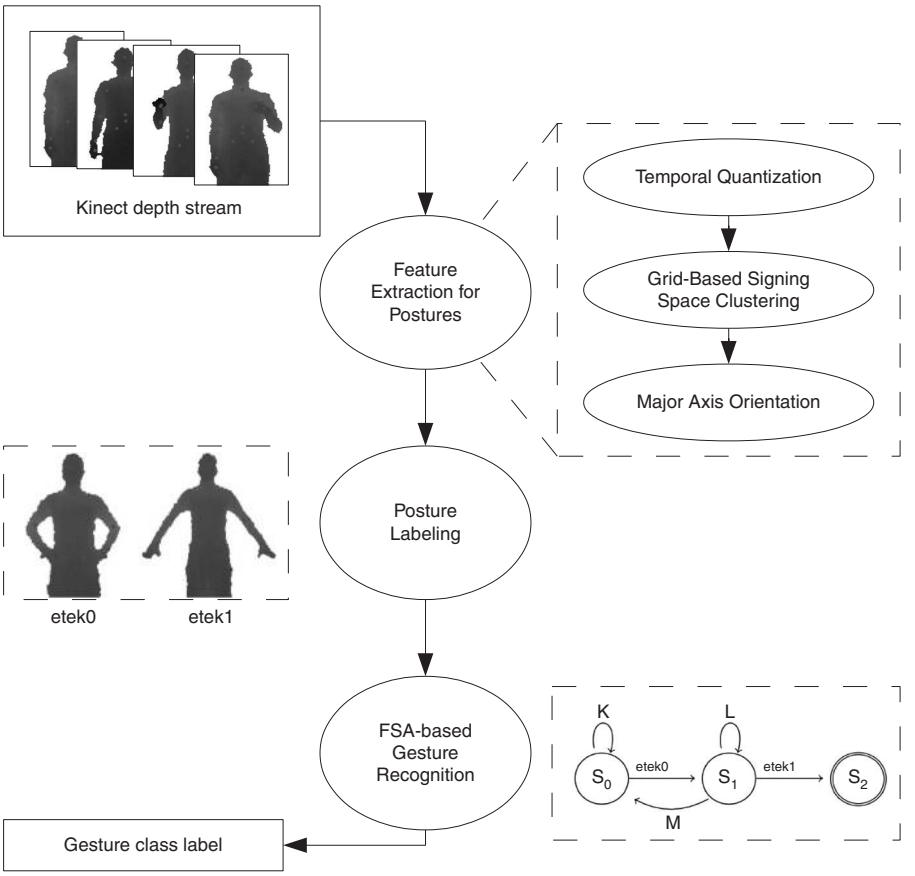


Figure 1.
The flow of execution in the posture labeling based gesture classification process

Notes: The input Kinect depth stream is passed through feature extraction step. Features based on grid-based signing space clustering scheme are used to assign a label for a frame that includes a posture. Then, finite state machine based recognition schemes are used to classify the gesture with respect to the known vocabulary

Temporal quantization is employed to be robust under sudden changes and variations. Then, this temporal sequence of posture labels is used by FSA to recognize the known gestures of TSL. Separate FSA is designed for each gesture in the known vocabulary. The posture labeling scheme is inspired from (Şaykol *et al.*, 2010), where a similar technique is shown to be effective in classifying a set of predefined video surveillance events.

- The 3D signing space is partitioned by the proposed grid-based signing space clustering scheme, where the body parts of the signer is divided into sub-parts by a ratio, which is computed by the shoulder and hip joint locations. In addition to that, the line segment connecting the left and right shoulder locations is used as the major axis of the torso, and the features are extracted after major axis orientation step. This way of signing space clustering and feature extraction provides rotation, position and scale invariance, and increases the robustness of the classification scheme.

- The overall classification scheme deals only with the posture labels instead of low-level features and spatio-temporal predicates. Hence, the search space and processing times are significantly reduced by lowering the total amount of data throughout the gesture classification process.

The remaining of the paper is organized as follows: Section 2 briefly summarizes the related studies on gesture recognition for various sign languages. Our posture labeling based gesture classification scheme is presented in Section 3 along with the major contributions of feature extraction, posture labeling, and FSA design for each gesture. The performance experiments and the explanations on the data set is given in Section 4 with the specialized tools devised for posture visualization, analysis, and gesture classification evaluations. Finally, Section 5 concludes the paper and states future work.

2. Related work

Gesture recognition systems for various sign languages have been proposed in the literature. An extensive survey on gesture recognition is presented in Mitra and Acharya (2007). Along with the accessibility of the depth sensors (e.g. Kinect), utilizing depth information to recognize/classify gestures has become popular. Here, a related literature review is to be provided to summarize various techniques using 2D/3D data in recognizing gestures.

Recognition of isolated Arabic sign language gestures is proposed in Shanableh *et al.* (2007) that uses spatio-temporal features in a serial execution. First, the temporal features of a video-based gesture are extracted through forward, backward, and bidirectional predictions. Then these predictions are accumulated into a single image. The experiments show that using nearest-neighbor metric yields comparable results to the classical HMM-based scheme, since the recognition rates range from 97 to 100 percent.

Linguistic sub-units are utilized to recognize sign language gestures in Cooper *et al.* (2012). Appearance data gathered from 2D/3D tracking is used in learning phase of three types of sub-units, which are then combined to classify sign gestures using Markov models. Experimentally, the approach is found to be robust to noise, and it performs well in signer independent tests with improved recognition rates to 76 from 54 percent.

A method utilizing a single depth image to predict 3D locations of skeletal joints is given in (Shotton *et al.*, 2011). A pixel-based body-parts model is used to classify human poses without temporal data. A large training data set is formed and the experiments show that the pixel-level classification scheme is invariant under body-parts, and the 3D joint predictions are accurate and stable.

