2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing

# 3D Modeling and Simulation of Human Activities in Smart Spaces

A. Helal[1], K. Cho[2], W. Lee[2], Y. Sung[1], J.W. Lee[1], and E. Kim[1]
[1] Computer & Information Science & Engineering Department, University of Florida, Gainesville, Florida, USA
[2] The Department of Multimedia Engineering, Dongguk University, Seoul, Korea
Corresponding Author: cke@dongguk.edu

*Abstract* — **For smart home researchers, it is essential to test activity recognition algorithms with various sets of sensory data. However, diverse sensory datasets are not always available due to several constraints, including limited budgets. Consequently, smart home simulators have recently grown in importance. However, there is still a need for realistic synthetic sensory data. This paper presents a simulator, 'Persim 3D', which relies on automatic scenario generation to create realistic sensory data. Persim 3D provides a 3D graphical user interface to help users' spatial perception, sensors that operate in real-time, similar to actual sensors, and a virtual character that lives in a virtual environment. These features allow users to generate longitudinal simulation data and eventually contribute to activity recognition research.**

***Keywords- Ubiquitous computing; Pervasive space; Activity Recognition; Simulation toolkit; Sensor-based simulation***

## I. INTRODUCTION

Smart home research has emerged as an important research topic due to the increasing demand on smart object and u-healthcare. Smart home research can be divided into activity recognition and context-awareness. Activity recognition research aims to recognize the activities of a resident. Context-awareness research examines service rules and operation for smart home applications [1].

Activity recognition research based within a smart house aims to precisely identify what activities are performed or are being attempted by a human being. The objective of such research is to help improve the quality of daily life of the elderly with mobility or other impairments, and to automatically contact a guardian by identifying abnormal activities. Activity recognition research can be especially applied to the rehabilitation of those with cognitive impairments or traumatic brain injury [2][3]. To improve the quality of activity recognition, a test environment for activity recognition is required, preferably one that is provided with low costs and convenience.

Persim [4][5] is a simulator for activity recognition research that simulates activities in a virtual environment without physical deployments. However, a challenge to the simulation of smart spaces is realism. Currently, Persim does not adequately create a realistic virtual environment for the accurate modeling of the physical activity entities in the real world. Reduced realism simplifies simulation. Therefore it is difficult to validate activity recognition. Persim also lacks easy methods to prescribe external activities to be simulated. Currently activities are specified and analyzed by Persim users, which is a cumbersome task, increasing the likelihood of inaccurate simulations.

This paper proposes to extend the Persim architecture to increase realism and to make it easier to use and to specify activities to be simulated. The new architecture facilitates the modeling of sensors required for the simulation of activities in a smart space. The main goal of simulator development is to aid activity recognition algorithm research by synthesizing realistic sensor data while at the same time reducing financial, time, and labor costs. To show how the proposed smart house simulator achieves the above goals, a simulation in 3D space using the proposed smart house simulator is described. In a simulation, environment and smart objects also effect the functions and values of sensors. However those are not dealt with in detail in this paper. We will model and simulate those effects in the future.

This paper is organized as follows. Section II analyzes the features and problems of simulation research of smart spaces. Section III describes the needs of our research. Section IV explains the architecture of the proposed Persim 3D simulator and simulation techniques. Section V describes our implementation. Section VI analyzes the results of a validation test of the proposed simulator. Section VII provides a discussion. Section VIII describes future research.

## II. RELATED WORK

Simulators for context-awareness aim to accurately simulate the operation rules of actuators and smart objects in a smart house, and the simulation of service computers activating those rules [4]. UbiREAL provides the interface to connect an actual device with a simulator [6]. UbiREAL is designed to enable a user to directly enter the calculation module for a physical context simulation as well as detection of a conflict of rules. A Context-Aware Simulation System for smart home (CASS) detects the conflicts of rules and provides the ability to control a character to move it [7]. The Interactive Smart Home Simulator (ISS) also detects conflicts of rules and works on the basis of the predefined scenario [8]. ISS also has the embedded weather module having impact on the environment in a smart house.

Simulators for activity recognition research in a smart house also aim to synthesize the data required by the activity recognition algorithm. One simulation technique is the event-driven approach, where each activity invokes one event in a virtual space and synthesizes sensory data considering activities. A 3D Open Source Smart Home Simulator for Activity Recognition (SIMACT) records an event log without generating the same sensor data as those generated in an actual environment while it is the simulator for activity recognition algorithm [2]. Persim is an event-driven simulator [4]. It defines an activity by a human being as one event and predefines the relationship between such event and

978-0-7695-4843-2/12 $26.00 © 2012 IEEE
DOI 10.1109/UIC-ATC.2012.35

a sensor which will work when each event is generated. A user defines the simulation scenario by connecting the generation of an event, and then Persim synthesizes and outputs the value related to a sensor working according to the generation of the event. Persim is less realistic because it assumes that all sensors necessary to detect event will work.

