Toward Smart Robot – A Survey of Smart Device based Sensing and Control Methods for Robotic Applications

*1 Yong-Ho Seo

1 *Department of Intelligent Robot Engineering, Mokwon University, Mokwon Gil 21, Seo-gu, Daejeon, Republic of Korea, yhseo@mokwon.ac.kr*

Abstract

Smart devices are becoming an affordable main computing resource for robotic applications in accordance to the fast growth of mobile internet environment. Methodologies for acquisition of remote sensory information and control of various types of robots using smart devices have been proposed recently. This paper reviews representative methods of remote sensing and control for robotic applications based on smart devices. These methods mainly deal with acquiring and processing sensory data from a robot and sending appropriate actions to the robot through wireless communication such as Bluetooth or WIFI with smart device. Experiments and performance evaluations of the sensing and control schemes show the feasibility and the effectiveness of smart device based robotic application toward smart robots.

Keywords*: Smart device, Remote sensing, Robot control, Robotic application, Bluetooth communication*

1. Introduction

Use of smart devices have become~~s~~ increasingly popular and the market size of smart devices is currently expanding as big as the Laptop and desktop computer. Main benefits of smart devices are mobility and electric power usage of battery. With the rapid increase of users of smart device, the computing power has also been remarkably enhanced.

Smart devices also have auxiliary hardware such as the digital camera, gravity sensor, Bluetooth, WIFI and touch screen. In consideration of this recent trend in mobile computing technologies and with the recent rapid diffusion of smart devices, interconnection of smart devices and robots via wireless communication will attract public attention and will have a major influence on robotic applications.

A mobile robot is typically equipped with various sensors to recognize the surrounding environment, and its main task is generally to travel to a given destination while avoiding obstacles. Smart devices can conduct autonomous tasks for robotic applications based on sensory information gathered from its own sensors ~~or~~ such as the gravity sensor or GPS or remote sensors attached to the robot. Therefore many researches on applying smart devices instead of the laptop computer on robotic application have been conducted [1].

Representative studies focus on data acquisition and navigation control of a mobile robot using smart device that can be used anytime anywhere, unlike the conventional approach of employing additional high-performance computing equipment to control a mobile robot [2]. Those studies realized a remote application by using a smart device running on smart device OS platforms such as Google Android, Microsoft Windows Mobile and Apples iOS In addition, Bluetooth communication and WIFI were mostly employed as the remote data transmission method between the mobile robot and the smart device for robotic application [3].

Received: Sep. 24, 2013, Revised: Oct. 10, 2013, Accepted: Dec. 18, 2013

For the smart device application, touch-based GUI was designed for visualization and remote control of acquired sensor information. Some approaches were carried out using a real smart phone for the performance verification of the developed robot system with smart phone application through the use of JAVA for an Android Dalvik virtual machine in an Eclipse environment and .NET Compact Framework with the Windows Mobile in a Visual Studio environment [4- 6].

2. Software Developments for Robotic Applications on Smart Device

There are two dominant operating systems for smart devices nowadays [7], the Microsoft Windows and the Google Android. The .NET Compact Framework is a version of the .NET Framework that is designed to run on resource constrained smart devices such as personal digital assistants (PDAs), mobile phones, and factory controllers. The .NET Compact Framework inherits the full .NET Framework architecture of the common language runtime for running managed code. It provides interoperability with the Windows CE operating system of devices so that the user can access native functions and integrate his or her favorite native components into applications [8].

The UI development is based on Windows Forms, which is also available on the desktop version of the .NET Framework. User interfaces can easily be created with Visual Studio by placing .NET Compact Framework controls such as buttons, text boxes, etc. on the forms. Features such as data binding are also available for the .NET CF [9].

Android is a software stack for mobile devices that includes an operating system, middleware, and key applications developed by Google. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using Java programming language [10].

Dalvik is the process virtual machine (VM) in Google's Android operating system. Dalvik is thus an integral part of Android, which is typically used on mobile devices such as mobile phones, tablet computers, and netbooks. Before execution, Android applications are converted into the compact Dalvik Executable format, which is designed to be suitable for systems that are constrained in terms of memory and processor speed. Figure 1 shows the .Net Compact Framework (left) and Dalvik VM (right) as representative latest smart device development environments.

