



# A hand gesture recognition technique for human–computer interaction <sup>☆</sup>



Nurettin Çağrı Kılıboz, Uğur Güdükbay <sup>\*</sup>

Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey

## ARTICLE INFO

### Article history:

Received 3 April 2014

Accepted 21 January 2015

Available online 2 February 2015

### Keywords:

Dynamic gesture recognition  
Hand gesture  
Finite state machine-based recognition  
Gestural interfaces  
Gesture-based interaction  
Human–computer interaction  
Intuitive interfaces  
Hand trajectory recognition  
Adaptive gestures

## ABSTRACT

We propose an approach to recognize trajectory-based dynamic hand gestures in real time for human–computer interaction (HCI). We also introduce a fast learning mechanism that does not require extensive training data to teach gestures to the system. We use a six-degrees-of-freedom position tracker to collect trajectory data and represent gestures as an ordered sequence of directional movements in 2D. In the learning phase, sample gesture data is filtered and processed to create gesture recognizers, which are basically finite-state machine sequence recognizers. We achieve online gesture recognition by these recognizers without needing to specify gesture start and end positions. The results of the conducted user study show that the proposed method is very promising in terms of gesture detection and recognition performance (73% accuracy) in a stream of motion. Additionally, the assessment of the user attitude survey denotes that the gestural interface is very useful and satisfactory. One of the novel parts of the proposed approach is that it gives users the freedom to create gesture commands according to their preferences for selected tasks. Thus, the presented gesture recognition approach makes the HCI process more intuitive and user specific.

© 2015 Elsevier Inc. All rights reserved.

## 1. Introduction

Various approaches to human–computer interaction (HCI) have been proposed in the last few decades as an alternative to the classic input devices of keyboard and mouse. However, these new techniques have not been able to supersede the old ones due to their lack of intuitiveness. Recently, HCI has regained popularity due to the intuitive and successful interaction techniques of devices such as tablet PCs, smart phones and even smart houses. All these applications use voice commands, mimics, and gestures to interact with humans.

Human–computer interaction with hand gestures plays a significant role in these modalities because humans often rely on their hands in communication or to interact with their environment. Therefore, hand-gesture-based methods stand out from other approaches by providing a natural way of interaction and communication [1]. Many studies evaluate gesture-based interaction techniques, their drawbacks, and propose ways to increase their effectiveness [2–4].

There exist various definitions of hand gestures in the literature. Some studies define gestures as only static postures [5], while

others consider hand motions and trajectory information as a part of the gestures [6]. In the scope of this study, we consider only the motion trajectory of the hand (excluding finger bending and orientation information) to define gestures.

Recognizing gestures is a comprehensive task combining various aspects of computer science, such as motion modeling, motion analysis, pattern recognition and machine learning [7]. Since the beginning of the 1990s, many hand gesture recognition techniques have been proposed. These studies can be divided into two categories, based on their motion capture mechanism: *vision-based* or *glove-based*. Vision-based techniques rely on image processing algorithms to extract motion trajectory and posture information [8–10]. Their success highly depends on the used image analysis approaches, which are sensitive to the environmental factors, such as illumination changes, and may lose fine details due to hand and finger occlusion [11].

Glove-based techniques generally provide more reliable motion data and eliminate the need for middle-tier software to capture hand positions and postures [12]. On the other hand, they require the user to wear cumbersome data gloves and position trackers, and usually carry a few connection cables. These factors reduce intuitiveness and usefulness of these methods and make them costly [12].

Recent developments in technology pave the way for more accurate and affordable motion capture technologies, namely

<sup>☆</sup> This paper has been recommended for acceptance by Yehoshua Zeevi.

<sup>\*</sup> Corresponding author. Fax: +90 (312) 266 4047.

E-mail address: [gudukbay@cs.bilkent.edu.tr](mailto:gudukbay@cs.bilkent.edu.tr) (U. Güdükbay).

depth camera sensors such as Kinect™ and Wii™. Hence, it is possible to retrieve more precise motion data without the limitations of traditional (vision-based and glove-based) approaches. Even small objects like human fingers can be effectively captured by these devices [13]. Researchers also propose dynamic hand gesture recognition algorithms that utilize these devices [14]. Similar to ours, these studies use hand trajectory based gesture recognition algorithms. Although these studies claim that their recognition rate is over 90% with relatively simple and small gesture sets, they use fairly large training sets. Unlike our approach, none of these approaches are capable of recognizing gestures on the fly. Additionally, our gesture recognizer does not require large training sets.

Studies in this field can also be classified by examining whether they recognize static or dynamic gestures. Although static gesture recognition is relatively simpler, it still requires much effort due to the complexity of gesture recognition. Most of the static gesture recognition research focuses on neural-network-centered approaches [15,16], but for dynamic gesture recognition, hidden Markov model (HMM)-based approaches are generally preferred because they yield better results [17–19]. Similar to our study, finite-state machine (FSM)-based techniques [20–22] are also used to recognize dynamic gestures. Other studies suggest using fuzzy logic [23] and Kalman filtering [24] for gesture recognition.

Many gesture recognition techniques such as neural-network [25] and HMM-based [26] approaches require a preliminary training phase in which an extensive training data is fed the system to form the recognizers. Our approach can achieve similar recognition rates without requiring a large training set or on-line training. The other advantage that we obtain from the FSM-based recognizer is that we can detect gestures in a stream of hand motion unlike the other methods [27] where the start and end positions of gestures should be specified explicitly.

We introduce an intuitive approach to teach a machine to recognize a hand gesture command so users can apply them to devices such as TVs or eReaders. This approach allows users to create their own gesture commands for a particular task according to how they think it suits the action.