A gesture segmentation technique based on Kinect depth data is mentioned in Bhattacharya *et al.* (2012). The technique has three steps; first, gesture classification is carried out from a known vocabulary based on the choice of SVM or linear kernel. Then, the technique is extended to detect and classify a gesture. Last, a rule-based filtering mechanism is used to eliminate the movements that were not intentional gestures. A set of aircraft marshaling gestures (e.g. liftoff, land) are used during the experiments.

A recognition technique for Swedish sign language is presented in Akram *et al.* (2012) for training the children with language disorders. The hands and face of the signer is captured via Kinect based on skin color and depth information. The 3D motion of the hands relative to the torso are used to train HMMs for classification. Performance tests show that 94 challenging words can be detected with a precision of up to 94 percent, and this percentage reduces to 47 percent when the features are utilized to be signer independent.

A TSL recognition system using spatio-temporal features is presented in Memiş and Albayrak (2013). The system uses video sequences and Kinect depth maps, and utilizes cumulative motion images that are generated based on motion variances. These motion images represent the temporal characteristics of dynamic signs and the whole motions of signers. Spatial features are obtained by 2D discrete cosine transform (DCT), which is applied to video data and depth data separately, and the feature vectors are formed by combining a certain amount of DCT coefficients with higher energy via zig-zag scanning. The recognition process employs k-nearest neighbor (kNN) with Manhattan distance, yielding a recognition rate about 90 percent on a sign database containing 1,002 signs that belong to 111 words in three different categories.

In Jaemin *et al.* (2013) and Takimoto *et al.* (2013), the authors propose a gesture recognition system using Kinect depth data. The system involves an extraction of hand shape features based on gradient value instead of conventional 2D shape features, and arm movement features based on angles between each skeletal joints. The depth data and 3D coordinates of six joints are utilized for recognition, where the hand joint position is used to extract hand shape features. Arm movements and hand gestures are recognized by utilizing a HMM. Evaluations are performed by using Japanese sign language gestures to validate the effectiveness of the technique.

A Kinect-based system for sign language recognition and verification for educational games for deaf children is presented in Zafrulla *et al.* (2011). The main motivation is to improve interactivity, user comfort, system robustness, system sustainability, and ease of deployment. The experiments are carried out on 1,000 American sign language phrases, and the Kinect-based system yields 51.5 and 76.12 percent sentence verification rates when the users were seated and standing, respectively.

Hand is smaller when compared to the entire human body, and more complex articulations are possible. Hence, hand gesture research is very challenging. An informative description of the hand poses can be used for gesture recognition, and the problem becomes more interesting with depth data (Ren *et al.*, 2011, 2013; Phadtare *et al.*, 2012; Nguyen *et al.*, 2013; Dominio *et al.*, 2013). In a typical model, the Kinect depth data is analyzed to fit the plane of hand point region, and the normal to this plane is defined as the orientation of the palm. Then, the 3D shape context is used to determine the hand shape by comparing it to the shapes in the database, and found to be correct in varying poses.

Due to the hardness of designing multi-dimensional features for traditional neural networks, convolutional neural networks (CNNs) are used in the literature. In Pigou *et al.* (2015), instead of constructing complex handcrafted features, CNNs are utilized to automate the process of feature construction for gesture recognition. The authors report that 20 gestures from the Italian sign language is successfully recognized with CNNs. In Huang *et al.* (2015), a novel 3D CNN is proposed to extract discriminative spatial-temporal features without any prior feature design. Color, depth, and body joints suggested by Kinect are used to recognize the gestures via integrating color, depth and trajectory information.

3. Posture labeling based gesture classification

Posture labeling is the process of determining the label of posture performed by a signer in a frame. These frames are then labeled with the corresponding postures, hence a temporally ordered event sequence representation is used to recognize the gestures in TSL. To this end, we design FSA schemes to classify gestures with respect to the known vocabulary. Having given a brief information on the basics of Kinect

depth data, we first focus on the extraction of the rotation, position, and scale invariant features for posture detection in a frame. Then, the posture labeling algorithm is explained along with the grid-based signing space clustering scheme. FSA based gesture classification is presented last, which uses the temporal posture label sequence representation to recognize a known gesture.