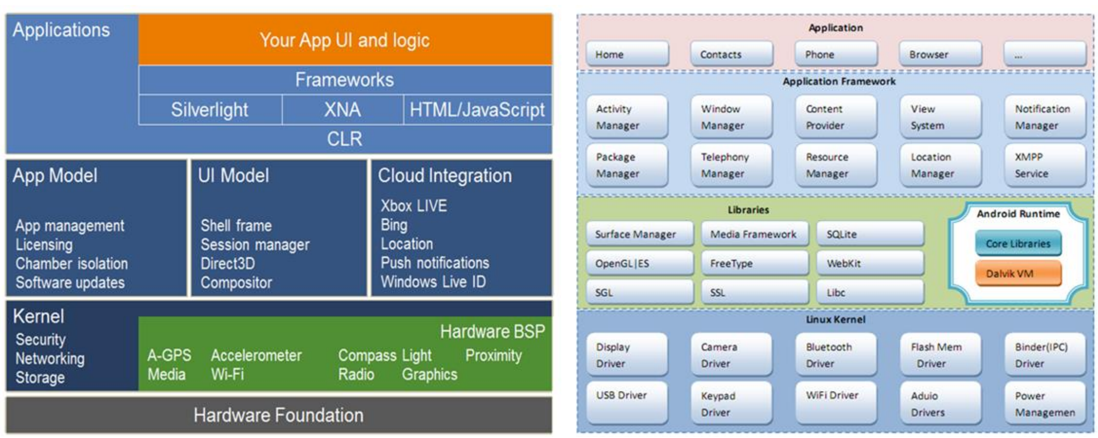

Figure 1. .Net Compact Framework (left) and Dalvik VM (right) based Smart Device Development Environment

2.1. Software Configurations for Smart Device based Mobile Robots

Software configurations of the two mobile robot systems are shown in Figure 2 below. The left software configuration is for the conventional mobile robot with two differential wheels and the right configuration is for the omni-directional mobile robot. Two systems are configured with C compiler based firmware, which transmits the robot's sensor measurements to the smart device and receives motor commands from the smart device. The software in the smart device is configured with .Net Compact Framework, which drives the robot based on the received sensor information from the robot and a C# based mobile application.

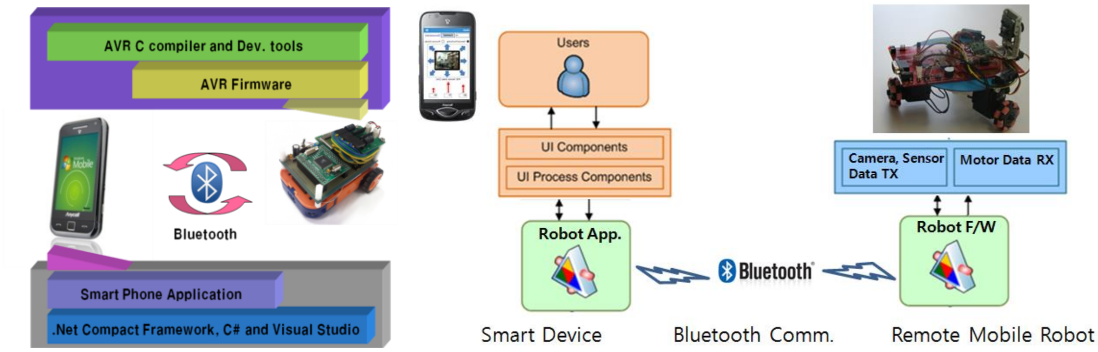

Figure 2. Software Configurations of Two Smart Device based Mobile Robots

2.2. Software Configurations for Smart Device based Mobile Robots

Sensor Data Visualization and LED dot-matrix Interfaces of a robot has been proposed in [6]. Among the features of the mobile robot, there is an 8x8 LED dot-matrix display. For use of this display, a touch event is received from the screen of the smart phone. Coordinate information corresponding to the touched LED is turned on both in the smart phone screen and the real dot-matrix on the mobile robot.