Similar to the other techniques, the proposed approach consists of two stages: *learning* and *recognition*. In the learning stage, the user is asked to repeatedly perform a particular gesture. The system records the motion trajectory of each gesture sample with a magnetic 3D position tracker attached to the user's hand. Unlike the other approaches [28], motion data is collected in gradient-like form. Instead of noting the absolute position of the hand, its position relative to the previous recording is noted. Additionally, threshold-based filtering is applied to the collected data to reduce noise caused by unintended vibrations and hardware errors. Next, collected motion data is filtered using a component-based sliding window technique for smoothing and further noise removal. Then, the filtered trajectory information is transformed into our gesture representation format, which is basically an ordered sequence of events (directional movements).

In the last step of the learning phase, our approach chooses a few event sequences (using the Needleman–Wunsch sequence-matching algorithm [29]) from the provided samples to form a base for gesture recognizers. The algorithm compares every pair of event sequences (gesture pairs) and computes a similarity score for them. The event sequences with the highest similarity scores are selected to form the bases for the gesture recognizers. Then, a recognizer finite state machine (FSM) is generated based on these chosen gestures. Because FSMs are sequence recognizers, each forward transition in a generated FSM corresponds to an event in the selected sequence in the respective order. This learning phase is repeated for every distinct gesture, with several FSMs produced for each.

In the recognition stage, continuous inputs from the tracker are processed in a similar manner as in the learning stage and fed to all

the recognizer machines. If one of the previously captured event sequences occurs during the session, the respective recognizer machine traverses all the states and reach the final (accepting) state. The resulting gesture recognition event triggers the action assigned for the gesture. With this approach, gestures can be recognized in real time.

One important feature of the proposed dynamic gesture recognition technique is that it can effectively detect gestures in a motion flow regardless of the motion capture technique. Vision-based approaches can be used in the proposed gesture recognition framework instead of the glove-based hand motion capture. The proposed gesture representation and recognition mechanism is especially suitable for vision-based hardware and algorithms. In fact, vision-based approaches may overcome the major problems of the hardware used because they will address the limitations of the device such as restricted motion capture range and carrying an uncomfortable attachment. For example, the results of hand follower algorithms proposed in [30,31] can be easily converted to our gesture representation and can be fed to the recognizer machines. It is even possible to extend the usage area of our approach to the public spaces using the hand segmentation and recognition approaches described in [32] that generate hand coordinates, which is sufficient for us to recognize hand gestures.

The rest of the paper is organized as follows: The proposed approach is described in detail in Section 2. The details of the conducted user study and experimental results presented in Section 3. Analysis and discussion on the experimental results are given in Section 4. Section 5 provides conclusions and future work.

## 2. Proposed approach

### 2.1. Gesture representation

In gesture recognition, representing gestures is a critical issue. We define gestures as a series of events performed consecutively. For trajectory-based dynamic gestures, this is a valid definition because trajectories are a series of directional vectors combined in a particular time interval. In our case, events are directional movements and a gesture is an ordered sequence of these directional movements (see Fig. 1).

In this study, we limit the trajectories to the *xy*-plane for simplicity. Our representation not only allows creating many interesting gestures, it also improves the robustness of the algorithm. It is possible to extend the event (gesture) alphabet with the third dimension, or with other features such as finger movements. Using

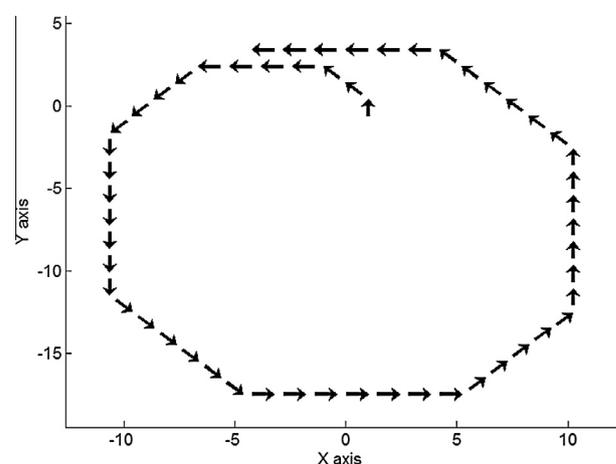


Fig. 1. A gesture (circle) is represented as an ordered sequence of directional movements.

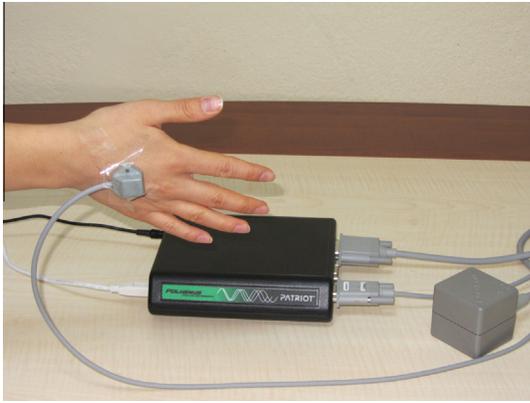


Fig. 2. The gesture recognition setup using a magnetic 3D position tracker.

only 2D, there are eight different directional movements:  $(+x)$ ,  $(-x)$ ,  $(+y)$ ,  $(-y)$ ,  $(+x, +y)$ ,  $(+x, -y)$ ,  $(-x, +y)$  and  $(-x, -y)$ , and they constitute a gesture space large enough to represent a variety of gestures.

