

Gesture-Driven System for Intelligent Building Control

Marcin Denkowski
Faculty of Computer Science,
Maria Curie-Sklodowska University
Lublin, Poland
marcin.denkowski@umcs.lublin.pl

Łukasz Sadkowski
SIM Ltd
Lublin, Poland
lukasz.sadkowski@sim.com.pl

Abstract—In recent years intelligent building systems have become standard in residential buildings as well as commercial buildings. This paper presents a novel system for controlling such buildings using natural user interfaces. Such a system consists of a set of depth cameras installed in buildings rooms. Due to advanced logic it is capable of identifying people occupying defined areas and recognizing their gestures. The system cooperates with a Domatiq System [1] (Intelligent Building Management System) to provide an overall control of the building. Together with the Domatiq users can define a set of gestures and associated events to realize specialized scenarios to control specific building components in a natural way.

Keywords—intelligent building, human-computer interaction, gesture recognition

I. INTRODUCTION

Intelligent buildings are buildings which, by their physical structure and specialized computing devices adapt to the changing needs of its users and organizations that use that building. Such a building should provide specific services to residents, manage the administration, operation and maintenance. Basically, smart homes are equipped with advanced automatic systems of sensors and actuators for various pre-programmed functions and tasks such as lighting, temperature control, multimedia, window and door operations, etc. Smart buildings allow residents to control various aspects of the behavior of a building by using a computer system. Users are able to control various devices such as the audio-visual media and other functions using a computer, touch pad, tablet or smartphone. Unfortunately, the main problem with this solution is that, although they improve convenience compared to a conventional home, users still have to physically interact with these devices and provide some techniques for the user authentication through a password, biometrics or RFID chips. On the other hand, humans communicate and interact with each other using both verbal and non-verbal languages. Verbal language is one of the most natural forms of human-human interaction. It has been used in many virtual systems [2] for human-computer interaction. However, speech recognition with its low accuracy and subtle rules of syntax and grammar of various languages is a very difficult task. These negative factors limit the practical use of verbal language. However, non-verbal communication such as gestures and body movements, as a secondary means of communication can be employed as another human-computer communication way to enhance the interaction performance. It is therefore logical to use this method of communication in dealing with the intelligent building. But, to our knowledge, few works pay attention to this topic in the virtual reality community [3].

In this context, several research projects have been developed. The DOMUS [4] project requires a smart control to gather information and impart instructions, one or more home automation devices to be controlled, and an internal communication network that allows the exchange of messages among the control system and the devices. A system described in [5] uses a Microsoft Kinect to detect a few simple gestures but does not provide a system to control the entire building or even a single room. 'Soft Remote Control System' described in [6] utilizes three CCD cameras to control multiple home appliances based on hand gesture. This human-friendly gesture-based interface system is able to recognize 10 different hand gestures but is very sensitive to lightness and the overall scene tint, and so more is simple and very restricted. Another very interesting smart home system is developed by Think Big Factory. Their Openarch [7] system provides the user with a possibility to control the house's components by the movement of the body, also to hold conferences from home, to know the power consumption at any time and activate any electrical household appliance from work or even share a live video. But it is still a prototype.

The proposal presented in this article assumes that the interface between the user and the building can be simplified and the process of communication can become more natural with the introduction of gesture-based control. However, it is a challenging task to apply this new way of interaction and to provide a good user experience and usability along with it.

The objective of this project was to design and implement an automated tracking and user identification system for use in a smart home environment. Our project seeks to eliminate conventional inconveniences of controlling smart homes and enable users to issue commands to their homes through simple non-verbal way.

