Human gesture recognition through a Kinect sensor

Conference Paper · December 2012

DOI: 10.1109/ROBIO.2012.6491161

4 authors:

Ye Gu
Oklahoma State University - Stillwater
18 PUBLICATIONS **375** CITATIONS
SEE PROFILE

Ha Do
Oklahoma State University - Stillwater
26 PUBLICATIONS **802** CITATIONS
SEE PROFILE

Yongsheng Ou
Chinese Academy of Sciences
203 PUBLICATIONS **2,445** CITATIONS
SEE PROFILE

Weihua Sheng
Oklahoma State University - Stillwater
251 PUBLICATIONS **5,108** CITATIONS
SEE PROFILE

Proceedings of the 2012 IEEE
International Conference on Robotics and Biomimetics
December 11-14, 2012, Guangzhou, China

Human Gesture Recognition through a Kinect Sensor

Ye Gu, Ha Do, Yongsheng Ou and Weihua Sheng

Abstract— **Gesture recognition can be applied to many research areas, such as vision-based interface, communication and human robot interaction (HRI). This paper implements a non-intrusive, real-time gesture recognition system using a depth sensor. Related features are obtained from the human skeleton model generated by the Kinect sensor. Hidden Markov Models (HMMs) are used to model the dynamics of the gestures. We conducted offline experiments to check the accuracy and robustness of the system. Online experiments were also performed to verify the real-time requirement. Final results indicate that the average recognition accuracy is around 85% for the subject who provides the training data and 73% for the other subject who does not. The system was also used to interact with a mobile robot through gestures. This application indicates that it is robust to work in real-time.**

I. INTRODUCTION

A. Motivation

A gesture is a motion of the body that contains information. Edward T. Hall, a social anthropologist claims that 60% of all our communications are nonverbal [1], and gestures are widely used from expressing emotions to conveying information. Therefore, gesture recognition has applications in many research areas, such as human machine interaction (HMI), human robot interaction (HRI) and social assistive robotics (SAR) [2], [3].

Currently, vison based sensors and motion sensors are widely used for gesture recognition. Vision based sensors include 2D and 3D sensors. However, there are some limitations on 2D-image based gesture recognition. Firstly, images may not be under consistent lighting. Secondly, items in the background may increase the difficulty of recognition. Also, it is hard to obtain the orientation information from 2D images when it comes to temporal gesture recognition. On the other hand, most of the motion sensors need to be attached to the human body, which may make the user feel uncomfortable. With the emergence of the Kinect sensor [4], the depth information obtained makes it possible and convenient to get not only position information, but also orientation information. Compared to motion sensors, it is totally non-intrusive. In this paper, our objective is to use a 3D sensor to build a gesture recognition system for human robot interaction.

Ye Gu Ha, Do and Weihua Sheng are with School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA. ye.gu, ha.do, weihua.sheng@okstate.edu

Yongsheng Ou is with Shengzhen Institute of Advanced Technology. Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China. ys.ou@siat.ac.cn

B. Related Work

Traditional gesture recognition uses vision information. Depending on the type of input data, the approach for interpreting a gesture could be done in different ways. The most widely used approaches are the appearance-based algorithms which use 2D images or videos for direct interpretation [5]. Meanwhile, the skeletal-based algorithms make use of 3D information to identify key elements of the body parts in order to obtain several important parameters, like palm position or joint angles.

Marcel *et al.* [6] proposed a hand gesture recognition method based on input-output Hidden Markov Models (HMMs), where gestures are extracted from a sequence of video images by tracking the skin-color blobs corresponding to the hand in a body-face space centered on the face of the user. Two kinds of gestures can be recognized. Sanchez-Nielsen *et al.* [7] proposed a fast segmentation process to extract the moving hand from the whole image. This method is able to deal with a large number of hand shapes against different backgrounds and lighting conditions, and can also identify the hand posture from the temporal sequence of segmented hands. Although 2D tracking offers the position information of the target (hands or joints), for temporal gesture recognition, the orientation information is also very critical. Therefore, 3D sensor based gesture recognition is gaining more and more attentions. Breuer *et al.* [8] described a pilot study of hand gesture recognition with a novel IR time-of-flight range camera. The system was able to recognize the seven degree of freedom of a human hand with a frame rate of 2-3 Hz. This is a promising result and defines a road map for further research. Sung *et al.* [9] performed detection and recognition of unstructured human activity in unstructured environments with a RGB-D sensor. The detection algorithm is based on Maximum Entropy Markov Model (MEMM).

