

Development of a Simulated Environment for Human-Robot Interaction

Jochen Hirth, Syed Atif Mehdi, Norbert Schmitz, Karsten Berns
Robotics Research Lab, Dept. of Computer Science, University of Kaiserslautern
D-67663 Kaiserslautern, Germany
e-mail: {j_hirth, mehdi, nschmitz, berns}@informatik.uni-kl.de

Abstract

Skenario interaksi manusia-robot adalah sangat rumit dan memerlukan definisi yang tepat dari variabel lingkungan untuk pengujian yang ketat aspek-aspek berbeda perilaku robot. Kondisi lingkungan mempengaruhi perilaku manusia dan robot, sehingga manusia dan robot merespon berbeda pada kondisi akustik atau pencahayaan berbeda. Pada penelitian ini dilakukan eksperimen berulang-ulang dengan manusia sebagai subjek pengujian yang menyebabkan perubahan perilaku pada mereka, sehingga responnya tidak ada yang mirip dengan percobaan yang sudah dilakukan. Karenanya, membuat tidak mungkin untuk melakukan skenario interaksi secara berulang. Pengembangan dan penggunaan simulasi 3D, di mana parameter yang berbeda dapat disesuaikan, adalah solusi yang paling menguntungkan dalam kasus tersebut. Ini tidak hanya menuntut pengembangan robot simulasi yang berbeda tetapi juga simulasi dari lingkungan yang dinamis termasuk mitra interaksi. Pada makalah ini dihadirkan kerangka simulasi yang memungkinkan simulasi interaksi manusia dengan robot termasuk mitra interaksi tersimulasi dan dinamikanya.

Keywords: kerangka perangkat lunak, interaksi manusia-robot, robot, simulasi

Abstract

Human-robot interaction scenarios are extremely complicated and require precise definition of the environment variables for rigorously testing different aspects of robotic behavior. The environmental setup affects the behaviors of both the humans and the robots, as they respond differently under varying acoustic or lighting conditions. Moreover, conducting several experiments repeatedly with the humans as test subjects also causes behavioral changes in them and eventually the responses remain no longer similar to the already conducted experiments. Thus making it is impossible to perform interaction scenarios in a repeatable manner. Developing and using 3D simulations, where different parameters can be adjusted, is the most beneficial solution in such cases. This requires not only the development of different simulated robots but also the simulation of dynamic surroundings including the interaction partner. In this paper, we present a simulation framework that allows the simulation of human-robot interaction including the simulated interaction partner and its dynamics.

Keywords: human-robot interaction, robot, simulation, software framework

1. Introduction

In recent years there is an increasing interest in developing robots as service and assistance systems. Compared to traditional application areas of robots these scenarios require improved human-robot interaction skills. Therefore, it is not surprising that human-robot interaction is one of the major research areas in robotics today. One problem in the development of control algorithms for that kind of robots is the realization of experiments during the development phase. For testing and improving the algorithms it is necessary to have the same conditions over and over again. Since the main tasks of these robots are the interaction with humans and to fulfill operations in home environments it is almost impossible to ensure the exactly same environmental conditions for the experiments. Therefore, a simulation framework that allows the simulation of various robot systems including actuators and sensors and also regarding their inaccuracies, the simulation of different environments, including apartments, and also the simulation of human interaction partners including their movements and behaviors, means a great benefit for the development of interactive robot systems.

A popular simulation framework for robot simulation is Gazebo Player [1]. It contains several models of real robots with a variety of sensors like camera, laser scanner, etc. Robots

and sensors are defined as plugins and the scene is described in XML format. The environment is modeled as a 3D world consisting of static objects, which, however, can be moved by the robots. This capability is based on a simulation of rigid-body physics which is also included in the framework, permitting physically plausible interaction between objects. Although this framework supports various kinds of dynamic interaction, dynamic objects as humans are not integrated in the framework. Another commonly used robotic framework is OROCOS [2]. This framework does not provide a 3D simulation environment at all. Yet, simulation modules can be integrated into this environment using a dedicated MATLAB/Simulink interface. However, it is not suitable to simulate complex 3D environments and dynamic objects like humans. Simulation environments for the development of robots are wide spread but in most cases the simulation of humanoid interaction partners is missing. For the development of such a simulation two main aspects have to be realized: On one hand the human model has to be animated and on the other hand the model has to be controlled to generate realistic behavior.