The introduction of a PrimeSense PS1080 chip-based sensor [8] made a depth sensor be easily available for every user, enabling higher levels of a human-computer interaction by the use of gestures and body movements. In our system we use Asus Xtion and Microsoft Kinect sensors to provide control over the building components together with the Domatiq System [1]. Such a solution enables the possibility to detect the occupancy of the room and to implement occupancy dependent lighting control system or gives the users the possibility to control their audio-visual media just through gestures. If the room is not occupied, the lighting or media can be switched off automatically if someone has neglected to switch it off manually. Such a system can significantly benefit older or disabled people, to whom this imperceptible and natural way of controlling and monitoring provides both a sense of security, privacy respect and autonomy in their living [9].

II. SYSTEM OVERVIEW

The whole system consists of two subsystems communicating with each other (see Fig. 1). The first is a Domatiq System [1]. It is a solution for intelligent building automation installation in any building. The Domatiq is a system with a distributed architecture based on dedicated modules that communicate on the common bus and implement definable functions which are necessary to properly control electrical or electronic devices. There is the number of modules divided into sensors and actors. They are represented in the form of virtual operator points and a dedicated configuration application allows to easily define even sophisticated scenarios for these operator points to adjust the functionality of the system at any time. For example, one can order the house to enter the night mode from a single location: switch off the lights, close a walk gate, close roller shutters, etc. The Domatic System can be controlled by the touch pad (tablet), the LCD touch button panel, the PC application, web application or an infrared remote controller.

The second subsystem is a NIU Service responsible for the detection and recognition of gestures, their interpretation and requesting execution of specific commands to the Domatiq System. The main task of the NIU Service is to provide knowledge about the exact location of people in a defined area. The system is able to track users and detect gestures made by them.

Our NIU Service distinguishes exactly three events: (1) the entry of a user into the field of view of the sensors, (2) the exit of the user from the field of view of the sensor, (3) the detection and recognition of a particular gesture performed by the tracked user.

Communication component of the NIU Service is able to map a specific user's actions to the appropriate functions of the Domatiq System through exchanging data frames with Domatiq system using Domatiq panel application.

The NIU Service consists of imaging devices (based on PrimeSense 1080 chip) and a control unit on which this service runs. A single instance of the NIU Service is able to monitor the area spanned by the field of view of their sensors. These sensors are placed in the specific and fixed positions in the room. Room configuration, operator points and the NIU sensors are defined in an external software system supplied with the Domatiq System.

Message sent from NIU service to the Domatiq System via the Domatic panel is called the NIU Action. Every Action message consists of:

1. Type of an action – allows to choose one from four types of an action:
 - *Point* – the Action executed for a specific operator point on the Domatiq plan, associated with a particular module,
 - *Scenario* – the Action executed for a specific scenario associated with a global Domatiq macro,
 - *User Entry* – the Action executed on the entry of any user in the field of view of the sensor,
 - *User Exit* – the Action executed when the user exits from the field of view of the sensor,
2. Operation – associates the Action with a particular operator point or a scenario.
3. Function – particular function of the Domatiq module, for example: switch on a relay or lower a value on a dimmer, etc.
4. Gesture – a defined activity gesture assigned to this Action.

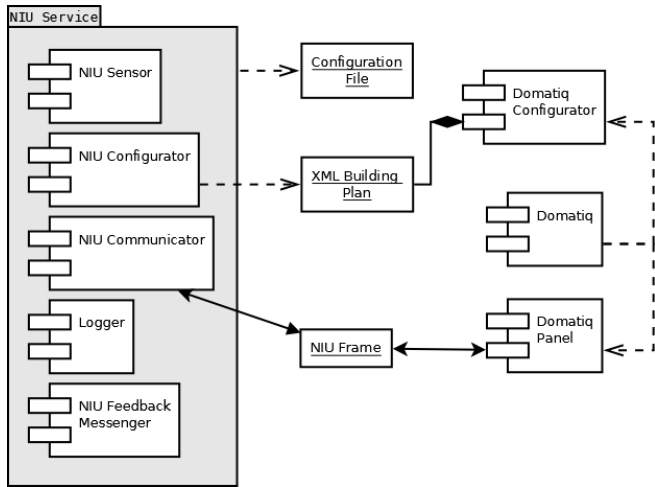


Fig. 1. Component diagram of NIU Service and its cooperation with Domatiq System.

