HANDY: A Configurable Gesture Recognition System

Mahsa Teimourikia
Department of Electronics,
Information and Bioengineering (DEIB)
Politecnico di Milano
Email: mahsa.teimourikia@polimi.it

Hassan Saidinejad
Department of Electronics,
Information and Bioengineering (DEIB)
Politecnico di Milano
Polo Regionale di Como
Email: hassan.saidinejad@polimi.it

Sara Comai
Department of Electronics,
Information and Bioengineering (DEIB)
Politecnico di Milano
Email: comai@elet.polimi.it

***Abstract*—With the growing usage of computer systems in daily life, a natural and intuitive Human Computer Interaction (HCI) method to support the embedding of computer systems in our environment seems necessary. Gestures are of utmost importance for the design of natural user interfaces. Hand gesture recognition to extract meaningful expressions from the human hand movements and postures is being used for different applications. However, the recognition of hand gestures that contain different hand poses can be challenging. In this paper, we propose a system (called HANDY) for hand gesture recognition that is flexible to be trained to recognize a variety of user-defined gestures defined as sequences of static hand postures. The system has been designed to be used in uncontrolled environments, to handle dynamic and cluttered backgrounds, and without the need of using any wearable sensor or any specific clothing. Evaluation results show a good average accuracy in gesture recognition.**

Keywords-interactive systems; gesture-based interface; natural HCI; personalizable system

I. Introduction

Natural User Interfaces (NUI) have attracted considerable amount of research nowadays. With the growing usage of computer systems in everyday life and the desire to embed them into the environment, employing the traditional input devices (like keyboard and mouse) seems to be a bottleneck in efficient and intuitive Human Computer Interaction. Hand gestures are a natural way to communicate in human-human interactions and can be adopted also in Human Computer Interaction (HCI). More recently, devices such as Microsoft Kinect [1], a low-cost Natural Interaction (NI) device with Red-Green-Blue (RGB) and Depth sensors, have been used increasingly in NUI and HCI research for hand gesture recognition. The most prominent gesture recognition frameworks either use wearable sensors, vision-based methods, or a combination of both. While wearable sensors can be costly, inaccessible for many users and also might cause health problems or fatigue, vision-based frameworks, like ToF (Time-of-Flight) cameras [2] or the Kinect, are instead affordable to the majority of the people.

Many frameworks have been proposed regarding hand pose and gesture recognition using vision-based techniques. However, the research on gesture recognition has focused mainly on static gestures that include a static hand pose estimation or on dynamic gestures that consider the trajectory of the hands [3], [4], [5]. Postures of the hand during dynamic gesturing can be important features in natural communication. However, only a small number of frameworks have successfully taken into consideration the hand postures in the recognition of the dynamic gestures [6], [7]. And still, a limited number of hand poses are understandable for the gesture recognition system in these works. Moreover, the posture is usually assumed to remain the same during the gesturing. Even in sign language recognition systems, the posture of the hands is usually neglected and other features such as velocity, direction, trajectory, and position of the hand with respect to the face are considered. Furthermore, the flexibility of the system to be configured with user-specific gestures is yet another key aspect that has received less attention in the recent literature.

In this paper, a system (called HANDY) for gesture recognition is proposed. The framework introduced here is flexible enough to be able to train and use the system with minimal effort for a variety of hand poses and gestures that meet the user's specific needs. Microsoft Kinect depth sensor is adopted because of its low cost, availability and ease of installation. Furthermore, by the usage of depth data, the system can be adopted in low illumination, dynamic backgrounds and in uncontrolled environments and also the user would not need to use any wearable devices, sensors or any specific clothing. The mentioned aspects make the system appropriate for usage in different contexts for natural interactions.

