Smart Home Appliance Control System for Physically Disabled People Using Kinect and X10

Md. Ahsan Iqbal, Sk Asrafuzzaman, Md. Mahfuz Arifin, SK Alamgir Hossain
Computer Science and Engineering Discipline
Khulna University
Khulna, Bangladesh
md.ahsan.iqbal@gmail.com, asraf_asif@yahoo.com, arifinshishir@gmail.com, alamgir@cseku.ac.bd

***Abstract*—Modern science and technology is all about making our lives more comfortable. As the study of various technologies is progressing, the definition of comfort in our lifestyle is evolving. Tasks that involved so much complexities and hard work a few years ago, can now be done by just pushing a button. But by being possessed with thoughts of our own comfort, we often forget the people who need it the most; the physically challenged people. They face so many limitations in everyday life that what may seem luxury to us, is ironically necessity to them. To provide a satisfactory resolution to this, we have built "Smart Home Appliance Control System for Physically Disabled People Using Kinect & X10". Our system's goal is to make their lives easy and comfortable by providing them with a self-dependable environment. To validate the system's acceptability we developed a prototype in our laboratory environment and performed real user study.**

Keywords—smarthome; physical disability; kinect; x10; gestures; voice commands.

I. INTRODUCTION

Science & Technology has become an essential part of our lives. Computer Science is now reaching every field we can imagine. It is now an integral part of human life serving in various ways. It provides us with better alternatives and has simple solutions to complex problems. Now-a-days Human Computer Interaction (HCI) is one of the major concerns in every aspects of our technology. As technologies are developing, the ways of interaction between Human and Computer are also evolving. Technology has taken us to the stars, made great depths of sea accessible and filled our lives with uncountable wonderful devices to provide us with smart and new ways of doing things. Yet, for some people even a normal daily life is still a luxury. People with physical disabilities still have to rely on other people's help to even do the simple things like turning a light on. Existing smart home frameworks are usually built for normal people in mind and are quite expensive.

Disability is a global issue. According to World Health Organization and the World Bank, 1.1 billion people live with some kind of disability [1]. After adding family and friends who provide the people with disabilities with daily support, it directly affects 1.9 Billion people. It is impossible to exclude such a significant proportion of the world population from development efforts. Societies cannot strive if such a large propotion of people are ignored because of their disabilities. The total number of adult people with significant difficulties in functioning is between 110 million and 190 million [2]. Among them 30.6 million suffer from difficulty of walking and climbing stairs. About 12.0 million people suffer from requirement of assistance by other people. And about 8.1 million people suffer from total or partial vision difficulty.

A Ullah et al. [3] developed "Augmented reality based marker tracking for Smart Home control" to address smart home for disabled people issue. This system proposes installation of different QR codes for different appliances in home and then the user can scan them with a phone to control the appliances or to avail them for further commands. The drawback is standard sized QR codes can only be properly scanned from a little distance. Users need to stand very close to them for interacting with the system.

RCAST, University of Tokyo in collaboration with Microsoft Inc. also developed "OAK, A Kinect based support system for severely disabled" [4]. They proposed a virtual air switch for controlling appliances. The Kinect [5] keeps on scanning the room for tracking color intensity changes in the area of virtual air switch to toggle the status of the appliance. Despite great usability, this system suffers from a major drawback; the inability to work in darkness. This is also applicable for the QR code based system. Only a normally lit environment is suitable for these systems.

A Hossain and others developed "Context-aware elderly entertainment support system in assisted living environment" [6] to address a similar needs for elderly people. Other similar system include "Ant-based service selection framework for a smart home monitoring environment.", [7] which was developed by M. S Hossain and his researchers.

Our main objective is to take these systems further by addressing and solving their limitations and to add more features to what they already offer to ultimately help the people with disabilities. The system we are introducing is developed with an architecture that makes it both cheap and easy to use in any environment, either dark or well lit.

978-1-5090-1269-5/16/$31.00 ©2016 IEEE

II. PROPOSED SYSTEM ARCHITECTURE

We have developed a smart home system for physically disabled people using Kinect [5] & X10 [8]. Our system provides most features of a smart home system and the user would be able to control them using hand gestures and/or voice commands. As we took advantage of Kinect's Depth Sensor which uses Infrared lights to see in the dark, our system can work in the absence of light with the same efficiency as it does in a normally lit environment. In this section we will present our proposed system architecture and the working process of its different modules. In section II-A we will describe the system architecture and finally in section II-B we will describe the gesture detection mechanism.