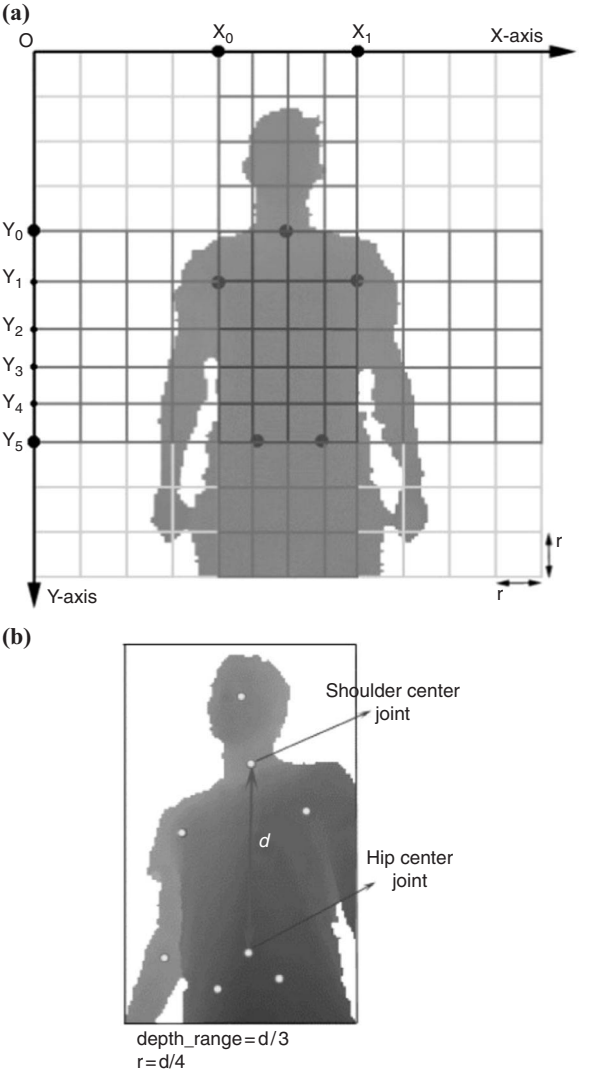
3.1 Feature extraction for postures

The Kinect sensor is capable of providing depth values, which are in fact distance measurements from Kinect to the salient object. Kinect can also provide human skeletal data, which is identified through positions of 20 skeletal body joints, computed by the depth data at a rate of 30 fps. The resolution of the color image is 640×480, whereas that of the depth image is 320×240. The viewing volume is specified by 43° horizontally, and 57° vertically where the typical distance of the signer is 1-3 meters.

The posture detection scheme basically relies on hand and elbow location values of the signer. Location is known to be one of the four phonological features of sign languages (Arik, 2012). Although using only the location feature for posture detection might not be adequate, a fair amount of postures, which do not overlap in location values, are recognized here. Hand positions are generally aligned with several reference areas on the body of the signer. Therefore, the signing space is clustered into small coordinate regions to facilitate feature extraction based on mainly locations of the hands and elbows. The remaining signing space is clustered with a range that is determined by the signer body. The feature vector has 12 features that correspond to the cluster numbers of the right hand, the left hand, the right elbow, and the left elbow skeletal joints, respectively. This is because of the fact that most of the gestures in sign languages contains actions in the upper part of the human body.

First, we perform a temporal quantization operation to increase the robustness of the extracted features under sudden variances. A temporal quantization of 0.2 seconds (six frames under 30 fps) is used to compute the moving average of the locations of the selected skeletal joints. The next step is the grid-based signing space clustering scheme, as outlined in Figure 2. First, the origin of the coordinates on the human body is set as the left shoulder joint for the rest of the computations. As shown in Figure 2(a), clustering on the x-y plane starts with dividing the x-axis between x_0 and x_1 into four equal parts. The y-axis between y_0 and y_5 is partitioned into five regions such that $\overline{y_0 y_1}$ is equal to $\overline{y_1 y_2}$, and the distance between y_2 and y_5 is divided into three equal parts. Hence, the chest of the signer is divided into two equal regions, whereas the abdominal area is divided into three equal parts. The remaining parts of the grid are divided into sub-parts by the cluster range r, as shown in Figure 2(b). This type of clustering with the predefined parameters help the extracted features be robust, since the locations of the hands and elbows are relatively small.

This way of clustering the signing space implicitly provides a simpler and reasonable level of scale invariance, since a data normalization with d is perforemd, as recommended in Bhattacharya *et al.* (2012). Position invariance is also preserved since clusters are originated from the upper left part of body, and cluster numbers are used features instead of locations. Rotation invariance is satisfied by rotating the locations of the skeletal joints such that the x-axis becomes parallel to the major axis of the signer. The major axis of the signer is defined as the line segment connecting the left shoulder joint location and the right shoulder joint location. This re-orientation preserves the rotation invariance.



Notes: (a) Clustering on the *x*-*y* plane: clustering for x values between x_0 and x_1 is done by dividing the distance between x_0 and x_1 into four equal parts. Clustering for y values between y_0 and y_5 is to divide the chest of the signer into two and the abdominal area into three equal parts. The remaining parts are divided by the cluster range r; and (b) calculation of r for x and y dimensions, and the *depth_range* for z dimension with d, as the distance between the shoulder center joint and the hip center joint of the signer

Figure 2.
Grid-based signing space clustering scheme

3.2 Posture labeling

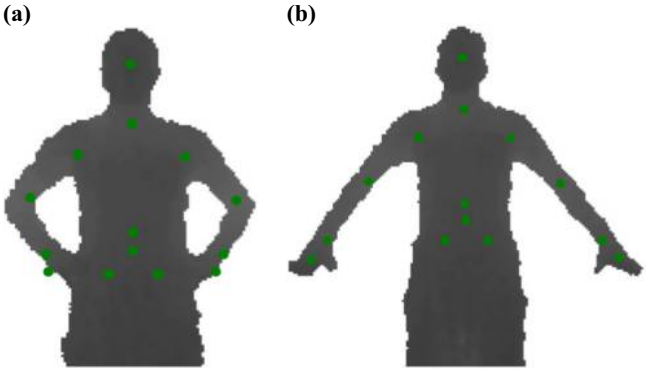
The posture labeling algorithm assigns a label for the processed posture based on the gestures in the known vocabulary. The selected subset of gestures, as the known vocabulary, from TSL includes six gestures, namely, *anne*, *bardak*, *bere*, *etek*, *masa*, and *neskafe*. These are Turkish words and their English meanings are mother, glass (as in "a glass of water"), hat, skirt, table, and nescafe, respectively. Only *bere* has one posture whereas the other five gestures have two consecutive postures. The postures for gesture *etek* is given in Figure 3 as an example.

The pseudo-code of the posture labeling algorithm is given in Figure 4. The algorithm checks the features for the current frame f with that of the existing postures to assign a label of the posture in f. This operation is not an exact match with the stored values, instead a distance-based similarity value is computed to find the label. If a match is not found, "not-a-known-posture" is returned as the posture label. The extracted features for a frame f in *extractFeatures*(f) function are the cluster numbers of the left hand, the right hand, the left elbow and the right elbow (line 2 in Figure 4). The details are explained in the previous subsection.

The cluster numbers of the left hand, the right hand, the left elbow, and the right elbow joints are stored for a later lookup to assign a label to a posture (lines 4-7 in Figure 4). These values are stored for each posture, and the entries correspond to previous observation from the training data. Table I shows a sample snapshot for the stored feature values of the postures, which are to be used in the similarity-based lookup operation.