Due to the advancement in MEMS and VLSI technologies, wearable sensors based gesture recognition has been gaining attention. Several researchers use multiple sensors worn on human body to record data of human movements. With inertial sensor (nIMU) from MEMSense [10], Zhu *et al.* [2] implemented a neural network based system for gesture spotting and a hierarchical HMM for context-based recognition. Bashir *et al.* [11] proposed a Reduced Dynamic Time Warping (RDTW) approach for handwriting recognition. The system can authenticate individuals and classify handwritten items like PIN words or just a short sequence of isolated characters. Xu *et al.* [12] developed a gesture recognition system based on HMMs using the Cyberglove

978-1-4673-2127-3/12/$31.00 © 2012 IEEE

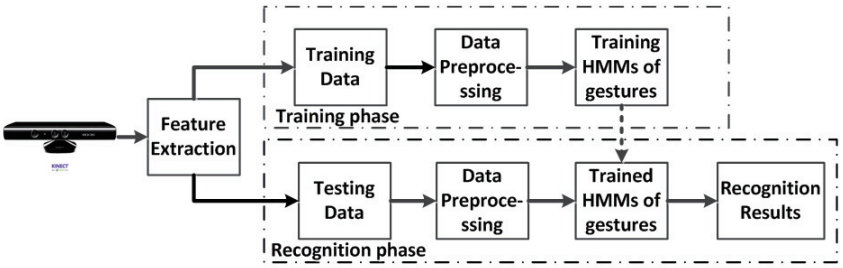

Fig. 1: Block diagram for supervised learning based gesture recognition.

[13]. They processed the data of different joint-angles in the hand, estimated from multiple sensors in the Cyberglove and recognized gestures from the sign language alphabet.

In this work, we perform temporal human gesture recognition using an inexpensive Kinect sensor. Despite the limitations of the device, it has caught on to a large (and growing) extent in the marketplace, which has brought 3D sensor based gesture recognition into the mainstream. Through using this sensor, the non-color based features of the human gestures can be extracted. The features are not sensitive to changing of lightening condition and ordinary image noise.

The rest of this paper is organized as follows: in Section II, the recognition approach is introduced; Section III presents the experiment implementations; in Section IV, the experimental results are analyzed; finally in Section VI, the conclusion is drawn and some future works are discussed.

II. METHODOLOGY

Most of the complete gesture recognition systems consist of three layers: detection, tracking and recognition. The detection layer is responsible for defining and extracting related features. The tracking layer is responsible for performing temporal data association between successive image frames. The recognition layer is responsible for grouping the spatiotemporal data extracted in the previous layers into particular classes of gestures [14].

In our system, the first two layers are implemented with the support of Kinect drivers and the middleware. Our work is focused on classification. HMMs [15] are adopted for the gesture modeling and recognition. The block diagram is shown in Fig. 1. The major steps are introduced in details.

A. Feature Extraction

Human gesture can be represented by several methods, such as hand shape or position, orientation and movement of the body. This work focuses on the movement of the upper limbs, specifically the left arm movement. Selecting good features is crucial to gesture recognition. Extracting the most relevant features while getting rid of the unrelated features can decrease the computation cost significantly. This is very important for real-time processing. The input sensor for our system is a Kinect sensor that gives a RGB image as well as depth at each pixel. With the *OpenNI* driver and a middleware called *NITE* [16], a person who is detected in the sensor view can be tracked as a rigid skeleton with fifteen joints (shown in Fig. 2). The algorithm extracts features that represent joint angles with respect to the person's torso.

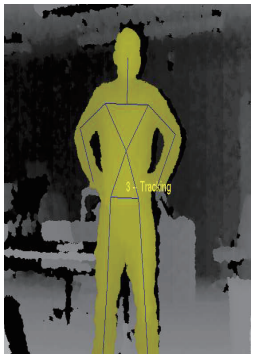

Fig. 2: Human skeleton model from the Kinect sensor.

