Mobile Robot Platform for Improving Experience of Learning Programming Languages

Jun Su Park and Artem Lenskiy
The Department of Information and Communication Engineering,
The Korea University of Technology and Education, Korea
Email: dufenwnf@naver.com; a.a.lensky@gmail.com

***Abstract*—It is known that learning programming can be cumbersome. The majority of courses on programming languages utilize console-based interface. This is especially true for C language courses. The console-based problems are simply "not-engaging". Moreover the problems itself are usually focused on a particular part of the language and do not have any interaction with the real world. Consequently, quality of education suffers from the lack of motivation. To improve educational environment and engage students into learning process we propose a methodology to mitigate and even prevent such a situation. In our methodology we follow the idea of making classes more interactive by incorporating mobile robots into the education process. However, cost of robots and their complexity often can be a factor that prevents students and/or institutions incorporating robots in education process. In this paper we describe a platform that aims to overcome these issues. Thus, in comparison to other similar projects, the advantages of our system is low implementation costs due to sensors simulation using a single camera complemented with the proposed computer vision algorithm. Moreover, robots are accessed via the network that allows students to program and test their algorithms from home.**

***Index Terms*—learning programming, mobile robots, tele-operation.**

I. Introduction

Industrial robotics, the commercial, personal and service robot became important worldwide[1]. Thus, it is beneficial to incorporate robots into learning process at the time when students just begin learning programming. This trend has been observed for the past few years. Mobile robots are getting to be adopted for various computer science, electrical engineering and mechanical majored classes.

Suvra et. al presented a paper where a curriculum development of a mechatronics class is discussed[2]. They found that involving mobile robots in the education process improves student satisfaction and quality of learning. Their focus was on robot design and the robot programming.

In Carnegie Mellon University, Moblie Robot Programming Laboratory class taught for twelve years. By providing students with hands-on experience with real robots, the class develops students' problem-solving, teamwork, and observation skill. It is noted[3] that flexibility is an important aspect, as it allows students to explore more particular subjects that arouse their curiosity.

Buiu[4] discussed an employment of mobile robots in cognitive robotics laboratory course during fall 2004 and 2005 classes. During each laboratory class he provided 20 students with 10 Khepera robots that were controlled over a network. The client software was developed in C#. At the end of course students reported about higher satisfaction of controlling real robots than simulated robots. Khepera robots come with a functionality of selecting turrets depending on the task. Although, the robot equipped with numerous sensors and relatively powerful processor, it makes the robot price quite high. The latest Khepera model runs as high as 3,660 USD. An advantage of the network approach is that it allowed students to monitor robots' movements with a camera. It is worth noticing that besides observational purposes the camera did not play any other role.

The other popular choice of mobile robots in education is LEGO NXT robots. Numerous educational projects employ these robots to study various subjects. High level of acquired competencies was achieved by adopting LEGO NXT mobile robots in the undergraduate course on mechatronics [5]. The authors used LabVIEW as a programming environment. Nevertheless, Java and C languages are available to program robots' behavior.

Another successful application of the LEGO robots is discussed in [6]. Before programming control algorithms, students had to assemble their own robots. Pinto et. al incorporated LEGO robots in "Mobile Robots" course to teach robot localization using Kalman filter [7].

One of the disadvantages of the LEGO NXT robots is absence of network robot control. The robot's sensors are limited to ultrasonic, light, push and sound sensors. The ultrasound sensor is capable of detecting objects located as far as 233cm, however the objects should be flat and large enough to be detectable.

All the discussed projects agree that involving robots enhances student engagement in academic atmosphere resulting in higher retention and quality of education. In

Manuscript received August 2, 2013; revised November 7, 2013.

doi: 10.12720/joace.2.3.265-269

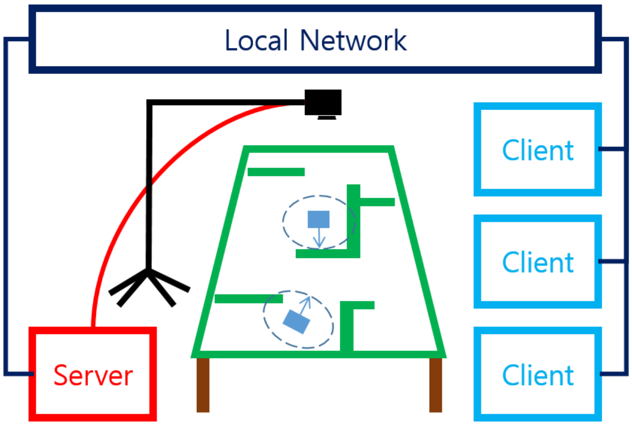
this paper we developed a uniform platform that has several advantages against other similar projects. First of all the robots' cost is very low, as no sensors on board are required. Secondly the developed computer vision algorithm allows tracking robots position that can be used to judge on robots performance. Thirdly all robots are controlled over a network, therefore students are able to control and observe robots from home at any time of the day.