Depending on the depth values of the posture, feature comparison operation has two alternatives. If the depth values of a posture can be compared against an interval, z-axis values are compared as a range (lines 14-19 in Figure 4). Otherwise, z-axis values are considered as points, similar to x-axis and y-axis values (lines 21-24 in Figure 4). This is decided by the observed values of the posture during training, and this decision will be elaborated on in the experimental evaluations section.

The time complexity of the posture labeling algorithm is as follows: The *extractFeatures*(f) function requires two-dimensional processing on the number of skeletal joints, say n, which can be written as $O(n^2)$. Then, for each posture in *posture−set*, a location check is performed in three-dimensions, which leads to $O(n^3 \times |posture-set|)$. Since the last term dominates, the overall algorithmic complexity is $O(n^3 \times |posture-set|)$.



Notes: The posture named as (a) *etek*0; and (b) *etek*1 in FSA design

Figure 3.
The postures that are labeled to recognize the gesture *etek*

```
Inputs:  f, a video frame;
         postureset, the set of the known postures;
Output:  label, the label of the posture;

1.    function ComputePostureLabel (f, postureset)
2.       fv = extractFeatures (f);
3.       for each posture p in postureset
4.          if (p.lefthandLocationset.Contains (fv.lefthandLocation) and
5.              p.righthandLocationset.Contains (fv.righthandLocation) and
6.              p.leftelbowLocationset.Contains (fv.leftelbowLocation) and
7.              p.rightelbowLocationset.Contains (fv.rightelbowLocation)) then
8.              return p.name as the label
9.          else
10.             return 'not-a-known-posture' as the label
11.         end if
12.      end for
13.   end function

14.   function LocationSet.Contains (LocationOI)
15.      if posture allows for ranged z values then
16.         if (LocationOI.Z ∈ LocationSet-of-JOI-of-Posture. Zranges) &
17.               LocationOI.XY ∈ LocationSet-of-JOI-of-Posture.XYValues) then
18.               return true
19.         end if
20.         return false
21.      else
22.         if (LocationOI.XYZ ∈ LocationSet-of-JOI-of-Posture.XYZ) then
23.            return true
24.         end if
25.         return false
26.      end if
27.   end function
```

Figure 4.
The pseudo-code of the posture labeling algorithm at a frame *f* with the known vocabulary postures in *postureset*

Notes: A label is assigned depending on the comparison against the feature values; and "not-a-known-posture" is assigned otherwise

L-Hand	R-Hand	L-Elbow	R-Elbow	P-Label
−1,6,2	2,1,1	−1,3,0	4,3,0	anne0
−1,6,0	1,0,1	−1,3,0	5,2,0	anne1
−1,5,1	4,0,2	−1,3,0	4,2,1	bardak0
−1,5,1	4,−2,1	−1,2,0	4,0,1	bardak1
0,−3,1	3,−4,2	−3,0,1	6,−1,1	bere0
−1,4,0	4,4,0	−2,2,0	5,2,0	etek0
−3,4,0	6,3,0	−2,2,0	5,2,0	etek1
3,1,3	4,1,3	−1,2,1	5,1,1	masa0
−1,6,0	3,0,1	−1,3,0	5,0,1	neskafe0
−2,5,1	4,0,1	−1,3,0	5,2,1	neskafe1

Table I.
The sample set of features based on the cluster numbers of four joints for each posture that are computed by the grid-based signing space clustering scheme

Notes: A single line of features for each posture is listed. Based on the training step, several instances are stored for the postures

3.3 FSA for gestures classification

A gesture can be modeled as a temporally ordered sequence of states representing separable postures in the scene. There are techniques using finite state machines for gesture recognition purposes in the literature (e.g. Bobick and Wilson, 1997; Davis and Shah, 1999; Hong *et al.*, 2000b). Here, we propose a recognition scheme using FSA to classify the gestures against the known vocabulary.

A deterministic finite state automaton (dFSA) is denoted as a quintuple (Σ, S, S_0, δ, F), where Σ is the input alphabet (a finite, non-empty set of symbols); S is a finite, non-empty set of states; S_0 is an initial state where $S_0 \in S$; δ is the state transition function such that δ: $S \times \Sigma \rightarrow S$; and F is the set of final states, where $F \subset S$.

We selected a set of six gestures, one of which has one posture, and the remaining five has two consecutive postures. The labels of these distinct postures are the symbols in the input alphabet, where $\Sigma = \{$*anne0, anne1, bardak0, bardak1, bere0, etek0, etek1, masa0, masa1, neskafe0, neskafe1, nes1an0, not-a-known-posture*$\}$. The selected gesture set is challenging, for example the gestures *anne* and *neskafe* have very a similar posture. Hence, a new posture, namely *nes1an0*, is introduced to improve the recognition rate of the FSA.

In each FSA, the execution starts with state S_0, and each posture label (symbol) is processed in order through the intermediate states. If the desired temporal order is achieved, then the gesture is recognized (accepted) via reaching the final state F.

This scheme is also extensible such that if a new gesture is to be added to the known vocabulary, an appropriate FSA can be augmented based on the postures of the new gesture. Below, the FSA models of each gesture in the known vocabulary is explained. Capital letters are used as a subset of Σ in the FSA drawings not to make them complex.