B. Data Preprocessing

In the training phase, the data preprocessing consists of two steps, segmentation and symbolization. Segmentation is to segment the training data from the raw data. Symbolization is the process of converting the feature vectors into finite symbols because only discrete HMMs are considered. The joint quantization of a block of parameters can be obtained through Vector Quantization [17]. On the other hand, the data preprocessing step in the recognition phase is to cluster testing data with the centroids obtained from the symbolization step.

C. Hidden Markov Models (HMMs)

Modeling the low-level dynamics of human motion is important not only for human tracking, but also for human motion recognition. It serves as a quantitative representation of simple movements so that those simple movements can be recognized in a reduced space by the trajectories of motion parameters [18]. HMM is a type of statistical model. It is characterized by the following components: the number of state in the model, the number of distinct observation symbols per state, the state transition probability distribution, the observation symbol probability distribution and the initial state distribution. It has already become a general method to modeling speech signals. Because of the similarities between temporal gesture signal and speech signal, HMM can be applied to gesture recognition.

Each activity is characterized by an HMM. For each HMM, two procedures are required. In the training phase,

Fig. 3: Snapshots for the gesture “wave”.

the aim is to adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$, where A is the state transition probability distribution, B is the observation symbol probability distribution, π is the initial state distribution. No tractable algorithm is known for solving this problem exactly, but a local maximum likelihood can be derived efficiently using the Baum-Welch algorithm [19] or the Baldi-Chauvin algorithm [20]. The Baum-Welch algorithm is an example of a forward-backward algorithm, and is a special case of the Expectation Maximization.

In the recognition phase, we try to efficiently compute $P(O|\lambda)$, the probability of the observation sequence given the model. Solving this problem allows us to choose the model which best matches the observations. We apply the Viterbi algorithm [21] to solve this problem.

III. IMPLEMENTATION

Five gestures are defined for the experiments: come, go, wave, rise up and sit down. The gesture for “wave” is shown in Fig. 3. Each gesture is encoded by an HMM. All the gestures are made using the left arm. Features are extracted from the four joint angles: left elbow yaw and roll, left shoulder yaw and pitch.

A. Training Phase

The flowchart of the training phase is shown in Fig. 4. Each model is trained with fifteen sets of training data with sampling rate of 20 Hz. A rule-based method is adopted for training data segmentation. The starting pose is defined, and then each training data set consists of thirty data points after the starting point. All the gestures have the similar duration

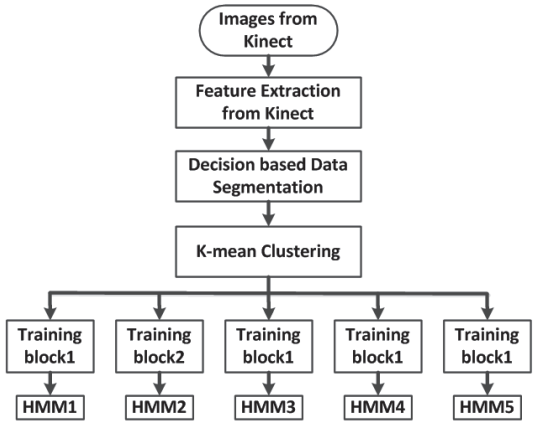

Fig. 4: Flowchart of the training phase.

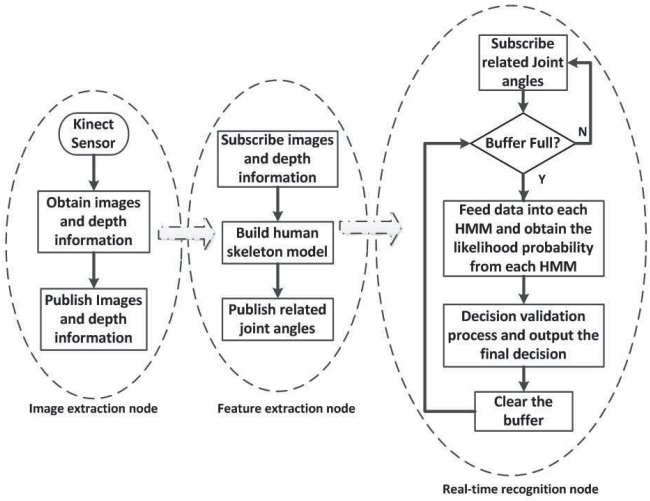

Fig. 5: Flowchart of the recognition phase.

to each other. Each gesture is about a second and a half long. Then, K-means clustering [22] is used to convert the vectors into the observable symbols for HMMs. The centroids from K-means are saved for further testing data. To balance the computational complexity, efficiency and the accuracy, we set up parameters for HMM as follows: the number of states in the model is 30; the number of distinct observation symbols is 6.