A. System Architecture

The Kinect is always connected to the Central Control Unit using USB interface. It tracks the elbows and wrists of the users with its depth sensor. The Kinect's depth data are then passed on to the Central Control Unit. The Central Control Unit then makes necessary decisions and then passes appropriate signals to the X10 Transceiver module. This module sends command signals thorough existing wires along with address data. The receiver X10 modules with appropriate addresses receive these data and then change the state of the devices concerned. The electrical and eletronic devices' addresses and current states are stored in a database for further assistance in controlling them. This architecture is presented in Fig. 1. More details about this procedure and modules are presented later in Section III.

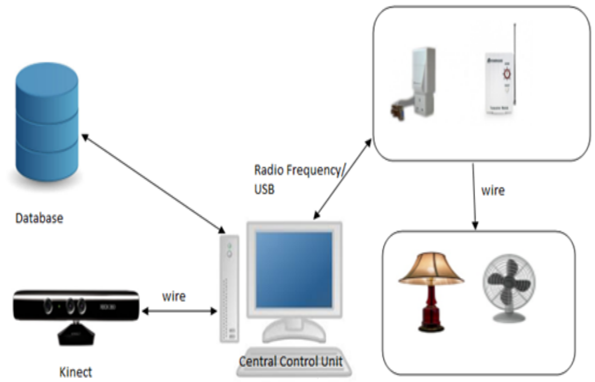

Figure 1. The System Architecture

As shown in the Fig. 2, the proposed system architecture is divided into 4 modules. Each module is marked with a different color.

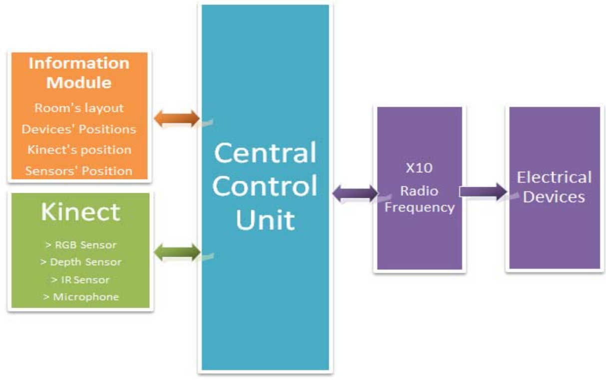

Figure 2. Block Diagram of System Architecture

The Information module is the one containing the database with necessary information e.g. all the devices' positions, the Kinect's position etc. These data are set when the whole system is initially installed in a room and can be updated at any time. The module below the Information module marks the Kinect sensor. The Kinect contains various sensors in it, namely RGB Sensor, Depth Sensor, Microphone etc. These sensors are the observers to collect information about the user.

The Central Control Unit solely consists of a Central Processing Unit (CPU). All the data available in the information module along with the information collected from the sensors are sent to this module. Depending on the information retrieved this makes the necessary decisions and in turn activates the X10 modules aka the Communication module to serve the user. The Central Control Unit also may receive information from the Device Control module, which comprises X10 and Electrical Devices in the room. The Device Control module sends the information pertaining to the electrical devices' current statuses to the Central Control Module whenever necessary.

The Central Control Unit communicates with X10 transceiver device using Radio Frequency channel or USB. Through either of them it can send the necessary signals to X10 modules to control the electrical devices. The X10 receivers check the status of the required device and then switch it to the state as commanded by the user.

B. Gesture

One of the major features of our system is the ability to simply point at any device in the room to select it for further commands. This feature is mainly targeted for users unable to perform voice commands. But it is open for anyone who might find a use for it. The Fig. 3 represents an empty room with different devices mounted on the wall or hanging from the roof.

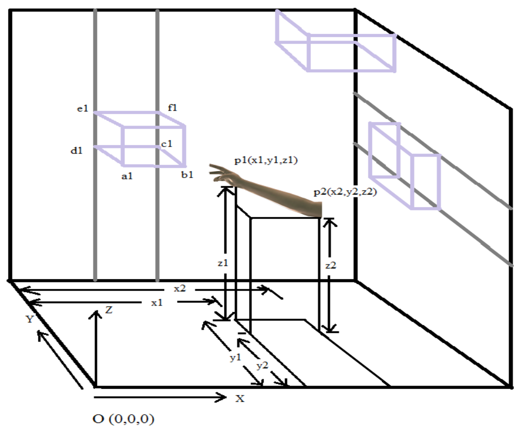

Figure 3. Measurement process while a user points his hand at a direction.