Feedback about the system state, users, currently executing gestures, identification states, etc. is a considerable challenge for the system designer. The simplest methods for this type of feedback are audible or visual messages. Since addition of the speech synthesis would unnecessarily complicate the whole system, therefore the NIU system confirms every action by the characteristic short sound only. If you install additional LCD panel, the diagram of the room, including all operator points and users would be displayed on it (see Fig. 3). Icons of these users are highlighted in the appropriate color depending on their state, and if the user selects a specific virtual point, an icon of this point is slightly magnified.

III. PRIMESENSE SENSOR

In our system prototype we use devices based on PrimeSense PS1080 chipset as visual sensors for user tracking. At the moment, there are two such devices freely available: Microsoft Kinect and ASUS Xtion. Their depth sensor consists of an infrared laser projector and IR camera which create depth image according to the structured light principle [8]. The depth camera, in both cases, runs in a native resolution of 640x480 with 30 frames per second. The range of the field of view is 43° in vertical and 57° in horizontal. The operational range of the depth sensor is approximately from 50 cm to 400 cm. Our system prototype is capable of using at most three of such devices in a single instance. Due to this limitation there is a possibility to monitor an area as large as a room of a size 800 cm x 400 cm (see Fig. 2).

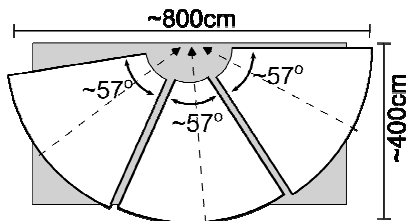


Fig. 2. Example coverage of an area of size approximately 800x400cm by three sensors of a PrimeSense type.

The result of the analysis of projected points (analysis of user tracking) depends directly proportional on the intensity of lighting conditions in the room. Better results will be achieved with darker conditions. However, the system should not have problems to work properly in the standard light (about 200-500 lux). If there are two sensors in one room, they should be arranged in such a way that the IR points projected by one of the sensors are not lighted into the field of view of the other sensor. Moreover, one sensor should not light directly onto the other sensor. It should be also avoided to target the sensor into the strong light source (such as a window). Best results will be obtained for sensors placed at a height of 1.5 m to 2.5 m, facing slightly downward to cover the entire human posture that is in their field of view.

Sensor devices, from the software side, are based on OpenNI library developed by a non-profit organization called OpenNI (Open Natural Interaction) run by PrimeSense. This library is provided along with motion tracking middleware called NiTE. NiTE

is able to track the human body giving the 3D positions of 15 specific body joints of each tracked user. A single sensor can track up to six users.

However, OpenNI libraries don't define any functionality to recognize and interpret gestures or to locate and recognize faces. For such functionality a FUBI framework was used [10,11]. FUBI (Full Body Interaction Framework) is a framework for recognizing full body gestures and postures in real time using the data from a depth sensor. Main functionality provided by FUBI can be reduced to: (1) static postures recognition – a specific configuration of several joints (positions or orientations) without any movement, (2) tracking of linear movements of several joints – linear movement of several joints with specific direction and speed, (3) combination of these two gesture categories – combines sets of 1 and 2 in a sequence of states and movements with specific time constraints. We also use an OpenCV library [12] as a main tool for all image processing tasks.

IV. GESTURE DEFINITION

In our system a gesture should be understood as a specific movement of a hand (or forearm) or just holding it in a certain position. All gestures can be performed using both left and right hand, To increase the accuracy of the gesture recognition procedure all movements should be smooth.