B. Recognition Phase

In the recognition phase, we test the trained HMMs by two different testing subjects, the subject who participated in the training phase, and the one who did not. For real-time processing, the Robot Operation System (ROS) framework is adopted. One of the advantages of the ROS framework [23] is that a system built using ROS consists of a number of processes, potentially on a number of different hosts, connected at runtime in a peer-to-peer topology. So instead of writing a whole program for all the functions, we can write each function as a node, and then run these nodes simultaneously. This parallel processing structure can enhance the efficiency of the program. In our case, there are three nodes. The image extraction node extracts the image and depth information from the Kinect sensor. The feature extraction node subscribes to the image extraction node and uses this information to create the human skeleton model. Then, the feature extraction node extracts and publishes the joint angles of the human skeleton. The real-time recognition node subscribes to the feature extraction node for the joint angles and save them into a buffer. This node then calculates the probability of the observation sequence given each model. The one with the maximum likelihood determines the type of the HMM. To get rid of noise and decrease false alarm rate, first we use the variance of the input to judge if it is a gesture or not; secondly, we set a threshold for each HMM. If the likelihood is smaller than the threshold, it is treated as noise. The flowchart of the recognition phase is shown in Fig. 5.

IV. EXPERIMENTAL RESULTS

In this section, we present the experimental results.

A. Recognition Results

For the offline experiment, we collected the testing data and save them to a file for post processing. Since it is offline-processing, we ignore the computation cost and use a sliding window with a size of 30 data points and a step size of 1 data point. One of the offline recognition results is shown in Fig. 6. Both the ground truth and the recognition results are shown in the figure. Each gesture is made with one stroke. The subject remains still for a couple of seconds between any two gestures. Most of the gestures are recognized. Only one missed the detection, which is labeled by a black circle in the figure. No false alarm occurred.

A robustness test has also been done. For each gesture, we run the experiment at a different speed to check how the recognition models perform. Each gesture is repeated three times, at normal, quick and slow speed respectively. Four out of five slow and normal gestures have been recognized. The duration of the slow gestures is much longer than the sliding window used. As long as it is detected once, it is classified as being detected. In contrast only one out of five fast gestures have been recognized which indicates the key features cannot be captured if the motion is too fast compared with the training data. The result is shown in Fig. 7.

The processing time of the recognition is very short compared to the data collection, therefore it won't cause data missing problems while executing the recognition algorithm. A sliding window of size 20 with 50 % overlaps is used. A majority voting rule is applied to the results of three consecutive windows to generate one gesture recognition decision.

For application purposes, two requirements are proposed; the system should allow the user to make any number of strokes they prefer, and most of the gestures should be recognized. The user should also be able to switch from one gesture to another directly without any pause. To test the robustness of the system, two experiments are designed. Fig. 8 shows that when the user did multiple repetitions of the

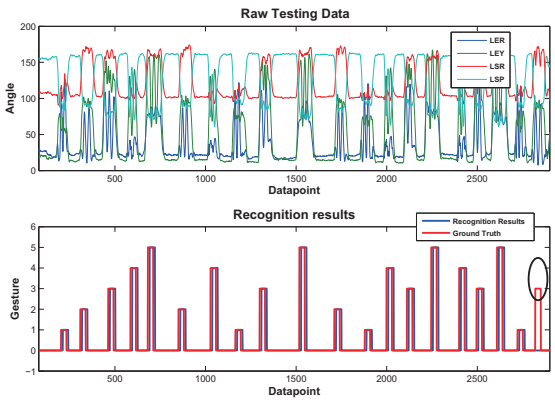

Fig. 6: Offline recognition results 1. Most of the gestures are recognized correctly.