The hand in Fig. 3 represents a user who is pointing at any arbitrary direction that may or may not point towards a device in the room. The system deduces if he is pointing at any of them using the X, Y and Z coordinates of the user's elbow and wrist, and also becomes sure about this by waiting 4 seconds to prevent unwanted selections.

The method to do that is to keep the Kinect on an always watching mode. After installation the Kinect will always observe the user by using both RGB and Depth Sensing cameras. Although the Kinect (version 1) is able track as far as 20 skeletal points of the user, we only focus on the wrist and elbow for selection stage. Using the 3-Dimensional positions of these points we identify if the user is pointing at any device in the room; of course we also use the positions of all the devices and space it occupies.

We naturally perceive every device as an individual sphere. Some interactions with system with different people showed human are not very much accurate with rectangular models while pointing. Whenever asked to point towards a device they are seeing, most people pointed at the device or its surrounding area not far from it. We decided to take this natural small margin of error into account and started testing with different shapes. We found out it's easier for human to perceive devices as spheres for pointing purposes instead of accurate 3D object models. So, we modified the system according to this and the selection result was already better. Users didn't even need to know anything before pointing at devices; they were simply asked to point at them and they did.

For implementation, just like the "Line-Plane Intersection" [9] method, a ray is drawn using the elbow and wrist's coordinates of the user and the devices are thought as spheres. Whenever the user adds any new device to the system he/she is simply asked to point out two opposing points on different side of the device when thought of them as a sphere. Then the distance between these two points is considered to be the diameter of the device. A sphere is drawn from this diameter and center, which is considered to be the middle point of those two points. Then every sphere present in the room is simply checked against the ray drawn from elbow and wrist positions every moment to see if the user is pointing at any of the devices.

The calculation is quite simple. A tangent is drawn from the user's elbow to the surface of the sphere and another one is drawn from users elbow to the centre of the device. Then the angle between them is calculated. This is the Threshold angle, θ. This varies depending on the distance of the user from the device in consideration and also with the size of the device. Now a line is drawn using the positions of elbow and wrist of the user. Then the angle between this line and the other one that was drawn from the elbow's position to the centre of the device/sphere under consideration is calculated. Let's denote this angle using α. If this angle i.e. α is less than the threshold angle θ, we can say the user is pointing towards that device. If that's not the case, then the user is not pointing at the device under consideration.

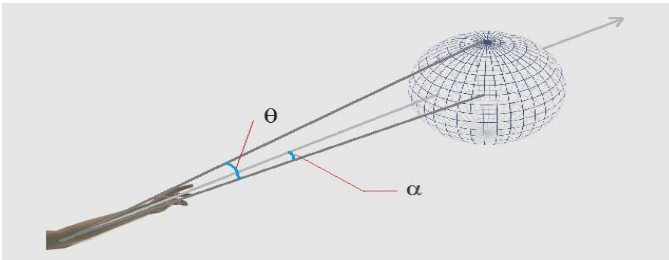

Figure 4. If $\alpha < \theta$, User is pointing towards the Device

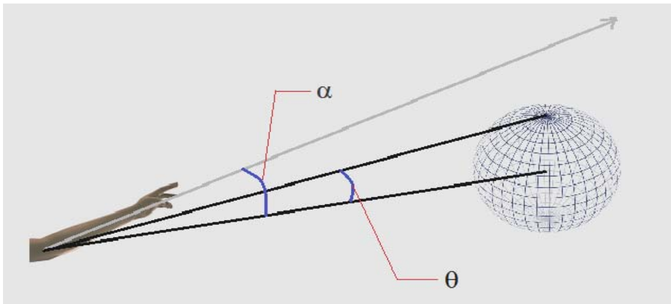

Figure 5. If $\alpha > \theta$, User is NOT pointing towards the Device

We used the "Law of Cosines" [10] for these calculations. Let's take figure 6 into account. A straight line EC is drawn from the user's elbow E to the device's centre C. Now another straight line EW is drawn from the user's elbow E to the user's wrist W. We then calculate the angle α between these two lines and check if it is less than θ. Let's assume points E, C and W forms a triangle EWC, where P = Length of EC, Q = Length of EW, R = Length of CW and α = Angle between EC and EW. According to the Law of Cosine,

$$R^2 = P^2 + Q^2 - 2.P.Q.\cos\alpha \quad (1)$$

Using this equation we can simply calculate α and check it against θ.