Due to the stochastic nature of human including immeasurable and hidden mental states along with the measurable and observable human actions, Hidden Markov Models (HMM) [8] can be used to model these processes. HMMs have been successfully used in many applications and studies for speech recognition [8] and hand writing recognition [9]. Also, they have been proven effective in sign language recognition and other complex hand gesture recognition processes [10], [11]. In this approach, HMMs are used for gesture recognition to enable recognition of gestures composed of sequences of states with tolerance to the changes in the hand posture, to exploit patterns, and to evolve recognition capabilities with the help of learning techniques.

The paper is organized as follows: Section II reports related work for hand pose and gesture recognition techniques. Then, in Section III, we introduce the HANDY system and its implementation details. Section IV focuses on the results. Finally, in Section V, we draw our conclusions and report future works.

II. Background and Related work

In this section, we introduce the background and review the state of the art related to the main techniques needed to identify the hand and its gestures: hand localization, segmentation, hand pose estimation and gesture recognition.

Approaches regarding the interpretation of gestures in HCI either use wearable sensors or employ vision-based methods [4]. An example of wearable sensors is represented by data gloves, which can provide accurate measurements of hand pose and movements. However, wearable sensors are commonly costly while they limit the hand movement and are not very comfortable to be used in everyday life. Mynatt et al. [12] use a wireless wearable device, called Gesture Pendant, that uses infrared illumination and a charge-coupled device (CCD) camera to recognize a simple set of gestures. P. Trindade et al. [13] propose using an Inertial Measurement Unit (IMU) sensor for recognizing the hand orientation. Some other approaches employ, colored hand gloves [14], data gloves [15], and forearm band or wrist band with accelerometer and electromyography (EMG) sensors [16] accompanied with a vision-based approach to simplify the hand localization, segmentation and tracking.

To avoid the use of any wearable or mechanical device, vision-based gesture recognition can be used. Vision-based gesture recognition techniques can be divided into two broad categories [17]: Model based approaches that focus on taking advantage of a 3D or 2D model of the hand for hand pose and approaches based on hand shape appearance which is used to extract the features of the visual data for gesture recognition. Model-based approaches usually suffer from high complexity in implementation and cannot be used in live applications. Appearance-based approaches use RGB or depth data or both as the input. Our framework falls into this category.

When only RGB images in the appearance-based approach are used, the tasks of locating and segmenting the hand from a cluttered and noisy background can become very challenging, especially when other skin-colored objects are present in the scene and in case of occlusions [18]. Moreover, RGB images can be sensitive to illumination changes and extracting the sophisticated features from the image to have an accurate recognition can be very costly regarding the processing time [4]. To overcome these issues, other approaches have shown that better results for hand segmentation can be obtained when using depth data [19]. However, as it is stated in [19], most of the proposals assume that the hands are the closest object to the scene. In our system, as we will see in Section III, we will take advantage of the body tracking information extractable from Kinect SDKs, so that depth thresholding can be done regardless of the position of the hand.

Also, Zhu and Pun [20] use Kinect depth data for extracting the trajectory data sequence of the hand movements, but, do not consider hand postures. In a similar way, [21] and [7] introduce a gesture recognition approach that considers the motion and shape information of the hand using depth data. However, the hand shape is considered to remain the same during the gesturing and pose estimation is done once at the beginning of the gesture. In another study by Chen et al. [6] HMM continuous gesture recognition is proposed considering the spatial and temporal features of the gestures. Yet, a small number of poses is recognized in this approach and the posture of the hand does not change while performing the gesture. HMM has been adopted also in the work by Starner et al. [22] for American Sign Language (ASL) recognition. For recognizing the sign language they have ignored the detailed shape and pose of the hand and have only considered, coarse hand pose, orientation, and the trajectory of the gesture through time, and used such information as input for the recognition system. In a more recent work, Molina et al. [23] propose an approach for static pose estimation and dynamic gesture recognition. They successfully recognize the gestures that include the change in the hand postures. They have obtained an accuracy of 90% for recognition of combination of static hand postures and dynamic gestures.