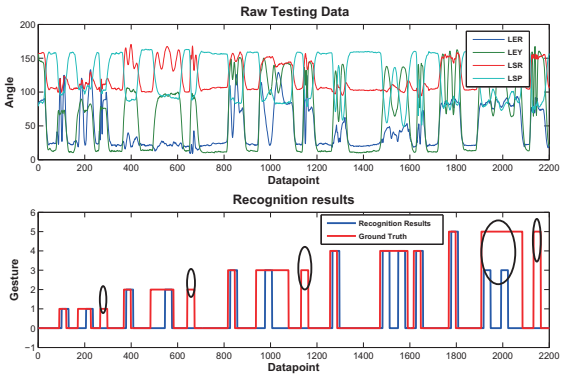

Fig. 7: Offline recognition results 2. Most of the normal speed and slow speed gestures can be recognized. However, several quick speed gestures are not recognized.

same gesture (Gesture 1) consecutively without any pause, all the gestures can be detected. On the other hand, it is shown in Fig. 9-(a) through Fig. 9-(g) that the user changed the gesture from 1 to 5 successively, and from Fig. 9-(g) to Fig. 9-(i), the user consecutively switched the gesture from 5 to 3 to 2 to 1 and then ended up with the initial position. The overall results show that none of the gestures misses the detection and no false alarm occurs. Except for the minor delay in detection, the recognition performance is robust.

Table I shows the likelihood values for five different sequences under different models. Each row is the likelihood values for one data set under different HMM parameters. The value in bold is the greatest likelihood among the five and the HMM index number corresponds to the type of the gesture. In addition, some statistical results are also collected. Table II and Table III show the accuracy for each gesture with two different subjects. The training data were collected from the first subject. The sum of each row i is the total number of the gestures that have been made. The column j of row i gives the number of the gesture i being detected. The sixth column gives the number of missed detections for each gesture. In the final column, the accuracy is given. The results show that no false alarm happened, since the gestures defined are very different from each other. The detection performance of the new user is not as good as the first user who provided the training data. This is mainly because of the gesture differences.

Another experiment is designed to check the effective range of the Kinect sensor. The user stands in different

TABLE I: Likelihood for Different Gestures Under Each HMM

Gesture type	HMMs				
	1	2	3	4	5
1	**-11.018**	-42.893	-inf	-inf	-inf
2	-337.303	**-10.451**	-inf	-inf	-inf
3	-inf	-inf	**-134.581**	-inf	-inf
4	-inf	-inf	-inf	**-13.541**	-inf
5	-inf	-inf	-inf	-inf	**-575.772**

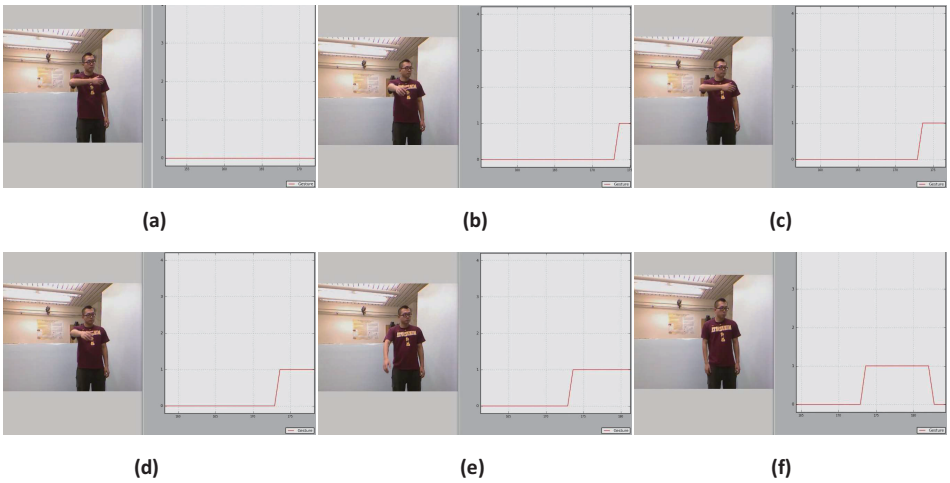

Fig. 8: Online recognition results 1. The user keeps doing gesture 1 without any pause. (images are mirrored)