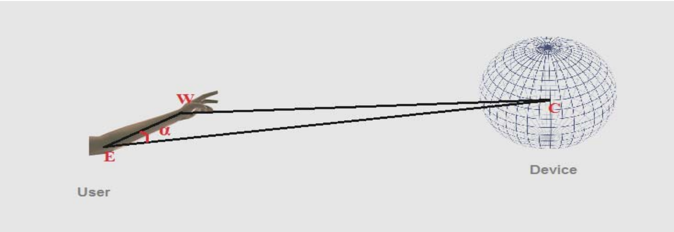

Figure 6. The Law of Cosine

The threshold angle θ can be calculated by following the same procedure. As we've discussed before, θ varies depending on the distance between the user and the device under consideration. It also varies with the size of the device.

III. IMPLEMENTATION

This section describes the implementation details of the prototype that reflects the proposed appliance control system presented in section II. Microsoft C# with .Net Framework version 4.0, the Kinect API V1, X10 API, and a laptop with Core i5 CPU and 4GB RAM were used to develop the prototype system. In section III-A, we will discuss what the gesture based commands look like in the final implementation. Section III-B and section III-C shade light on implementation procedures of voice commands and X10 modules, respectively. A snapshot of our implemented prototype is presented in Fig. 7.

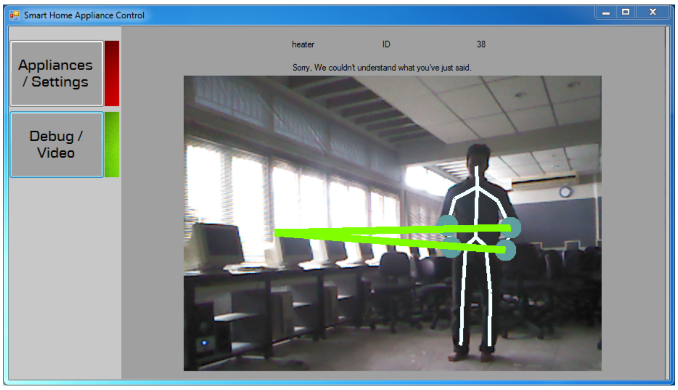

Figure 7. A Snapshot from the prototype system where one subject is standing in front of the kinect sensor to control the appliances using gestures.

A. Gesture Customizability

In our prototype implementation one of our major goals was to keep the customization option open for gestures, and we did not compromise with that goal. The user can select different gestures for different devices, even for same type devices with different IDs i.e. he/she can select "Right Wrist above Shoulder" command to turn Fan number 01 on and "Left Wrist above Shoulder" for Fan number 02. Given below is a chart that illustrates this:

Table 1. Gesture Based Commands Combination for User

Gesture	Action
Point at <Device>	Select <Device>
Right Hand Wrist Above Shoulder	Turn ON Selected Device
Right Hand's Wrist Below Hip	Turn OFF Selected Device
Left Hand's Wrist Below Hip	Toggle Selected Device's State

B. Voice Commands

The Kinect always keeps on listening for the user commands. If any of the spoken words matches any valid commands it then further checks if the command is successfully hypothesized. If the system recognizes any predefined command, it will do the task associated to that command, be that sending appropriate command to X10 or doing something else.

Some of these commands are just for selecting a device for further commands e.g. "*Air-conditioner One", "Television Three", "Light Four*" etc. And some are for performing the preferred action for selected device(s) e.g. *"Turn Down", "Decrease Heat"* etc. And if the user prefers it, then these two phases of commands can be combined into a single one e.g. *"Light Two OFF"* or *"Fan Two Down"* or *"All Devices On"* etc. The customization option was again left open for the user, just like gestures. So the table below is a possible scenario if the user prefers to customize the system this way.

Table 2. Voice Command List for User A

Command	Action
Light Two	Select Light Two
Turn OFF everything	Turn Everything OFF
Increase Heat	Increase Electrical Heater's Temperature
Light One OFF	Turn Light One OFF

Setting different combinations of Voice Commands and Gestures is also possible in the system. The user will have the option of controlling devices using either one of them. One example of such combinations of gesture based and voice based commands are given below in Table 3.