Persim uses sensors, but applies the event-driven simulation technique. Furthermore, its time cost is significant because a user needs to record each sensor. UbiREAL [6] CASS [7] and ISS [8], the simulators for context-awareness, are for simulation of service rules, not the synthesis of sensor data. SIMACT [2] only applies an activity-based approach, not a sensor-based approach for simulation. In this paper, we propose a 3D simulation architecture for generating activity data.

## III. THE NEED FOR HIGH-REALISM ACTIVITY SIMULATION

Activity research within the context of a smart city is critical to many human-centric ubiquitous applications and helps dictate the criteria for the design of smart spaces and ubiquitous computing environments. Activity models, activity recognition algorithms, and activity recognition sensor platforms are active areas of research. This technology will be invisibly embedded into our everyday environments through a pervasive transparent infrastructure, capable of recognizing, responding and adapting to individuals in a seamless and unobtrusive way.

There are many challenges facing activity recognition research. Access to meaningful collections of sensory data is one of the major impediments to human activity recognition research. Researchers often need data to evaluate the viability of their models and algorithms. But useful sensory data from real world deployments of pervasive spaces are scarce. This is due to the significant cost and elaborates groundwork needed to create actual spaces. Additionally, human subjects are not easy to find and recruit.

Powerful and realistic simulation tools could be used to support the growing demand for test data. Simulations enable researchers to create focused synthetic replications of important events and activities under study. It can be easily changed and refined allowing researchers efficiently to experiment, analyze and fine-tune their models and associated algorithms. Simulation also allows a wider community of researchers to engage and collaborate to solve a specific problem. Hence, a design based on preliminary simulation studies would most likely to be a more robust and inclusive design. Also, a simulation model that mimics an existing real world pervasive space is most likely to answer more questions (and generate much more data) than the target actual space. This early stage simulation can help researchers evaluate their ideas and algorithms quickly, cost-effectively, and with reasonable accuracy. Such capability would be an indispensible tool to use in building and rolling out smart city projects.

Our team consists of members of the pervasive computing laboratory at University of Florida where Persim has been developed and is currently being extended. The team also consists of the RISE group at Dongguk University, South Korea, working on improving the realism of Persim. The members of RISE GROUP have significant experience with smart characters and virtual reality, as well as in programming by demonstration (PbD) research [9], which will be utilized to accurately generate action and scenario primitives. This collaboration is crucial to enhancing the expressiveness, accuracy and realism in Persim.

### *A. Human Characters and 3D Spaces Capture*

We refer to the version of Persim with the enhancements described in this paper as Persim 3D – a version which aims to simulate human activities as they occur in a real smart space.

A key enhancement that we propose for performing high-realism simulation is to build the virtual space as similar as possible to the real world providing 3D environment and realistic characters. 3D environments can be configured easily using mouse clicks for constructing walls, deploying windows and installing doors, etc.. Simulated spaces must also capture restrictions for the proper simulation of activities. These restrictions are proposed to enhance simulation realism in virtual spaces. Activity capture depends on the deployment of sensors and actuators, thus Persim 3D needs an approach in deploying sensors/actuators in a smart space that does not compromise reality. A sophisticated, yet simple to use approach is proposed to allow only realistic placements of sensors in the smart space. Moreover the range of sensors are visually viewed. Finally, we propose a virtual character to represent real human in virtual spaces. Sensors and actuators are invoked according to the movement of the virtual characte mimicking the real environment.

### *B. Persim 3D as a Toolset*

A major outcome of Persim 3D is a high-realism simulation and modeling toolset for experimentation and analysis of human activities in smart and pervasive environments. Therefore, after configuring the 3D environment and modeling sensors, activity researchers could generate the datasets of sensors and actuators by controlling a virtual character. These datasets are stored as a simulation output in the Sensory Dataset Description Language format (SDDL) [10]. The activity researchers can use this simulation output for a variety of purposes in activity recognition research.

The resulting tools will help activity recognition researchers to validate their activity recognition algorithms and models before embarking on costly deployments or recruiting human subjects in evaluative studies. The intuitiveness and accuracy by which Persim 3D models and simulates a smart environment and its associated activities will result in toolsets much needed in designing and validating u-city and other smart city applications and services. Persim 3D will also benefit smart home automation and control in many application areas such as health telematics and elder care, in which smart homes must be programmed to react to complex situations that must be sensed and recognized accurately.