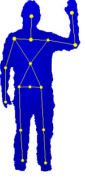


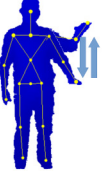
There is also a problem that the user can be anywhere and in any direction in the room. However, the gesture analyzer requires that the user is in a particular position in space, and is facing forward in the direction of the Z axis. To overcome this problem, after the OpenNI library has designated all the joints of the user, the virtual character is translated into the center of the coordinate system. Then, it is rotated around the X axis to compensate the tilt of the sensor relative to the plane of the floor on which the user is standing. Finally, the character is rotated around a vertical axis at the angle between the axis of the sensor's imaging and the normal vector defining the user's front. With this treatment we obtain the independence from the mutual relation between the sensor and the user.

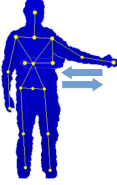
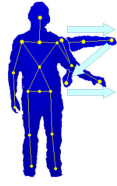
We have defined a relatively small number of gestures inside the system, which was dictated by practical reasons – the users don't have to learn and remember too many moves. Anyhow, the knowledge of the combination of gestures, together with operators and functions, is quite a large database of actions to be remembered. However, the gestures were carefully selected to imitate the natural movements of the human, such as turning on/off lights or increasing/decreasing the volume of audio devices.

Another important selection criterion were the analysis of the typical, everyday hand movements of users at home environment. On one hand, the gestures used to control could not be too simple, which might result in the frequent accidental execution or on the other hand, too complicated, which would lead to problems in their everyday use. Based on that assumption all gestures which were just a single movement of the hand and a lot of hand gestures requiring holding the hand in a particular place were rejected. To eliminate the randomness of gestures a static gesture called Comman Mode which enters the command execution mode has also been introduced.

Complete list of defined gestures is presented in Table 1. Thanks to a method for modular definitions of gestures contained in the XML file used by FUBI library this list can be modified according to your needs. Gestures were distinguished into two types: (1) *Command Gestures* (CG) and (2) *Activity Gestures* (AG). Command Gestures always perform the same command, whereas Activity Gestures can be assigned to any system Action or Scenario by users. We have defined two Command Gestures: (1) *Command Mode* which is used to activate and deactivate the Command Mode and (2) *Point Selection* which is used to select a specific operator point virtually located in the room. There are also four Activity Gestures defined: (1) Circle Gesture, (2) Up/Down Gesture, (3) Left/Right Gesture and (4) a gesture of Zorro. Circle, Up/Down and Left/Right gestures occur in two versions: clockwise/counterclockwise, move up/down and move to left/to right respectively. Users can assign these gestures to any Action or Scenario from the configuration menu.

TABLE I. A LIST OF DEFINED GESTURES.

	<p>Command Mode (CG)</p> <p>To activate command mode, user should lift a hand higher than the head joint so that the forearm is in the vertical position. Deactivation of the command mode is possible after 3 seconds after enabling command mode with the same gesture. There is no difference whether the gesture is performed by a left hand or a right hand. Proper execution of the gesture will be signaled by a system.</p> <p>Duration of a gesture: minimum 2 seconds.</p>
	<p>Point Selection (CG)</p> <p>In order to select a desired operator point, for which next activity gesture will be performed, user should aim a hand in the direction of that point, in such a way that the whole arm would be straighten on the height of a shoulder. System will select an operator point located nearest to the line connecting the head joint with the hand joint but not farther than an adjustable angle threshold.</p> <p>There is no difference whether the gesture is performed by a left hand or a right hand. Proper execution of the gesture will be signaled by a system.</p> <p>Duration of a gesture: 1 second.</p>
	<p>Circle (AG)</p> <p>Circle gesture consists of two clockwise or counterclockwise revolutions of a forearm. The move of the hand joint should start from the position nearest to the body and take two full circles. The radius of a circle should be greater than 15 cm and smaller than 50 cm. There is no difference whether the gesture is performed by a left hand or a right hand but the direction of revolutions determines if the gesture will be Left Circle or Right Circle.</p> <p>Duration of a gesture: minimum 3 seconds, maximum 6 seconds.</p>
	<p>Up/Down (AG)</p> <p>The Up gesture consists of two movements of a forearm from the bottom to the top and back. In the initial phase of a gesture, the hand should be placed below the elbow, and in the final above it. The Down gesture is analogous to the Up gesture but it starts from the top position above the elbow. The length of a movement should be about 50 cm. There is no difference whether the gesture is performed by a left hand or a right hand but the direction of movements determines if the gesture will be the Up or Down gesture.</p>