## 2.2. Motion capture

To capture hand motions, we use a six-degrees-of-freedom (DoF) magnetic motion tracking device (Polhemus Patriot™) attached to the user's hand (see Fig. 2). The device has a 60 Hz update rate for each sensor, but in our experiments, we observe that a 20 Hz rate is sufficient to teach and recognize gestures. Although we use hardware-based tracking, it is possible to employ computer-vision-based tracking for a more intuitive solution. Because the required motion capture technique does not need a fast update rate or high accuracy, it is also well suited for camera tracking. The cheaper motion tracking devices could be utilized for the motion capture stage for the proposed approach. Gestures are represented as small directional movements so there is no need to maintain the absolute position. This advantage therefore makes the offered solution naturally applicable to accelerometer based motion tracking algorithms. Collected motion data in absolute position format is converted to relative position data (gradient form) while recording. In other words, when the tracker sends a new position reading, its position relative to the previous reading is noted and the direction of the movement is calculated. However, to prevent noise that may be caused by small vibrations in the hand and/or by tracker inaccuracies, relatively small changes from the previous recording are not recorded (cf. Parameters 1 and 2 in Table 2).

## 2.3. Smoothing

Although filtering is applied during the motion capture phase, the collected trajectory data may still contain events that are not part of the gesture due to user reaction error during the initial and final moments of the recording. There also exist a few events that do not fit the natural flow of the trajectory especially at points where a major direction change occurs (see Fig. 3(a) and (b)). To eliminate these minor reaction errors, the beginnings and endings of the trajectory records are discarded (cf. Parameter 3 in Table 2) and a smoothing process is applied to the collected motion data. We use a simple sliding window filter for smoothing. The windows run on the collected data for majority-based filtering. In majority-based filtering, we change the raw input data according to the number of neighboring inputs. If the majority of the inputs in this neighboring window belongs to a single input type (i.e.,  $(+x)$ ), the point at the center of the window is converted to this type (cf.

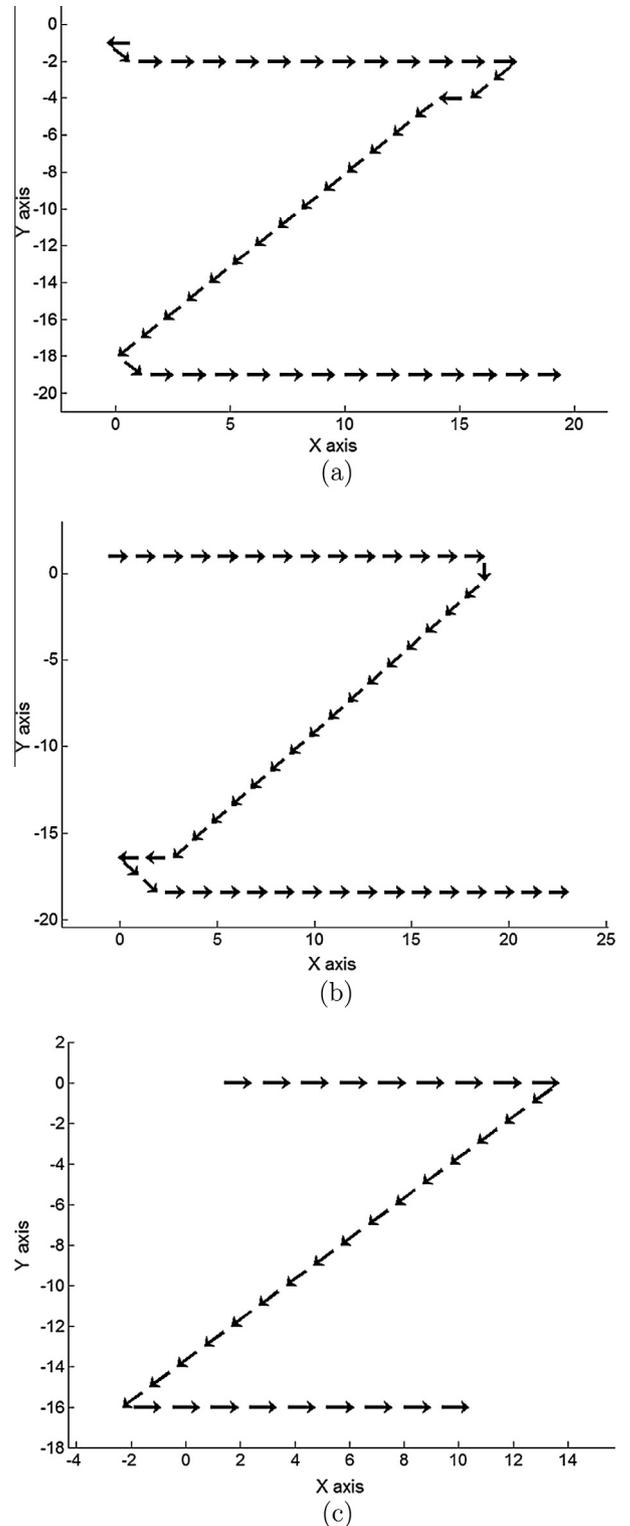


Fig. 3. Two raw gesture motion data (a and b), and the result of filtering (c).

Parameter 4 in Table 2 for the neighboring window size). An input gesture motion data and the results of the applied filtering are shown in Fig. 3.