Table 3. Combination of Voice Commands & Gestures

Gesture/Command	Action
Point at <Device> / <Say Device's Name and ID>	Select <Device>
Turn OFF everything (Voice)	Turn Everything OFF
Turn ON everything (Voice) / Clap Hands	Turn Everything ON
Light One OFF (Voice)	Turn Light One OFF

C. X10 Modules

Communication with X10 can be done in many ways. X10 Transceiver modules are plugged into USB Ports or serial Ports of the CPU. The receiver X10 modules are plugged into power outlets and require no other connection to the transceiver module as they can communicate through existing wires of any home, as long as they are under the same load as the transceiver module. Each X10 receiver device has 8 bits of address code available for identifying up to 256 different electrical devices. These addresses take on forms like A1, B12, G9, C10 etc. The appliances to be controlled are plugged into these receiver modules, which in turn are plugged into any power outlets. X10 receiver modules also come in the shape of normal switches, which can replace the traditional switches of appliances and provide them with necessary unique device codes. Electrical devices will then be automatically connected to them. Next phase contains identifying each device using different addresses from the available 256 addresses. The devices' locations marked by X, Y and Z coordinates are also logged into database for future gesture based selections. Upon receiving a command from the Central Control Unit for a specific device, the X10 PC Module will send necessary command to the X10 device connected to that address. And it will perform the task commanded by the Central Control Unit.

This system's communication procedure with X10 is quite straightforward. The CPU sends the desired command along with an address to the transceiver unit. And the data relayed by transceiver unit is caught by the receiver unit with that particular address. And then it performs the desired function by turning the particular device either on/off or by changing its sound or heat level by controlling voltage. We followed traditional ways of programming for the X10 module of the system. We used X10 instead of similar technologies like Insteon [11] or Z-wave [12]. This is mainly because X10 is cheaper than the competitions and can work using existing wires of any home, thus ensuring hassle-free installation.

IV. RESULTS

The goal of our evaluation is to determine the performance of our smart home control system for physically disabled people. We experimented on different users' activities to measure the detection capabilities of our prototype. In the following section i.e. section IV-A, we will present the prototype's performance by calculating the response time of different modules and in section B we will present the user study.

A. Response Time

Response time is one of most important factors in a smart home system. Response time of our system largely depends on 3 major things. They are:

- Detection time by Kinect, represented by α.
- Communication Time between Kinect & CPU, represented by β.
- Propagation delay of the X10 devices, represented by P.

Now, if response time is R, then the equation is-

$$\boldsymbol{R = \alpha + \beta + P} \qquad \textbf{(2)}$$

The value of $\boldsymbol{P}$ in equation (2) depends on the noise present in the wiring. The X10 devices communicate with each other through existing copper wires of any home by sending digital signals in between the analog signals of normal appliances. If there are a lot of analog activities going on, communication between X10 devices gets delayed. Depending on this noise, the standard delay may vary from 2 to 5 seconds. In our system, average detection time by Kinect is 1.1s, average communication time between Kinect and CPU is 356ms and average propagation time of X10 is 2.3s. So the total average time is, R = 1.1 + .356 + 2.3 = 3.75s (approximately).

B. Usability Study

Prior to conducting the usability test we designed a test plan, where we defined our evaluation objectives, developed questions for the participants, identified the measurement criterion and decided upon the target users of the system.

The test took place in our university laboratory with 10 volunteers. Four of the participants were in the age group 40-55 years, three of them were in the age group 56-65, and the remaining three were in the age group 65+ during the time of the test. These volunteers suffered from different types of disabilities. The Table 4 presents some data about them.

Table 4. Different Types of Disability, Interaction Method & Success Rates

Disability type	**Preferred Interaction Method**	**Rate of Successful Interaction with System**
Lower Limb Disability	Voice & Gesture	71%
Upper Limb Disability	Voice	64%
Speech Disability	Gesture	81%

For complexity reduction the volunteers stayed with the system only at day time from 10 am to 1 pm each day. Each subject stayed for two days in our experimental room.

Table 5. Questionnaire for Measuring the Usability of the System.

Question 1	Is the system easy to adapt or is it user friendly?
Question 2	Is it comforting for disable people?
Question 3	Is it suitable for our daily life?
Question 4	Is the response time satisfactory?

The users' activities were monitored throughout the experiment and recorded for analysis. Later, based on their interaction experiences, the users completed the questionnaire listed in Table 5, where they were requested to provide ratings for the adaptability, comfortbility, suitability and performance. The user responses are shown in Likert Scale [13] in Fig. 8. The ratings of the questionnaire is in the range of 1–5; the higher the rating, the greater the satisfaction was. The average values of the responses were calculated in percentage form and measured after the usability tests. From the table we can see that more than 70% of users agree with us that our system is easily adaptable. More than 60% of them agreeed that this system is very useful for disabled people. Concerning these responses, we tried to make our system as much efficient as possible.