	Duration of a gesture: minimum 1 second, maximum 3 seconds.
	<p>Left/Right (AG) The Left gesture consists of two movements of a forearm from the right to the left and back. In the initial phase of the gesture the hand should be placed on the right of the shoulder and in the final left of it. The Right gesture is analogous to the Left gesture but it starts on the left of the shoulder and ends on the right of it. The length of the movement should be about 50 cm. There is no difference whether the gesture is performed by a left hand or a right hand but the direction of movements determines if the gesture will be the Left or the Right gesture. Duration of a gesture: minimum 1 second, maximum 3 seconds.</p>
	<p>Zorro (AG) The Zorro gesture consists of three hand movements arranged in a “Z” letter. The gesture starts with a hand placed on the height of a shoulder and goes right horizontally then moves left down to the hip and again goes right horizontally. The length of a movement should be about 50 cm. There is no difference whether the gesture is performed by a left hand or a right hand. Duration of a gesture: minimum 1 second, maximum 3 seconds.</p>

The user must be in the field of view of the sensor while performing a gesture, and should be careful while moving the hand so it will not be obscured by any obstacles including user's body. What is more, too fast execution of the gesture might not be registered by the sensor because of too slow frame rate of the camera. It is also recommended to perform consecutive gestures at intervals of at least 1 second.

After the sensors start, a tracked user can activate command mode by performing a Command Mode gesture. Changing to this mode is indicated by a sound signal and a color change of a user icon to green on a Domatiq panel application. Depending on the type of the action, there are two ways to issue a command by gestures and two by pseudo-gestures:

1. In case of an action of *Point* type, after activation of the command mode, the user has to select a specific operator point (by the Point Selection gesture) for which a command action will be performed. After choosing the correct operator point the Domatiq panel highlights this point on a plan and highlights the user icon in yellow. This selection is also confirmed by a sound signal. Then, the system waits seven seconds for the user to execute the Activity Gesture. If the system recognizes such a gesture, it is assigned to the defined action and the system executes that action. The execution is confirmed by a sound signal and the operator point is deselected.
2. In case of an action of *Scenario* type only an Activity Gesture is required. After detection of such a gesture, the system assigns it to the defined scenario and executes that scenario. The whole procedure is confirmed by a sound signal.
3. In case of a user entry (pseudo-gesture), after detection of a new user, the system checks if there are any scenarios assigned to that user entry event, and if there are any, then a defined scenarios are executed.
4. In case of a user exit (pseudo-gesture), after losing a user from the sensors field of view the system checks if there is any scenario assigned to that user lost event and if there are any, a determined scenarios are executed.

V. SHOWROOM CONFIGURATION AND TESTS

Our experiments were conducted in a showroom presented on a schema on Fig. 3. The whole room was 5m x 3m in size. There were two wardrobes and a desk with a chair. The sensor was placed in the corner of the room, opposite the entrance at a height of approximately 1.9 m. Its field of view covered about 70% of the whole area of the room. Moreover, we set four relay modules, two of them turn on/off the upper left and upper right lights, and the other two switch on/off the air vents (see Fig. 3).

For the purpose of the tests we have recruited 12 users, 9 men and 3 women. All users ranged in age from 25 to 48 with an average of 32. Three of them were classified as advanced users, i.e. those who have already dealt with gesture interaction in a virtual environment, 7 other users had to deal with gestures for the first time, or occasionally.