Differently from these approaches, we consider hand poses and gestures composed of their combinations: hand poses are modeled by their skeleton, and time series analysis algorithms and HMM techniques are used to recognize sequences of hand poses. Using these techniques user-defined gestures can also be defined and more flexibility can be added in case of gesture evolution.

III. The HANDY System

The architecture of the HANDY framework for gesture recognition is depicted in Fig. 1. The system extracts from the

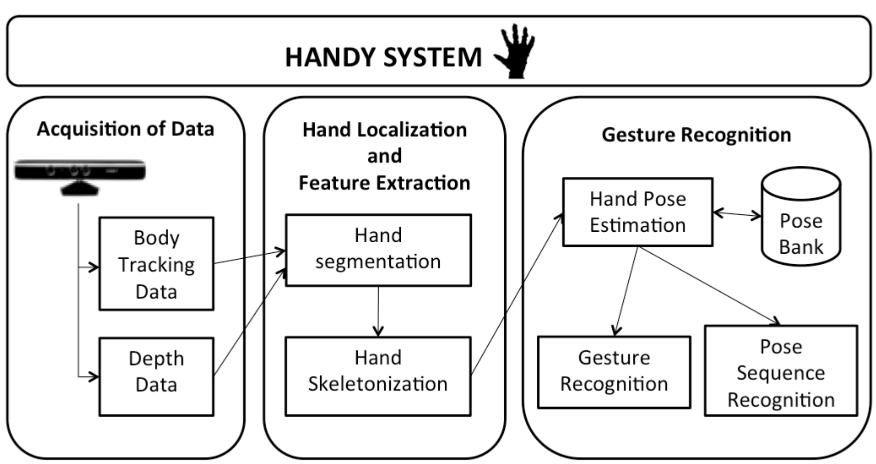


Figure 1: The Gesture Recognition Scheme

 ISBN: 978-1-61208-325-4

Kinect the depth data and the position of the center of the palm: the OpenNI [24] framework together with NiTE [25] skeletal tracking middleware have been used to extract the needed information. Then, the segmented hand is first identified and its skeleton is calculated as a feature that will be used for hand pose estimation. Hand poses can be stored in a pose bank, to be used in gesture recognition. Both static gestures and pose sequences are considered in our system: in both cases their recognition is based on the hand pose estimation. In the sequel, each step of this scheme is explained in detail.

A. Hand Localization and Segmentation

One of the essential components of hand gesture recognition is hand localization and tracking. Hand localization refers to the positioning of the hand and to its segmentation from the background. In other words, it means to understand which pixels of the input image belong to the hand. Using the NiTE skeletal tracking it is possible to acquire the position and depth information of the center of the palm. We employ this information to segment the hand from the background using depth thresholding. Depth thresholding is an easy and quick way for real-time hand segmentation and it can be greatly beneficial for separating the hand from the background to exclude the effects of cluttered or dynamic backgrounds. Unlike other approaches that need the hand to be the closest object to the camera or that need the person to keep his/her hand in a predefined position, here the depth thresholding is done automatically by knowing the depth information of the palm center in each frame. And therefore, no restrictions are applied for positioning the hands correctly.

Fig. 2a shows the depth data of the user that is taken from Kinect depth sensor and hand localization using the NiTE skeletal tracking. Different levels of depth are shown with different shades of gray to better show how depth thresholding can be used to segment the hand. The obtained segmented hand is shown in the upper part of Fig. 2b.