II. Robot Platform Architecture

We selected an inexpensive hamster robot (~180$) that is equipped with 6 IR sensors and works on an all-in-one framework, Roboid Studio[8]. The sensors are designed to detect obstacles in close proximity with the robot. Two sensors positioned at the front and one sensor is at left and right sides of the robot. The remaining two sensors detect black regions below the robot. However, these sensors have two disadvantages that make them undesirable for many robot control tasks. Firstly, the sensors are sensitive to colors. Brighter colors generate a stronger sensor response while the response for darker objects is lower. Therefore, it becomes impossible estimating precisely distance to obstacles. Moreover the position of the sensors often prevents detecting obstacles located at the corners, this leads to situations when the robot is stuck.

To resolve these problems and simulate such sensors as GPS or laser range scanner we apply computer vision algorithms to analyzing the video stream captured from the camera mounted above the field.

The proposed system consists of two parts, a client side and the server side. The above mentioned computer vision algorithms is located on the server part. At the server side the video form the camera is captured and processed. As result the algorithm returns robot's position and orientation. This information is sent to clients that are user's computers. On the client side the received information is employed to take decision on robot's further movements. This control info is sent back to the server, where it is retransmitted to the particular robot (Fig. 1).



Figrure. 1. System overview.

III. The Communication Protocol

The communication between the server and clients is performed according to a designed protocol. The communication protocol consists of a hand-shaking part and the part responsible for information exchanging. During the first phase clients register at the server side either as observers or controller. The observers are capable of receiving the whole map including all obstacles and robots but do not have permission to send control messages. The controllers on the other hand permitted to control the robots, although they only restricted to receiving local robot's surrounding. The messaged are designed using XML.

Fig. 2 show four types of messages circulating between clients and the server.


```
<?xml version="1.0" encoding="EUC-KR"?>
- <message type="clientRegister">
  - <student>
      <ID>2007160062</ID>
      <name>박준수</name>
    </student>
  - <from>
      <ip>111.11.1.1</ip>
      <prot>111</prot>
    </from>
  </message>
```


(a)


```
<?xml version="1.0" encoding="EUC-KR"?>
- <message type="clientAskAuthority">
  - <student>
      <ID>2007160062</ID>
    </student>
  - <robot>
      <robotID>박준수</robotID>
    </robot>
  - <from>
      <ip>111.11.1.1</ip>
      <prot>111</prot>
    </from>
  </message>
```


(b)


```
<?xml version="1.0" encoding="EUC-KR"?>
- <message type="clientControl">
  - <student>
      <ID>2007160062</ID>
    </student>
  - <robot>
      <robotID>robotID</robotID>
      <leftWheel>leftWheel</leftWheel>
      <rightWheel>rightWheel</rightWheel>
    </robot>
  - <from>
      <ip>111.11.1.1</ip>
      <prot>111</prot>
    </from>
  </message>
```


(c)


```
<?xml version="1.0" encoding="EUC-KR"?>
- <message type="serverContorlInformation">
  - <robot>
      <robotID>1</robotID>
      <robotX>20</robotX>
      <robotY>20</robotY>
      <theta>20</theta>
      <leftWheel>20</leftWheel>
      <rightWheel>20</rightWheel>
      <aroundRobot>11111111</aroundRobot>
      <radius>20</radius>
    </robot>
  - <from>
      <ip>ip</ip>
      <port>port</port>
    </from>
  </message>
```


(d)

Figure 2. a) registration message, b) request control message, c) control message, d) robot status message, this message also includes robot's surrounding.

The main advantage of the designed XML-based protocol is the independence of the choice of programming languages. It is up to the client what programming language to select. The only restriction is availability of the network socket library and any XML parsing library. Fortunately socket library comes as a standard library with all well-know programming languages. Regarding the XML parser, there are multiple open source XML parsing libraries that are bound with many languages. Currently we implemented wrappers for java and C languages, however it easy to add other languages if above stated requirements are satisfied.