Fig. 3. Example view of a Domatiq panel application. Locations: left top edge - a single PrimeSense sensor; top right and left bottom vents operator points; in the middle of the top and bottom light operator points. There are also visible three users in different states: (from top) the tracked user, the user in a command mode, and the user who selected an operator point.



For each user we conducted 20-30 minutes training explaining how the system works, what are the possible command gestures and how the feedback from the system works. This time users had the opportunity to familiarize with the showroom configuration and test and train all of the gestures. After the training, each user had to perform several tasks:

- 1) enter the room, activate the command mode, use a gesture to turn on and turn off the right light;
- 2) enter the room, activate the command mode, sit on a chair, use a gesture to turn on the air-in vent;
- 3) enter the room, sit on a chair, activate the command mode, use a gesture to turn on the air-out vent;
- 4) use a gesture to turn on every module from 6 marked points in the showroom while being in a standing position;
- 5) use a gesture to turn off all the lights and vents, and deactivate the command mode;
- 6) Tasks 1) – 5) with 5 other users in the room.

While performing these tasks all the users' gestures were registered together with the effect of their execution. This allows to objectively assess the accuracy of gestures interaction in virtual environment and their accidental execution. After completing the scenario, the participants filled in a questionnaire in which we asked for a subjective opinion about:

- 1) the gesture is easy to perform: a) Command Mode, b) Point Selection, c) Circle, d) Left/Right, e) Up/Down, f) Zorro;
- 2) the gesture is natural: a) Command Mode, b) Point Selection, c) Circle, d) Left/Right, e) Up/Down, f) Zorro;
- 3) gestures are easy to learn/remember;
- 4) gesture control is reliable,
- 5) gestures seem to be tiring;
- 6) gestures are often executed accidentally;
- 7) system feedback.

And two open questions:

- 8) which gestures, body poses, and user positions give false results;
- 9) suggest any changes to the system.

First seven questions were answered by rating the statements on a ten point Likert scale ranging from 1 (completely disagree) to 10 (completely agree). Questions 8 and 9 were formulated as open questions to make it possible for the testers to provide feedback concerning the problems that may occur while using the system and to make any suggestions for changes or improvements.

VI. RESULTS

Figure 4 shows the accuracy for individual gestures divided between all users and advanced users. The Command Mode gesture and the Point Selection gesture were used in every case by all users, whereas the Activity Gestures were used according to their

preference. In every case, advanced users performed gestures more accurately, reaching the reliability from 4% to 25% (with average 14%) higher compared to non-advanced users. The average reliability of gestures performed by all users was 78%, performed by the non-advanced users was 70%, while advanced users reached 84%. Such a statistic shows quite clearly that when a user gets used to and becomes fluent in performing gestures and avoiding of specific situations for which the gesture recognition gives false results, the overall accuracy of the gesture control procedure significantly (almost 15%) improves.

Graph 5 shows the reliability of gestures depending on the standing or sitting pose. For each gesture performed in a sitting position worse results of reliability were achieved (which is from 3% to 12%) in comparison to the standing position, and for the Zorro Gesture even more than 40%. The average reliability for seated gestures was 68% whereas for a standing poses reliability reached 80%. The decrease in the quality of gesture recognition in a sitting position is largely due to difficulties in determination of the main body joints by the NiTE library at partial or complete occlusion of the lower parts of the body.

Figure 6 shows the popularity of individual gestures chosen by the users. The most popular Activity Gesture is a Circle Gesture, which is selected in more than half of the cases. Slightly less popular is the Up/Down Gesture chosen in over twenty to one hundred cases. The Zorro Gesture and the Left/Right Gesture are selected in only about 15% of the cases.

