The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

A Hand-Gesture-Based Control Interface for a Car-Robot

Xing-Han Wu[1], Mu-Chun Su[1], and Pa-Chun Wang[2]
[1]Department of Computer Science and Information Engineering, National Central University, Taiwan, R.O.C.
[2]*Cathay General* Hospital, Taiwan
muchun@csie.ncu.edu.tw

***Abstract*—In this paper, we introduce a hand-gesture-based control interface for navigating a car-robot. A 3-axis accelerometer is adopted to record a user's hand trajectories. The trajectory data is transmitted wirelessly via an RF module to a computer. The received trajectories are then classified to one of six control commands for navigating a car-robot. The classifier adopts the dynamic time warping (DTW) algorithm to classify hand trajectories. Simulation results show that the classifier could achieve 92.2% correct rate.**

I. INTRODUCTION

Recently, with the technological progress of industrial production, accelerometers have already been commercialized and widely used in various applications such as robot balance [1]-[2], fall detection [3], etc. For example, Kun and Miller adopted an accelerometer in a robot for adaptive dynamic balance [1]. Nagasaki et al. designed a running experiment of humanoid biped with accelerometers [2]. A personal emergency response system with a tri-axe accelerometer which could differentiate falls from human daily activities (e.g., stand-to-sit, sit-to-lying, etc) was proposed in [3]. In addition, a very popular application of accelerometers is the Wiimote which is the control interface for Nintendo Wii game consoles. A Wiimote consists of a 1024×768 infrared camera and a 3g 8-bit 3-axis accelerometer. It is a kind of somatosensory interaction which allows a user to play a video game in an easy and intuitive way.

The somatosensory interaction is one of the most user-friendly interactive interfaces for controlling objects. Motivated by the idea of a Wiimote, we try to implement a interface which allows a user to navigate a car-robot in a somatosensory interactive way. An easy way is to directly use a Wiimote to control a car-robot; however, the price a Wiimote is not very low and Wiimote's size is not very small either. Therefore, the interface developed by us adopts a small sized accelerometer module instead of the traditional Wiimote. To navigate a car-robot, we need at least six different hand gestures shown in Fig. 1. An immediate problem to be solved is the hand gesture recognition problem.

Recently, there have been many different hand gesture recognition systems, such as vision-based trajectory recognition systems [4]-[5], and inertial-based trajectory recognition systems [7]-[11]. No matter cameras or accelerators are used in the hand gesture systems; the core module is a hand gesture recognition algorithm. The dynamic time warping algorithm (DTW) and the Hidden Markov model (HMM) are two most popular algorithms employed to recognize hand gestures. For example, Niezen and Hancke adopts the DTW to recognize hand gestures for a mobile phone [9] and Shiqi et al. uses the HMM to recognize handwritten characters [10].

The HMM is a very efficient algorithm for gesture recognition; however, the price paid for the high efficiency is the length training time and a large amount of training data. As for the DTW algorithm, it is efficient for recognizing a small amount of hand gestures since it doesn't have the HMM's drawbacks. Since only six different hand gestures need to be classified, we decided to adopt the DTW algorithm as the core recognition module.

The hand-gesture-based control interface proposed by us is introduced in Section II. The Simulation results are given in Section III. Finally, Section IV concludes the paper.

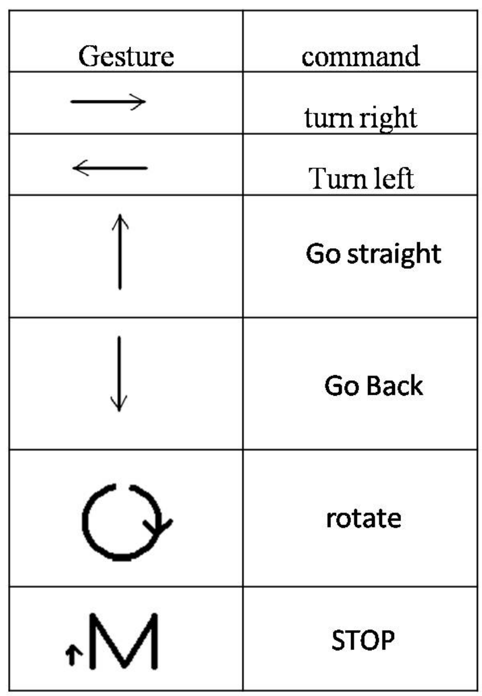


Fig. 1. The six hand gestures used to navigate a car-robot.