IV. Computer Vision Module

The computer vision algorithm running on the server side consists of the following algorithms executed one after another a) a field detection algorithm, b) a robot position and orientation detection, and c) a robot's neighborhood extraction and labeling algorithm. The robot's field is detected and cut out of the image. For this purpose we implemented a rectangle detector. The part of the image that corresponds to the detected rectangle is cut out and rotated to align the image axis (Fig. 3).

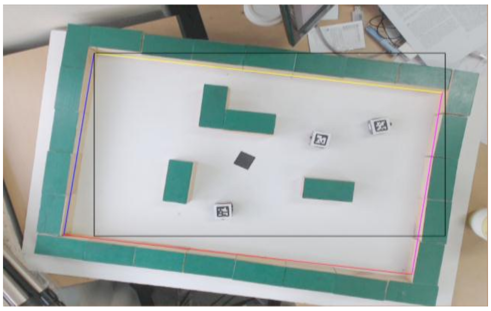


Figure. 3. A camera view of the robot field. The rectangle with colored sides is the detected field.

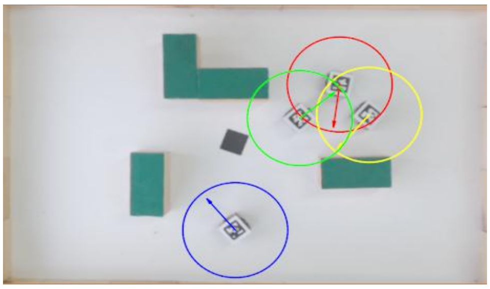


Figure. 4. Detected robots and their orientations.

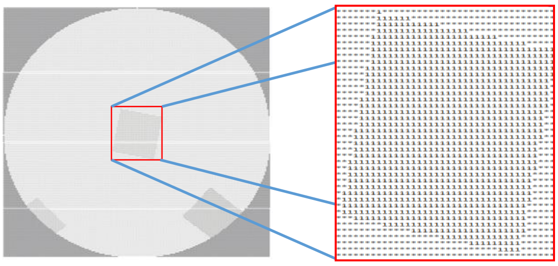


Figure 5. The result of labeling algorithm applied to the robot's surrounding shown by the red circle in Fig. 3.

The next step is a detection of the robot's position and orientation. For this purpose we employed augmented reality toolkit ARToolKitPlus[9]. One of the main tasks of the toolkit is to detect position and orientation of specially designed markers. By attaching a marker on the top of the robot, we are able to extract robot's position and orientation (Fig. 4). After detecting position of the robot we extract and label robots and obstacles that are in close proximity to the robot. Fig. 5 shows the result of labeling. It can be seen that each pixel gets its label that corresponds to the robot's ID.

V. Sensor Simulation

The proposed computer vision approach allows us to simulate various types of sensors including laser range scanners, sonars and IrDA. We found that for programing courses for freshman students the ideal sensors are simple left, right, front-left, front-right sonars. Fig. 6 depicts the idea of such sensor implementation setup. This setup is simple enough to be able to implement obstacle avoidance algorithms. The black square represents the robot. The red half-ellipse represents the region of interest, everything beyond this region will not be processed by the robot. We chose half-ellipse to emphasize the importance of the front part over the sides. Fig. 7 depicts obstacle detection example based on the four sub-regions. If no obstacles are observed within each of the sub-regions it is considered that there are no obstacle in the robots proximity.

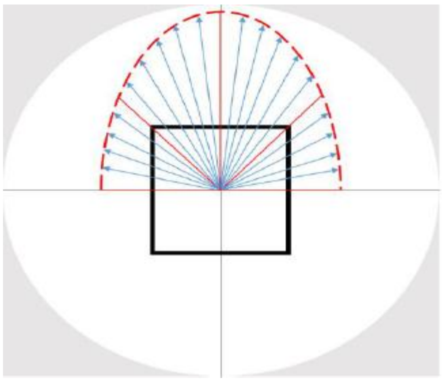


Figure 6. Sensor simulation.

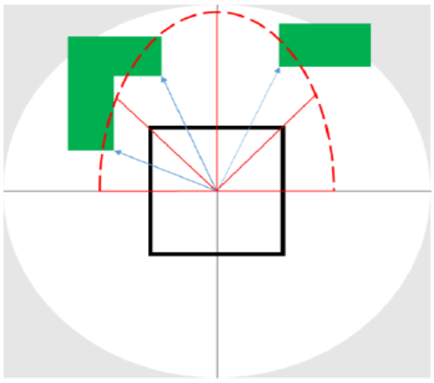


Figure 7. Obstacle detection.