In the 64 LEDs comprising the dot-matrix, an 8 bit control address is assigned and images are arranged in accordance with the LED address on the screen of the smart phone. Figure 3 shows the LED control layout for the 8x8 dot-matrix location output on the screen of the smart phone and the bit address according to the LED coordinates sent to the mobile robot through Bluetooth communication.

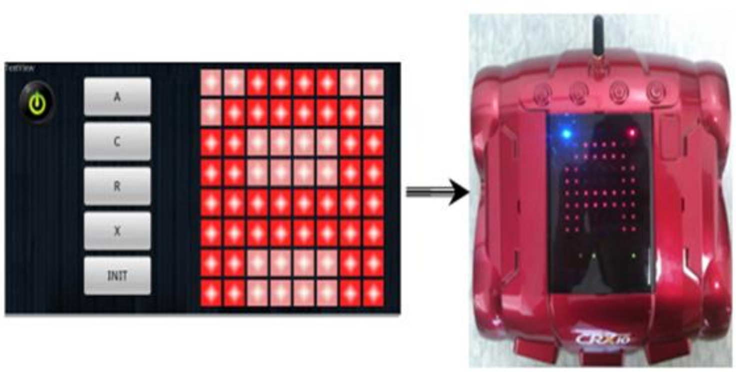

Figure 3. User Interface for 8x8 dot-matrix location indication

The basic layout of the user interface for manual control of a mobile robot is shown in Figure 4. The arrow in the figure indicates the proceeding direction of the mobile robot, which depends on the location of the motor. The user interface is composed of a track ball and direction buttons for remote mobile robot control. In addition, buttons are arranged in line with each direction and the speed value for the motor is sent to the mobile robot only when the relevant button is pushed.

Figure 4. Track ball and Direction buttons for Remote Control of Mobile Robot

The user interface of the smart device application for an omni-directional mobile robot control has been proposed in [5]. This user interface was developed using a smart application interface based on C# and .Net Compact Framework 3.5. Various buttons are placed to manually control the robot and to make it easy to connect to Bluetooth, select which mobile mode to run, and verify current PSD sensor measurements. The developed robot software is available on various smart devices and Windows CE based embedded devices. The following Figure 5 shows its user interface.

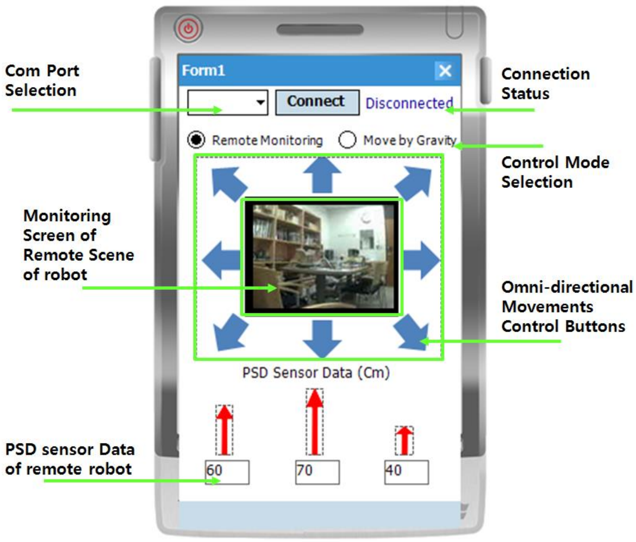

Figure 5. User Interface for Omni-directional Mobile Robot

These user interfaces support a user interface that has various control mode options so that the user can select a specific mode among the control mode and the gravity steering mode. When the mobile robot moves according to the mobile mode when selected via the smart device, it collects information of the surrounding environment through IR sensors and transmits the information to the smart device via Bluetooth. The smart device then analyzes the data from the robot and generates new motor control commands and transmits them to the mobile robot.

3. Sensing and Control Methods using Smart Devices

The types of control methods using smart devices are composed of an autonomous task method and a manual control method using touch based interface or the gravity sensor in smart devices. When a button of any mode is touched, the composed layout screen appears on the smart device, and proper actions can be performed by the mobile robot via connection through Bluetooth.