The animation of the humanoid models is often limited to non-natural characters (see, e.g., [3]). For the animation of interaction partners motion capture systems can be used to store motion primitives which can be applied to a broad range of human characters (see, e.g., [4]). An additional challenge to the generation of motions is variable human models which either varies in size or joint hierarchy (see, e.g. [5]). To control human characters there exist different approaches. In [6] a so called cognitive control for human characters is explained. Environment knowledge of the agent is used to determine the human character's behavior. They developed a high level language to define the axioms necessary for the human character's decision making, in order to determine the most appropriate behavior. Compared to reactive approaches as explained in [7] or rule based approaches like in [8] they realize a more sophisticated behavior selection mechanism. Since affective and emotional reactions are characteristic for human beings some approaches, like e.g. [9], focus on their realization. Thalmann et al. describe a behavior-based method to realize not just goal directed behavior, but also spontaneous gestures as reactions to internal and external stimuli.

Although most of the simulation frameworks support a realistic 3D simulation of robots with standard sensors and support for system dynamics, there is still a need of a more flexible, allowing usage of custom objects, and extensible framework. Therefore, an extension for the SimVis3D [10] framework has been developed that allows the simulation of human-robot interaction. This article provides an extended description, also including new aspects, of the system introduced in [11]. Besides supporting a variety of robots and environments, the extension of SimVis3D allows the realization of various movements of autonomous human characters. The paper is arranged in the following way: At first the simulation framework and the developed capabilities for human-robot interaction are introduced. Afterwards experiments dealing with simulated interaction are presented. Finally a conclusion and an outlook are given.

2. Simulation Framework

The framework, developed at Robotics Research Lab at the University of Kaiserslautern, used for the simulation exceeds the state of the art in simulating robots and their environments in that it combines the robot system, its static environment, as well as its dynamic environment (Figure 1). The scenes consisting of the robot system and its environments are defined with an XML based description file. In this way all elements of a scene can be positioned and their characteristics like rotation or translation axes etc. can be described. The framework provides an unified interface to get and set the position parameters of an object (x, y, z, roll, pitch, yaw) which allows the motion of different objects like the head of the robot can be measured or realized respectively in a simple way.

2.1. Simulating the Robot's Sensors and Actuators

In the SimVis3D framework it is possible to simulate different sensor systems, e.g. laser scanner, infrared distance sensor, camera, etc. and also a lot of different actuators, like dc-motors, stepper-motors, or servo motors either with a velocity or with a position controller. One robot, which is realized in the simulation, is the humanoid robot ROMAN (RObot-huMAN interaction machine) (Figure 2). ROMAN is designed for studying human-robot interaction; especially nonverbal communication skills should be investigated. Therefore, it is equipped with

two stereo camera systems and six microphones. Each of them is also simulated using the framework.

The simulation framework allows free positioning and fixation points of cameras. Since the simulation framework is able to handle several cameras multiple views at the scene are possible. The images of the cameras in the simulation are transferred in a so called Blackboard structure which is used by the perception system of ROMAN. That way they can be processed like the data stream of any other camera and the vision processing algorithms of ROMAN can be tested. In order to test sound processing algorithms in the simulation it is necessary that a given input sound signal is transformed with respect to acoustical properties of the simulated scene into the signal resulting in variable target locations. A system developed by E. Deines [12], which is able to solve this problem, is included into the simulation environment of ROMAN. The humanoid robot ROMAN is equipped with two types of motors. On the one hand dc-motors with a velocity controller and on the other hand servo-motors with an internal position controller are used. The dc-motors realize the movement of the upper-body and the head, the servo-motors control the facial expressions and the eye movement. For the simulation of the dc-motors all typical characteristics like gear reduction, inaccuracy of the controller, time delay, etc. are regarded. Since the servo-motors have internal position controllers only a delay is modeled. This time delay is also realized within the servo-motor simulation. That way the interface of the simulated motors and the real motors is the same and they also behave in the same way.