## IV. PERSIM 3D DESIGN

### A. Overall Approach

Persim [4][5] is based on an event-driven approach and is implemented as a 2D space simulator. For many simulations, however, the goal requires high degree of realism and details. An event-driven approach cannot simulate complex activities especially if high degree of realism is required. Long-lasting activities (e.g., daily activity) are also complex to simulate in an event-driven approach because it involves significant processing of low-level events. Persim would therefore not scale in these cases. Therefore an alternative approach which could drive the simulation at a higher level and eventually reduces the complexity is needed for Persim. To mimic complex human activity in a virtual space, the space should be reflective and sentient to human activities. A 3D virtual space and virtual characters are being exploited to enhance realism and to support the simulation of complex activities.

To address the issue of scale, we propose a context-driven approach (as compared to event-driven) to achieve scalable simulation with less complexity. Context is a higher level structure which covers all the status information of environmental entities such as space, objects, and virtual characters including human subjects. Given a context, activities which are comprised of actions are selected and driven. Also a virtual character-based approach is being developed in tandem with the context-driven approach. A virtual character is programmed to model a human being. However, it is difficult because of the complexity of human behavior. Applying artificial intelligence into a virtual character can make the latter as smart as real a human in action.

Our two approaches for simulation and modeling, while able to resolve the problems described above, are incapable of describing how a smart character performs activities in a virtual space. In order to resolve this issue, we conjoin simulation and modeling together. Under such conjoining (shown in Figure 1), we first model activities which can be performed in a given context, configure them and schedule them, to be performed by a smart character (the modeling part). To animate the activities, we first decompose them into several primitive actions, which are in turn modeled and configured as well. Then the actions are finally plugged in animation predefined in a smart character (the simulation part). Another way to describe our approach is through the two intersecting pyramids of simulation and modeling. The simulation modeling is based on the context-driven simulation engine (base), which as it advances the simulation steps through key contexts each associated with one or more activities. Such activities are composed of actions (pyramid tip). The modeling pyramid, on the other hand consists of several models, including the 3D space model including furniture and objects (at the base), followed by models for all sensors in the space. Additionally, component models for characters and their animations are utilized (even though not created by this research team but rather procured as assets to Persim) to define programmable and autonomous smart characters (pyramid tip).

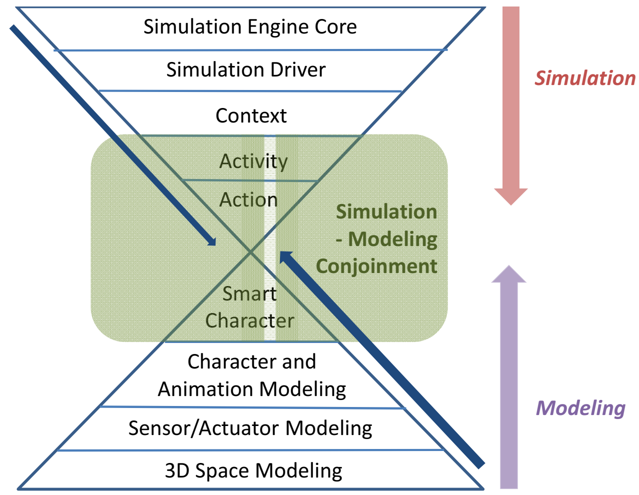


Figure 1. Overall Approach for Persim 3D.

In order to produce a realistic simulation, the modeling of space, sensors, actuators, and virtual characters must follow an integrated and "affective" approach. For instance if a vibration sensor is used, then vibration must be an attribute of each object as well as the virtual characters. Another example that demonstrates completeness is having the required animation to allow a character to sleep if a bed is included in the space (having a bed but no sleeping animation would not be integrated nor affective). Our integrated and affective approach naturally leads to higher degree of realism. Also virtual characters can perform activities (be programmed) based on realistic sensors and actuators models, which can affect and influence the virtual characters behaviors.

### B. Persim 3D Architecture

Persim 3D consists of the Virtual Environment Manager that builds the smart space (e.g., smart house), the Scenario Manager that specifies and generates scenarios, and the Simulation Manager. Such architecture facilitates the addition of spatial objects including sensors and other entities as the design expands to include diverse scenarios.

The Virtual Environment Manager configures a smart space model with input from a user. The space model comprises walls, doors, windows, objects arranged in a house, furniture and sensors. Then the Virtual Environment Manager analyzes the smart space model, stores information for simulation persistently in a project-specific information table and sensor information table.

The Scenario Manager assists in the generation of scenarios which govern the simulation of virtual characters' activities. Although this scenario manager is not implemented yet, scenarios can be generated by a user or by analyzing datasets before starting simulation. With Scenario Manager, user can generate diverse activities according to generated scenarios.