II. SYSTEM OVERVIEW

Fig. 2 shows the hand-gesture-based control interface for controlling a car-robot. Based on the interface, a user with a 3-axis accelerometer module attached to his or her wrist can directly use hand gestures to navigate a car-robot. The 3-axis accelerometer module senses the hand trajectories and then wireless transmitted to a PC via a RF module. Then the core hand gesture recognition module adopts the DTW algorithm to recognizes the trajectories. In the following, it sends a control command wirelessly to the car-robot's receiving module. The robot is then navigates according to the received command.

A. Hardware

On the PC side, the interface consists of a 3-axis accelerometer (x-axis = frontal, y-axes = sagittal, z-axes = vertical) ADXL330 accelerometer (range: ±3g, sensitivity: 300mV/g, typical bandwidth: 1600Hz , noise density: 280µg/√Hz rms, operating voltage range: 1.8~3.6 V), a ATMEGA168V microcontroller (small size , high performance , low power, 6-channel 10-bit ADC, 16k bytes of in-system self-programmable flash program memory), and an AM24L01BS-U-based RF (2.4G Hz wireless link) module.

On the car-robot side, each car-robot also has an ATMEGA168V microcontroller and an RF module. The RF module receives commands from the computer and the microcontroller ATMEA168V on the robot coverts the received command to a PWM signal to control robot's motors. Then the car-robot navigates according to the corresponding received command.

Fig. 3 shows the accelerometer module designed by us and Fig. 4 shows the car-robot developed by our Lab.

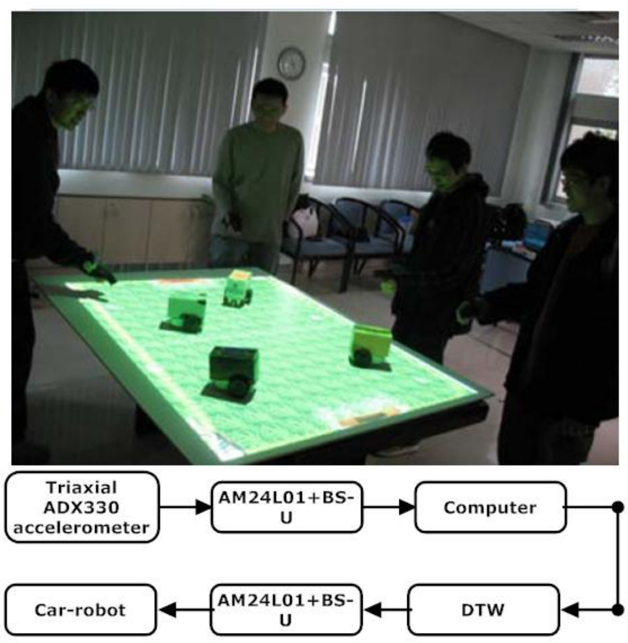


Fig.2. The hand-gesture-based interface for a car-robot.

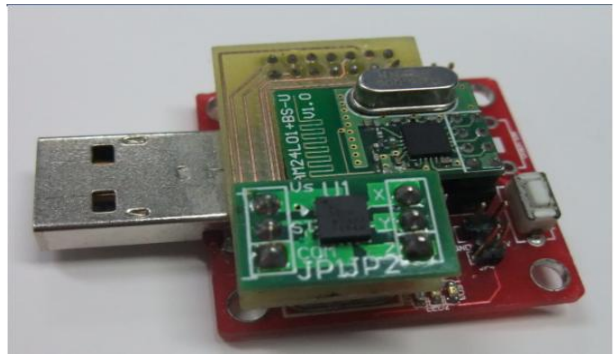


Fig. 3. The accelerometer module.