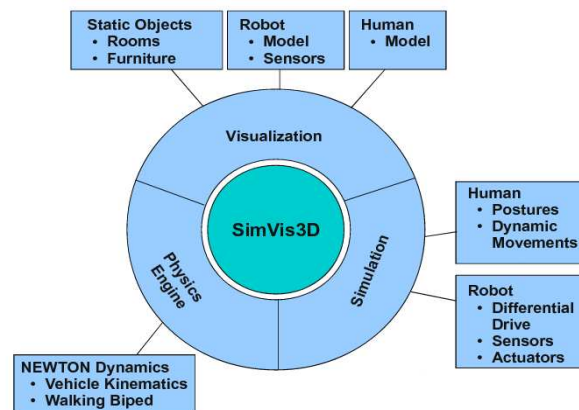


Figure 1. Components of SimVis3D

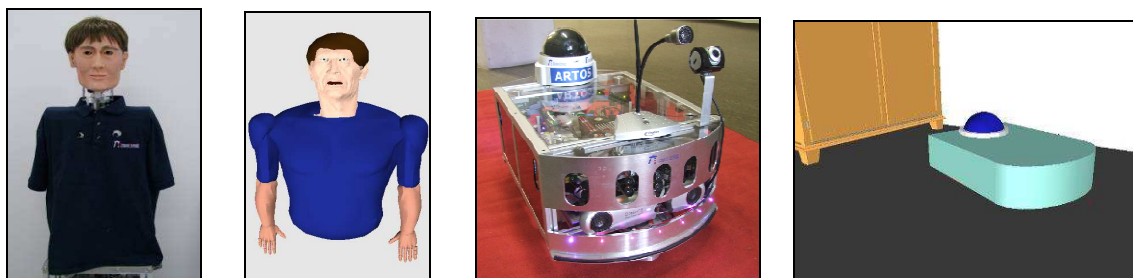


Figure 2. The real and simulated robots: The humanoid robot ROMAN (left) and the service robot ARTOS (right).

Another robot, namely ARTOS (Autonomous Robot for Transport and Service) (Figure 2), has also been realized in this simulation framework. It is especially developed as a service robot for home environments in assisted living scenarios. That's why the human-robot interaction is also a very important task for this robot. Like the real robot, the simulated ARTOS is equipped with a simulated pan-tilt camera and a simulated laser range finder. To simulate the

camera the same SimVis3D component than for ROMAN's cameras is used. For simulating the laser range finder the distance sensor of SimVis3D is used. The angle of the beam as well as the number of distance values that should be provided and a maximum distance can be defined for a certain distance sensor. The distance is calculated from a given center to the nearest object in the surrounding. To simulate the movement of the robot the same motor simulation can be used for both, ROMAN and ARTOS.

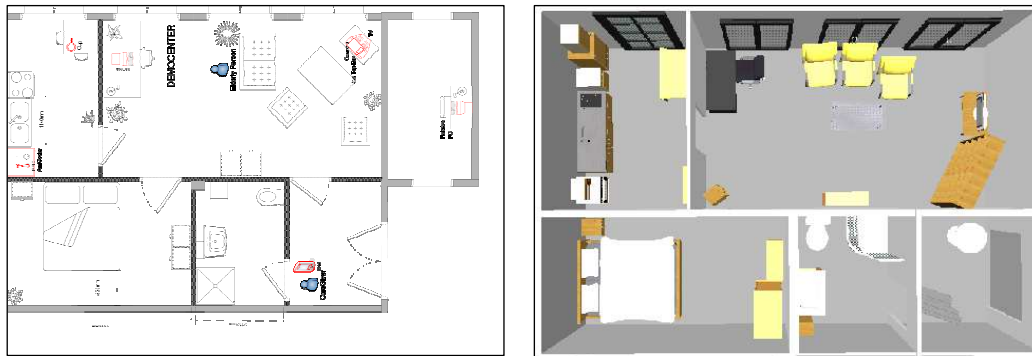


Figure 3. In the left the map of a real apartment is displayed. The right figure shows the simulated apartment including all static elements.

It is noteworthy that the control systems of both robots remain unchanged no matter whether they are acting in real world or in simulation. Furthermore, a big advantage of the simulation framework is that the components developed for the simulation can be used for building up completely different robot systems.