3.1. Gravity Sensor Steering

In the gravity steering control method proposed in [4], the user can control the mobile robot as if a user drives a car with the steering wheel of car. In this mode, the smart device simulates the physical car handle by estimating current pose using the 3-axis gravity sensor. The user also can feel the speed of the mobile robot through vibration of smart phones driven by the embedded vibration motor. Figure 6 shows the axis of the gravity sensor and its data plot when the rotation movements according to the axis occur [11].

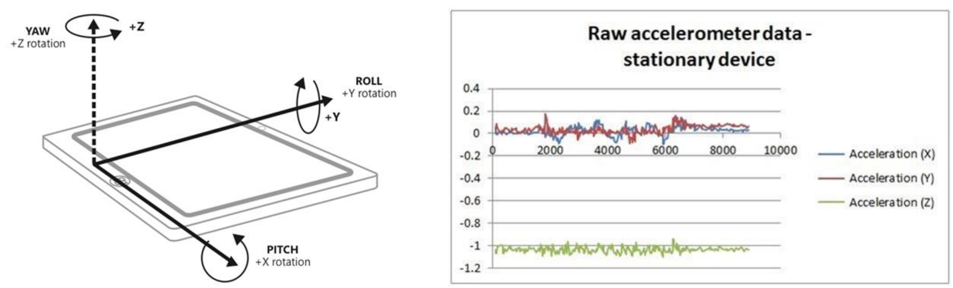

Figure 6. Axis of Gravity Sensor and its Data Plot

3.2. Touch-based Odometric Information Generation and Mobile Robot Navigation

Touch events of smart devices can be defined as shown in Table 1. Touch-based odometric information generation for a mobile robot control has been proposed in [6]. The generation of odometric information from the input order of a curve inputted by the user's touch is shown in Figure 7. From the point when the user begins to touch the screen with a finger, to the point when the finger is removed from the screen, the points are saved in an array. Then, from the touch curve drawn by the user, five major points are chosen and marked [12]. The straight line segments are then extracted from the saved points.

Then, a mobile robot is moved to the location designated by the coordinates and a message indicating that the mobile robot arrived at the point is sent from the smart device. The smart device sends proper wheel commands to the robot based on the value of the coordinates and the angle of the next point through via Bluetooth communication. Figure 8 shows a navigation experiment which shows that the robot follows the given odometric information using smart devices.

Table 1. Types of Touch Events

Action Item	Detailed Explanation
Action_DOWN	Push the screen by a finger
Action_UP	Move a finger away from a screen
Action_MOVE	Move the finger on the screen while a finger contacting a screen

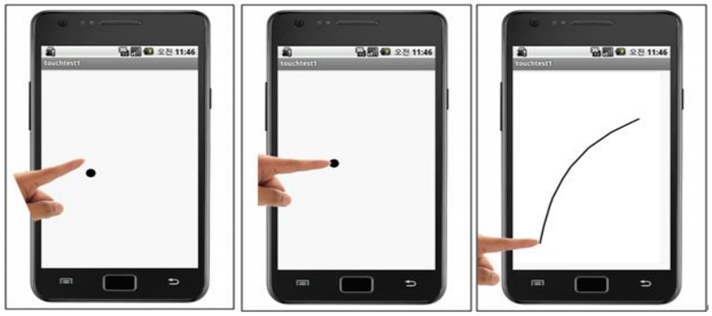

Figure 7. Generation of Odometric Information from a user's touch event

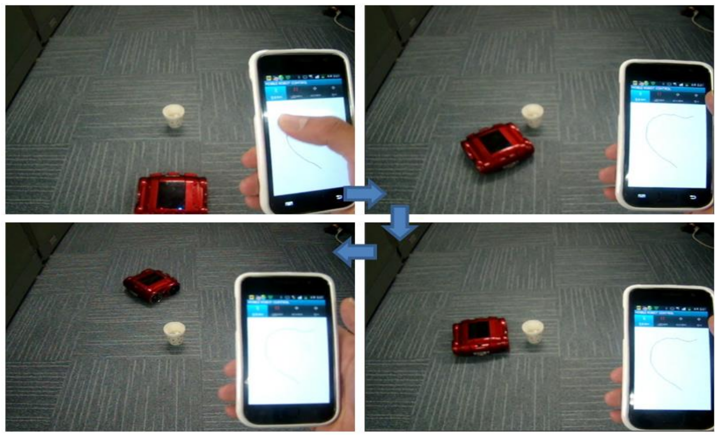

Figure 8. Touch-based odometric information generation and mobile robot navigation experiment