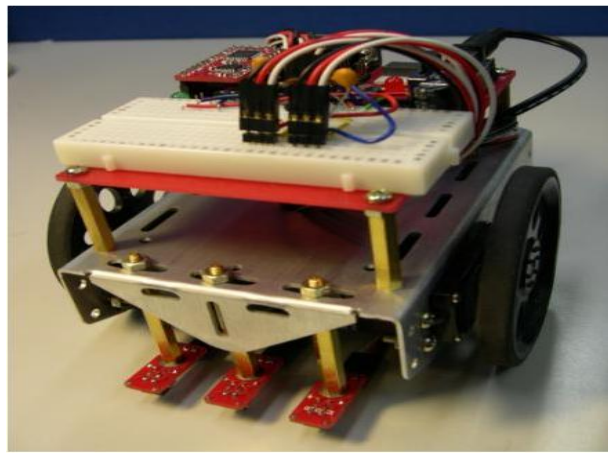


Fig. 4. The car-robot used in the experiments.

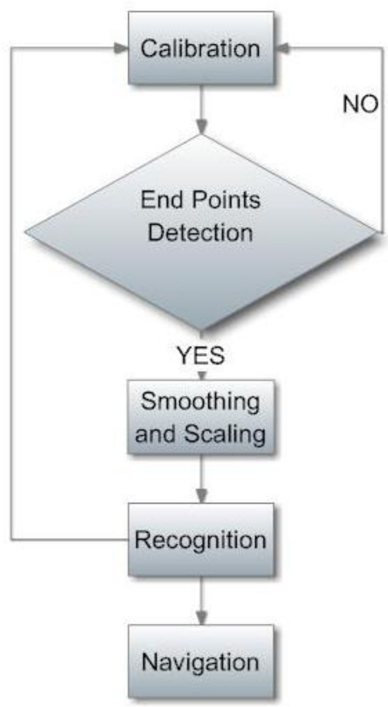


Fig.5 The flowchart for recognizing a hand trajectory.

B. Software

Fig. 5 shows the flowchart for recognizing a hand trajectory. The recognition procedure involves 5 steps as follows:

Step 1. Calibration: The way how a tri-axes accelerometer module is attached to a user's wrist varies from person to person. Even more, it may vary from time to time during the control procedure for the same user. For example, a user may wear the module with a certain tilting angle. Without any hand movement, the accelerator already has some g values on the sensor. Therefore, the module has to be calibrated for each user at the beginning of the control procedure. A kind of calibration procedure has been introduced in [11]. Similar to the method in [11], we calibrate the module in the following way. To calibrate the accelerator module as shown in Fig. 6, we first measure the tilting and rotation angles by the following formula:

$$\theta_x = \sin^{-1} g_x \tag{1}$$

$$\theta_y = \sin^{-1} g_y \tag{2}$$

where θ_x is the tilt angle of x-axe, θ_y is the rotating angle of y-axe, g_x is the g value of x-axe accelerator. g_y is the g value of y-axe accelerator.

After the tilting and rotation angles have been computed, all three axis accelerator sensed data has to employ the following transformation:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \times \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \times \begin{bmatrix} A_x' \\ A_y' \\ A_z' \end{bmatrix} \tag{3}$$

where , A_y , and A_z are the accelerometer data after multiplying rotation matrix, respectively. A_x' , A_y' , and A_z' are the accelerometer data before the transformation.