2.2. Simulating the Robot's Static Environment

For the simulation of an interaction situation, the static environment e.g. furniture is also of much importance. Since the data format used by SimVis3D is Open Inventor or VRML it is quite easy to include static 3D objects in the simulated scene. Construction tools like Pro Engineer provide the possibility to convert a drawing into the Open Inventor format. It is also possible to use 3D-graphics tools like Blender to generate objects in the appropriate format. Arbitrary static objects can be positioned in the scene relative to an anchor point. The position of a single furniture object is defined by pose offset in the XML description file. For easier positioning of these static objects more anchor points can be defined in the scene. Therefore again the 3D-graphics tool Blender can be used. In the laboratory example, anchor points for all tables, chairs, PCs, and cabinets are defined. Afterwards the furniture objects can be placed at these anchor point positions without calculating the pose offset. Figure 3 depicts a simulated apartment including furniture.

2.3. Simulating Human Characters and the Robot's Dynamic Environment

Besides sensors, actuators and the static environment a simulation framework should also provide the possibility to simulate the dynamic objects. The dynamic objects include all those parts of the environment which are able to react, influence and modify the current state of the simulation. Since our framework is designed for natural human-robot interaction the main focus for dynamic objects is to simulate humans although other objects like animals, flowers, or other robots are possible to realize with an acceptable amount of effort. Focusing on human simulation various problems arise. How are human motions modeled? How is the integration realized? And how do these models interact with the scene? All these questions will be answered in this section. In essence the simulation of humans can be divided into 3 major parts: (i). the modeling of human motions has to fulfill many requirements. A uniform tool for modeling has to be available and the output of that tool should be transferable to a broad range of characters. Ideally the modeling should not depend on the character or avatar; (ii) the next step is to connect the previously modeled motions and the avatar. Various motions have to be combined and complex behaviors have to be realized. If parameters of modeled motions are

available they have to be mapped to the virtual character. These parameters may include speed or range of motion, and (iii). Finally the avatar has to be able to interact with the environment. This interaction requires a uniform interface for the collection of information and the initiation of motions.

Modeling of human motions: The modeling of human motions is based on the well known human modeling standard H-Anim (Figure 4). Various versions of H-Anim are published and several models based on this standard are already available. Additionally major toolboxes like Poser partially support this free standard. The choice of this standard offers the possibility to integrate new avatars without changes of the motion modeling. The second aspect is the modeling tool itself. Various tested tools like Poser are limited to a specific character when motions are modeled. Therefore we decided to use Blender with simplified H-Anim conforming avatar to model the desired motion primitives. The levels of articulation (LOA), with values between 0 and 3, are defined by the H-Anim group and describe the amount of joints that are available within a specific group. The modeling tool supports the most detailed level 3 and is able to export any desired LOA level without a remodeling of any motion. Similar to various motion modeling tools the user specifies key frames in a time-line. At these key frames model positions are specified using drag and drop and the inverse kinematics tool integrated in Blender. When the modeling process is completed an export plug-in for the simulation framework is used to interpolate the specified key frames and export it as sequence of joint angles. Additionally to the LOA3 models lower detail levels are also exported. This allows the modeling of a motion for a broad range of avatars.

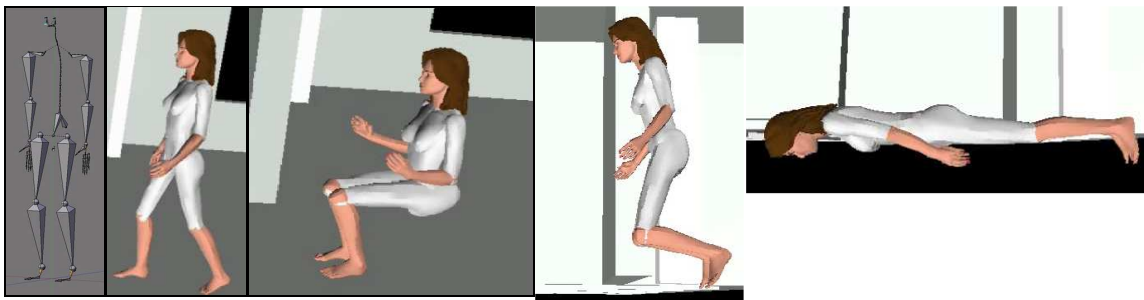


Figure 4. Realized motions for a simulated person on the basis of the H-Anim Model.