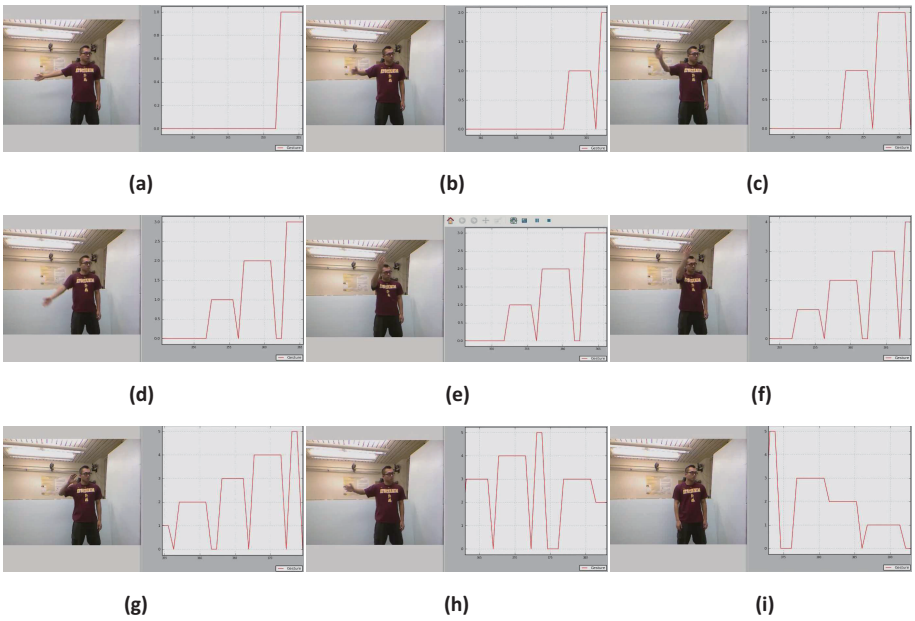

Fig. 9: Online recognition results 2. The user switches from one gesture to another without pause. (images are mirrored)

distance and angles with respect to the Kinect sensor. The results indicate that, as long as the human skeleton model can be created reliably, the angles with respect to the user's own torso can be extracted accurately. Then the detection performance will be reasonably good.

TABLE II: Accuracy for different gestures with the trainer

Ground truth	Gesture Recognized						Test accuracy
	1	2	3	4	5	6	
1	34	0	0	0	0	7	**.8262**
2	0	36	0	0	0	8	**.8182**
3	0	0	35	0	0	6	**.8537**
4	0	0	0	42	0	8	**.8400**
5	0	0	0	0	45	5	**.9000**

B. *Gesture based Mobile Robot Control*

To check our recognition system in real-world applications, our system is applied to control a mobile robot. The platform is shown in Fig. 10 which contains a Kinect sensor, a fit-PC2 and a Pioneer robot. The fit-PC2 is a small energy-efficient fanless PC [24]. The Pioneer robot is

TABLE III: Accuracy for different gestures with the non-trainer

Ground truth	Gesture Recognized						Test accuracy
	1	2	3	4	5	6	
1	17	0	0	0	0	4	**.8095**
2	0	19	0	0	0	7	**.7307**
3	0	0	15	0	0	7	**.6818**
4	0	0	0	21	0	8	**.7241**
5	0	0	0	0	14	5	**.7368**

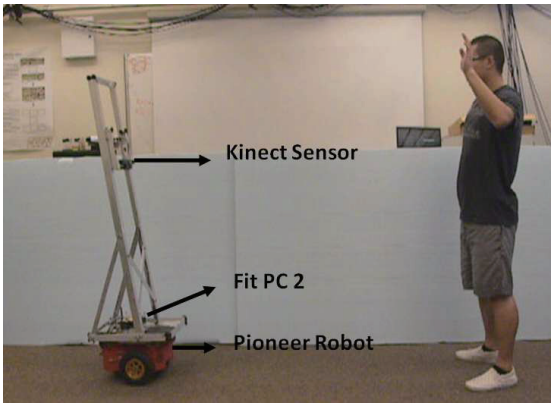

Fig. 10: Controlling a mobile robot with gestures.

fully programmable [25]. Four left arm gestures are defined to control the mobile robot. The commands are "moving forward", "moving backward", "turning left" and "turning right". The results show that despite of the monior delay, the robot can follow the human's command correctly and stably.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, a non-intrusive human gesture recognition system is implemented with a Kinect sensor. Both the online and offline tests show promising results. The average recognition accuracy is around 85% for the subject who provided the training data and 73% for the other subject. The recognition system has the following features: (1) Easy training: the user can simply train the system by recording the gesture to be detected. (2) Person independent: the system can be trained by one person and used by others with acceptable performance. (3) Orientation and distance independent: the system can recognize gestures even if the trained and recorded gestures do not have the same orientations or distance with respect to the sensor. (4) Speed flexibility: the system is able to recognize gestures if they are performed faster or slower (within certain ranges) compared to the training data.