## 2.4. Selection of best gestures

In the ideal case, when the same gesture is performed, it would yield the same event sequence so the recognizer could be formed

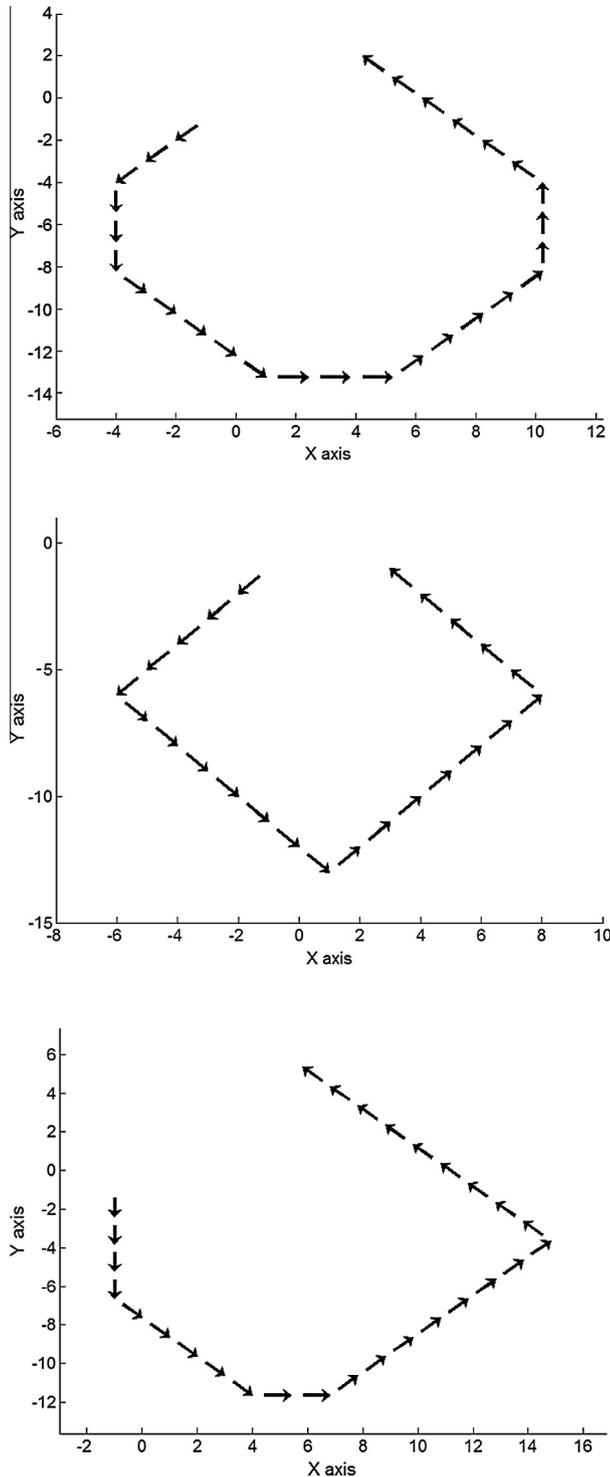


Fig. 4. Captured gesture samples may be different due to the nature of trajectory-based gestures and filtering errors.

from just one gesture sample. However, due to the nature of trajectory-based gestures and filtering errors, the captured gesture samples may not be identical in terms of the resulting event sequences (Fig. 4). To determine the correct series of events that a gesture contains, the system needs several samples of trajectory information, from which “the best” event sequences are chosen. These choices are made by the Needleman–Wunsch [29] sequence matching algorithm that produces a similarity score, which is a global sequence alignment algorithm commonly used in bioinfor-

matics to align two protein or nucleotide sequences. The alignment procedure also computes a similarity score between two sequences. Similarity scores are calculated according to a similarity matrix/function for the characters in alphabets (events). Since events are vectors in our case, the similarity of two “characters” is calculated using the distances between vectors. The gap penalty value for the sequence matching algorithm is set to be over maximum distance value to achieve the same length gesture sequences (cf. Parameter 5 in Table 2).

A total similarity value for each sequence is acquired by summing its pairwise similarity scores. Then, the highest  $n$  (cf. Parameter 6 in Table 2) event sequences are selected to later create recognizers. In other words, gestures that are located closer to the center of the gesture cluster are selected because they are more likely to generate a more generic sequence of events, which can then be used to form the bases for gesture recognizers.

### 2.5. Generating recognizers

Since strings and gestures are represented in the form of event sequences, an analogy between string and gesture recognition problems can be made. When we convert the gesture sequence in Fig. 3(c) into a string, we obtain the following expression:

(+x) (+x) (+x) ...  
 (-x, -y) (-x, -y) (-x, -y) ...  
 (+x) (+x) (+x) ... ,

which can be expressed with the following regular expression:

(+x) + (-x, -y) + (+x) + .

Because our gestures can be represented as regular expressions, an FSM-based recognizer becomes a natural and suitable solution among alternatives. To establish the recognizer machine, we use the gestures (sequences) that were selected in the previous step (see Fig. 5 for a sample gesture recognition machine for the gesture in Fig. 3(c)).

Using FSMs as recognizers ensures that the resulting machines are scale invariant, which means that if trajectories are repeated on a higher or lower scale it can still be recognized. As long as the order of events is preserved, the number of repetitive events does not affect the recognition result.

During the learning phase, a total of  $n \times m$  recognizer machines are generated separately, where  $m$  is the number of gestures and  $n$  (cf. Parameter 6 in Table 2) is the number of selections in the previous stage.

### 2.6. Online gesture recognition

Online recognition of dynamic gestures is achieved using the previously generated sequence recognizers. When the position tracker attached to the user’s hand is activated, it starts to continuously transmit position information to the system. The received absolute position data is converted to the relative (gradient) form and filtered as in the learning phase to reduce the effects of small trajectory errors and to improve the robustness of the algorithm.

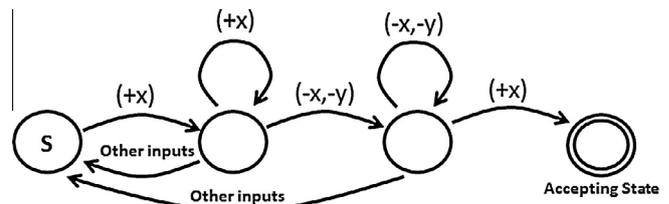


Fig. 5. A sample recognition machine to recognize the gesture in Fig. 3(c).