The students receive an array of four values. Each element of the array contains a distance to the closest obstacle in corresponding section.

VI. The Target Search Problem

Based on the described sensor models we presented two implementations of the target search problem as a tutorial example for the students and then students were asked to implement their own solutions.

Our first implementation is based on a random movement similar to the Brownian motion. The robot wanders around until either an obstacle is met or the goal point is reached. In the other algorithm the robot is simply moves straight until it reaches either an obstacle or the goal point. At the time when obstacle appears on the robot's path, a decision is made randomly to change the direction. Fig. 8 and 9 shows the trajectory of the robot's path.

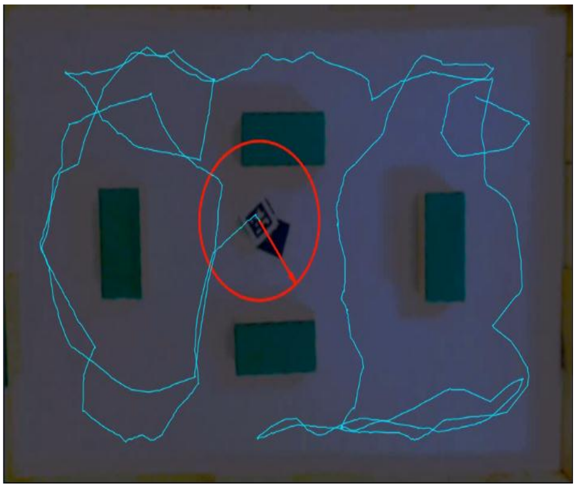


Figure 8. The robot searches for the goal point based on the random movement. The goal point is the black square.

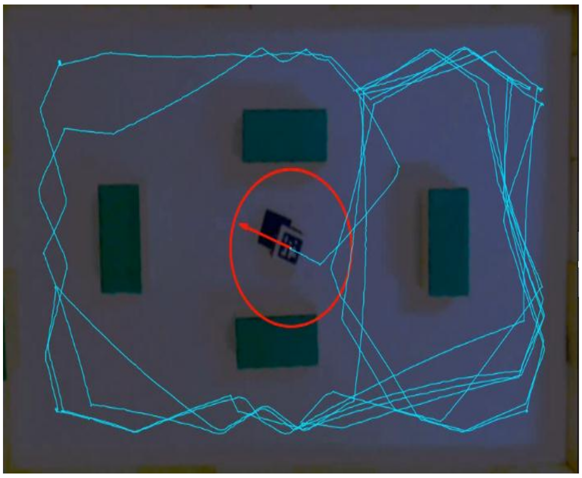
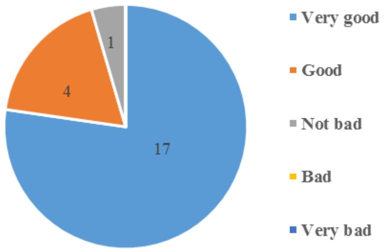


Figure 9. The robot searches for the goal point based on the straight movement. The goal point is the black square.

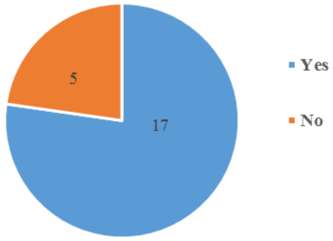
Although we do not show the comparison statistics, following the random movement allows the robot to find the goal point faster.

VII. Survey Result

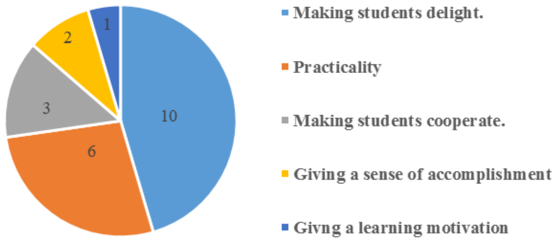
To measure students' satisfaction we took a survey of 22 freshmen after the lab exercise. The majority of students was satisfied with the class and expressed positive. Fig. 10 show the results of the survey.