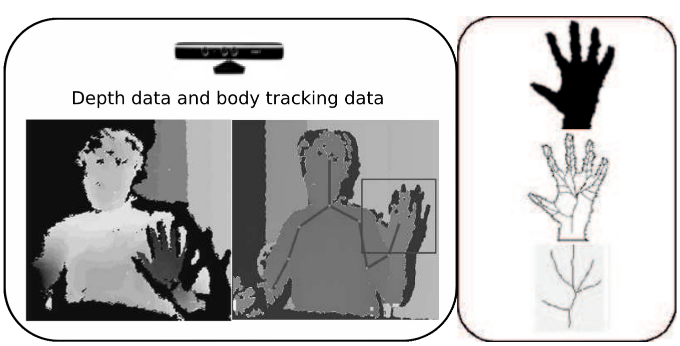
B. Feature Extraction

After hand segmentation, the system needs to extract some hand features to define and then compare different postures. In the proposed approach, skeletons are considered as the features to be used in pose estimation. A skeleton is a graph that summarizes the shape of an object and can be employed as an efficient shape descriptor for object recognition and image analysis. To obtain the skeletons of the hand (skeletonization), the Voronoi Diagram algorithm [26] is applied on the boundary points of the segmented hand. Voronoi diagrams (VD) partition the space in a number of regions based on a set of geometrical points (called seeds), in such a way that each region consists of the points closer to its seed. By applying the VD technique to the boundary points of the segmented hand, a diagram including also the main skeleton of the hand is obtained as shown in the middle part of Fig. 2b, which can be cut out by pruning the extra branches. An example of hand skeleton is shown in the bottom part of Fig. 2b.

Using this approach, defining new postures can be as easy as saving a snapshot of the posture into the system. In the HANDY framework, the user can perform the custom poses in front of the Kinect and the system saves the skeleton data of the specified poses for later use in hand pose and gesture recognition. In Fig. 3, it is possible to view some samples of hand poses that can be defined with our framework and their extracted features (skeletons).

C. Hand Pose Estimation

Hand pose estimation refers to the recognition of a single posture of the hand. This process uses the data provided by the feature extraction module, and the output will be the recognized hand pose for the performed posture. In the proposed approach, it is possible to define a set of hand poses by saving the extracted skeleton of the specific posture in a bank of hand poses. After defining hand poses, each of the performed pose skeletons will be compared with all the poses available in this bank and the most similar one will be chosen as the estimated hand pose. Since the skeletons of the hands can be slightly different even for the same hand posture performed by the same user, a method for comparing the skeleton graphs should be adopted. In our approach, to estimate the similarity of two skeletons, we adopt the Dynamic Time



(a) (b)

Figure 2: (a) Depth data and body tracking data, (b) Hand segmentation (top), application of VD on boundary points (middle) and the resulted skeleton via pruning (button)

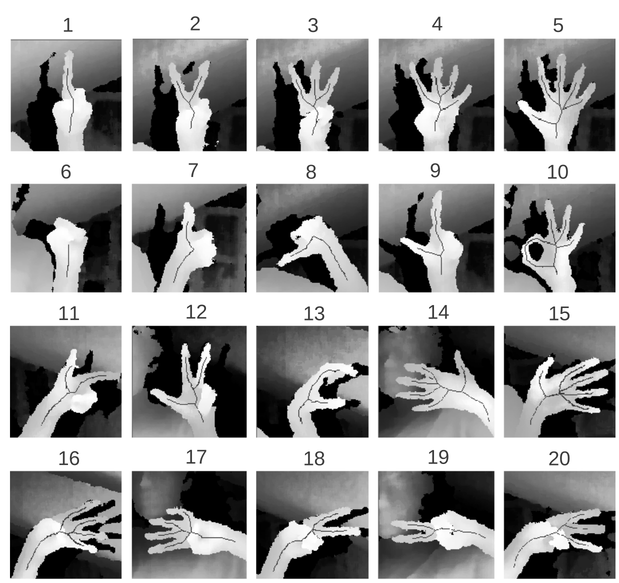


Figure 3: Some sample hand poses defined in the HANDY framework

Warping (DTW) algorithm [27], which allows to compare time series. Our algorithm takes into account the internal order of the points in the skeleton and, for a fair comparison, it normalizes the skeletons to make the the hand pose estimation system invariant to the distance of the hand to the depth sensor and also to the absolute position of the hand in the camera field of view. For further details about the process adopted in our framework the reader may refer to [28].

