TOWARDS DEVELOPMENT OF A ROBOTIC GUIDE DOG

Manish Walia, Taha Lukmanji, Ronnie Farrell, James R. Green
Department of Systems and Computer Engineering, Carleton University, Ottawa, ON

INTRODUCTION

According to Statistics Canada, there are more than 87,000 legally blind people living in Canada, and 37 million worldwide [1-2]. Many visually-impaired people use guide dogs to assist them in their daily lives. Guide dogs are expensive in terms of both money and time. It takes about 6 to 8 months of training and costs about $20,000 to fully train a guide dog [3], however this cost is normally subsidized by organizations such as CNIB and CCB. In addition, guide dogs have a limited service life of 8 to 10 years [4]. Furthermore, the minimum age requirement for the owner is 16 years in Canada [5]. Thus, traditional guide dogs do present a number of limitations and this motivates the current work.

There are a number of technology-based products available to assist visually impaired people achieve independence. These include the Tormes Navigation Aid for the Blind [6], the GuideCane [7], the Canadian Bank Note Reader (Brytech Inc., Ottawa), and the Trekker Talking GPS [8]. In the present study, we introduce a research initiative aimed at developing an alternate solution to replace guide dogs – The Robotic Guide Dog (RGD). The main concept behind it is to have the RGD do everything a real guide dog can with the addition of smart technologies like GPS, sensors, internet tracking, voice recognition, and image processing. This has similar goals to other electronic travel aids (reviewed in [7]), however the RGD will be an autonomous companion rather than a worn device. Upon completion of this multiyear initiative, the robotic guide dog will be able to execute user specific voice commands, detect obstacles, determine its location, read and interpret street signs, and navigate typical environments. It will have to do all of this while safely guiding the user to their destination. The present study reports on the first year on this project which has three principle objectives: 1) identify and acquire a suitable robotic platform for future development, 2) implement a voice recognition subsystem to verbally control the dog, and 3) select a suitable microcontroller board to interface with the robot and support the addition of future sensor subsystems.

METHODS

This year's robotic guide dog project is divided into three main stages: Robotic Dog Platform, Expansion board, and Voice Recognition.

A. Robotic Dog Platform

After evaluating a number of robotic dogs available on the market, a configuration of the Bioloid Comprehensive Kit from Robotis [9] was ultimately selected as the robotic dog platform. While the Robotis kit enables arbitrary configuration of its components, for our purposes, the kit was configured as a dog

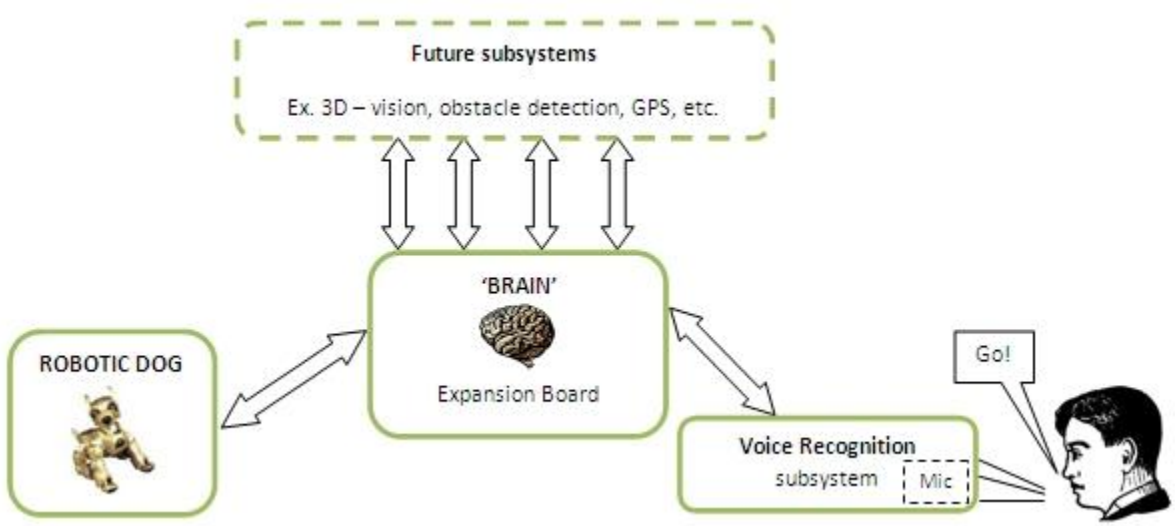

Figure 1 – Robotic Guide Dog flow diagram

because it aesthetically resembles a true dog, which may improve adoption by visually impaired users and acceptance from the general public. The Robotis platform was chosen over other robotic dogs, such as the Sony AIBO or the Genibo-QD Robotic Dog, since its motions and control strategy are completely customizable and it can be interfaced with external sensor subsystems via a serial port.

The kit consists of three main hardware components: CM-5, AX-12+ and AX-S1 [9]. The CM-5 is the microcontroller module which acts as the control centre for the robot. The robot contains 19 AX-12+ servo actuators which serve as the joints of the robot. Each AX-12+ controls its speed, monitors it position, and senses both temperature and load. The robot has one AX-S1 sensor module that acts as the eyes and ears of the robot. It contains with a three direction infrared sensor and can sense distance, brightness, temperature, and sound.

Three software components allow the robot to operate and function as the user desires. The Behaviour Control Program (BCP) sets the rules for the robot on situation recognition, judgment and behavior. This package allows one to develop new control strategies for the robot using a form of graphical programming. The Motion Editor helps the user create complicated motions. These motions can then be called as part of control programs developed uing the BCP. The Robot Terminal is used for robot management and acts as a serial communication terminal that can be used when the user wants to send data to and from the robot. Low level communication with individual components is also possible via this interface.

Initially, the Bioloid robotic kit was configured using a template provided by Robotis. This configuration is illustrated in Figure 2. We require the RGD to move at approximately human walking speed (3-4kph or 80-110cm/s). However, the walking speed was much to slow with speeds of approximately 8cm/sec. To enhance the speed of the dog we added wheels to act as the paws. With the default wheels provided (diameter=5cm), the speed of the dog doubled to about 16cm/sec. As the AX-12+ servos are capable of high torque (16.5kg-cm at 10V) at a maximum of 110 revolutions per minute, we believe that further increasing the speed of the dog can be achieved by increasing the diameter of the wheels [9]. Currently, the robotic dog has been programmed to move forward, move backward, turn both left and right, to stop, sit, and stand. Basic communication is also possible using audible tones from the speaker in the AX-S1.

Figure 2 – Robotis Bioloid Dog Configuration

B. Expansion Board

Expansion board is an important part of the Robotic Guide Dog as it acts as a 'brain' of the dog. The reason why we need an external expansion board is because the robot is not capable of concurrently handling numerous sensor subsystems such as GPS, stereoscopic image processing, and proximity sensors. The expansion board will form the basis for future developments of the RGD to make the dog more environmentally aware and intelligent. Given our requirements (overall size, power consumption, IO ports, processor power, memory, A/D conversion), the following 3 potential candidates were identified: 1) Freescale M52259EVB Board, 2) MIT Handy Board and 3) RoBoard RB-100 [10]. Ultimately, the RoBoard RB-100 (see Figure 3 below) was selected due to the following features:

- Small form factor – 9.6x5.6cm – which enable us to mount this board on the top of CM-5 (9x5.8 cm).
- Processor - Vortex86DX, a 32-bit x86 CPU running at 1000MHz with 256MB DRAM which is capable of running different OS from a Micro SD card – Windows, DOS and Linux
- I/O Ports – 4 COM, 3 USB connections, 10/100M LAN, Mini PCI socket, 24 GPIO, 24 PWM channels, 8 A/D channels, and microphone input and audio output ports. The latter components are required for the voice recognition subsystem. At least one serial port was needed to communicate with the CM-5.
- API – RoBoard provides an open source C++ API for general purpose I/O functions with sensors and actuators.
- Low power usage – 5V @ 400 mA

We tried to install different operating systems including MS Windows XP, BartPE, Ubuntu 9, Damn Small Linux, Fedora 10, CentOS 5, and Puppy Linux. However, only Fedora 10 and MS Windows XP were successfully installed due to driver compatibilities. For development purposes, we selected Fedora 10 because Linux allows us to exercise more control over

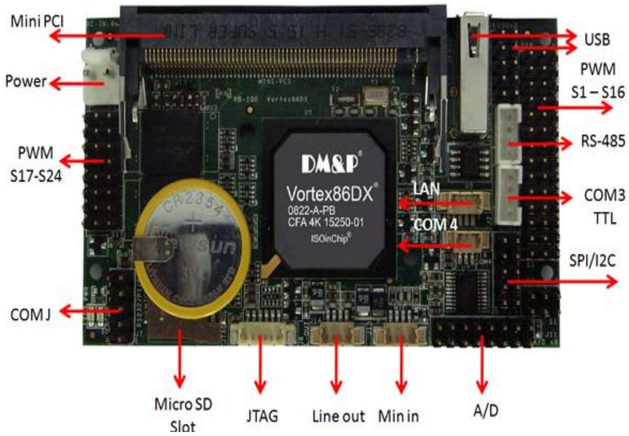

Figure 3 – RoBoard RB-100 [10]

the processes in the OS. Memory usage of Windows XP and Fedora have been compared in table 1.

We are currently using Linux run level 1 (i.e. text-only mode) due to its low memory usage which enables us to develop different applications on the board. It is clearly indicated from the Table 1 that the memory consumption of OS for runlevel 1 is the minimum which leaves 90% of memory for future applications. The interface between the RoBoard and the Robotis CM-5 uses only using one RS-232 port leaving the remaining hardware for use with future subsystems.

Table 1: Memory Usage on Roboard

Memory Usage	**Fedora 10 Runlevel**			
	1	***3***	***(GUI) 5***	**XP**
MB	18	27	84	94
% of total	7.0	10.5	32.8	36.7

C.Voice Recognition Subsystem

The main task of Voice Recognition Subsystem is to recognize user voice commands and send an appropriate control signal to the CM-5 via a RS-232 serial connection. We initially considered a number of hardware-based solutions such as the Sensory RS-4128 (Sensory Inc.) and open source software-based solutions such as the HTK toolkit (University of Cambridge) and Sphinx [11]. We selected a software based solution to avoid additional hardware, to achieve reduced cost (up to $5000 for the SDK), and to allow for future expansion of the vocabulary of user voice commands. Among the software based solutions, Sphinx-4 was selected as is extensively documented. Sphinx is an open-source project initiated by Carnegie Mellon University and industrial collaborators such as Sun Microsystems and Mitsubishi.

A Sphinx program has 3 components: the Front End, the Decoder, and the Linguist [11]. The Front End contains various digital signal processing functions to convert the input speech into sound units called features. The Linguist converts any type of language model or grammar, along with information from the dictionary and Acoustic Models, into a search graph. The Decoder uses the features from the Front End and the search graph from the Linguist to perform the actual decoding which produce results.The language model or grammar specifies a set of valid words and phrases which should be recognized by the program. The dictionary contains words with its phonemes and the Acoustic Model module provides a mapping between a unit of speech (phoneme) and an HMM which will be evaluated against incoming features provided by the Front End. Sphinx has provided 4 acoustic models: Wall Street Journal (WSJ) for a sampling rate of either 8 kHz or 16 kHz, TIDIGIT and RM1. The dictionary contains more than 100,000 words and has more than one set of phonemes for many words to account for different dialects. There are 2 main grammar formats: Good Turing Discounted Ratio and Java Speech Grammar Format (JSGF).

Following a sample program provided by Sphinx we developed a voice recognition system using WSJ (16 kHz) as our acoustic model and JSGF as our grammar format. JSGF is a very simple grammar format similar to regular expressions used in some programming languages such as Perl, Python, and Java. Good Turing Discounted Ratio, on the other hand, involves complex methods to calculate probabilities for each word. Using the configuration mentioned above, the memory consumed by the program was 120 MB while the available memory on the board was only 256 MB. To reduce the memory, we reduced the number of words in the dictionary from more than 100,000 to just 18 words which are currently required by our program. This reduced the memory footprint to 80 MB, a saving of 40 MB.

We conducted a test to find out the word recognition error rate for our program. We developed a script in Python where voice commands were delivered every 2 seconds from a computer line-out directly to the microphone in jack on the expansion board. Out of the 4000 samples, without any noise, 4 words were recognized incorrectly giving us an accuracy of 99.9% for our program. During this test, the average CPU usage by the program was roughly 35% while the maximum CPU usage was 60%. In the dormant stage, when no voice commands are generated, the average CPU usage is 5% and the maximum CPU usage is 7%. Even though it may seem that the CPU usage is high when the program is active, it is important to note that the program is active only for the duration of the voice recognition and it returns to a dormant state after recognition. Hence, this leaves the majority of CPU available for the OS and other applications.

CONCLUSION

This paper has introduced the Robotic Guide Dog initiative, which aims to develop an electronic travel aid for visually impaired users in the form of an autonomous robotic companion. In the first year of the RGD initiative, the Robotis Bioloid kit was selected as the robotic dog platform, the RoBoard RB-100 microcontroller board was selected as the sensor expansion board, and an open-source HMM-based voice recognition system was implemented and interfaced with the robot through the expansion board. In future years, more subsystems can be added such as obstacle detection, global positioning system (GPS), and 3G internet connection. Ultimately, the RGD system will help visually impaired users navigate complex situations such crossing an intersection by interpreting traffic lights or the ability to find important locations in the city. By adding more systems, the robot will be more environmentally aware of its location and guide the visually-impaired person more effectively than current means.

REFERENCES

[1] Statistics Canada (2006). “Participation and Activity Limitation Survey 2006 Facts on Seeing Limitations,” *Social and Aboriginal Statistics Division*. Ottawa.

[2] New Media Journalism, “Statistics on the Blind”, http://www.fims.uwo.ca/NewMedia2007/blindstats.aspx, accessed September 19, 2009.

[3] The Canadian Press. (2009). “Schools for guide dogs struggling,” *MetroNews*. Vancouver. Retrieved from http://www.metronews.ca/vancouver/work/article/352214.

[4] Canadian Guide Dogs for the Blind, “Frequenty Asked Questions”, September 19, 2009, http://www.guidedogs.ca/faq.htm

[5] CVC 2009 Charity Profile, November 15, 2009, http://www.cvc.vipnet.org/cgi-bin/cvc view.cgi?org_id=1a090632215552c2200

[6] European Space Agency, “More autonomy for blind people thanks to satellite navigation”, *ESA Navigation*, June 4, 2003.

[7] J. Borenstein and I. Ulrich, "The GuideCane-a computerized travel aid for the active guidance of blind pedestrians," *Proceedings of International Conference on Robotics and Automation*, IEEE, 1997, pp. 1283-1288.

[8] (n.d.). “Trekker talking GPS - HumanWare USA.” Retrieved from http://www.humanware.com/en-usa/products/blindness/talking_gps/trekker/_details/id_88/trekker.html.

[9] Robotis Co. Ltd., “Bioloid User’s Guide”, 2007

[10] DMP Electronics Inc. “RoBoard RB-100”, http://www.roboard.com/RB-100.htm, (Accessed Oct 03, 2009).

[11] Walker, W.F., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., Wolf, P. and Woelfel, J. “Sphinx-4: A Flexible Open Source Framework for Speech Recognition.” Sun Microsystems, http://research.sun.com/techrep/2004/smli_tr-2004-139.pdf, 2005. (Accessed 5/3/2010).