The results of objective tests are confirmed by the subjective feelings of the users gathered in a questionnaire, which was completed after the tests. The subjective feelings of the users concerning the reliability of the gesture control is around 7.0 on a scale out of 10, while the ease and naturalness of the gestures are 7.5 and 7.9 respectively on the same scale. What is more, the respondents reported that they had no major problems with the learning of the gestures – the easiness of gesture learning rated as 8.3. However, the respondents assessed gesture control as tiring at 3.0, whereas the accidental execution of the gestures, according to the testers, is believed to be very rare, which was also assessed at 2.1 in both cases on a scale from 1 to 10. In most cases the respondents found that the system communicates its status sufficiently and gives useful feedback (average 8.11 on a scale from 1 to 10).

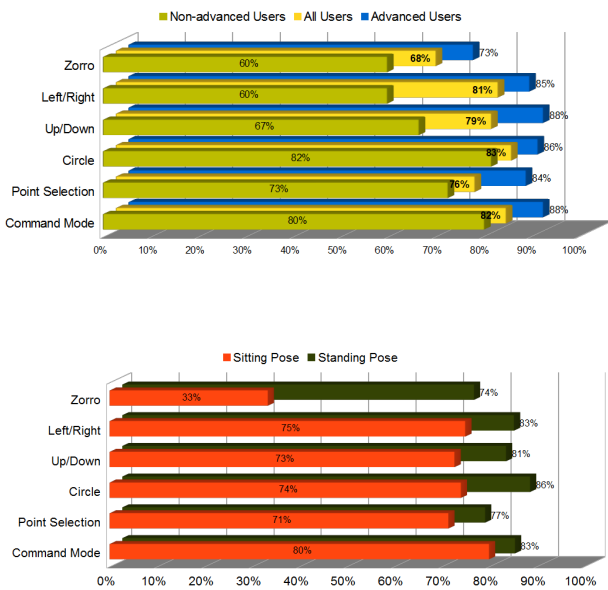


Fig. 4. Accuracy for individual gestures divided for All Users, Non-advanced Users and Advanced Users.

Fig. 5. Reliability of gestures depending on the Sitting or the Standing Pose.

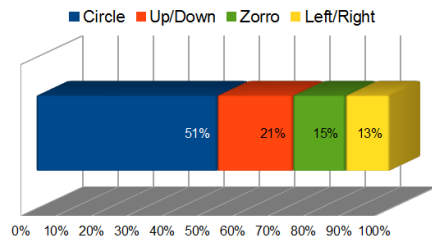


Fig. 6. Popularity of Activity Gestures chosen by the users.

In the open question concerning the difficulties with the system usage, the main problem indicated by the users was referred to the recognition of the gestures, which were made in the line of sight of the sensor approximately behind the body. Another observed problem was connected with tracking the gestures in case when the user was located on the edge of the field of view of the sensor. Moreover, several users also signaled difficulties with tracking and recognizing hand gestures in a sitting position. Responders often suggested to increase the field of view of the sensor, and to introduce some new gestures. The most interesting idea was to introduce a gesture that inquires about the status of the system.

VII. PERFORMANCE ANALYSIS

The demand for computing power of the system is quite large and the system requires at least an Intel I3 class processor. This is due to the depth image analysis, continuous tracking of the users, detecting their gestures and managing the transfer of data between the NIU system and the Domatiq panel application.

The main application was tested on a desktop computer equipped with an Intel I5 processor and 4GB RAM. For instance, running the application in an idle mode, when there is no user to track, takes about 6% of the overall CPU time. What is more, tracking and recognizing gestures use another 2-3% of the CPU time for every user's skeleton. Adding more sensors increases the CPU usage by further 5-6% for each sensor. With three active sensors, NiTE library is able to track up to 18 users, consuming on average 70% of the CPU time. In the case of detection or the loss of a user, there is a temporary but significant increase in the CPU usage reaching up to 50% of free CPU resources.

The PrimeSense sensor itself is demanding in the context of the USB bus bandwidth and saturates it in over 60 percent (USB version 2.0), which enforces the following sensors to be connected to a separate USB controllers.