D. Gesture Recognition

As mentioned before, the goal of the HANDY system is to recognize both static gestures consisting of a single pose (e.g., the poses depicted in Fig. 3) or generic gestures composed of sequences of different hand postures. Some examples of these kinds of gestures are shown in Fig. 4. In both cases, it seems convenient to employ the results of the hand pose estimation as the input of the gesture recognition system.

The hand pose estimator based on DTW is able to recognize the saved postures with a reliable accuracy; however, if we see a gesture as the composition of single poses and for each sub-pose we try to recognize it by comparing it with the bank poses, there may be cases in which the hand pose does not exist in the bank (e.g., for intermediate poses); moreover, when the hand is moving between the poses, the ability of the pose estimator degrades and might not be able to recognize the correct pose. In such cases, the most similar hand posture (skeleton) will be assigned to the performed pose. Furthermore, as already noticed, there are variations in features even when the same gestures are being performed by the same user. In these cases, machine learning methods such as HMMs can be applied to handle the variations and also to recognize the gestures in which the pose estimator is unable to estimate the correct pose sequences. It is worth noting that misrecognition of an intermediate hand pose and assignment of the closest hand pose from the bank is tolerated by the gesture recognition mechanism thanks to HMMs, which are based on stochastic processes. Indeed, the HMM is a collection of states connected by transitions, that can be used to represent the statistical behavior of observable symbol sequences in terms of network states. Each observed symbol (an estimated hand pose) happens corresponding to its probability function in an HMM state. Once a symbol is observed, the HMM can stay in the same state or move to another state due to transition probabilities associated with each state [6].

Using HMMs for gesture recognition, training is needed before the system is able to recognize the gestures. As we will see in the evaluation section, a little amount of training data (in our tests, 20 for each gesture) is needed for the system to obtain an acceptable accuracy in gesture recognition. We assume that the user needs to wave, for the system to start looking for meaningful gestures. While user's hands are in the rest mode (e.g., hands are on the arms of the armchair) the gesture recognizer also goes into the rest mode. The gesture duration is considered to be fixed and we sample the data every 100 milliseconds up to 1 second and the hand pose for each acquired frame will be estimated. In this way, we will have a sequence of 10 estimated hand poses which will be used as the input for training the HMMs and for gesture recognition. The duration of the gesture and the interval of the sampling can be increased and decreased when convenient.

Fig. 5 shows a snapshot of the application that has been developed for testing the introduced system with different users. In the application, it is possible to define and train different hand poses and gestures for any user. In the next section, the results of testing the system are discussed in more detail.

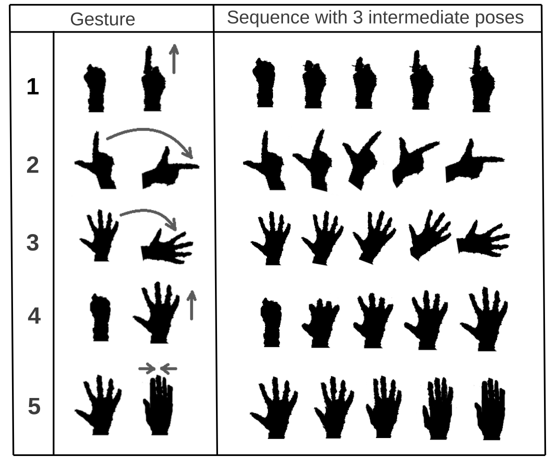


Figure 4: Set of gestures for test

IV. Results and Discussion

For the evaluation of the system, 7 subjects were asked to train and test the system with a set of predefined gestures. All the subjects were right-handed but two of them used their left hand (thus, introducing also a lower self-confidence and fidelity in the performance of the gestures). A set of 20 hand postures, shown in Fig. 3, were recorded for each user in the pose bank. In a first experiment, the recognition of the single isolated poses was tested. In this case, the average accuracy of the pose estimation system for the sample poses in Fig. 3 was around 83.52%. The poses that include the rotation of the wrist showed less accuracy since the pose estimation system is not rotation invariant and the mentioned poses can vary regarding rotation even when performed by the same user. The gesture recognition is tested according to the before mentioned user specific data. However, as we will see next, when a gesture defined as a sequence of poses is considered and during the gesture intermediate hand poses are sampled with a fixed interval, the accuracy can be increased.