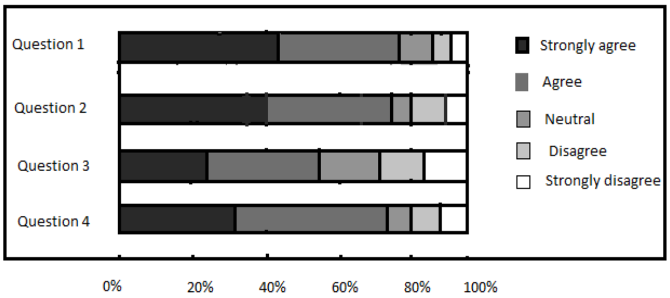

Figure 8. Experimental Result Based on the Question of Table 5.

For testing the voice recognition capabilities of our system we performed 3 different tests of our system in 3 different environments. Each time the test was performed using 50 voice commands to the system. While performing the test in different environments, we used different confidence levels of Kinect's speech recognition. We used different levels in between 40% and 70% in different environments. From the experiment we found that 60% confidence level is optimal for our system. So, the confidence level was set at 60% for the final system. The description of 3 different environments & the result of the tests in thsoe environments are given below in Table 6.

Table 6. Result of Voice Command System

Description of the environment of the test	Accuracy of the system in %
Normal household environment	73%
A little crowded environment	67%
Household environment with background music or noise	60%

It is worth mentioning that, the volunteers were always under the supervision of medically trained people.

V. CONCLUSION

Smart home system is one of the best creations of modern technologies. It has proved to contribute to increase independence and safety and made our everyday house-hold tasks easier. Most of the smart home systems are developed for normal people. But as we already discussed, a large portion of our total population is physically disabled. Most of these disabled people are living insecure and dependable life. So, being highly motivated by the suffering of these people, we developed a system which will give them a secured and independent life and provide them with a comfortable and interactive smart home system. We presented the implementation details of a preliminary prototype as per the requirements. From our usability study we received suggestions from both old people and their caregivers to improve the prototype e.g. "Add multiple Kinect in different location in a room" and "Add SMS or email based control" etc. As it is an on-going research, we aim to incorporate these features and many more in the future.

Disability is a curse for disabled people. And we hope our customizable and feature-rich system will play an important role to lessen the sufferings this curse brings.

REFERENCES

[1] Disabled World™, "http://www.disabled-world.com/disability/statistics/". Last visited 25th February, 2016.

[2] WHO, Disability and Health "http://www.who.int/mediacentre/factsheets/fs352/en/". Last visited 25th February, 2016.

[3] Ahmed Mohmmad Ullah, Md. Rashedul Islam, Sayeda Farzana Aktar, SK Alamgir Hossain". "Remote-Touch: Augmented reality based marker tracking for Smart Home control", February 2014. Presented during: ICCIT 2012, Chittagong, Bangladesh, December 2012.

[4] RCAST, University of Tokyo in collaboration with Microsoft Inc. "OAK, A Kinect based support system for severely disabled." Published on "http://www.assist-i.net/" (Subsidiary of University of Tokyo), September 2013.

[5] Microsoft Inc., "http://www.microsoft.com/en-US/kinectforwindows" , Last visited 15th February 2016.

[6] Hossain, M. Anwar, Atif Alamri, Ahmad S. Almogren, S. K. Hossain, and Jorge Parra. "A Framework for a Context-Aware Elderly Entertainment Support System." *Sensors* 14, no. 6: 10538-10561, 16th June, 2014.

[7] Hossain, M. Shamim, SK Alamgir Hossain, Atif Alamri, and M. Anwar Hossain. "Ant-based service selection framework for a smart home monitoring environment." *Multimedia tools and applications* 67, no. 2: 433-453, November 2013.

[8] SMARTHOME®, "http://www.smarthome.com/sc-what-is-x10", Last visited 15th February, 2016.

[9] Line-Plane Intersection, "http://mathworld.wolfram.com/Line-PlaneIntersection.html", Last visited 26th February, 2016.

[10] Law of Cosines, "http://mathworld.wolfram.com/LawofCosines.html", Last visited 26th February 2016.

[11] Insteon, "http://www.insteon.com/", Last visited 28th February, 2016.

[12] Z-Wave, "http://www.z-wave.com/", Last visited 28th February, 2016.

[13] Likert R (1932) A technique for the measurement of attitudes. Arch Psychol 140:1–55.