Before the filtered event data is fed to all recognizer machines in a continuous manner, online filtering is applied to the newly received data to determine whether it is consistent with the previous events. Inconsistent events are not sent to recognizers because they are not part of the intended gestures. The received events cause state transitions in the recognizers. When a machine reaches its accepting state, a gesture recognition event is triggered immediately.

If no state transitions are detected for a particular time interval, a timeout (cf. Parameter 7 in Table 2) mechanism is triggered and the gesture recognizer is reset to the initial state to prevent unnaturally long waits for a gesture recognition event. In the proposed approach, there is no need to specify the start and end positions of gestures because the machine returns to its initial state automatically in the event of an incorrect gesture input or a timeout.

### 3. Experiment

We conducted a user study on a simple virtual reality application in order to assess the usability of the proposed gesture recognition technique. The application is a computer-aided design application in which users can design basic clay models. The design process takes place in the virtual environment that contains volumetric deformable model, design tools and a virtual hand that is driven by the data glove and the tracker. The users manipulate

the design tools and the deformable model via the virtual hand. The deformation on the model is done by stuffing or carving material (voxels) with the help of the tools or directly by the virtual hand.

We selected a pre-trained gesture vocabulary that consists of eleven gestures (see Fig. 6) for evaluation. The recognizable gesture space can be easily extended by the fast learning method described in Section 2. The parameters used in the learning stage to establish the recognizers for the gesture recognition library are shown in Table 2. Each gesture in the vocabulary is mapped to a specific task/action that can be performed in the application (see Table 1).

We assess the technique in terms of performance and attitude criteria [33]. The performance criterion for the method is the gesture recognition rate. To measure the recognition rate, we carefully observe the each participant individually and count the number of trials for a gesture to be recognized. In case of attitude evaluation, we used the following criteria: usefulness, learning, memory, naturalness, comfort, satisfaction and enjoyment. Questionnaires were filled by the participants using a Likert scale from 1 (strongly disagree) to 5 (strongly agree) to assess attitude points of the proposed HCI approach.

A total of 30 volunteers with the average age of 28 (5 female, 25 male) were recruited to participate in the study. The participant's occupation varied; the group included computer scientists, engineers, accountants and economists. None of the participants reported previous experience with a gestural interface or similar

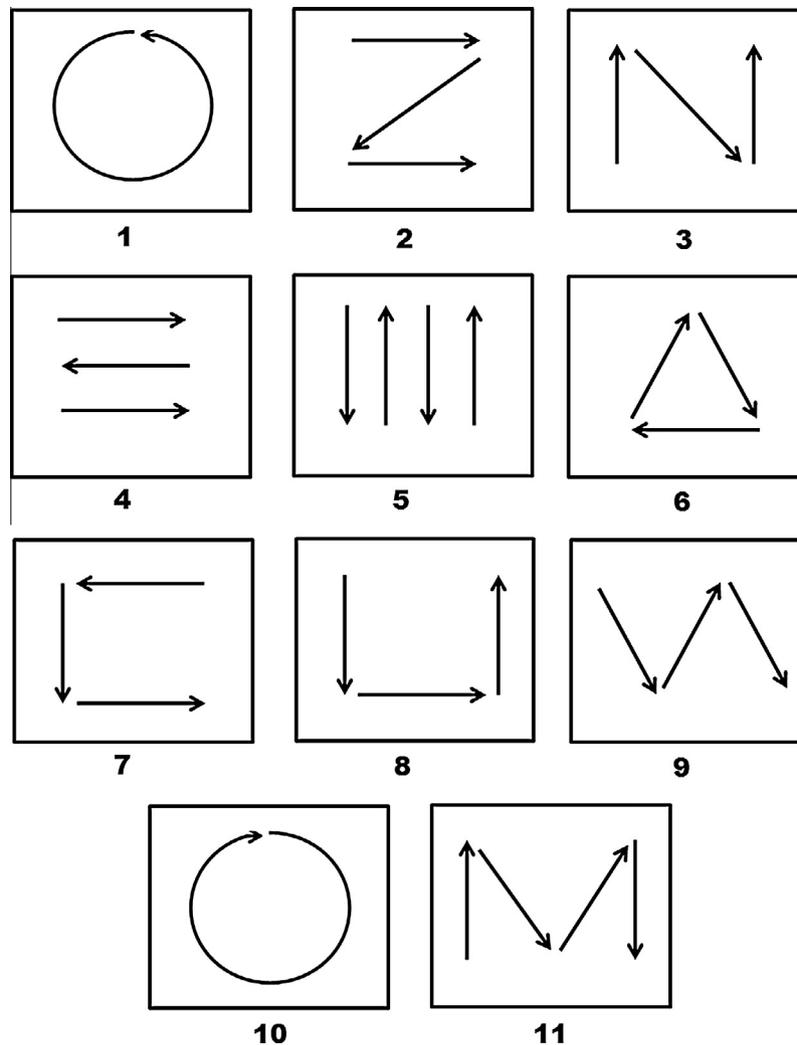


Fig. 6. Gesture vocabulary.

**Table 1**  
Gesture-Action mapping.

Gesture no	Action
0	Rotate the model counter-clockwise
1	Rotate the model clockwise
2	Activate/deactivate tool
3	Change tool mode (Stuffer/Carver)
4	Increase tool size
5	Decrease tool size
6	Activate/deactivate hand deformation
7	Save the model
8	Load the model
9	Activate/deactivate hand mouse
10	Exit the program

**Table 3**  
Gesture recognition rates.

Gesture No	# Gesture trials	# Successful recognitions	Recognition rate
0	321	223	0.69
1	306	198	0.67
2	220	135	0.61
3	112	101	0.90
4	238	193	0.81
5	249	204	0.82
6	106	75	0.71
7	89	69	0.78
8	101	84	0.83
9	218	137	0.63
10	38	34	0.89
Total	1998	1453	0.73

but, all of them were familiar with the classical input devices because they use desktop computers on a daily basis. The experimental setup we use consists of a standard laptop computer (Intel Core 2 Duo CPU (T6600 2.2 GHz), 3 GB RAM, Windows 7 (32 bit), 5DT Data Glove 14 Ultra with USB interface and Patriot™ tracker (Polhemus™) with two tracking sensors.