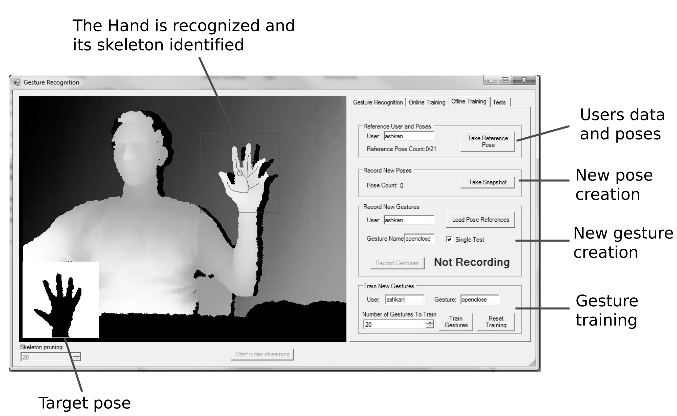


Figure 5: The application for testing the HANDY gesture recognition system

The poses to be stored in the system should be, however, defined according to the nature of the gestures. The more poses are defined in the system, the more similar they get to the postures that are used in gestures, the more accurate will be the inputs of the gesture recognition system. However, there is a trade-off between the accuracy of the input pose sequence and the number of the poses that are defined, as with a large pose bank the pose estimation system can get slower.

In our tests, we have used the 20 poses depicted in Fig. 3 and, based on these poses, tried to recognize the five gestures shown in Fig. 4. These five gestures were chosen taking into account that:

1) Poses with similar skeletons exist in the gesture. Having similar poses can confuse the pose estimation and reduce the accuracy in producing the gesture recognition inputs. For example, for the first gesture in Fig. 4, it is possible to see that its beginning and final poses (pose no. 6 and pose no. 1 in Fig. 3) have very similar skeletons that are composed of a single vertical branch.
2) Some of the poses in the gesture might not exist in the hand pose bank used for pose estimation. For example, the gesture number 5 in Fig. 4 includes poses like the closed hand, the final pose in the sequence, which is not included in the gesture set recorded in the pose bank (depicted in Fig. 3). If a pose does not exist in the pose bank, the most similar pose from the bank will be selected in the pose estimation step.
3) Gestures contain some intermediate poses with lower pose estimation accuracy. As an example, in the gesture number 3 depicted in Fig. 4 the final hand pose (that corresponds to pose number 15 in Fig. 3) has shown a low accuracy of only 40% in the pose estimation system. This is also true for the second gesture in Fig. 4.

These gestures were recorded by a single user as isolated gestures and then used for training and testing the system. In the test, gestures have a fixed length of 1 second, and are sampled 10 times during their performance. In a real application, the number of of sampling and the duration of the gestures can be configured in the system according to the nature of the gestures and the user specific considerations. Evaluation of the HANDY was done in two steps: first, the system is trained for a single user using 20 training data for each gesture. Later, the system is tested for the same user.

Table I shows the accuracy of each gesture per user: the first column represents the user; the second indicates if the left or right hand was used; the next columns represent the five gestures specified in Fig. 4: for each gesture the percentage indicates the accuracy in the recognition of 30 performances of that gesture.

Table II demonstrates the average accuracy of the same results for the left and right-hand users: from the results, no considerable differences can be concluded in the accuracy of the system in the two cases.

Finally, Table III reports the average confusion matrix for the tested gestures with the different users: for example, gesture G1 was recognized correctly in the 94.28% of the cases, it was confused 3.33% with gesture G2 and 2,86% with gesture G4. The average accuracy of 95.57% was concluded during these tests which is acceptable for many applications.