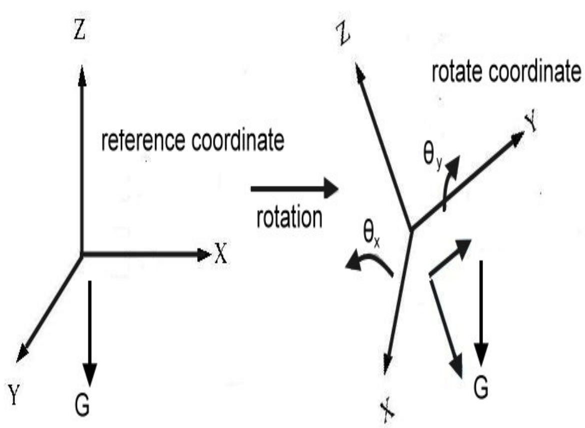


Fig. 6 Calibration via the coordinate transformation.

Step 2. End Points Detection: Before we can use the DTW algorithm to classify the recorded trajectories, the end points of the received trajectories have been detected first. The end points mark the beginning and the ending of the trajectory sampling data.

The acceleration signal changes accordingly as the acceleration module moves. Therefore, the amount of acceleration changes (AAC) can be used to detect the end points:

$$AAC(t) = \sqrt{[A_x(t)-A_x(t-1)]^2 + [A_y(t)-A_y(t-1)]^2 + [A_z(t)-A_z(t-1)]^2} \tag{4}$$

where $A_x(t)$, $A_y(t)$, and $A_z(t)$ represent the acceleration data at time t in *x*-,*y*-, and *z*-axes, respectively.

When the AAC exceeded a certain pre-specified threshold, it represents that a starting point of motion is detected. The end point is detected when the AAC falls below the threshold for 0.5 seconds.

The performance of the end points detection depends on the value of the pre-specified threshold. If the threshold is set too low, a small shaking movement would be detected as a new trajectory. On the contrary, if the threshold is set too high, a slow moving trajectory would not be detected. From our many experiment, the 0.1g would be a good threshold.

Step 3. Smoothing and Scaling: After detecting the end points, we smooth the data to reduce the impact of noise. In our system, we use the second-order Hanning filter to smooth the data as follows.

$$y(t) = \frac{1}{4}[x(t) + 2x(t-1) + x(t-2)] \tag{6}$$

where $y(t)$ represents the smoothed data and $x(t)$ represents the original data at time t.

After smoothing, we need to scale the smoothed data to lie in the interval [0, 1]. Without the scaling process, trajectories resulted from the same hand gesture may be different.

Step 4. Recognition: The DTW algorithm is a time-warping algorithm for recognizing spatial-temporal sequences. It works as follows. First, for each class, one or more than one sequences are stored as template sequences. Then the test sequence is compared with the pre-stored template sequences, and the corresponding similarity score (or distance) for each template is computed. An important step in these comparisons is the alignment of the test sequence in time with each template sequence due to the variations in the sequence lengths. Fig. 7 illustrates a typical warping path for the DTW algorithm.

In our recognition module, for each hand gesture, we randomly select one trajectory as the corresponding template trajectory for the gesture as shown in Fig. 8. The resulting test trajectory is then compared to the template sequence for each hand gesture. The two trajectories to be compared are dynamically time aligned and the resultant alignment path of maximum similarity is computed. Finally, the test trajectory is claimed to be the class with the largest similarity.

Step 5. Navigation: After the trajectories have been classified, the corresponding command is then transmitted to the car-robot.