3.3.1 FSA for neskafe. FSA to recognize *neskafe* is given in Figure 5. Formally:

$$dFSA_1 = (\Sigma, \{S_0, S_1, S_2, S_3, S_4\}, S_0, \delta_1, \{S_4\}), \tag{1}$$

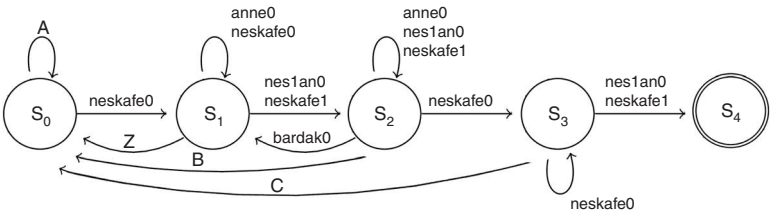
$$\delta_1: \{S_0, S_1, S_2, S_3, S_4\} \times \Sigma \rightarrow \{S_0, S_1, S_2, S_3, S_4\}. \tag{2}$$

The details of δ_1 is only explained graphically in Figure 5 to clarify the explanations. As given in Equation (1), five states are used, where S_0 is the initial state, S_4 is the accepting state.

3.3.2 FSA for anne. FSA to recognize *anne* is given in Figure 6. Formally:

$$dFSA_2 = (\Sigma, \{S_0, S_1, S_2\}, S_0, \delta_2, \{S_2\}), \tag{3}$$

$$\delta_2: \{S_0, S_1, S_2\} \times \Sigma \rightarrow \{S_0, S_1, S_2\}. \tag{4}$$



Notes: S_0 is the initial state, S_4 is the accepting state. A = $\sum$ − {neskafe0}; B = $\sum$ − {anne0, bardak0, neskafe0, neskafe1, nes1an0}; C = $\sum$ − {neskafe0, neskafe1, nes1an0}; and Z = $\sum$ − {anne0, neskafe0, neskafe1, nes1an0}

Figure 5. Finite state automaton for *neskafe*

The details of δ_2 is only explained graphically in Figure 6 to clarify the explanations. As given in Equation (3), three states are used, where S_0 is the initial state, S_2 is the accepting state.

3.3.3 FSA for bardak. FSA to recognize *bardak* is given in Figure 7. Formally:

$$dFSA_3 = (\Sigma, \{S_0, S_1, S_2\}, S_0, \delta_3, \{S_2\}), \tag{5}$$

$$\delta_3\colon \{S_0, S_1, S_2\} \times \Sigma \rightarrow \{S_0, S_1, S_2\}. \tag{6}$$

The details of δ_3 is only explained graphically in Figure 7 to clarify the explanations. As given in Equations (5), three states are used, where S_0 is the initial state, S_2 is the accepting state.

3.3.4 FSA for bere. FSA to recognize *bere* is given in Figure 8. Formally:

$$dFSA_4 = (\Sigma, \{S_0, S_1, S_2\}, S_0, \delta_4, \{S_2\}), \tag{7}$$

$$\delta_4\colon \{S_0, S_1, S_2\} \times \Sigma \rightarrow \{S_0, S_1, S_2\}. \tag{8}$$

The details of δ_4 is only explained graphically in Figure 8 to clarify the explanations. As given in Equation (7), three states are used, where S_0 is the initial state, S_2 is the accepting state.

3.3.5 FSA for etek. FSA to recognize *etek* is given in Figure 9. Formally:

$$dFSA_5 = (\Sigma, \{S_0, S_1, S_2\}, S_0, \delta_5, \{S_2\}), \tag{9}$$

$$\delta_5\colon \{S_0, S_1, S_2\} \times \Sigma \rightarrow \{S_0, S_1, S_2\}. \tag{10}$$

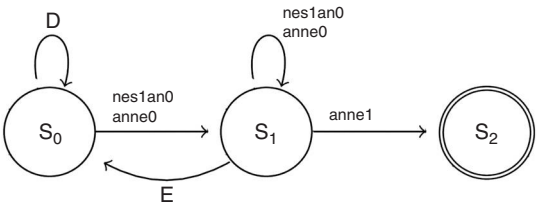


Figure 6.
Finite state automaton for *anne*

Notes: S_0 is the initial state, S_2 is the accepting state. D=∑−{anne0, nes1an0}; and E=∑−{anne0, anne1, nes1an0}

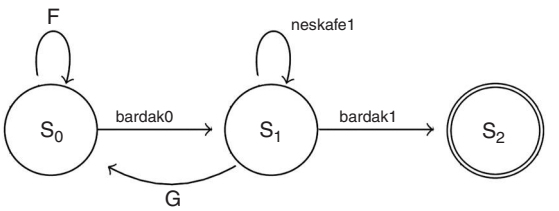


Figure 7.
Finite state automaton for *bardak*

Notes: S_0 is the initial state, S_2 is the accepting state. F=∑−{bardak0}, and G=∑−{bardak1, neskafe1}

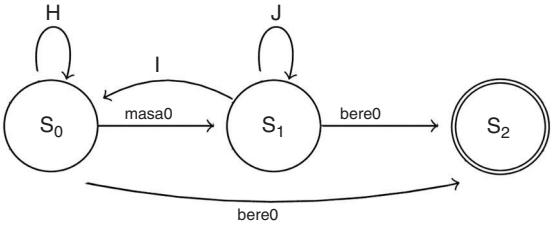
The details of δ_5 is only explained graphically in Figure 9 to clarify the explanations. As given in Equation (9), three states are used, where S_0 is the initial state, S_2 is the accepting state.

3.3.6 FSA for masa. FSA to recognize *masa* is given in Figure 10. Formally:

$$dFSA_6 = (\Sigma, \{S_0, S_1, S_2\}, S_0, \delta_6, \{S_2\}), \tag{11}$$

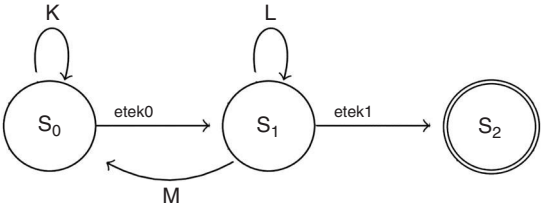
$$\delta_6: \{S_0, S_1, S_2\} \times \Sigma \rightarrow \{S_0, S_1, S_2\}. \tag{12}$$

The details of δ_6 is only explained graphically in Figure 10 to clarify the explanations. As given in Equation (11), three states are used, where S_0 is the initial state, S_2 is the accepting state.