Integration into the simulation framework: Based on the motion files exported with Blender's export script the connection to the avatars and the combination of motions has to be realized. Each avatar in a scene is connected to a motion module which handles all activities that an avatar can perform. These motion modules handle the so called simple and complex motion primitives. Simple motion primitives are single movements generated and exported with the modeling tool. Examples for motion primitives are hand waving, a step forward, sitting down, standing up, and many more. Besides these primitive motions it has become obvious that further complex motions are required. Complex motions are defined as combinations of simple motions. An example for a complex motion is walking or picking objects. The walking motion combines several step forward motions and a rotation to allow an avatar to reach any point in a plane. The realization of simple motions, complex motions and the interface to the visualization will be explained with the help of the walking motion example. The only simple motion which is required is the single step forward motion. This motion is loaded from file which is the output of the previously described modeling process. It contains a *Position-Interpolator* which moves the *Humanoid Root* (the position where the avatar is located in the scene) of the avatar during the step motion and several *Orientation-Interpolators*, which contain the orientation of each joint of the avatar during the step. Begin and end pose of the model during that motion are identical, except the location in the scene, to allow a smooth loop execution of a single step. The complex motion walk consists of various steps which are sequentially executed: (i). the model is initialized, which brings the model to the pose which is required to smoothly start the single step; (ii). a path to the target location and orientation is planned. This path is planned using splines. Currently obstacles avoidance is not modeled in this process; (iii). single steps are executed

while translating and rotating the model along the planned curve; and (iv). a final step is performed to bring the model in a convenient position. Whenever sharp turns are planned, the rotation of the model in place results in a non natural turning motion.

This can be fixed by applying a second turning motion and overlay both motions for the realization of the complex walk motion. Last but not least the motion module is assigned to any H-Anim conforming model of avatar in the visualization environment, see Figure 4. The visualization environment provides access to all joint angles and the avatar position. For the development of the robot control system the Modular Control Architecture KL (MCA2-KL) is used. Therefore, SimVis3D provides an interface to run within this framework. Since the MCA2-KL is executed synchronously in loops the model is updated in every loop by querying an interpolation of the model pose at the requested time step. These values are then transferred to the visualization, which shows the playback of the motion.

Avatar module interface: The access to the motion modules from any module in the framework is realized using a textual interface. In general every interested module can connect to the motion module and send commands to the avatar. The interface for each of the motions – either complex or simple – is kept simple. Each motion can be executed by sending the name of the motion and optional parameters to the avatar module. The walk motion for example is called using the command “walk <posx> <posy>”.

Generally, human beings perform multiple actions while moving from one location to the other. They may sit, fall down etc. To accommodate this behavior of human movement a complete sequence of motion for an avatar will consist of multiple simple and complex movements. These motions can be randomly combined but it will generate a chaotic pattern of activities when two human movements cannot be performed in a sequential order, for example, a standing human can fall but cannot walk without standing up first. Therefore, it is required to generate the movements in a sequence in which a real human may perform these actions. Generation of a motion sequence has been accomplished by using Markov chain. Probabilities have been assigned to different movements and these movements are then combined based on these probabilities. In case, more than one action has the same probability, one of such action is chosen randomly or autonomous movements of the avatar in a more human like way, the simulated character walks in the environment from one place to the other. One approach can be to randomly select a location in the simulated environment and move the simulated human to that location. In order to make this movement more realistic, probabilities of presence of a human being at different locations have been generated. These probabilities represent the presence of a human being at different places in the environment based on time. Using these probabilities makes it possible to move the simulated character based on patterns that represent the real human being and thus the movement to different locations is not completely random.