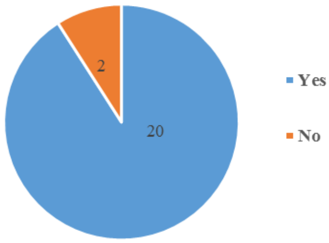
(a) What is your opinion on the lab.



(b) Was it helpful for you to cooperate with teammates in solving robot problem?



(c) What advantages dose this class have?



(d) Would you prefer to have more practical exercises with mobile robots in the next semester?

Figure 10. Survey results.

VIII. Conclusion

The developed platform has a number of advantages against other similar projects. Our system does not require expansive robots with multiple sensors. The camera setup allows simulating various sensors, such as an analog of GPS and laser range finder. Students can access robots form any place with the Internet access. The platform has already motivated students to be passionate about programming.

However, during the two semester programming course we have also found that the hardware part could be and should be improved. Particularly, the hamster robot has low component quality. It is often the case when one wheel rotates faster than the other while the velocities should be equal. This made some of the algorithms hard or impossible to implement. Moreover, among 11 robots, 5 have broken down. Therefore, as for the future work we are assembling a new mobile robot based on arduino board[10] with a bluetooth module. We are also planning to develop interfacing software based on robot operating system [11] that will allow us to replace mobile robots with other types of robots.

REFERENCES

[1] H. Weinert and D. Pensky, "Mobile robotics in education–south african and international competitions," in *Proc. 5th Robotics and Mechatronics Conference of South Africa*, 2012.

[2] D. Shuvra, A. Sandra, and M. K. Yost, "A 10-year mechatronics curriculum development initiative: Relevance, content, and results—part I," *IEEE Transactions on Education*, vol. 53, no. 2, pp. 194-201, May 2010.

[3] J. F. Lalonde, C. P. Bartley, and I. Nourbakhsh, "Mobile robot programming in education," in *Proc. IEEE International Conference on Robotics and Automation Orlando*, Florida, May 2006.

[4] C. Buiu, "Hybrid educational strategy for a laboratory course on cognitive robotics," *IEEE Transactions on Education*, vol. 51, no. 1, pp. 99-108, Feb 2008.

[5] J. M. G. D. Gabriel, A. Mandow, J. Fernández-Lozano, and A. García-Cerezo, "Using LEGO NXT mobile robots with LabVIEW for undergraduate courses on mechatronics," *IEEE Transactions on Education*, vol. 54, no. 1, pp. 41-47, Feb 2011.

[6] H. K. Seung and J. W. Jeon, "Introduction for freshmen to embedded systems using LEGO mindstorms," *IEEE Transactions on Education*, vol. 52, no. 1, pp. 99-108, Feb 2009.

[7] M. Pinto, A. P. Moreira, and A. Matos "Localization of mobile robots using an extended kalman filter in a LEGO NXT," *IEEE Transactions on Education*, vol. 55, no. 1, pp. 135-144, Feb 2012.

[8] K. J. Kim, I. H. Suh, and K. H. Park, "Roboid studio: A design framework for thin-client network robots," *IEEE International Conference on Advanced Robotics and its Social Impacts*, Taipei, Taiwan, Aug 23-25, 2008.

[9] D. Wagner and D. Schmalstieg, "ARToolKitPlus for pose tracking on mobile devices," in *Proc. 12th Computer Vision Winter Workshop*, Sankt Lambrecht, Austria, Feb 2007.

[10] J. Sarik, "Lab kits using the arduino prototyping platform, ioannis kymissis," in *Proc. 40th ASEE/IEEE Frontiers in Education Conference*, Washington, DC, October 27-30, 2010, pp. T3C-1—T3C-5.

[11] J. Kerr and K. Nickels, "Robot operating systems: Bridging the gap between human and robot," in *Prco. 44th IEEE Southeastern Symposium on System Theory University of North Florida*, Jacksonville, March 11-13, 2012.

Jun Su Park received BSc in wireless communication from Korea University of technology, Korea. He is currently working towards his master's degree at the school of electrical, electronics & communication engineering.
His research interests are computer vision, mobile robots and teaching methods in computer vision.

Artem Lenskiy received BSc and MSc degrees in computer science form Novosibirsk State Technical University, Russian in 2002 and 2004, respectively. After teaching at the same university for and year he joined doctor course at University of Ulsan, Korea. He was awarded the Ph.D. in 2010 from the same university. He joined Korea University of Technology and Education as an assistant professor in 2011.
His research interests include machine learning and self-similar processing applied to various research and engineering fields.