3.3. Remote Line Tracing Task and Obstacle Avoidance

When a mobile robot has its own sensors, smart devices can gather proper sensory information to conduct an autonomous task remotely. Remote line tracing task has been proposed in [5]. In their line tracing task, line tracer algorithm is carried out on the smart device using received floor-facing 3 IR sensors measurements from the mobile robot. Navigation functions for the line tracing mode are forward, turn left, turn right, acceleration, immediate left and right turn and setting initial direction for line tracing. Figure 9 demonstrates a remote line tracing task using smart device, which successfully tracked the given circled line.

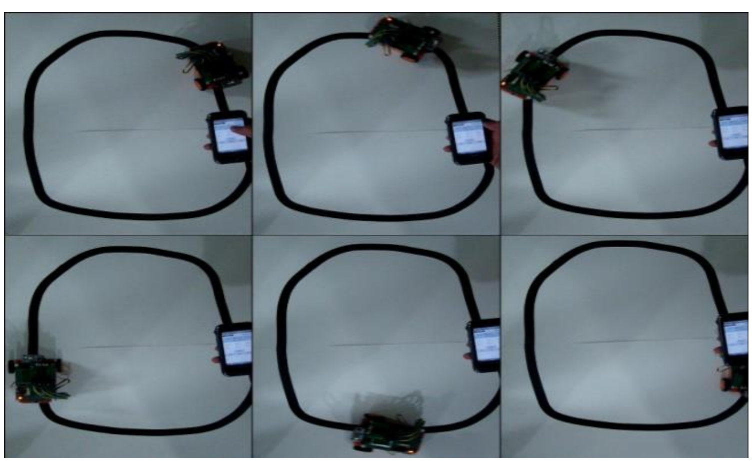

Figure 9. Remote Line Tracing Task using Smart Device

When PSD sensors are placed on the left, right and center in front of a mobile robot, the distance values between the mobile robot and an obstacle can be acquired remotely at the smart device. The information of surrounding obstacles can be used to conduct an avoidance task which the robot navigates autonomously. The distance values from sensors are displayed on the screen of the smart device and an obstacle avoidance experiment can be conducted, as shown in Figure 10.

Figure 10. Remote sensor data acquisition and obstacle avoidance experiment

3.4. Remote Line Tracing Task and Obstacle Avoidance

Remote Control of the Omni-directional Mobile Robot and its Monitoring In remote control of the omni-directional mobile robot and its monitoring task, the user can control the mobile robot while monitoring the remote scene captured from the camera on the robot. The user can also see the fore-facing 3 PSD sensor measurements from the robot. Control functions in the remote monitoring mode are 8 desired directions for the robot to move without rotating it using the three omni-directional wheels mechanism and additional left and right turning. Figure 11 shows images from the experiment of the remote monitoring mode, where the robot was controlled successfully according to the user's commands.