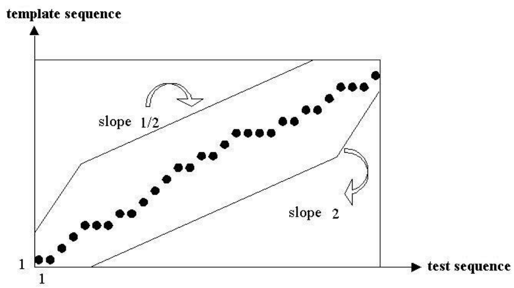


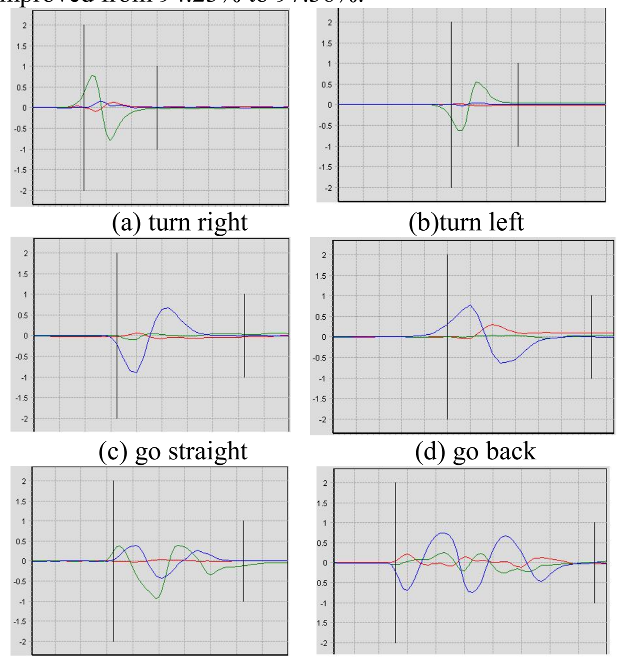
Fig. 7. Typical warping path for the DTW algorithm.

III. SIMULATIONS

In order to test the performance of the proposed hand-gesture-based interface for navigating a car-robot, we need to generate a comprehensive data base to test the interface. There are several factors making the problem of hand gesture recognition become very challenging. First of all, human variability causes the same type of hand gestures to be conducted differently each time. Even for the same person, different instances of the same type of hand gestures may not be identical. Secondly, the same hand gesture can be performed in different positions and / or with different sizes. Thirdly, the same hand gesture can be performed in different speed. Therefore, we asked 8 subjects to generate the data base. Each subject was asked to draw 6 types of hand gestures as shown in Fig. 1. Each subject was asked to draw the same type of hand gesture with 3 different sizes corresponding to a small size, a medium size, and a large size. Each size of a hand gesture was drawn for 5 times. Therefore, each hand gesture had 120 data. Finally, the data base consists of 720 data.

For each hand gesture, we randomly select one trajectory as the corresponding template for the hand gesture to be used in DTW algorithm. Table I is the experimental result of the recognition results. The average recognition rate was 88.8%. By further analyzing the hand gestures, we found that the six hand gestures involved only planar trajectories. Therefore, we could remove the dimension where the variation of the acceleration signals was the least amount. In the experiments, we considered only the y-axe and z-axe acceleration data and used them for hand gesture recognition. Table II shows the recognition results after we took out the x-axe information. The average recognition rate was improved from 88.8 % to 92.2%.

We also used the HMM algorithm to classify the same data set for comparison. Table III is the experimental results based on the HMM algorithm. For considering only the y-axis and z-axis information, the average recognition rate was improved from 94.23% to 97.36%.



(a) turn right (b)turn left

(c) go straight (d) go back

(e) rotate (f) stop

Fig.8. An example of the trajectory for each hand gesture. A long vertical line indicates a starting point and a short vertical line indicates an ending point. Blue lines mean the z-axe, red lines mean the x-axe, and green lines mean the y-axe. (a) Turn right (b) Turn left (c) Go straight (d) Go back (e) Rotate (f) Stop.

TABLE I.
THE RECOGNITION RESULTS USING TRI-AXIS INFORMATION

		Actual trajectory					
Recognized trajectory		→	←	↑	↓	O	M
	→	109	7	4	0	2	1
	←	0	95	3	1	3	0
	↑	7	18	96	4	1	2
	↓	1	0	3	114	2	3
	O	0	0	0	0	112	0
	M	3	0	14	1	0	114
	individual recognize rate	90.8%	79.1%	80%	95%	93.3%	95%
	average recognition rate	88.8%					