Each participant was trained on the aim of the VR application and how to perform the gestures to command the application before the experiment. Then, the participants of the user study are asked to design simple models that require the usage of these actions that are mapped to dynamic hand gestures so that the participants experience and evaluate the new technique while designing basic models. The experiments approximately took 20–25 min (including the training phase) for each participant. The gesture recognition rates of the proposed approach are depicted in Table 3. The statistics of the user survey are listed in Table 4.

## 4. Analysis and discussion

### 4.1. Performance

The experimental results show that the average recognition rate of the algorithm is approximately 73% from a stream of motion (cf. Table 3). This indicates that a standard user should perform a gesture  $1/0.73 = 1.37$  times to trigger an application functionality. Although it can still be improved, most of our test subjects find this ratio *satisfactory* for interaction. Thus, we can claim that recognition rate of the proposed technique is high enough to be used as a reliable human–computer interface.

The experiments also show us that sensor incapability of the magnetic tracker is one of the main reasons behind the unrecognised gestures. When the distance between the sensor and the magnetic source exceed certain point, the accuracy of the tracker drops dramatically so after this point, the tracker cannot detect the position of the hand accurately enough to correctly form the gesture

**Table 2**  
The parameters used for the gesture recognition experiments.

No	Parameter	Value	Description
1	Motion capture threshold	3 cm	If the displacement in hand position is lower than the motion capture threshold, it is ignored for the learning and recognition stages
2	Component angle threshold	25°	If the angle between the motion vector and its x, y components is less than the component angle threshold, the respective component of the movement is ignored for the learning and recognition stages
3	Skipped inputs	5	The number of skipped inputs at the start and end of the motion capture
4	Smoothing window size	11	The previous and subsequent five records are considered with the processed input, and the majority of these records are assigned to the processed input
5	Gap penalty	3	The gap penalty value for the Needleman–Wunsch algorithm
6	Selection count	3	The number of best sequences selected from the recorded trajectory motion data
7	Recognition time-out	1500 ms	If no state change is detected in a gesture recognizer by the end of the time-out period, the state machine is reset to the initial state
8	Gesture sample count	8	The number of trajectory motions recorded for the learning stage

**Table 4**  
The statistics of the user survey.

Criteria	Average	Standard deviation
Usefulness	4.24	0.62
Learning	4.38	0.61
Memory	3.90	0.76
Naturalness	3.79	1.06
Comfort	3.17	0.87
Satisfaction	4.21	0.66
Enjoyment	4.59	0.49

sequence. This problem can be eliminated by using a more powerful position tracker or alternative position detection approaches.

We also observe from our experiments that the most of the unrecognised gestures occur in the initial learning phase due to the ill-formed gestures. After the users are adopted the new interface, the performed gestures become healthier (more detectable) and recognition rates increases dramatically. This indicates that higher recognition rates that can reach up to 90–95% can be an estimated from an experienced user.

### 4.2. Attitude

The outcome of the attitude criteria denotes that the participants of the user study find the proposed human computer interface *useful* (4.24) and *satisfactory* (4.21) with very high attitude scores. This is promising for the proposed technique because it indicates that the presented gestural interface can be an alternative interaction approach for classic HCI interfaces.

A surprising result of the attitude evaluation is the relatively low *naturalness* score (3.79) with respect to the other criteria because our claim is to achieve more natural and intuitive HCI interface. However, the critical information for this attitude criterion is the high standard deviation factor. While some participants

think that the pre-selected gestures are very natural for the assigned actions, the others find them quite unnatural. This shows that naturalness is fairly relative to the user. Because the proposed approach is highly adaptable with the fast learning algorithm, the user can replace the assigned actions and gestures with more suitable and natural ones for themselves. This makes the presented technique more superior than the other gesture recognition algorithms that are hard to train.

The lowest attitude criterion is *comfort* (3.17) because of the following two reasons: the users have to perform gestures repeatedly, which can be exhausting after some time, and they have to wear cumbersome motion capture hardware, which is not comfortable. The cumbersome equipment problem can be solved by using an alternative motion capture approach that does not require users to wear gloves or attach motion tracking devices. The fatigue problem is relatively insignificant for applications that do not necessitate continuous interaction. However, it might be a good idea to establish a hand supporting/resting instrument for applications that need constant interaction for a long time.

As a consequence, sufficiently high recognition rates prove the effectiveness of the presented approach to recognize simple and natural gestures that are suitable to command applications. The assessment of the attitude criteria points out that proposed HCI technique can be utilized as an intuitive and natural interface.

#### 4.3. Analysis of learning parameters

We emphasize that the selected parameters have a critical effect on recognition rate. We observe that small values for *motion capture* and *component angle threshold* (cf. Parameters 1 and 2 in Table 2) decrease the recognition rate dramatically because the system records small changes in trajectory that are not part of the intended gesture. On the other hand, choosing a large value for these parameters causes to miss some events that are a part of the intended gesture.

*Smoothing window size* (cf. Parameter 4 in Table 2) also has an important effect on the accuracy. A larger window size causes fine details of the motion to disappear, while a smaller window size may not be able to achieve sufficient smoothing to form gesture recognizers.