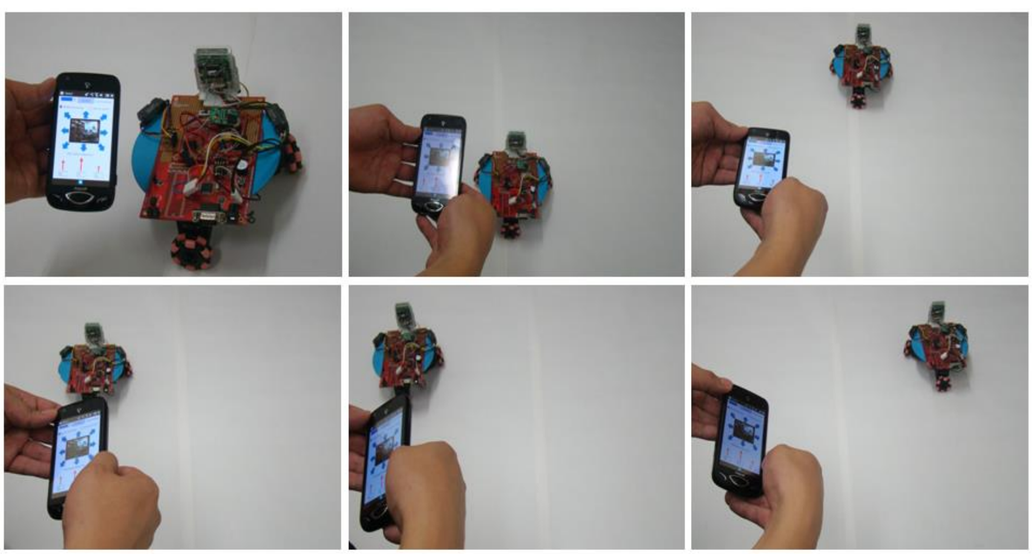

Figure 11. Experiments of Remote control by Gravity sensor and Monitoring of Omni-directional Mobile Robot

4. Performance Evaluations

Performance evaluations of the reviewed sensing and control schemes are needed to show the feasibility and the effectiveness of smart device based robotic applications. In section 3, the experimental tasks have in real-time successfully demonstrated, using the show, the feasibility of smart device based sensing and control scheme of a robot through wireless communication such as Bluetooth or WIFI.

4.1. Performance Evaluation of Remote Control

In [4], the authors conducted a performance evaluation of a remote control with smart devices by measuring the wireless data transmission latency. Unlike traditional PC based robot control, to evaluate performance of wireless smart phone based robot control system, average wireless data transmission latency needs to be measured at each communication speed.

In their remote line tracing experiment, the average wireless data transmission latency was measured after 10 byte data packets were sent and received 20 times in a loopback manner. Table 2 shows the average data transmission latency according to different Bluetooth communication speeds to evaluate the wireless data transmission latency of the wireless smart device based robot control system. Figure 12 also shows the latency plot according to two different communication speeds and three different distances in Bluetooth connection.

Table 2. Data Latency according to Bluetooth Communication Speeds

Comm. Speed of Bluetooth (bps)	Average data transmission latency (mSec)		
	Near	5M	10M
19,200	63.2	67.9	70.9
57,600	33.6	36.8	58.9
115,200	22.3	25.5	33.9

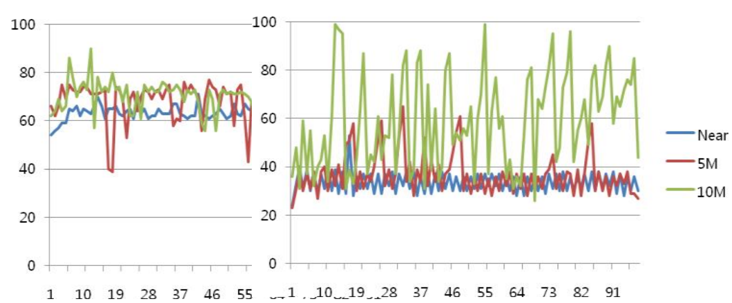

Figure 12. Data Latency according to Bluetooth Comm. Speeds and Distances

4.2. Performance Evaluation of Remote Monitoring

In [5], the author made a mobile robot which has a Jpeg serial port CMOS camera module. It can capture pictures of various resolutions such as VGA, QVGA and 160*120, and send JPEG images to smart devices through Bluetooth communication. The author conducted a performance evaluation of remote control and monitoring with a smart device by measuring the wireless data transmission latency and the frame rate of the captured images at 160*120 resolutions.