In this first evaluation for each user, we considered the hand poses and the gestures previously defined and trained by the user him/herself. In a second evaluation, we tried to evaluate the accuracy of gesture recognition considering the case of a new user who has not trained the system before. For this evaluation the system was trained with the data from all the subjects except the user who is going to test the system. Only the data from right hands were used since we did not have enough training data for left hands. The results of this evaluation are reported in Table IV. The average recognition accuracy of 96.26% was obtained. This suggests that with more training data some universal gestures could be recognized without initial training.

While testing the system the users reported that they did not feel any delay in the gesture recognition system and therefore that the system is able to respond in real-time. Moreover, subjects were asked to rank the gestures based on their difficulty. Those involving difficult hand manoeuvres like wrist rotation (gestures 2 and 3, see Fig. 4) were commented to be the most difficult ones. However, in general, they found the system easy to use.

V. Conclusions and Future Works

In this work we have presented an approach for a flexible and configurable hand pose and gesture recognition system that

TABLE I: RECOGNITION ACCURACY OF TRAINED GESTURE RECOGNITION SYSTEM FOR SINGLE USERS

Users	R/L-hand	G1	G2	G3	G4	G5
U1	R	100%	90%	100%	96.67%	100%
U2	R	100%	100%	100%	100%	100%
U3	R	100%	100%	96.67%	96.67%	100%
U4	R	90%	100%	100%	100%	100%
U5	R	83.33%	100%	100%	76.67%	96.67%
U6	L	86.67%	80%	100%	100%	100%
U7	L	96.67%	100%	100%	96.67%	93.33%

TABLE II: AVERAGE RECOGNITION ACCURACY OF TRAINED GESTURE RECOGNITION SYSTEM FOR SINGLE RIGHT-HAND AND LEFT-HAND USERS

R/L-hand	G1	G2	G3	G4	G5
R	94.66%	99.95%	99.33%	94%	93.33%
L	91.67%	90%	100%	98.33%	96.66%

TABLE III: AVERAGE CONFUSION MATRIX OF GESTURE RECOGNITION SYSTEM FOR SINGLE USERS

Gestures	G1	G2	G3	G4	G5
G1	94.28%	3.33%	0%	2.86%	0%
G2	0%	95.71%	0.48%	0.48%	2.86%
G3	0%	0%	99.59%	0%	0.48%
G4	0%	0.95%	0%	95.24%	3.81%
G5	0%	0.48%	0.48%	0.48%	98.57%

TABLE IV: RECOGNITION ACCURACY OF TRAINED GESTURE RECOGNITION SYSTEM FOR SINGLE USERS

Users	G1	G2	G3	G4	G5
U1	100%	90%	100%	100%	93.33%
U2	100%	100%	100%	96.67%	100%
U3	100%	100%	100%	86.67%	96.67%
U4	100%	96.67%	93.33%	80%	90%
U5	96.67%	93.33%	93.33%	100%	100%

could be adapted to the specific needs of the user. Specific hand postures can be defined and recognized in this system according to the needs of the user, the application and nature of the dynamic gestures that are based on these postures. In our tests, 20 different hand poses have been considered: the comparison of the performed pose with the skeleton stored in the pose bank led to accurate results in 83.52% of the cases. In addition, the system can be trained with dynamic gestures, considered mainly as a continuous sequence of hand poses, with small number of training gestures (20 to 30) to achieve a good recognition accuracy. We obtained a good accuracy of 95.57% in our tests, involving even gestures with intermediate poses having lower estimation accuracy.

As future work, we plan to improve the hand pose estimation to be able to also recognize the postures in which the fingers might not be visible. The gesture recognition considered in this paper can be extended to consider also hand trajectories, to be able to recognize a larger variety of gestures. We have in mind also to do a comparative evaluation of our recognition system with the similar ones. Further extensions of this system may include also facial expression recognition and vocal orders to support an even wider set of applications that use natural user interfaces.