Currently we have only recognized left arm gestures. However, it is very convenient to extend the whole system to other joints or even tracking the whole body movement with all the joints included. With more Kinect sensors, it is plausible to do human daily activity recognition. Voice and gesture inputs complement each other and when used together, create an interface more powerful than either modality alone. Both the gesture recognition and voice recognition module will be integrated with our robot platforms. Instead of using them separately, an appropriate decision fusion method should be proposed.

VI. ACKNOWLEDGMENTS

This project is partially supported by the NSF grant CISE/CNS 0916864, CISE/CNS MRI 0923238 and IIS1231671. Thanks to Jeremy Evert for setting up the mobile robot platforms.

REFERENCES

[1] G. Imai. Body language and nonverbal communication. [Online]. Available: http://www.csupomona.edu/~tassi/gestures.htm/
[2] C. Zhu, W. Sun, and W. Sheng, "Wearable sensors based human intention recognition in smart assisted living systems," in *Information and Automation, 2008. ICIA 2008. International Conference on*, june 2008, pp. 954 –959.
[3] M. Billinghurst, *Chapter 14 Gesture Based Interaction*, 2011.
[4] Z. Zhang, "Microsoft kinect sensor and its effect," *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4 –10, feb. 2012.
[5] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 677–695, 1997.
[6] S. Marcel, O. Bernier, J.-E. Viallet, and D. Collobert, "Hand gesture recognition using input-output hidden markov models," in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 456 –461.
[7] E. Sánchez-Nielsen, L. Antón-Canalís, and M. Hernández-Tejera, "Hand gesture recognition for human-machine interaction," in *WSCG*, 2004, pp. 395–402.
[8] P. Breuer, C. Eckes, and S. Mller, "Hand gesture recognition with a novel ir time-of-flight range camera: a pilot study," in *Proceedings of the 3rd international conference on Computer vision/computer graphics collaboration techniques*, 2007, pp. 219–226.
[9] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Human activity detection from rgbd images," *CoRR*, vol. abs/1107.0169, 2011.
[10] Memsense. [Online]. Available: http://www.memsense.com/index.php/inertial-sensor-modules/
[11] M. Bashir, G. Scharfenberg, and J. Kempf, "Person authentication by handwriting in air using a biometric smart pen device," in *BIOSIG*, 2011, pp. 219–226.
[12] C. Lee and Y. Xu, "Online interactive learning of gestures for human/robot interfaces," in *IEEE International Conference on Robotics and Automation*, 1996, pp. 2982–2987.
[13] Cyberglove. [Online]. Available: http://www.cyberglovesystems.com/products/cyberglove-ii/overview
[14] X. Zabulis, H. Baltzakis, and A. Argyros, *Vision-based Hand Gesture Recognition for Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc. (LEA), 2009.
[15] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.
[16] Openni. [Online]. Available: http://www.openni.org/Documentation/
[17] D. O. T. Jr., "Hidden markov models for gesture recognition," *Masters Thesis*, pp. 1–52, 1995.
[18] Y. Wu, T. S. Huang, and N. Mathews, "Vision-based gesture recognition: A review," in *Lecture Notes in Computer Science*. Springer, pp. 103–115.
[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society Series B Methodological*, vol. 39, no. 1, pp. 1–38, 1977.
[20] P. Baldi and Y. Chauvin, "Smooth on-line learning algorithms for hidden markov models," in *Neural Computation*, vol. 6, no. 2, 1994, pp. 307–318.
[21] M. S. Ryan and G. R. Nudd, "The viterbi algorithm," Coventry, UK, Tech. Rep., 1993.
[22] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, 2000, pp. 456 –461.
[23] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
[24] Compulab. [Online]. Available: http://www.fit-pc.com/web/fit-pc/
[25] Adept mobilerobots. [Online]. Available: http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx