



Engaging Students with Visual Impairments in Engineering and Computer Science through Robotic Game Programming (research-to-practice)

Dr. Chung Hyuk Park, Georgia Institute of Technology

Chung Hyuk Park is a postdoctoral research fellow at the Georgia Institute of Technology. He received his B.S. and M.S. degrees in Electrical Engineering and Computer Science from the Seoul National University, Seoul, Korea, in 2000 and 2002 respectively, and his Ph.D. degree in Electrical and Computer Engineering from the Georgia Institute of Technology in 2012. His research interests include machine learning, networked control, computer vision, haptics, multi-modal fusion, human-robot interaction, and assistive robotics.

Dr. Ayanna M Howard, Georgia Institute of Technology

Ayanna Howard is the Motorola Foundation Professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. She received her B.S. from Brown University, her M.S.E.E. from the University of Southern California, and her Ph.D. in Electrical Engineering from the University of Southern California in 1999. Her area of research is centered around the concept of humanized intelligence, the process of embedding human cognitive capability into the control path of autonomous systems. This work, which addresses issues of autonomous control as well as aspects of interaction with humans and the surrounding environment, has resulted in over 130 peer-reviewed publications in a number of projects – from scientific rover navigation in glacier environments to assistive robots for the home. To date, her unique accomplishments have been highlighted through a number of awards and articles, including highlights in USA Today, Upscale, and TIME Magazine, as well as being named a MIT Technology Review top young innovator of 2003, recognized as NSBE Educator of the Year in 2009, and receiving the Georgia-Tech Outstanding Interdisciplinary Activities Award in 2013. From 1993-2005, Dr. Howard was at NASA's Jet Propulsion Laboratory, California Institute of Technology. Following this, she joined Georgia Tech in July 2005 and founded the Human-Automation Systems Lab. She also serves as Chair of the multidisciplinary Robotics Ph.D. program at Georgia Tech.

Engaging Students with Visual Impairments in Engineering and Computer Science through Robotic Game Programming

Abstract

This paper presents an approach for engaging students with visual impairments to learn computer programming through our multi-modal feedback system and carefully devised curriculum using a robotic platform. Our goal of research is to promote precollege students' interests in the fields of engineering such as computer science and robotics. The multi-modal feedback system includes multiple haptic and auditory signals that are designed to transfer distinctly defined messages during the robot's actions. The curriculum consists of a set of task-based tutorials for learning basic programming functions to control the robot, as well as intriguing challenge activities such as games. This system and curriculum have been deployed in five camps for middle and high school students with visual impairments over four different cities over two years. We explain the details of our system and the activities during the camps, and present the results that show positive impacts of our approach in students' interest in computer science and robotics

Introduction

The ratio of entering college freshmen with disabilities has been increasing in the academic environment, including a recent reported growth of 9 percent^{1,2}. Among those, students with visual impairments account for almost 16 percent, but only 3.9 percent of them majored in computer science¹. The contributing factor to this can be found in the disparity in education of precollege math and science classes for students with visual impairments, which provide a foundation for pursuing a degree in computer science. The reason for this disparity is related to the unavailability of information and lack of accessible formats and alternative teaching methods such as the utilization of non-visual curriculum³. Currently, there are only a few efforts focused on encouraging students with visual impairments to pursue higher education and computing career opportunities at the precollege level, which include the National Center for Blind Youth in Science⁴, the Access Computing Alliance⁵, and Project ACE (Accessible Computing Education)⁶.

One of the popular activities that encourage K-12 students to focus their future career goals in the computer science and engineering fields is the utilization of robotic platforms⁷. With the multidisciplinary characteristics of the robotics field and its basis in math and science, the robotics curriculum can provide an intriguing and challenging environment for students. Unfortunately, for students with visual impairments, there is still lack of opportunities in

studying basic computing concepts with robotic-based curriculum. As such, our research is focused on engaging students with visual impairments by incorporating robotic platforms and commercially available non-visual interfaces such as gaming devices to deliver practical knowledge in computing. We follow the basic philosophy from Ludi's work⁸, in which students with visual impairments were provided with a mobile robotic platform and a set of programming instructions to achieve a set of goals to control the robot. To provide more accessibility and perceptual modalities to the students, we added an approach that resembles the methodologies in the field of assistive technology^{9,10} that provides feedback signals to provide environmental perception and notification to the user.

We have developed a mobile robotic system that is linked to a PC and hand-held commercial gaming devices (the Wii remote, namely Wiimote), along with a set of tutorials that teach a student with visual impairments how to program a robot and enable the robot to perform a programmed set of tasks. The programming curriculum is devised to teach both basic and advanced knowledge of programming skills through multi-level tutorials. At every step of the tutorial, our system (PC and the Wiimote) is activated to send haptic and auditory signals to the student so that the student can understand how the robot is moving and how well the student's code is functioning. Furthermore, several robotic games are provided for the students to both motivate and challenge their programming skills after learning basic programming processes. By using robotic games as a means of engagement, our hypothesis is that as long as alternative interface technologies can be employed, a student can become an active participant in robotics-based computing activities, with the goal of encouraging the student to consider future possibilities in computing. In this paper, we introduce the platform and interface modalities of our system, and also present our accumulated results taken over five camps for students with visual impairments hosted in four cities (Atlanta, Baltimore, Cleveland, and Berkeley). Along with the results from the camps, we discuss the basic curriculum for the camp and resulting game challenges that are used to engage the students while proving a means to evaluate their learned robotic programming skills.

Robotic Platform and Accessible Device

In order to provide an intriguing and user-friendly robotic platform, the LEGO MindstormTM was selected for use in the camp (Figure 1). The robots for this session were pre-built for the students to provide identical hardware platform for all participants. The robots were composed of one LEGO Brick computing block, two motors with wheels and built-in encoders for odometry calculation, two touch sensors to detect user input and bumping incident, one light sensor to detect a goal on the floor, and one ultrasound sensor to detect an object in front of the robot.

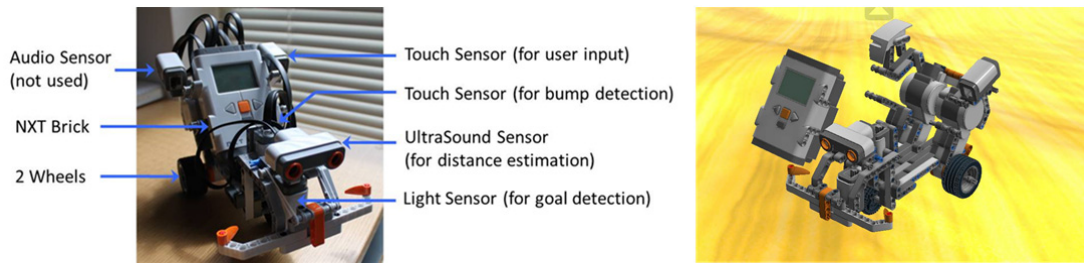


Fig. 1. The mobile robot designed for our camp that is built with the LEGO Mindstorm™ set (left image) and the graphic model that shows the structure (right image).

For transferring non-visual feedback to the student, we adopted the Wii remote controller (called the Wiimote) as the primary interface between the robot and the user. Wiimote is an interactive game controller that has several buttons for input and a motor for creating vibration feedback. In our system, we only use the motor for haptic feedback generation. The communication between the Wiimote and the LEGO NXT robot is governed by a PC via Bluetooth connection^{11,12}, and the PC is in charge of generating auditory feedback for the student. Figure 2 illustrates how the students program the robot and use our system to get the feedback while testing the robot.



Figure 2. A typical desk setup for the students while in programming (left image), and a scene of a testing time when a team of students are testing their robot while holding wiimotes to receive feedback from the robot (right image, Wiimotes are marked with red circles).

Communication between the user, the robot, and a PC (used to decode the communication protocols) is accomplished wirelessly via Bluetooth (BT) connectivity. The NXT robot has a BT module that can transmit and receive signals wirelessly at a sufficiently fast transfer rate. The Wiimote also has a BT module that can be interfaced wirelessly with the host PC. The PC that controls the communication thus opens two BT connections, one with the NXT robot and the other with the Wiimote, handles the signals and manages the multi-modal feedback process.

Using this platform, programming instructions were developed that utilized screen readers to relay visually depicted information on the computer console and a lesson plan that provided

1-on-1 instruction on basic programming syntax, compiling and downloading program to the robot, etc. The teaching protocol used was a modification on the NXT resources¹³.

Programming Instruction Contents

The common programming process for robots involves the following steps: 1) write a program code based on available command set (library); 2) compile the code; 3) download the compiled code onto the robot; 4) run the code on the robot; and 5) evaluate the robot's action and fix/update the initial code. In every stages of this process, visual information transfer between the programming platform (PC), the robot, and the programmer is essential. Certain initiatives in graphical programming take efforts in simplifying the steps described above by using more intuitive graphical symbols to make programming easier¹⁴ as shown in Figure 3. Given that these programming systems involve capitalizing on visually-based constructs, an alternative approach needed to be utilized for teaching programming skills to students with visual impairments.

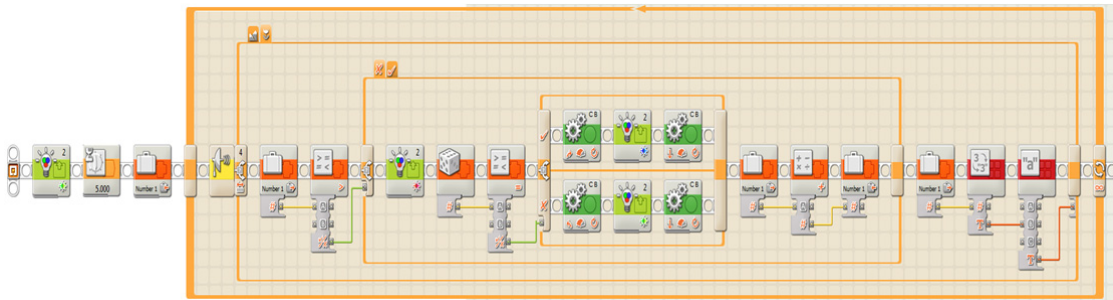


Figure 3. An example of a graphic programming for LEGO MindstromTM.

Our strategy to teach programming to students with visual impairments can be defined by the following goals: 1) provide a simple command-set (library) that can be built upon in stages so that students can progress into learning more complicated coding more easily; 2) provide text-based programming tutorials that can work with screen reader software such as JAWS; and 3) provide multi-modal feedback signals so students can easily test their programmed robot and fix/update their codes.

As a solution for our first goal, we provide five commands that are designed to control the robot as listed below:

List of Commands

- **start_robot()** - Turn the NXT Robot on.
- **move_up(x)** - Move the NXT Robot forward x amount of times.
- **turnleft(x)** - NXT Robot turns left x amount of times.
- **turnright(x)** - NXT Robot turns right x amount of times.
- **stop_robot()** - Turn the robot off.

Each command has a predefined code that controls the robot through more complex steps, and these steps are hidden from the novice student to provide easier access to the robot in learning how to program. Figure 4 shows a typical code using these commands.

```

task main()
{
  start_robot();      //start the robot

  moveup(1);         //move up 1 moves 10 cm
  turnleft(2);       //turn left 2 make the robot turn 90 degrees to the left
  turnright(2);      //turn right 2 make the robot turn 90 degrees to the right

  stop_robot();      //stop the robot
}

```

Figure 4. An example code that controls the robot with simple commands.

As for the second goal, we placed a detailed instruction on every tutorial code in the form of comments (text marked after “//” or in-between “/*” and “*/”), as shown in Figure 5. In this way, the student can simply open a tutorial code template, listen to the instruction with screen software as JAWS, and start programming right away.

```

/*
The following is a very simple program which turns the robot on and then moves up once, and
then turns the robot off
To run this program you should first press the control key and F5 key at the same time and call
the instructor.
To make your program start running you must press the touch sensor on top.
*/
task main()
{
  start_robot();
  moveup(1);
  stop_robot();
}

```

Figure 5. A screenshot of Tutorial_1 code template.

The full list of tutorial tasks and the intended skills for learning are described in Table 1.

Table 1. Programming tasks for the tutorials and the intended goals for skill acquisition.

Programming Task	Intended Skill for Learning
Turn the robot on, move up once, then turn the robot off	Creating your first program

Turn the robot on, move up 80 cm, then turn the robot off	Associating real-world straight-line movement to programming units
Turn the robot on, turn left 45 degrees, then turn the robot off	Associating real-world turn movements to programming units
Move forward 300 cm, turn left 90 degrees, then move forward 100 cm	Trying learned skills together for robot control
Move the robot in the shape of a square twice	Applying learned skills to an application, (e.g. correlating learning program commands to real-world robot movement)

Multimodal Feedback Signals

Finally, to address the third goal, we have designed a multimodal feedback framework that is intended to provide information for environmental feedback that represents the following environmental elements and associated with five feedback primitives:

- 1) Sense of distance traveled by the robot (Primitive: Travel distance feedback)
- 2) Sense of direction the robot is turning (Primitive: Turning left/right feedback)
- 3) Sense of distance to an object located in front of the robot (Primitive: Objective distance feedback)
- 4) Sense of whether the robot has reached a goal or not (Primitive: Bump feedback)
- 5) Sense of whether the robot has bumped into an obstacle (Primitive: Goal feedback)

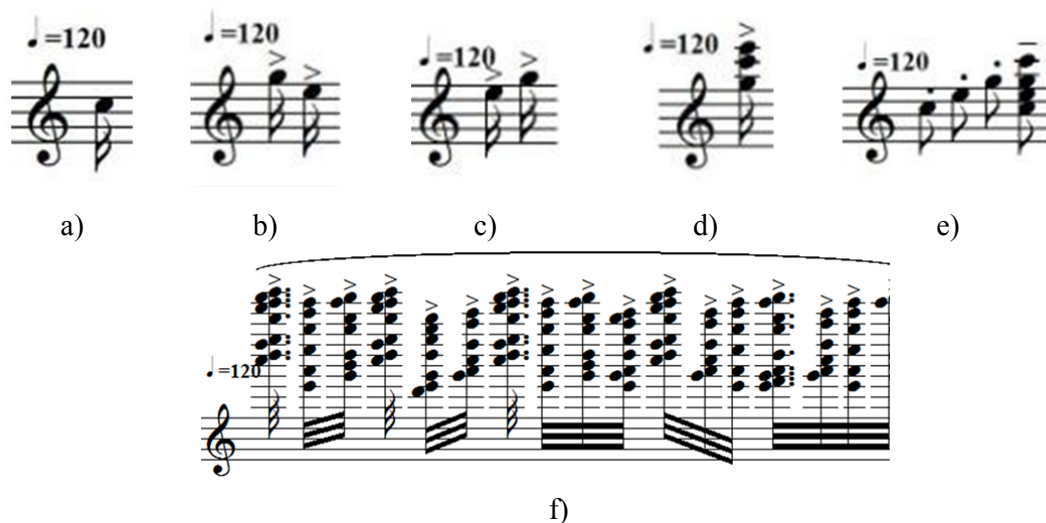


Fig. 6. Notes for audio feedback: a) travel forward, b) turn left, c) turn right, d) object detected, e) goal achieved, and f) bump.

Using these sensory elements, the five feedback primitives were designed to provide in-situ

information of the robot via auditory and haptic sensory feedbacks as presented in Figures 6 and 7.

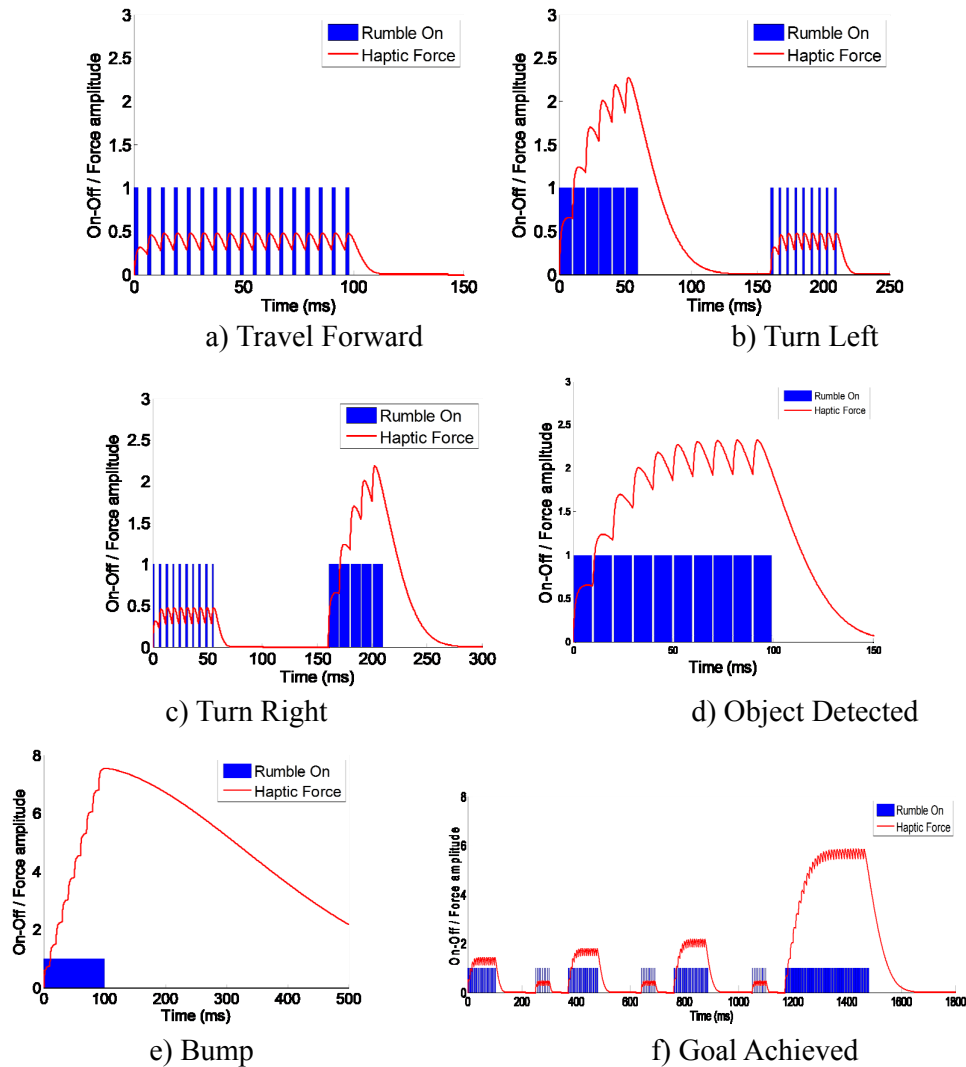


Fig. 7. Haptic feedback force profiles.

Learning How to Program through Extra Activities (Games)

After finishing the tutorials (and becoming familiarized with the feedback signals in the process), the students are provided with challenge activities. Through these activities, the student can exercise the learned programming skills to promote deep learning. Several activities have been designed as described in Table 2.

Table 2. Extra Programming Activities and Games

Activities	Description
Draw a shape	As an extension from the tutorial task, the students are encouraged to program the robot to move along various geometrical shapes such as triangle, pentagon, octagon, etc.
Write a letter	Taking one further step from moving along shapes, some students even challenged to make the robot move along a letter shaped path. We provided pens to attach it behind the robot so it can write a letter, but due to the mechanical design, writing results were not as good as expected.
Corners	This is a popular game we noticed that the students are actually playing in their free time. Adopting the concept from it, the students programmed their robots to move toward certain corners in an arena, and the instructor picks a corner without seeing the arena. The robot located at the corner is removed, and the last robot to stay in the arena receives credit.
Musical chairs	Similar to the musical chair game, except that in this game the robots move around while the music is played and the robot closest to a certain position (specified by the instructor) in the arena is removed at the end of the music.
Robot dance	The students are encouraged to apply all their knowledge in the robot's movement to choreograph dance movements for their robots. The dance can be evolved into a group robot dance.
Kick-the-can	Multiple objects, or "cans," are placed in the arena, and the robot is programmed to move around kicking (hitting) the cans. Two types of objects are used (Jenga blocks and water bottles), and each knock-down of objects add scores (5 points for a Jenga block and 10 points for a bottle) to the robot. So the goal is to sweep as much area as possible.

These activities are selectively provided to students as they finished their tutorials, and the most popular game, Kick-the-Can, formed a regular competition session after the tutorial session. Figure 8 shows the scene of the competition, and Figure 9 depicts two common strategies planned by students.

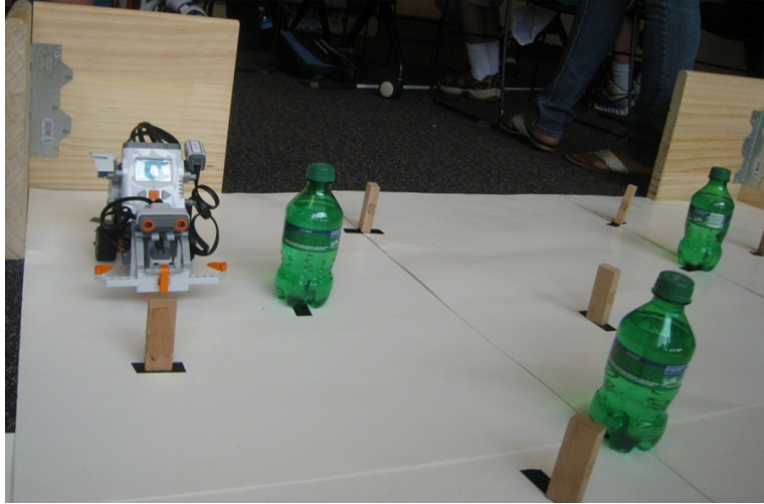


Figure 8. A scene from the Kick-the-Can game.



Figure 9. Two common strategies planned by students.

Measurements and Questionnaire

To gain knowledge of how well the students learned from our camps and how much they were motivated to pursue a professional career in the fields of computing and robotics, we recorded their scores during the competition as well as collected answers from the students on a questionnaire. The scores from the competition (the Kick-the-Can game) were recorded twice from each team, since each team was allowed one test run and two actual trials in the arena. Each run took about a minute, but the time limit was not explicitly set in order not to give extra stress to the students. The questionnaire was composed of pre-session questions and post-session question as below:

Pre-session

Q1. How much experience do you have with programming?

- a. a lot. b. some. c. a little. d. none.

- Q2. How much have you considered working with computers or robotics when you grow up?
a. a lot. b. some. c. a little. d. none.

Post-session

- Q3. How much do you think this workshop helped show you that you are capable of working with computers or robotics?
a. a lot. b. some. c. a little. d. none.
- Q4. How much has this workshop encouraged you to consider working with computers or robotics when you grow up?
a. a lot. b. some. c. a little. d. none

Results

During the five one-day camps held in the four cities (Atlanta, Baltimore, Cleveland, and Berkeley), a total of 32 students (from both middle and high schools) participated. Students were given two hours to finish the tutorials (and do extra activities if they had time left), and given two hours to come up with their own strategies for the Kick-the-Can game and compete with other teams through 1 test run and 2 actual trials for competition. We easily noticed that the students enjoyed the challenge provided by this game. Although they were given only two chances, they showed strong focus and interest in scoring in the competition. Figure 10 explicitly shows how steep the progress curves are between two trials of every team on the Kick-the-Can challenge during one of the camps.

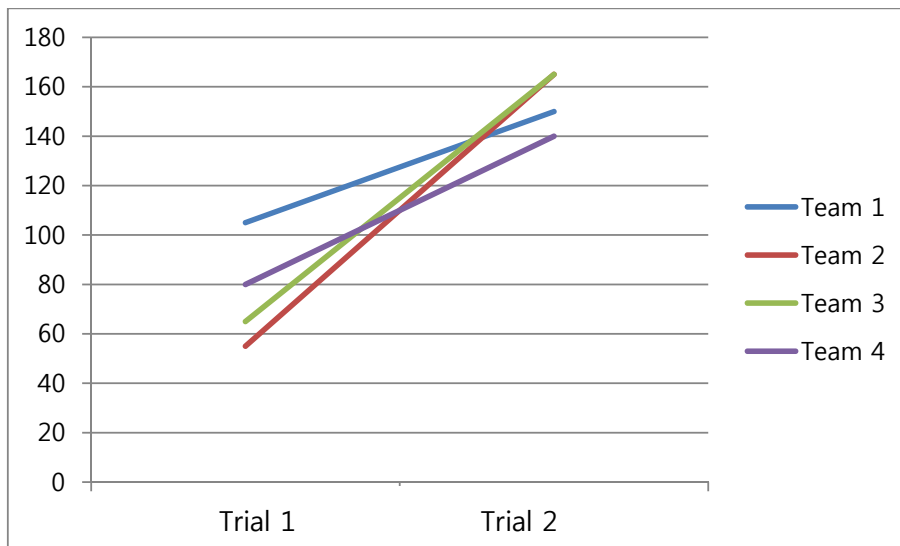


Figure 10. Scores on the Kick-the-Can game in a camp with four teams.

Responses to the first question reveals that 75% of the students had little or no experience with programming before (as shown in Figure 11), but about 56% of the students said they had considered working in the field of computing or robotics in the future (Figure 12).

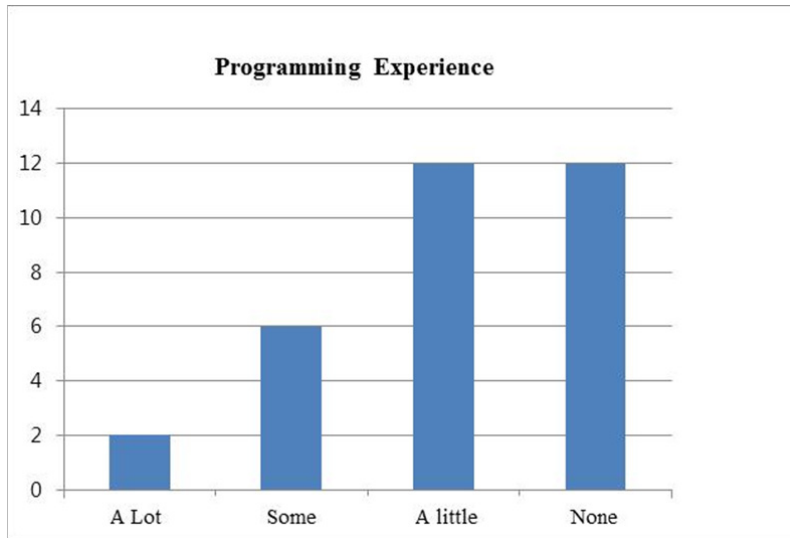


Figure 11. Survey results for Question 1: How much experience do you have with programming?

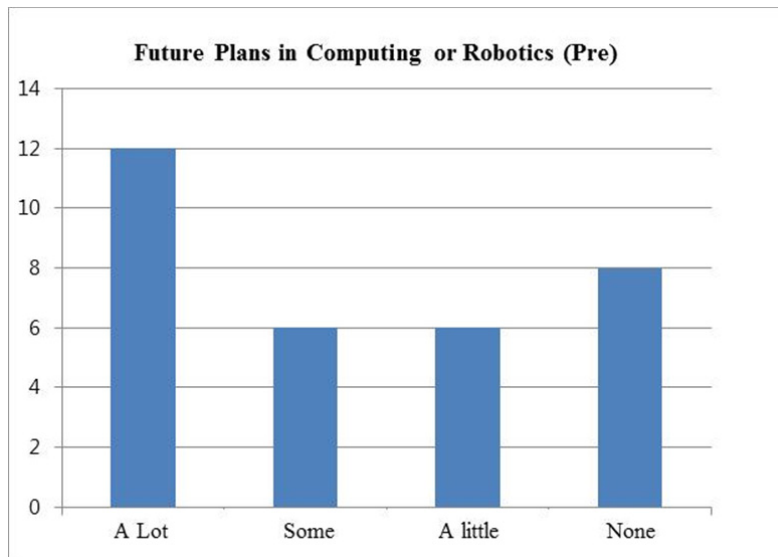


Figure 12. Survey results for Question 2: How much have you considered working with computers or robotics when you grow up?

After the camp (programming tutorials and game activities), the students' responses show that about 97% of the students thought that the camp helped them to understand that they were capable of working with computers or robots (Figure 13), about 69% of the students

said the camp encouraged them (somewhat or a lot) to pursue careers in computing or robotics, and about 28% of the students felt that this camp was at least helpful in encouraging them to pursue a career in computing or robotics (Figure 14).

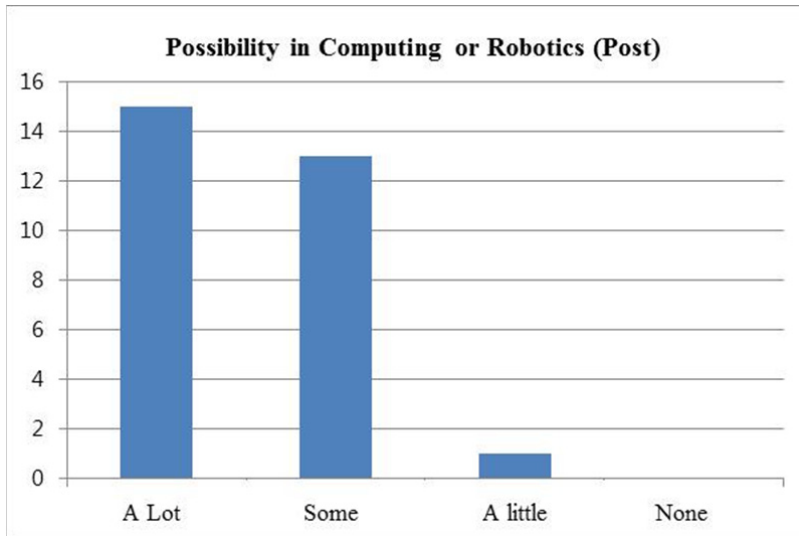


Figure 13. Survey results for Question 3: How much do you think this workshop helped show you that you are capable of working with computers or robotics?

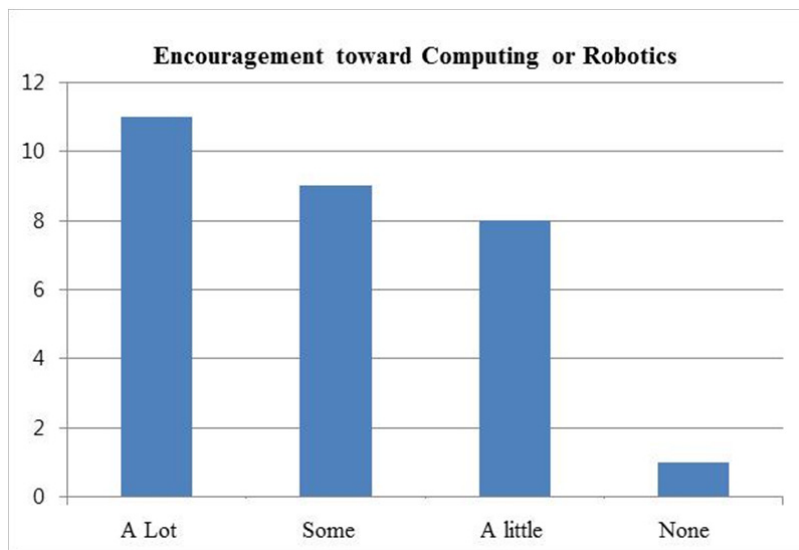


Figure 14. Survey results for Question 4: How much has this workshop encouraged you to consider working with computers or robotics when you grow up?

Conclusion

In this work, we have developed a multi-modal framework for transferring the environmental feedback of a robot to individuals with a visual impairment, and devised a curriculum for teaching computer programming using robotic platforms. We designed the hardware system to incorporate mobility and remoteness, with the haptic and auditory feedback signals tailored to the capability of the system. The signals are not additive, but rather consist of a sequence of real-time signals from which the student can build up knowledge of how well the robot is functioning as programmed. Results show that people with disabilities can perform even highly visual tasks, such as programming a robot, if provided with extra sensory feedback about the problem and the environment, and extra activities such as games can encourage them to work hard to achieve goals and also learn the material with more passion. Future work will focus on expanding the command set to provide feedback associated with other sensory modes, such as robot vision, and other robot tasks, such as manipulation.

Bibliography

- [1] Nat'l Science Foundation, Division of Science Resources Statistics, Women, Minorities, and Persons with Disabilities in Science and Engineering: 2002, Sept. 2003.
- [2] C. Mull, P. Sitlington, and S. Alper, "Postsecondary Education for Students with Learning Disabilities: A Synthesis of the Literature," *Exceptional Children*, vol. 68, no. 1, pp. 97-118, 2001.
- [3] B. Bech-Winchatz and M. Riccobono, "Advancing Participation of Blind Students in Science, Technology, Engineering, and Math," *Advances in Space Research*, vol. 42, no. 11, pp. 1855-1858, 2008.
- [4] Nat'l Federation of the Blind Youth Slam, <http://nfbyouthslam.org>, Feb. 2011.
- [5] The Alliance for Access to Computing Careers, <http://www.washington.edu/accesscomputing>, May 2010.
- [6] Project ACE: Accessible Computing Education for Visually Impaired Students, <http://www.se.rit.edu/~imagine-it/aboutus.html>, May 2010.
- [7] F. Michaud et al., "Assistive Technologies and Child-Robot Interaction," *Proc. AAAI Spring Symp. Multidisciplinary Collaboration for Socially Assistive Robotics*, 2007.
- [8] S. Ludi and T. Reichlmayr, "Developing Inclusive Outreach Activities for Students with Visual Impairments," *Proceedings of SIGCSE Conference*, Portland, OR. February, 2008.
- [9] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran, "RFID in robot-assisted indoor navigation for the visually impaired," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1979-1984, 2004.
- [10] I. Ulrich and J. Borenstein, "The guidecane: applying mobile robot technologies to assist the visually impaired," *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 31, pp. 131-136, 2001.
- [11] "WiiYourself! - native C++ Wiimote library v1.15," from http://wii_yourself.gitter.org/
- [12] "C++ communication library for NXT 2.0," from <http://www.norgesgade14.dk/bluetoothlibrary.php>
- [13] "ImagineIT: Resources," <http://www.se.rit.edu/~imagine-it/resources.html>
- [14] "NXT-G programming environment" developed with NI LabView, <http://www.ni.com/academic/mindstorms/>.