Considering memory requirements, the most demanding component is the NIU Sensor responsible for managing every single Kinect or Xtion device, as well as for detecting, tracking and gesture recognition. Each instance of this component takes about 250MB RAM. Each tracked skeleton takes an average of about 25MB. Therefore, with three sensors and the maximum number of tracked users, the system needs almost 1.5GB of free RAM.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a system based on state-of-the-art intelligent building technology. Our approach covers all aspects of controlling an intelligent building with gestures, ranging from image analysis, recognition and interpretation of gestures to finally controlling devices. The results are very promising. This indicates, that it may be worthwhile designing new and more natural ways of interaction and controlling the intelligent buildings. People's lives can be greatly enhanced by these home-based assistive technologies, especially those who may be physically or cognitively challenged. This includes older persons, who especially should experience the benefits from this technologies.

For the further study, we will focus on enhancing the robustness of the system by using user-adaptation technology especially in recognizing gestures in the sitting pose. We also have done some initial work to recognize and identify users based on their face using face recognition algorithms with the database of their faces. Some drawback of our system is also a limited number of sensors used simultaneously. There is some work done in [5,13] to link Kinect to a network via the ethernet or wireless WiFi with a mesh topology.

REFERENCES

- [1] Domatiq Main Site, (2013), <http://www.domatiq.pl>.

International Conference on Automation, Control, Engineering and Computer Science (ACECS'14)
Proceedings - Copyright IPCO-2014, pp.187-196
ISSN 2356-5608

- [2] J. Gratch, et al., "Can virtual humans be more engaging than real ones?", In *Human-Computer Interaction*, HCI Intelligent Multimodal Interaction Environments, LNCS, vol. 4552, 2007, pp. 286–297.
- [3] S. Gobron, J. Ahn, D. Garcia, Q. Silvestre, D. Thalmann, R. Boulic, "An event-based architecture to manage virtual human non-verbal communication in 3d chatting environment", In *Articulated Motion and Deformable Objects*. Springer, 2012, pp. 58–68.
- [4] P. Carner, "Project Domus: Designing Effective Smart Home Systems", Dublin Institute of Technology, 2009.
- [5] A. Ben Hadj Mohamed, T. Val, L. Andrieux, A. Kachouri, "Assisting people with disabilities through kinect sensors into a smart house", *International Conference on Computer Medical Applications (ICCMA)*, 2013, pp. 1-5.
- [6] S. Yang, J. Do, H. Jang, J. Jung, Z. Bien, "Advanced Soft Remote Control System in Human-friendliness", *SCIS&ISIS2006*, Tokyo, Japan, 2006, pp. 218-222.
- [7] Openarch Main Site, (2013), <http://www.openarch.cc>.
- [8] PrimeSense Ltd: US Patent Application Publication No. US2010/0118123, <http://www.freepatentsonline.com/20100118123.pdf>, 2010.
- [9] C. Wolf, "Maintaining older people at home: Scientific Issues and technologies related to computer vision", Etia, 2011.
- [10] F. Kistler, B. Endrass, I. Damian, C. Dang, E. André, "Natural interaction with culturally adaptive virtual characters", *Germany Journal on Multimodal User Interfaces*, Verlag: Springer, Berlin/Heidelberg, 6 (1-2), 2012, pp. 39-47.
- [11] F. Kistler, D. Sollfrank, N. Bee, E. André, "Full Body Gestures enhancing a Game Book for Interactive Story Telling", *Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011*, Vancouver, Canada, November 28 – 1 December, 2011. Proceedings, pp. 207-218.
- [12] OpenCV, <http://opencv.org>.
- [13] A. Ben Hadj Mohamed, T. Val, L. Andrieux, A. Kachouri, "Using a Kinect WSN for home monitoring: principle, network and application evaluation", *ICWCUCA's, IEEE International Conference on Wireless Communications in Unusual and Confined Areas, Clermont-Ferrand, France, 2012*, pp. 1-5.