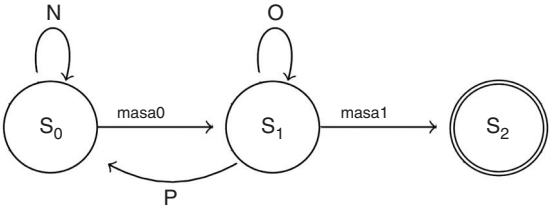
Notes: S_0 is the initial state, S_2 is the accepting state. H=∑−{bere0, masa0}, I={not-a-known-posture}; and J=∑−*I*−{bere0}

Figure 8. Finite state automaton for *bere*



Notes: S_0 is the initial state, S_2 is the accepting state. K=∑−{etek0}, M={not-a-known-posture}; and L=∑−*M*−{etek1}

Figure 9. Finite state automaton for *etek*



Notes: S_0 is the initial state, S_2 is the accepting state. N=∑−{masa0}, P={not-a-known-posture}, and O=∑−*P*−{masa1}

Figure 10. Finite state automaton for *masa*

4. Performance experiments

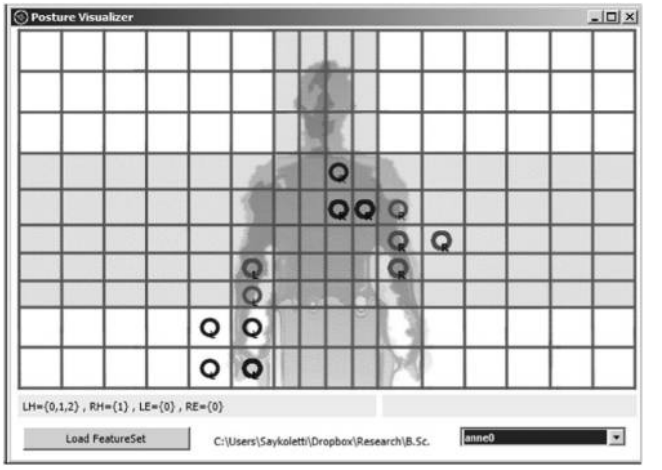
The phonologic properties and linguistic nature of TSL is investigated and a set of challenging gestures is chosen based on the analysis of posture expression to evaluate our posture labeling and gesture classification schemes. Since most gestures have a single posture in TSL, these techniques can be easily tailored to recognize more gestures.

The data set that we experiment on has 129 gestures (20 *anne*, 20 *bere*, 20 *etek*, 20 *bardak*, 20 *masa*, 29 *neskafe*), and 1,355 corresponding postures. We have devised a separate tool for visualization and analysis of the postures in the data set. The main screen of this tool is shown in Figure 11. In the bottom part, right under the grid-based region, there is a separate line displaying depth information of the postures. As discussed earlier in similarity-based lookup to assign a posture label, if the depth range for a joint has a consecutive set of clusters, then we say that this joint supports comparison against an interval (lines 14-19 in Figure 4).

The performance experiments are two-fold: One is the evaluation of posture labeling algorithm, hence the grid-based signing space clustering scheme is also evaluated. Since a distance-based similarity metric is used for the lookup of cluster locations in posture labeling, we select a distance-based classifier, kNN. Euclidean distance is used for computing distances, and the Weka implementation is used for kNN. The other set of experiments is to evaluate the gesture classification performance.

4.1 Performance of posture labeling

Table II shows the recognition rates of the posture labeling algorithm, and a comparison of our technique and kNN with spherical coordinates is provided. The experiments show that our posture labeling scheme has a considerable level of classification rate when compared to that of a distance-based classifier kNN. Hence, assigning a correct label to a posture can be considered as classification of postures. The temporal quantization is performed with $\tau = 6$. The overall classification performance for our technique is 97.7 percent, whereas it is 90.3 percent for kNN



Notes: The circles in the grid denote the locations observed for a posture. The depth locations are shown as a line right under the grid-based display

Figure 11.
The main screen of the application developed for posture visualization and analysis

classifier, on the average. Among the 226 instances, kNN classifies 204 of them correctly. Table III gives the confusion matrix for kNN.

An interesting observation in Table II is the classification rate of kNN for the posture *neskafe1*. As discussed earlier, *neskafe1* and *anne0* postures are very similar and we intentionally select these postures to make the data set challenging. Hence, kNN has 0 correct classifications for *neskafe1*, and this case can be seen in the confusion matrix as well (see Table III). As expected, kNN classified these instances as *anne0*. However, in the FSA design, we create a new posture *nes1ann0* to alleviate this issue. This is the main reason behind the huge difference in the classification rates of *neskafe1*.

4.2 Performance of gesture classification

The gesture recognition performance is given in Table IV. Although the accuracy is more than 97 percent in posture labeling scheme, it lowers to 93 percent on the average in gesture classification. Since a gesture is modeled as a sequence of postures and utilized FSA for recognition, the expected classification rate for a gesture is the minimum rate of its postures. This holds for every gesture except *anne*, where the

Posture Label	Posture labeling (%)	kNN (%)
anne0	100	100
anne1	100	100
bardak0	100	92.3
bardak1	90	100
etek0	100	100
etek1	95	84.6
neskafe0	100	96.4
neskafe1	95	0
masa0	95	91.6
masa1	100	100
bere0	100	100
Average	97.7	90.3

Note: Euclidean distance with respect to spherical coordinates is employed for kNN