After generating spatial data and scenario data required to simulate a 3D environment using Virtual Environment Manager and Scenario Manager, the Simulation Manager utilizes these data and then generates simulation output in a

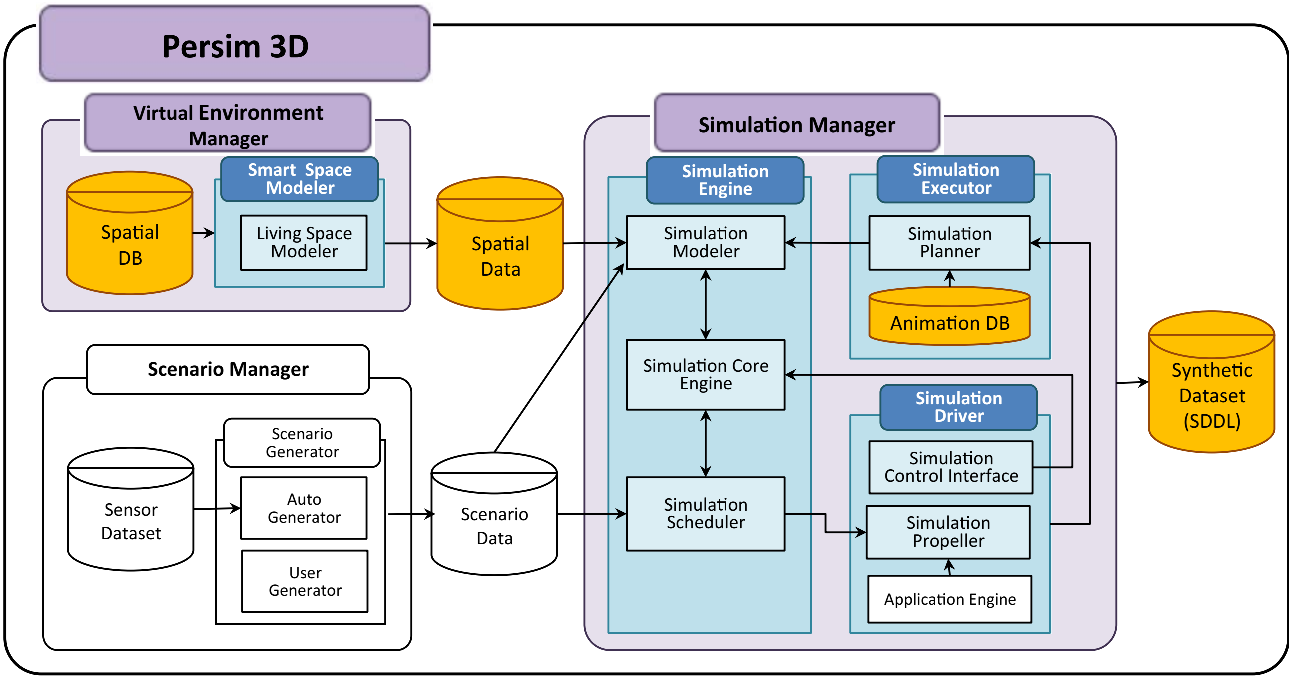


Figure 2. Persim 3D Architecture.

standard SDDL [10] format. The Simulation Modeler and Simulation Core Engine handle and analyze these data firstly, and then the Simulation Scheduler determines the next activities of virtual characters, the next actuations of smart objects and the next variations of environment. The Simulation Driver coordinates virtual characters, smart objects and environment to execute planned schedules through the Simulation Executor.

### C. Sensors Simulation in Persim 3D

Persim 3D simulation includes a variety of popular sensors usually found in actual smart spaces. Sensor effect range is also simulated. Accordingly, Persim 3D is able to synthesize simulation data that is very similar to sensor data in an actual space. Table 1 summarizes the types, applications and functions of the sensors currently modeled and included in Persim 3D.

The Motion Sensor is used to identify the location of a resident. When a movement is detected, it is assumed that a resident is in that detected place. The Pressure Sensor, when arranged in a grid pattern on the floor, is able to identify the location and path of activity of a resident. The Vibration Detection Sensor is to detect delicate movements. It can be utilized on a bed to identify tossing or turning of a resident on a bed. The Temperature Sensor detects change of temperature in a room in which it is installed, when temperature is affected by either human activities or natural effect (e.g., open front door or window). The RFID Tag can be attached to a cup or a kettle to check its position. The RFID Receiver can be worn (as pendant) or attached to a resident's hand, in which case it can identify if a resident holds or come close from a certain tagged object. The Contact Detection Sensor can be attached to doors, windows or drawers to identify opening or closing of those objects.

Table 1. Sensors modeled and used in Persim 3D.

| | **Sensor Types** | **Applications in a Smart House** | **Functions in Persim 3D** |
|---|---|---|---|
| **Location-based Sensor** | **Motion Sensor** | Detect the movements in the detection scope by being attached to walls or ceilings | Return "True" if a movement is detected |
| | **Pressure Sensor** | Detect the pressure in the detection scope by being attached to the floors | Return "True" when a character is inside a detection scope |
| | **Vibration Detection Sensor** | Detect vibration by being attached to furniture including beds or chairs | Return "True" if vibration is made by a character in a detection scope |
| **Physical Sensor** | **Temperature Sensor** | Measure room temperature | Calculate and return temperature depending on environment variables |
| **Object Sensor** | **RFID Tag** | Send signal to an RFID receiver. Attached to objects or walls | Standard to make a RFID calculate the signal strength |
| | **RFID Receiver** | Identify if an object is used by receiving signals from an RFID Tag, measuring the signal strength and estimating the distance to a Tag | Measure the distance to a RFID Tag and return the signal strength |
| | **Contact Detection Sensor** | Check opening or closing. Usually attached to objects which can be opened and closed including doors and windows | Comprise in a pair. When a pair loses its contact, "True" is returned. |

To incarnate sensors in a virtual space, they should be modeled first following their real-world behaviors. For instance, the Vibration Detection Sensor model can be used in a Smart House to detect delicate movements. The Vibration Detection Sensor is usually attached to beds, sofas or chairs to detect tossing or turning of a resident. It can also detect vibrations transferred through an object where it is attached. Based on the model, the Vibration Detection Sensor is implemented in Persim 3D and detects vibration if any is generated by a virtual character. Figure 3 shows the operation of the Vibration Detection Sensor. Each object has a collision box, which can be used to cause a collision between the object and a virtual character. When a collision between the collision box of an object and the mesh of a virtual character occurs, a vibration happens and is transferred through the object. The Vibration Detection Sensor on the object detects the vibration by applying a detection algorithm.

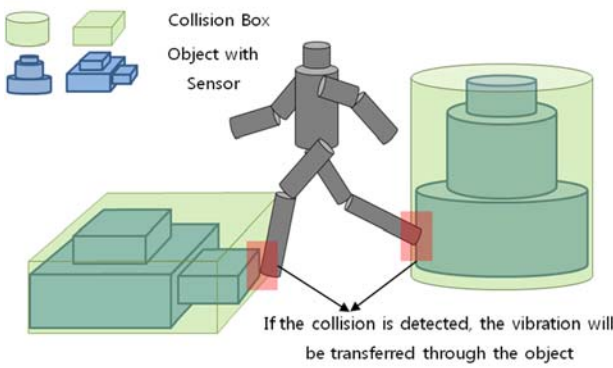


Figure 3. Vibration Detection Sensor Model.

Likewise, other sensors can be modeled from knowledge of their real-world behavior. Of course the fidelity of these models is limited which is the case in any simulation. For example, temperature distribution in a room is not even but a Temperature Sensor will not capture these slight variations. Currently, highest fidelity that is practically possible to implement is considered only as the models affect human activities.

Sensor modeling in Persim 3D does not only entail adding sensor models to a library. In fact, adding a sensor to Persim 3D requires the addition of one or more attributes to all existing and future objects and humans in the space. For instance, temperature becomes an added attribute to all objects and humans once a temperature sensor is added. Attributes could be directly or indirectly related to the sensor. For instance, weight becomes the attribute to be added to objects and humans when the vibration sensor is introduced and modeled. Such affective modeling approach ensures high degree of fidelity in capturing the space realism.

## V. IMPLEMENTATION

Persim 3D provides two kinds of views to help the intuitive understanding of a user on the space perception of a smart house comprising a virtual space. The left view in Figure 4 is an overview of the floor plan. The right view is a third person view enabling a user to move his/her point of view to any place depending on the user's needs.

The house in Persim 3D is constructed as follows. The structure of a house is built using the Space Element menu. A user can configure a house by selecting and clicking a wall. When the first outer wall is built by clicking to configure a smart house, a ceiling and a roof are automatically built. A user builds additional walls in the space surrounded by the outer wall and can add windows or doors if required.

Persim 3D provides basic furniture for living as space objects including beds, desks, tables, stoves refrigerators, telephone, spoons, cups and bowls. Furniture and objects have types for expansion. For example, a chair is basically provided as a round chair or a stool type. The items marked as 'Single' are provided only as basic types, but can be simply added when additional 3D models are added. Figure 4 (right) shows the screen in which furniture and objects are selected and arranged by the Persim 3D user in a smart house environment.

The next step after a space is designed and furniture and objects are arranged is to instrument the space with sensors to recognize movements and activities of a resident. Persim 3D provides a library of sensors commonly found in a smart home deployment. The arrangement method of sensors is unified. When a mouse is clicked in an empty space the pop-up menu 'Deploy a new sensor' is displayed. After the menu button is selected, the pop-up menu shows the list of sensors available in Persim 3D. After selecting the desired sensor from the list, a location at which to set the sensor is clicked, which results in insertion of the sensor into the 3D model. A user can change the attribute values of the deployed sensors, by simply clicking the sensors at which point the menu to change the attribute values is displayed on the right side of the screen. Each sensor has attributes such as name, user's memos and a detection scope. Figure 4 shows a smart house with deployed sensors.