Other critical elements in the learning process are the number of recognizer machines and number of gesture samples for each gesture. One of the advantages of the proposed approach is that it does not require a large set of training data. However, preparing a good training set has a crucial importance. A training set that include different versions of the same gesture yields better results because it generates unlike machines that can recognize possible alternatives of the same gestures. In the learning phase of the experiments, we use eight trajectory data for each gesture (cf. Parameter 8 in Table 2). Out of these eight motion data, three of the most alike trajectory sequences are selected to generate recognizers (cf. Parameter 6 in Table 2.) Increasing the number of gesture samples and the number of selected trajectories may improve the recognition rate because it covers different versions of gestures. On the other hand, this may cause some confusion during the recognition phase because it increases the chance of generating similar recognizer machines for various versions of different gestures. Hence, an evaluation should be made to balance the recognition and confusion rate for these parameters.

## 5. Conclusions and future work

We present a simple yet powerful technique to detect and recognize trajectory-based dynamic hand gestures in real time. Gestures are represented with an ordered sequence of directional

movements in 2D space. Gesture data is collected by a magnetic position tracker attached to a user's hand, but the proposed method is also applicable to motion data gathered using vision-based approaches, inertial motion capture algorithms or depth sensors. Motion data in absolute position format is converted to our representation during the motion capture phase.

We introduce a fast learning methodology to facilitate adding new gestures to the recognizable gesture set. A few sample gestures are sufficient to form the gesture recognizers. The learning samples are smoothed to eliminate errors generated by the imperfect nature of human capture data. From these filtered samples, the best sequence of directional movements is selected. The selected learning samples are later processed to generate the gesture recognizers, which are basically FSM sequence recognizers.

The experimental results show that the proposed approach can recognize dynamic hand gestures with an average of 73% accuracy in real time for a vocabulary of eleven gestures from continuous hand motion. The proposed technique's high accuracy and online recognition mechanism make it easily adaptable to any application for gesture-based HCI so that such applications will become more intuitive in how they interact with their users. Assessment of the attitude points shows that the presented gestural interface is an *enjoyable* and *satisfactory* alternative to the classical HCI interfaces.

Another contribution is that a user can create a gesture command set specific to him or her without the need for extensive training, unlike neural-network and HMM approaches. In this way, the HCI process becomes more natural and intuitive. The other favorable functionality that other dynamic gesture recognition approaches do not provide is that the proposed approach can detect and recognize trajectory based dynamic hand gestures without specifying the start and end positions of gestures, thanks to FSM recognizers. This advantage improves the intuitiveness of the interface.

Although the number of studies on gesture recognition is very high, there are very few public datasets that can be used to directly compare different techniques. Most of the publicly available datasets consist of image sequences or videos. However, we need to post process the data with computer vision algorithms to extract trajectory data and experiment with these data sets. Such processing operations can affect the recognition performance drastically. Additionally, the proposed approach can be used together with different motion capture techniques, such as using depth cameras or vision-based techniques, which can collect hand motion trajectory data. Hence, a direct quantitative comparison with the existing studies would not be appropriate. On the other hand, an indirect comparative analysis with the existing approaches can be made. Even though our approach has a relatively low recognition rate compared to the recent studies [19,26], the recognition performance is satisfactory for practical usage according to the conducted user study. Furthermore, the recognition performance improve dramatically after a short adaptation period. In the comparison of the recognition performance, number of training samples should be taken into account because our approach achieves the presented recognition rates with only a small amount of training data.

One important advantage of our approach over the others is that it provides a gesture recognition mechanism that can detect meaningful gestures on the fly. This functionality is essential for practical usage of gesture recognition systems because it significantly simplifies the interaction and make it user friendly.

As a future work, gesture space dimensions can be extended to increase the number of recognizable gestures. For this purpose, depth (z-coordinate) can be added as the third dimension. To add some constraints to recognized gestures or to recognize orientation-based gestures, palm orientation can be utilized. We plan to combine the proposed dynamic gesture recognition approach with

dynamic posture recognition by using a data glove that can capture finger-bending values. The recognition rate can be improved by using more advanced filtering techniques and by deploying an online filtering mechanism for captured data. Utilizing cheaper and more comfortable motion capture techniques, such as depth cameras, inertial sensors, or other vision-based approaches, could be a useful extension.

### Acknowledgment

The first author is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under BİDEB 2210 Graduate Scholarship.

### Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.jvcir.2015.01.015>.