REFERENCES

[1] "Kinect for windows, official website, http://www.microsoft.com/en-us/kinectforwindows/," accessed: 2014.01.28.

[2] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor-system description, issues and solutions," in Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on. IEEE, 2004, pp. 35–35.

[3] S. Mitra and T. Acharya, "Gesture recognition: A survey," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 37, no. 3, 2007, pp. 311–324.

[4] G. Murthy and R. Jadon, "A review of vision based hand gestures recognition," International Journal of Information Technology and Knowledge Management, vol. 2, no. 2, 2009, pp. 405–410.

[5] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," Urbana, vol. 51, 1999, pp. 103–115.

[6] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden markov models," Image and Vision Computing, vol. 21, no. 8, 2003, pp. 745–758.

[7] N. D. Binh, E. Shuichi, and T. Ejima, "Real-time hand tracking and gesture recognition system," Proc. GVIP, 2005, pp. 19–21.

[8] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, 1989, pp. 257–286.

[9] A. Kundu, Y. He, and P. Bahl, "Recognition of handwritten word: first and second order hidden markov model based approach," Pattern recognition, vol. 22, no. 3, 1989, pp. 283–297.

[10] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on. IEEE, 1992, pp. 379–385.

[11] J. Schlenzig, E. Hunter, and R. Jain, "Recursive identification of gesture inputs using hidden markov models," in Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on. IEEE, 1994, pp. 187–194.

[12] E. D. Mynatt, A.-S. Melenhorst, A.-D. Fisk, and W. A. Rogers, "Aware technologies for aging in place: understanding user needs and attitudes," Pervasive Computing, IEEE, vol. 3, no. 2, 2004, pp. 36–41.

[13] P. Trindade, J. Lobo, and J. P. Barreto, "Hand gesture recognition using color and depth images enhanced with hand angular pose data," in Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on. IEEE, 2012, pp. 71–76.

[14] C. Keskin, A. Erkan, and L. Akarun, "Real time hand tracking and 3d gesture recognition for interactive interfaces using hmm," ICANN/ICONIPP, vol. 2003, 2003, pp. 26–29.

[15] F. Camastra and D. De Felice, "Lvq-based hand gesture recognition using a data glove," in Neural Nets and Surroundings. Springer, 2013, pp. 159–168.

[16] Y. Li, X. Chen, J. Tian, X. Zhang, K. Wang, and J. Yang, "Automatic recognition of sign language subwords based on portable accelerometer and emg sensors," in International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, no. 17. ACM, 2010.

[17] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition," World Academy of Science, Engineering and Technology, vol. 49, no. 1, 2009, pp. 972–977.

[18] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," Computer Vision and Image Understanding, vol. 108, no. 1, 2007, pp. 52–73.

[19] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in RO-MAN, 2012 IEEE. IEEE, 2012, pp. 411–417.

[20] H.-M. Zhu and C.-M. Pun, "Real-time hand gesture recognition from depth image sequences," in Computer Graphics, Imaging and Visualization (CGIV), 2012 Ninth International Conference on. IEEE, 2012, pp. 49–52.

[21] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," in Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on. IEEE, 2004, pp. 529–534.

[22] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 20, no. 12, 1998, pp. 1371–1375.

[23] J. e. a. Molina, "Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models," Machine Vision and Applications, vol. 24, no. 1, 2013, pp. 187–204.

[24] "Openni official website, http://www.openni.org/," accessed: 2014.01.28.

[25] "Nite official website, http://www.openni.org/files/nite/," accessed: 2014.01.28.

[26] F. Aurenhammer, "Voronoi diagrams: A survey of a fundamental geometric data structure," ACM Computing Surveys (CSUR), vol. 23, no. 3, 1991, pp. 345–405.

[27] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in KDD workshop, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[28] M. Teimourikia, "Design and development of a kinect-based multi purpose gesture interface for fragile people," Master's thesis, Politecnico di Milano, October 2013.