Table II. The recognition rate of our posture labeling technique based on the data set used in performance experiments

a	b	c	d	e	f	g	h	i	j	k	
12	0	0	0	0	0	0	0	0	0	0	a
0	10	0	0	0	0	0	0	0	0	0	b
0	0	12	0	0	0	0	1	0	0	0	c
0	0	0	17	0	0	0	0	0	0	0	d
0	0	0	0	14	0	0	0	0	0	0	e
0	0	0	0	2	11	0	0	0	0	0	f
0	0	0	0	0	1	27	0	0	0	0	g
17	0	0	0	0	0	0	0	0	0	0	h
0	1	0	0	0	0	0	0	11	0	0	i
0	0	0	0	0	0	0	0	0	12	0	j
0	0	0	0	0	0	0	0	0	0	78	k

Notes: a, anne0; b, anne1; c, bardak0; d, bardak1; e = etek0; f, etek1; g, neskafe0; h, neskafe1; I, masa0; j; masa1; k, bere0

Table III. The confusion matrix of kNN for posture classification based on the Weka implementation of kNN

postures *anne0* and *anne1* are detected at 100 percent, but the gesture classification rate is 90 percent. That is the main cause of the reduction in gesture classification and the main reason behind this is the fact that the training samples that we used for *anne* is not adequate enough to design a FSA to have an accuracy higher than 90 percent. Extending the training set with more samples is likely to solve the issue with this gesture.

There exist various types of techniques in the literature working on different sign language gestures. Each technique evaluates its performance with a specialized data set, which hardens to evaluate the performance in terms of classification accuracies. Here, we would like to provide an evaluation based on the underlying techniques between our scheme and the state of the art. The existing studies can be broadly classified into neural network based and hidden markow model based techniques. Due to the hardness of designing multi-dimensional features for traditional neural networks, CNNs are used in the literature (e.g. Pigou *et al.*, 2015; Huang *et al.*, 2015). These CNNs are more suitable for hand-based features and has limitations on the registration of the locational variances of the same postures performed by same or different signers. The existing techniques based on HMMs require many samples for training purposes to overcome the limitations of NNs and CNNs. However, the major limitation on these HMM-based techniques is the temporal registration of the consecutive postures of the gesture, due to the temporal variances among same or different signers.

Our posture labeling based gesture classification technique utilizes grid-based signing space clustering scheme to overcome the locational variances of the signers, which is a limitation for CNNs. Moreover, we provide a representation of grid labels relative to the signer's torso, which improves the locational invariance of our technique. This grid-based scheme is used to label the corresponding postures, and then the recognition is performed on the sequence of posture labels. This type of temporal processing helps us overcome the temporal variances, which is a limitation for HMM-based techniques. Hence, we can utilize FSA to recognize a set of predefined gestures continuously via processing the posture labels in temporal order. Another advantage of our scheme is its extensibility since designing FSAs is rather simple when compared to the existing approaches.

5. Conclusion

We propose a posture labeling based gesture classification technique for TSL recognition using Kinect to acquire skeletal features and depth data. A grid-based signing space clustering scheme is proposed, and the cluster numbers are used as features for a set of joints. A posture labeling algorithm is proposed to recognize a predefined set of gestures in TSL. The labels given to the postures are used to classify

Table IV. The peformance of gesture classification

Gesture name	Total	Correctly classified	Recognition rate (%)
anne	20	18	90
bardak	20	19	90
etek	20	18	95
neskafe	29	26	90
masa	20	19	95
bere	20	20	100

the gestures with respect to a known vocabulary using FSA. A set of challenging gestures is chosen to evaluate our technique; however, our scheme is also extensible for a new gesture by simply providing an appropriate FSA based on its postures. The overall classification scheme deals only with the posture labels instead of low-level and spatio-temporal features, which reduces the space and time requirements.

Two sets of experimental evaluations show that both the posture labeling scheme has a considerable level of classification rate when compared to that of kNN and the gesture recognition performance is around 93 percent on the average. Even though extending the training set with more samples is likely to improve the gesture recognition accuracy, the achievements are very reasonable for TSL gestures.

As a future work, we plan to design an extension scheme based on the devised tools for experimental purposes. As a consequence, more gestures will be added to the vocabulary. Another future study is to enhance the location-based feature extraction by augmenting palm orientation. This study will also improve the classification rates since palm orientation is an important clue for sign language gestures.

References

Agarwal, A. and Thakur, M. (2013), "Sign language recognition using microsoft kinect", *Proceedings of the Sixth International Conference on Contemporary Computing (IC3)*, *Noida*, pp. 181-185.

Akram, S., Beskow, J. and Kjellström, H. (2012), "Visual recognition of isolated swedish sign language signs" The Computing Research Repository (CoRR), abs/1211.3901.

Arik, E. (2012), "Space, time, and iconicity in Turkish sign language (TID)", *Trames: A Journal of the Humanities and Social Sciences*, Vol. 16 No. 4, pp. 345-358.

Bhattacharya, S., Czejdo, B. and Perez, N. (2012), "Gesture classification with machine learning using Kinect sensor data", *Proceedings of the Third International Conference on Emerging Applications of Information Technology (EAIT'12)*, *Kolkata*, pp. 348-351.

Bobick, A. and Wilson, A. (1997), "A state-based approach to the representation and recognition of gesture", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 No. 12, pp. 1235-1337.

Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X. and Zhou, M. (2013), "Sign language recognition and translation with Kinect", *Proceedings of the 10th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2013), Demo Paper*, *Shanghai*.

Cooper, H., Ong, E.-J., Pugeault, N. and Bowden, R. (2012), "Sign language recognition using sub-units", *Journal of Machine Learning Research*, Vol. 13, July, pp. 2205-2231.

Davis, J. and Shah, M. (1994), "Visual gesture recognition", *IEE Proceedings on Vision, Image and Signal Processing*, Vol. 141 No. 2, pp. 101-106.

Davis, J. and Shah, M. (1999), "Toward 3-D gesture recognition", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 13 No. 3, pp. 381-393.