Table 3. Data Latency and Frame Rate according to Bluetooth Communication Speeds

Comm. Speed of Bluetooth (bps)	Frame Rate (Frames/Sec)
57,600	2~3
115,200	4~5

5. Conclusion

This paper reviewed methods to acquire sensory information from sensors on a smart device or on a remote robot and control the robot using smart devices remotely. The reviewed papers present methods of the remote control and monitoring of various types of mobile robots using Bluetooth or WIFI with a smart device. They also show the experiments of various types of remote controlled tasks and autonomous tasks of robots using smart device and perform evaluations to verify applicability of the developed control and monitoring system using smart devices.

In their experiment, the control algorithms for the remote mobile robot were implemented at the smart device side and the smart device controlled the mobile robot by sending motor control signals through wireless connection. To validate the wireless control system, data transmission latency was measured at different communication speeds and different distances between the robot and the smart device.

In addition, one study suggested technology for generation of odometric information of a mobile robot and following technology based on the use of a touch event of smart device. Simply drawing a curve with a finger provides an easier way to control a robot than clicking a button or using additional sensors.

The reviewed smart device based remote robot control schemes are considered to give meaningful contributions to the field in that it is new types of robot applications, successfully integrating and exploiting a traditional PC based mobile control method and the distinct features of a smart device, mobility in particular. Localization of a mobile robot using a GPS and acceleration sensor information and HRI (Human Robot Interaction) using a camera, 3D depth sensor, audio of smart device are expected to be the focus of future research. When considering the rapid advancement of smart devices with various embedded sensors, it is expected that smart devices will soon be able to replace desktop PCs when controlling robots and the service variability of intelligent service robots will be widened through their connection with smart devices and their use of various features and contents available on smart devices.

6. Acknowledgments

This work was also supported by the NAP (National Agenda Project) of the Korea Research Council of Fundamental Science & Technology.

7. References

[1] A. C. Santos, L. Tarrataca, J. M. P. Cardoso, "The Feasibility of Navigation Algorithms on Smartphones using J2ME," *Mobile Networks and Applications*, Vol. 15, No. 6, pp. 819–830, 2010.

[2] H. H. O. Nasereddin, A. A. Abdelkarim, "Smartphone Control Robots through Bluetooth," *International Journal of Research and Reviews in Applied Sciences*, Vol. 4, Iss. 4, 2010.

[3] CNRobot Inc [Online]. Available: http://www.cnrobot.co.kr

[4] Y. H. Seo, S. S. Kwak, and T. K. Yang, "Mobile Robot Control Using Smart Phone and Its Performance Evaluation," in Proc. of International Conference on Advanced Communication and Networking (ACN 2011), Brno, Czech Republic, Aug. 2011, pp 362–369.

[5] Y. H. Seo, "Remote Control and Monitoring of an Omni-directional Mobile Robot with a Smart Device," in Proc. of International Conference on Convergence and Hybrid Information Technology (ICHIT 2011), Daejeon, Korea, Sept. 2011, pp. 286–294.

[6] Y. H. Seo, H. Y. Jung, C. S. Lee, and T. K. Yang, "Remote Data Acquisition and Touch-Based Control of a Mobile Robot Using a Smart Phone," in Proc. of International Conference on Future Generation Communication and Networking (FGCN 2011), Jeju, Korea, Dec. 2011, pp. 219–226.

[7] F. Lin and W. Ye, "Operating System Battle in the Ecosystem of Smartphone Industry," in Proc. of the International Symposium on Information Engineering and Electronic Commerce, Washington D.C., May 2009, pp. 617–621.

[8] Microsoft .NET Framework Developer Center [Online]. Available: http://msdn.microsoft.com/netframework

[9] .NET Compact Framework [Online]. Available: http://msdn.microsoft.com/en-us/library/9s7k7ce5(v=vs.80).aspx

[10] Android developers [Online]. Available: http://developer.android.com/guide/basics/what-is-android.html

[11] Supporting sensors in Windows 8, Developer support for sensors [Online]. Available: http://blogs.msdn.com/b/b8/archive/2012/01/24/supporting-sensors-in-windows-8.aspx

[12] S. Singh and D. H. Shin, "Position Based Path Tracking for Wheeled Mobile Robots," in Proc. of IEEE Intelligent Robots and Systems, Tsukuba, Japan, Sept. 1989, pp. 386–391.