Figure 4. The configured smart house in Persim 3D.

Compared to Persim 1.5 [4][5], Persim 3D provides the following additional features. Firstly, user can configure smart spaces in detail. Diverse sizes and shapes of rooms can be defined. The relation between rooms can also be described like real environment. Second, sensors can be deployed in specific places in the space. Finally, users can play (see) the process of simulation visually, and therefore users can check whether a virtual character is performing as expected and if sensors interact as desired or expected.

## VI. Experiment and Performance Analysis

In order to validate the performance of Persim 3D, we performed an experiment for comparison with Persim 1.5 [4][5] and Persim 3D. In this experiment, we simulate a simple scenario using both simulators, and validate similarities between generated datasets using each simulator and a real dataset. The real dataset is collected in the Gator Tech Smart House (GTSH) [11]. For the comparison, we apply fuzzy logic [4].

### *A. Breakfast Scenario for Validation*

The scenario describes a daily routine of a fictitious female character called Matilda. It includes 4 activities from sleeping to having breakfast. Times, locations, and sensors used to detect the activities are in the Table 2. Between each activity, there is a common activity "walking" between locations. This activity takes 5 to 10 seconds and does not appear in the Table 2.

Table 2. Activities in the sample scenario.

| *Activity* | *Start time*<br>*End time* | *Location* | *Sensors* |
|---|---|---|---|
| Sleeping | 7:30:30<br>7:35:22 | Bedroom | Pressure sensor<br>Vibration sensor |
| Using a<br>toilet | 7:35:25<br>7:36:10 | Bathroom | Contact sensor |
| Making<br>hot tea | 7:36:30<br>7:40:35 | Kitchen | Pressure sensor, RFID<br>Temperature sensor |
| Having<br>Breakfast | 7:41:40<br>7:47:15 | Dining<br>Room | Pressure sensor<br>RFID |

The scenario starts at the moment in which Matilda is sleeping on the bed. The sleeping activity is detected by vibration sensors attached on the bed. A few minutes later, she gets up and goes to bathroom. When she gets out of the bed, pressure sensors on the floor detect the movement. In the bathroom, she uses the toilet. The toilet activity can be detected by a contact sensor on the toilet because the contact sensor is triggered when she touches the flushing button. Afterwards, she gets to the kitchen to make hot tea. To make hot tea, she pushes a button on the stove and the temperature in the kitchen increases. Such activities are detected by a pressure sensor and temperature sensor respectively. When she approaches a cup to make hot tea, an RFID tag attached on the cup sends signals. Then she brings it to the dining room and starts eating breakfast which is prepared on the table. RFID tags are attached to the utensils and a pressure sensor is attached to the bowl. We simulate the above described scenario using Persim 3D. Figure 5 shows how the scenario performs in the virtual GTSM.

(a) (b)

Figure 5. Activities in the scenario: (a) Sleeping. (b) Having Breakfast.

### *B. Result and Analysis*

In order to validate Persim3D, we computed and compared a simulated dataset and a real dataset. We computed both micro and macro level similarity. First, for a micro level similarity comparison, we examined the sensor type and temporal position similarity of every individual sensor event. To illustrate, for simulated sensor event $s_i$ as in (1), we determined the most similar real sensor event $r_j$ and computed temporal position similarity after checking if their sensor types are the same. Second, for a macro level similarity analysis, we examined the features of the two datasets rather than individual sensor events. For example, we compared the number of sensor events for each sensor type in each dataset because if two datasets are similar, their sensor types and number of sensor events for each sensor type will also be similar.

For computing the temporal position similarity of two sensor events, Fuzzy Logic was used. In Fuzzy Logic, a fuzzy set *A* of the Input *I* is defined by its membership function (denoted also by *A*) $A: I \rightarrow [0,1]$[12]. In our analysis, there are two datasets *R* and *S*. *R* is a real dataset and *S* is a simulated dataset. We regard *S* as an input *I* and compute the membership of *S* corresponding to real dataset *R*.

$$R = \{r_1, r_2, r_3, \ldots, r_n\}$$
$$S = \{s_1, s_2, s_3, \ldots, s_m\} \quad (1)$$

We use the notation $A_{ij}(s_i) = (r_j, \alpha, \beta)$ to define membership of $s_i$ to $r_j$ where, $\alpha$ and $\beta$ define the maximum possible distances from $r_j$ towards its left and right, respectively. In our analysis, $\alpha$ and $\beta$ are calculated using start time and end time of each dataset. We found the fuzzy set for sensor type ($A_{ij}^{type}$) and sensor position ($A_{ij}^{position}$) using fuzzy membership functions in (2) [12][13]. We then computed the membership of $s_i$ using min-max fuzzy operation as in (3) [13].