### References

- [1] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, Visual interpretation of hand gestures for human-computer interaction, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (6) (1997) 677–695.
- [2] Siddharth S. Rautaray, Anupam Agrawal, Vision based hand gesture recognition for human computer interaction: a survey, *Artif. Intell. Rev.* 43 (1) (2015) 1–54.
- [3] Donald A. Norman, Natural user interfaces are not natural, *Interactions* 17 (3) (2010) 6–10.
- [4] Gang Ren, Eamonn O'Neill, 3D selection with freehand gesture, *Comput. Graphics* 37 (3) (2013) 101–120.
- [5] Deng-Yuan Huang, Wu-Chih Hu, Sung-Hsiang Chang, Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination, *Expert Syst. Appl.* 38 (5) (2011) 6031–6042.
- [6] Gerhard Rigoll, Andreas Kosmala, Stefan Eickeler, High performance real-time gesture recognition using hidden markov models, in: *Gesture and Sign Language in Human-Computer Interaction*, Lect. Notes Comput. Sci., vol. 1371, Springer-Verlag, Berlin, Heidelberg, 1998, pp. 69–80.
- [7] Ying Wu, Thomas S. Huang, Vision-based gesture recognition: a review, in: *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction (GW'99)*, Springer-Verlag, London, UK, 1999, pp. 103–115.
- [8] Xiaohui Shen, Gang Hua, Lance Williams, Ying Wu, Dynamic hand gesture recognition: an exemplar-based approach from motion divergence fields, *Image Vis. Comput.* 30 (3) (2012) 227–235.
- [9] Jiatong Bao, Aiguo Song, Yan Guo, Hongru Tang, Dynamic hand gesture recognition based on SURF tracking, in: *Proceedings of the International Conference on Electric Information and Control Engineering (ICEICE '11)*, IEEE, 2011, pp. 338–341.
- [10] Meenakshi Panwar, Pawan Singh Mehra, Hand gesture recognition for human computer interaction, in: *Proceedings of the International Conference on Image Information Processing (ICIIP '11)*, IEEE, 2011.
- [11] Chen-Chiung Hsieh, Dung-Hua Liou, David Lee, A real time hand gesture recognition system using motion history image, *Proceedings of the 2nd International Conference on Signal Processing Systems (ICSPS '10)*, vol. 2, IEEE, 2010, pp. 394–398.
- [12] Sushmita Mitra, Tinku Acharya, Gesture recognition: a survey, *IEEE Trans. Syst. Man Cyber - C: Appl. Rev.* 37 (3) (1999) 311–324.
- [13] Zhou Ren, Junsong Yuan, Zhengyou Zhang, Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera, in: *Proceedings of the 19th ACM International Conference on Multimedia (MM '11)*, ACM, 2011, pp. 1093–1096.
- [14] Youwen Wang, Cheng Yang, Xiaoyu Wu, Shengmiao Xu, Hui Li, Kinect based dynamic hand gesture recognition algorithm research, *Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC '12)*, vol. 1, IEEE, 2012, pp. 394–398.
- [15] Cemil Oz, Ming C. Leu, Human-computer interaction system with artificial neural network using motion tracker and data glove, in: S.K. Pal et al. (Eds.), *Pattern Recognition and Machine Intelligence*, Lect. Notes Comput. Sci., vol. 3776, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 280–286.
- [16] Claudia Nolker, Helge Ritter, Visual recognition of continuous hand postures, *IEEE Trans. Neural Networks* 13 (4) (2002) 983–994.
- [17] Yanghee Nam, KwangYun Wohn, Recognition of space-time hand-gestures using hidden markov model, in: *Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST '96)*, ACM, New York, NY, 1996, pp. 51–58.
- [18] Stefan Eickeler, Andreas Kosmala, Gerhard Rigoll, Hidden markov model based continuous online gesture recognition, *Proceedings of the Fourteenth International Conference on Pattern Recognition (ICPR '98)*, vol. 2, IEEE, Piscataway, NJ, 1998, pp. 1206–1208.
- [19] Zhong Yang, Yi Li, Weidong Chen, Yand Zheng, Dynamic hand gesture recognition using hidden Markov models, in: *Proceedings of the 7th International Conference on Computer Science & Education (ICCSE '12)*, IEEE, 2012, pp. 360–365.
- [20] Pengyu Hong, Matthew Turk, Thomas Huang, Gesture modeling and recognition using finite machines, in: *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, IEEE, Piscataway, NJ, 2000, pp. 410–415.
- [21] M. Yeasin, S. Chaudhuri, Visual understanding of dynamic hand gestures, *Pattern Recogn.* 33 (11) (2000) 1805–1817.
- [22] Aaron F. Bobick, Andrew D. Wilson, A state-based approach to the representation and recognition of gesture, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (12) (1997) 1325–1337.
- [23] Oliver Bimber, Continuous 6DOF gesture recognition: a fuzzy logic approach, in: *Proceedings of WSCG'99*, vol. 1, 1999, pp. 24–30.
- [24] Aditya Ramamoorthy, Namrata Vaswani, Santanu Chaudhury, Subhashis Banerjee, Recognition of dynamic hand gestures, *Pattern Recogn.* 36 (9) (2003) 2069–2081.
- [25] John Weissmann, Ralf Salomon, Gesture recognition for virtual reality applications using data gloves and neural networks, *Proceedings of the International Joint Conference on Neural Networks (IJCNN '99)*, vol. 3, IEEE, Piscataway, NJ, 1999, pp. 2043–2046.
- [26] Xiaoyan Wang, Ming Xia, Huiwen Cai, Yong Gao, Carlo Cattani, Hidden-Markov-models-based dynamic hand gesture recognition, *Math. Probl. Eng.* 2012 (2012) 11. Article No. 986134.
- [27] Mingyu Chen, Ghassan AlRegib, Biing-Hwang Juang, Feature processing and modeling for 6d motion gesture recognition, *IEEE Trans. Multimedia* 15 (3) (2013) 561–571.
- [28] Ciprian David, Vasile Gui, Pekka Nisula, Veijo Korhonen, Dynamic hand gesture recognition for human-computer interactions, in: *Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics*, 2011, pp. 165–170.
- [29] Saul B. Needleman, Christian D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, *J. Mol. Biol.* 48 (3) (1970) 443–453.
- [30] Marco Rocchetti, Gustavo Marfia, Marco Zanichelli, The art and craft of making the tortellino: playing with a digital gesture recognizer for preparing pasta culinary recipes, *Comput. Entertainment* 8 (4) (2010) 5–24.
- [31] Marco Rocchetti, Angelo Semeraro, Gustavo Marfia, On the design and player satisfaction evaluation of an immersive gestural game: the case of tortellino xperience at the shanghai world expo, in: *Proceedings of the 29th ACM International Conference on Design of Communication (SIGDOC '11)*, ACM, 2011, pp. 45–50.
- [32] Marco Rocchetti, Gustavo Marfia, Angelo Semeraro, Playing into the wild: a gesture-based interface for gaming in public spaces, *J. Vis. Commun. Image Represent.* 23 (3) (2012) 426–440.
- [33] Brian Shackel, Usability – Context, Framework, Definition, Design and Evaluation, Cambridge University Press, New York, NY, 2009. Chapter 2.