3. Experimental Evaluation of the Framework

The robot humanoid ROMAN is developed to study human-robot interaction. One test scenario is a tangram game situation, see Figure 5. Tangram is a table top game where an image containing a silhouette is shown to a player. The task of the player is to order colored tiles in a way that their shape matches the shown silhouette. The robot can help by evaluating the current solution and providing hints e.g. about correctly or incorrectly placed tiles. To show that the proposed simulation framework allows the simulation of interaction scenarios a tangram game has been realized in simulation. To play the game the robot has to detect a person in its surrounding to play the game. Once a human plays the game the robot has to recognize and evaluate the solution currently provided by the human player. Figure 5 shows the real tangram game and the simulated game. It is worth mentioning that the whole control architecture of the robot remains the same and just the sensors and the actuators are replaced by the simulated ones. That way no changes in the control system are needed and it is guaranteed that the same algorithms are used in simulation and in real world. That means e.g. the image processing algorithms get their input image from the simulated cameras and the control data is sent to the simulated actuators instead of the real cameras and motors. The images show the results provided by the same evaluation algorithms once using the input of the real world and one of the simulated word. It can be seen that the results are the same. For more information see [11].

To test the simulated environment use ARTOS a typical application scenario searching the human being by the robot (see [13] for details). To accomplish this task, the robot has to

drive autonomously in the simulated environment and detect the human face using the camera. For autonomous navigation, it is necessary that the grid-map is build using the simulated laser scanner, containing information about the obstacles in the scene, and the path is planned avoiding these obstacles. In order to measure the performance of the robot for searching the human, certain points are marked as reference points in the environment. To make the scenario more interesting, it is not always possible for the robot to directly see the person from these reference points even if she is present around the same area. This is consistent with a real life situation where sometimes a human being can not be identified due to lightning conditions or orientation of the human or the robot. In this case the desired behavior of the robot is that it should move to another place and try to find the human there.

The experimental results show that the robot autonomously navigates to different locations to find the human character in the simulation (Figure 6). In some cases, due to orientation and positioning of the human, the robot was not able to find the human in the environment but in such cases it navigated to the other rooms as was desired.

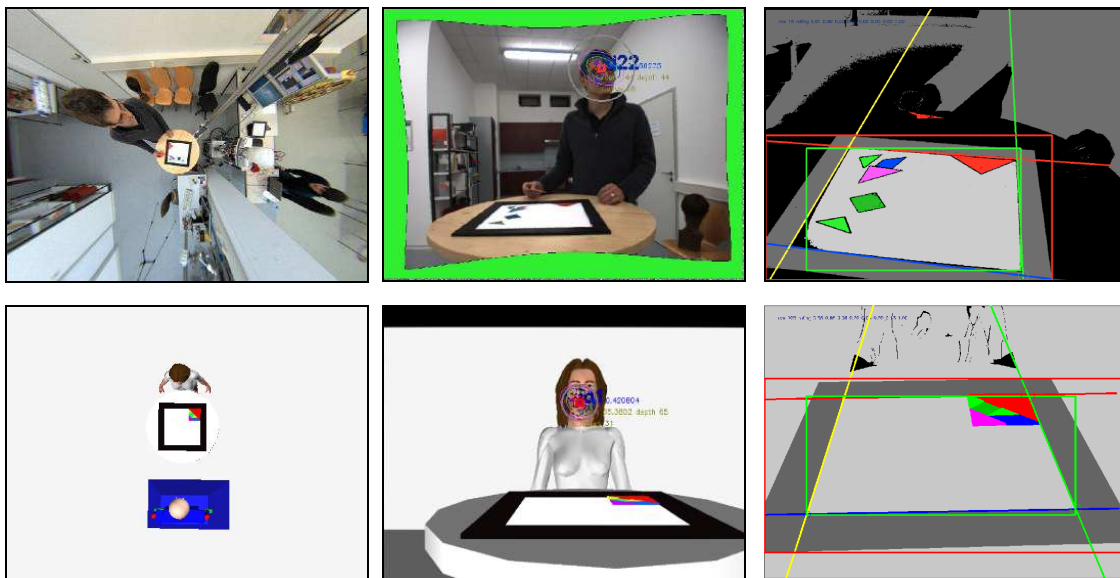


Figure 5. Comparison of real (upper row) and simulated (lower row) tangram game situation: From left to right the images show the whole scene, the face detection and the tangram game evaluation using color detector.



Figure 6. Comparison of real and simulated home environment: In the left-most image the robot plans a path to reach the person in the real world. The middle-image shows the simulated situation and in the right-most image the robot finds the simulated person.