Dominio, F., Donadeo, M., Marin, G., Zanuttigh, P. and Cortelazzo, G. (2013), "Hand gesture recognition with depth data", *Proceedings of the 4th ACM/IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream (ARTEMIS'13), ACM*, *Barcelona*, pp. 9-16.

Gao, W., Fang, G., Zhao, D. and Chen, Y. (2004), "A Chinese sign language recognition system based on SOFM/SRN/HMM", *Pattern Recognition*, Vol. 37 No. 12, pp. 2389-2402.

Hong, P., Turk, M. and Huang, T. (2000a), "Constructing finite state machines for fast gesture recognition", *Proceedings of the 15th IEEE International Conference on Pattern Recognition (ICPR'2000)*, Vol. 3, *Barcelona*, pp. 691-694.

Hong, P., Turk, M. and Huang, T. (2000b), "Gesture modeling and recognition using finite state machines", *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, *Grenoble*, pp. 410-415.

Huang, J., Zhou, W., Li, H. and Li, W. (2015), "Sign language recognition using 3D convolutional neural networks", *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'2015)*, pp. 1-6.

Jaemin, L., Takimoto, H., Yamauchi, H., Kanazawa, A. and Mitsukura, Y. (2013), "A robust gesture recognition based on depth data", *Proceedings of the 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV'2013), IEEE*, *Incheon*, pp. 127-132.

Keskin, C., Kraç, F., Kara, Y. and Akarun, L. (2011), "Real time hand pose estimation using depth sensors", *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops 2011)*, *Barcelona*, pp. 1228-1234.

Khoshelham, K. and Elberink, S. (2012), "Accuracy and resolution of Kinect depth data for indoor mapping applications", *Sensors*, Vol. 12 No. 2, pp. 1437-1454.

Lee, G., Yeh, F.-H. and Hsiao, Y.-H. (2016), "Kinect-based Taiwanese sign-language recognition system", *Multimedia Tools and Applications*, Vol. 75 No. 1, pp. 261-279.

Memiş, A. and Albayrak, S. (2013), "Turkish sign language recognition using spatio-temporal features on Kinect RGB video sequences and depth maps", *Proceedings of the 21st IEEE International Conference on Signal Processing and Communications Applications Conference (SIU 2013)*, *Lefkosa, North Cyprus*.

Mitra, S. and Acharya, T. (2007), "Gesture recognition: a survey", *IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, Vol. 37 No. 3, pp. 311-324.

Mohandes, M., Deriche, M. and Liu, J. (2014), "Image-based and sensor-based approaches to Arabic sign language recognition", *IEEE Transactions on Human-Machine Systems*, Vol. 44 No. 4, pp. 551-557.

Nguyen, L., Thanh, C., Ba, T., Viet, C. and Thanh, H. (2013), "Contour based hand gesture recognition using depth data", *Advanced Science and Technology Letters (SIP 2013)*, Vol. 29, pp. 60-65.

Öz, C. and Leu, M. (2011), "American sign language word recognition with a sensory glove using artificial neural networks", *Engineering Applications of Artificial Intelligence*, Vol. 24 No. 7, pp. 1204-1213.

Phadtare, L., Kushalnagar, R. and Cahill, N. (2012), "Detecting hand-palm orientation and hand shapes for sign language gesture recognition using 3D images", *Proceedings of the Western New York Image Processing Workshop (WNYIPW'12)*, *New York, NY*, pp. 29-32.

Pigou, L., Dieleman, S., Kindermans, P.-J. and Schrauwen, B. (2015), "Sign language recognition using convolutional neural networks", in Agapito, L., Bronstein, M. and Rother, C. (Eds), *Computer Vision – ECCV 2014 Workshops, Part I*, Vol. 8925, Springer, Zurich, pp. 572-578.

Ren, Z., Yuan, J. and Zhang, Z. (2011), "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera", *Proceedings of the 19th ACM International Conference on Multimedia (MM'11), ACM*, *Scottsdale, AZ*, pp. 1093-1096.

Ren, Z., Yuan, J., Meng, J. and Zhang, Z. (2013), "Robust part-based hand gesture recognition using Kinect sensor", *IEEE Transactions on Multimedia*, Vol. 15 No. 5, pp. 1110-1120.

Şaykol, E., Baştan, M., Güdükbay, U. and Ulusoy, Ö. (2010), "Keyframe labeling technique for surveillance event classification", *Optical Engineering*, Vol. 49 No. 11, 12 pp.

Shanableh, T., Assaleh, K. and Al-Rousan, M. (2007), "Spatio-temporal feature extraction techniques for isolated gesture recognition in Arabic sign language", *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol. 37 No. 3, pp. 641-650.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. and Blake, A. (2011), "Real-time human pose recognition in parts from single depth images", *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2011)*, pp. 1297-1304.

Sun, C., Zhang, T., Bao, B.-K., Xu, C. and Mei, T. (2013), "Discriminative exemplar coding for sign language recognition with kinect", *IEEE Transactions on Cybernetics*, Vol. 43 No. 5, pp. 1418-1428.

Takimoto, H., Jaemin, L. and Kanagawa, A. (2013), "A robust gesture recognition using depth data", *International Journal of Machine Learning and Computing*, Vol. 3 No. 2, pp. 245-249.

Weinland, D., Ronfard, R. and Boyer, E. (2011), "A survey of vision-based methods for action representation, segmentation and recognition", *Computer Vision and Image Understanding*, Vol. 115 No. 2, pp. 224-241.

Yeasin, M. and Chaudhuri, S. (2000), "Visual understanding of dynamic hand gestures", *Pattern Recognition*, Vol. 33 No. 11, pp. 1805-1817.

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H. and Presti, P. (2011), "American sign language recognition with the Kinect", *Proceedings of the 13th International Conference on Multimodal Interfaces (ICMI'11), ACM, Alicante*, pp. 279-286.

Corresponding author
Ediz Saykol can be contacted at: ediz.saykol@beykent.edu.tr