TABLE II.
THE RECOGNITION RESULTS USING TWO-AXIS INFORMATION

		Actual trajectory					
Recognized trajectory		→	←	↑	↓	O	M
	→	109	1	4	0	1	1
	←	0	113	1	0	0	0
	↑	7	6	95	5	0	2
	↓	1	0	3	115	1	2
	O	0	0	0	0	118	1
	M	3	0	17	0	0	114
	individual recognize rate	90.8%	94.1%	79.1%	95.8%	98.3%	95%
	average recognition rate	92.2%					

TABLE III.
THE RECOGNITION RESULTS OF HMM ALGORITHM

Data set							
		→	←	↑	↓	O	M
tri-axes	training	98.8%	94.3%	90.9%	87.5%	93.1%	97.7%
	testing	100%	96.8%	100%	78.1%	100%	96.8%
	average recognition rate	94.23%					
two-axes	training	100%	100%	98.7%	91.2%	100%	98.7%
	testing	97.5%	97.5%	92.5%	87.5%	100%	100%
	average recognition rate	97.36%					

IV. CONCLUSION AND FUTURE WORK

In this paper, we introduced a hand-gesture-based interface for navigating a car-robot. A user can control a car-robot directly by using his or her hand trajectories. In the future, we will directly use a mobile phone with an accelerometer to control a car-robot. We also want to add more hand gestures (such as the curve and slash) into the interface to control the car in a more natural and effectively way.

Acknowledgement:

This study was partly supported by the joint research fund (Project number: 99 CGH-NCU-A4) of the Cathay General Hospital and the National Central University of Taiwan and by the National Science Council under the 98-2221-E-008-085-MY2, 98-2221-E-008-094-MY3, and 98-2631-S-008-003-.

REFERENCES

[1] A. Kun and W. T. Miller, "Adaptive dynamic balance of abiped robot using neural networks", *IEEE Conference on Robotics and Automation,* pp. 240-245, 1996.

[2] T. Nagasaki, S. Kajita, K. Kaneko, K. Yakoi, and K. Tanie, 2004, "A running experiment of humanoid biped", *Proc. of IROS2004*, pp. 136–141, 2004.

[3] A. Y. Jeon, I. C. Kim, J. H. Jung, S. Y. Ye, J. H. Kim, K. G. Nam, S. W. Baik, J. H. Ro, and G. R. Jeon, "Implementation of the Personal Emergency Response System", *International Journal of Biological and Medical Sciences 4*:1 2009

[4] M. Singh, M. Mandal and A. Basu, "Visual Gesture Recognition for Ground Air Traffic Control using the Radon Transform", *to be presented at IEEE/RSJ IROS 2005.*

[5] K. K. Kim, K. C. Kwak, and S. Y. Ch, "Gesture Analysis for Human-Robot Interaction", *Proc. of the 8th Int. Conf. on Advanced Communication Technology,* Vol. 3, pp. 1824–1827. 2006

[6] R. Plamondon and S. S, "On-line and off-line handwriting recognition: A comprehensive survey", *IEEE Transactionson Pattern Analysis and Recognition*, 22(1):63–84, January2000.

[7] E. Choi , W. Bang , S. Cho, J. Yang, D. Kim, and S. Kim, "Beatbox music phone: gesture-based interactive mobile phone using a triaxisaccelerometer", *IEEE international conference on industrialtechnology,* ICIT 2005, pp. 97–102. 2005

[8] C. Zhu and W. Sheng, "Online Hand Gesture Recognition Using Neural Network Based Segmentation", *IEEE/RSJ IROS* 2009

[9] G. Nieze and G.P. Hancke, "Gesture recognition as ubiquitous input for mobile phones", *International Workshop on Devices that Alter Perception (DAP 2008)*, conjunction with Ubicomp 2008 (2008)

[10] Z. Shiqi, Y. Chun and Z. Yan, "Handwritten character recognition using orientation quantization based on 3D accelerometer", *the 5th Annual International Conference on Mobile and Ubiquitous System*s, Dublin, Ireland, 25 July 2008.

[11] J. Lee and I. Ha, "Real-Time Motion Capture for a Human Body using Accelerometers", *Robotica,* vol. 19, pp. 601-610, 2001.