4. Conclusion

Testing the interaction mechanisms between robots and humans is a difficult and work-intensive task. Especially, it is extremely difficult to ensure reproducible environmental

conditions. In a natural environment there are always changes and disturbances which influence both the human interaction partner and the robot system. In order to avoid such problems and to provide a test environment that always allows testing under freely definable and reproducible conditions, the simulation framework SimVis3D has been extended to realize interaction situations. To test the quality of the developed simulation two in reality existing robot systems, the humanoid robot ROMAN and the service robot ARTOS, have been implemented in the SimVis3D framework. Afterwards, the same experiments have been realized in simulation and in real world. It must be mentioned that the control algorithms were not changed. Only the real hardware and environment have been replaced by the simulated ones. The results of these experiments showed that the robots behave in the same way in the real world as well as in simulation. Therefore, with the presented simulation framework a very powerful tool that permits the easy definition of reproducible test scenarios for robot control software, which makes the development processes much more effective has been developed.

An important requirement for a simulation framework for human-robot interaction, the aspect of interactive simulated human behavior has not been addressed yet. Although the interface is already realized only scheduled motions are possible up to now. This aspect is of interest for future implementations. Furthermore, the simulated person should be enabled to realize also verbal interaction with the robot. Therefore, an interface between the simulation and the speech synthesis and speech simulation system should be generated. Another important aspect for making the simulated human more realistic is the realization of collision detection as well as their effects to the human and the environment. Besides this, additional standard motion patterns of the human character will be developed to increase the level of realism during testing of methodologies being developed for the robot.

Acknowledgement

We gratefully acknowledge DAAD and HEC-Pakistan for partially supporting this work.

References

- [1] Koenig N, Howard A. *Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan. 2004: 2149–2154.
- [2] Cáceres D. A, Barberá HM, Izquierdo MAZ, Balibrea LMT. *A Real-Time Framework for Robotics Software*. Proceedings of the International Conference on Computer Integrated Manufacturing (CIM-03). Wisla, Poland. 2003.
- [3] Greggio N, Silvestri G, Menegatti E, Pagello E. *A Realistic Simulation of a Humanoid Robot in Usarsim*. Proceeding of the 4th International Symposium on Mechatronics and its Applications (ISMA07), Sharjah, U.A.E. 2007.
- [4] Babski C, Thalmann D. *Real-time Animation and Motion Capture in Web Human Director (WHD)*. Proceedings Web3D - VRML 2000. Fifth Symposium on the Virtual Reality Modeling Language. ACM. Lausanne, Switzerland. 2000.
- [5] Gleicher M. *Retargetting Motion to New Characters*. Proceedings of the 25th annual conference on Computer graphics and interactive techniques. New York, USA. 1998: 33–42.
- [6] Chen L, Bechkoum K, Clapworthy G. *A Logical Approach to High-Level Agent Control*. Proceedings of the International Conference on Autonomous Agents (AA). Montreal, Canada. 2001: 1–8.
- [7] Basori AH, Tenriawaru A, Mansur ABF. *Intelligent Avatar on E-Learning Using Facial Expression an Haptic*. TELKOMNIKA Indonesian Journal of Electrical Engineering. 2011; 9(1): 115-124.
- [8] Schweiss E, Raupp-Musse S, Garat F, Thalmann D. *An Architecture to Guide Crowds using a Rule-Based Behavior System*. Proceedings of Autonomous Agents. Washington, USA. 1999: 334–335.
- [9] Thalmann D. *Control and Autonomy for Intelligent Virtual Agent Behaviour*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg. 2004: 515–524.
- [10] Braun T, Wettach J, Berns K. *A Customizable, Multihost Simulation and Visualization Framework for Robot Applications*. 13th International Conference on Advanced Robotics (ICAR07). Jeju, Korea. 2007:1105–1110.
- [11] Schmitz N, Hirth J, Berns K. *A Simulation Framework for Human-Robot Interaction*. In Proceedings of the International Conferences on Advances in Computer-Human Interactions (ACHI), St. Maarten, Netherlands Antilles. 2010: 79-84.
- [12] Deines E. *Acoustic Simulation and Visualization Algorithms*. University of Kaiserslautern; 2008.
- [13] Mehdi S. A, Berns K. *Probabilistic Search of Human by Autonomous Mobile Robot*. 4th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA). Crete, Greece. 2011.