$$A_{ij}(s_i) = \begin{cases} 1 - \dfrac{r_j - s_i}{\alpha} & if\, r_j - \alpha \le s_i \le r_j \\ 1 - \dfrac{s_i - r_j}{\beta} & if\, r_j \le s_i \le \beta \\ 0 & otherwise \end{cases} \quad (2)$$

$$A_{ij} = \min\left(A_{ij}^{type}, A_{ij}^{position}\right)$$
$$A_i = \max\left(A_{i1}, A_{i2}, A_{i3}, \ldots, A_{in}\right) \quad (3)$$

To compute similarity of a simulated dataset, *A(S),* we used the hamming distance method, which employs the distance between fuzzy sets *A* and *B* as shown in (4) [12]. In our analysis, we assign 1 to the fuzzy set *B* of a real dataset *R* because we assume that the membership of real dataset is always *true*.

$$d(A, B) = \frac{1}{m}\sum_{i=1}^{m}|A_i - B_i| \quad (4)$$

$$Similarity(A, B) = 2 - d((A \cap B),[1]) - d((A \cup B),[0])$$

For computing the similarity of the number of events for each sensor type, we used a similar method used in the computation for temporal position similarity. First, we count the number of events for each sensor type of both a real dataset and a simulated dataset. Then we compute similarity using (2) and (4). For this similarity computation, $\alpha$ and $\beta$ are the minimum number and maximum number of sensor events respectively.

The analysis result is shown in Table 3. Persim 3D outperforms Persim 1.5 for both temporal position similarity and event feature similarity of sensor events. To illustrate, Persim 3D shows 96% similarity for temporal position of sensor events whereas the similarity is 67% for Persim 1.5. For event feature similarity, Persim 3D shows 89% similarity whereas Persim 1.5 shows only 53% similarity. Total averages of the similarities of both tools are 81.5% and 71.3 respectively.

Table 3. Similarity of Simulated Dataset.

| Data Set | Having a breakfast in the morning | | Comparison of Similarity | |
|---|---|---|---|---|
| | Start Time | End Time | Temporal Position | Event Feature |
| Real | 07:30:30 | 07:50:12 | 1 | 1 |
| Persim 1.5 | 07:30:30 | 07:50:12 | 0.67 | 0.53 |
| Persim 3D | 07:30:30 | 07:50:27 | 0.96 | 0.89 |
| Average | | | 0.815 | 0.713 |

Figure 6 compares the similarity between simulated datasets and real datasets for both Persim 1.5 and Persim 3D. For both similarities, Persim 3D shows higher similarity than average.

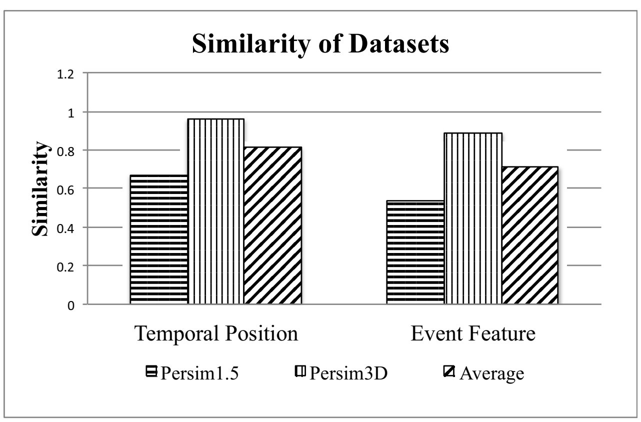


Figure 6. Similarities of Simulated Datasets by Persim 1.5 and Persim 3D.

## VII. DISCUSSION

One of the challenges facing human activity simulation is the high computational complexity of the simulation due to the various algorithms and the demanding implementations of virtual environment and living virtual characters. Persim 3D needs to collect data from a large domain to achieve a high degree of realism. When we collect data from a real environment for a long period, we can acquire many accurate "seed" activities, which ensures more realistic simulation. However, handling a large domain is computationally demanding and running a simulation may increase the complexity exponentially. We are currently addressing this problem via complexity analysis, and through a novel context-driven approach in which groups of selected sensor values (contexts) define the lowest level processing of the simulation core. Processing contexts instead of events is analogous to processing vectors instead of scalars. The context-driven approach to simulation is driven by contexts in a predefined structure called a context graph, instead of sensors events. When simulating by contexts, the complexity in computing is remarkably reduced, because irrelevant contexts are ignored and relevant ones are computed. This context-driven approach is work in progress and will be adapted in the simulation engine in Persim 3D.

Another challenging objective of the modeling aspect of this research is developing an Intelligent Character capable of living a daily-life autonomously. The current version of Persim 3D is not adequate for such purpose. Therefore, additional research is required. We will develop a daily-life Intelligent Character using automatically generated activity scenarios and the database created by PbD approach [9]. To develop an Intelligent Character capable of living a daily-life autonomously, additional research is required. For example, the Intelligent Character must be an autonomous character that is smart enough to live a daily-life without being controlled from the outside world. In order to create and add this capability to Persim 3D, an integrated design between motivation theory based system that we developed (called S-AVA [14]), the scenario database, and other algorithms must be carefully coordinated and developed.

The current version of Persim 3D supports only a single user activity simulation. Spporting multiple users and concurrent activities is an ultimate generality which will undoubtedly empower Persim 3D. However, ensuring successful simulation of single user scenarios before moving on to additional complexities is a highest priority in our approach.

## VIII. CONCLUSION AND FUTHER RESEARCH

Activity recognition research is in dire need of datasets to be used to verify and validate recognition algorithms and models. Such datasets are not plentiful due to the cost and labor associated with their creation. We have shown how Persim 3D - a simulator for human activity recognition research - can virtually create a smart space at low cost to synthesize the datasets needed for activity recognition research.

This paper reports on the simulation architecture and interface for simulating and generating datasets of activities performed by a virtual character in a smart house configured in a virtual space.

We showed how the smart house was designed based on Persim 3D's intuitive GUI. Furthermore, we showed how sensors were modeled and injected into the smart house. The simulation process was designed to enable a user to intuitively check the sensors activated during simulation by real-time visualization.

We conducted and presented an experimental study and verified the feasibility of implementing a real-time simulation.

## REFERENCES


[1] Y. Oh, C. Shin, Y. Seo, H. Yoon, J. Han, and W. Woo, "UCAM 2.0 : A Context Aware Application Model for U-city Environment Construction," Korea Multimedia Society, Vol .11, No.3, pp. 39-56, 2007

[2] K. Bouchard, A. Ajroud, B. Bouchard and A. Bouzouane, "SIMACT: A 3D Open Source Smart Home Simulator for Activity Recognition," Lecture Notes in Computer Science, Vol. 6059, pp. 524-533, 2010

[3] M. E. Pollack, L. Brown, D. Colbry, C. E. McCarthy, C. Orosz , B. Peintner, S. Ramakrishnan, and I. Tsamardinos, "Autominder: an intelligent cognitive orthotic system for people with memory impairment," Robotics and Autonomous Systems, Vol. 44, pp. 273–282, 2003

[4] S. Helal, J.W. Lee, S. Hossain, E. Kim, H. Hagras, and D. Cook, "Persim – Simulator for Human Activities in Pervasive Spaces," Proceedings of the 7th International Conference on Intelligent Environments, 2011

[5] Persim Official Webpate:
http://www.icta.ufl.edu/projects_persim.htm

[6] H. Nishikawa, S. Yamamoto, M. Tamai, K. Nishigaki, T. Kitani, N. Shibata, K. Yasumoto, and M. Ito, "UbiREAL: Realistic Smarspace Simulator for Systematic Testing," Lecture Notes in Computer Science, Vol. 4206, pp. 459-476, 2006

[7] J. Park, M. Moon, S. Hwang, and K. Yeom, "CASS : A Context-Aware Simulation System for Smart Home," 5th ACIS International Conference on Software Engineering Research, Management & Applications, 2005

[8] T. Van Nguyen, J. G. Kim, and D. Choi, "ISS: The Interactive Smart Home Simulator," Proceeding of ICACT $11^{th}$ International Conference on Advanced Communication Technology, 2009

[9] Y. Sung, K. Cho, "A Sequential Action Generation Framework for Intelligent Agents in Cyberspace, " Journal of Internet Technology, Vol. 12, No. 6, pp.969-978, 2011.

[10] S. Helal, A. Mendez-Vazquez, and S. Hossain, "Specification and Synthesis of Sensory Datasets in Pervasive Spaces," Proceedings of the IEEE Symposium on Computers and Communications, 2009

[11] Gator Tech Smart House Official Wegsite:
http://www.icta.ufl.edu/gt.htm

[12] D. Park, S. Lee, E. Song and D. Ahn, "Similarity Computation of Fuzzy Membership Function Pairs with Similarity Measure," Advanced Intelligent Computing Theories and Applications. 2007, Volum e 4682/2007, 485-492

[13] A. Abraham, "Rule-based Expert Systems," in Handbook of Measuring System Design, edited by P. Sydenham and R. Thorn John Wiley & Sons 2005.

[14] M. Cha, K. Cho, and K. Um, "Design of Memory Architecture for Autonomous Virtual Characters using Information of Visual Attention and Quad-Graph," ACM Int'l Conf. Proc., ACM, Vol. 403, pp. 691-696, 2009