DESIGNING A MIXED REALITY FRAMEWORK FOR ENRICHING INTERACTIONS IN ROBOT SIMULATION

Ian Yen-Hung Chen, Bruce A. MacDonald

Department of Electrical and Computer Engineering, University of Auckland, New Zealand
{i.chen, b.macdonald}@auckland.ac.nz

Burkhard C. Wünsche

Department of Computer Science, University of Auckland, New Zealand
burkhard@cs.auckland.ac.nz

Abstract: Experimentation of expensive robot systems typically requires complex simulation models and expensive hardware setups for constructing close-to-real world environments in order to obtain reliable results and draw insights to the actual operation. However, the test-development cycle is often time-consuming and resource demanding. A cost-effective solution is to conduct experiments by replacing expensive or dangerous components with simulated counterparts. Based on the concept of Mixed Reality (MR), robot simulation systems can be created to involve real and virtual entities in the simulation loop. However, seamless interaction between objects from the real and the virtual world remains a challenge. This paper presents a generic framework for constructing MR environments that facilitate interactions between objects from different dimensions of reality. In comparison to previous frameworks, we propose a new interaction scheme that describes the necessary stages for creating interactions between real and virtual objects. We demonstrate the strength of our MR framework and the proposed MR interaction scheme in the context of robot simulation.

1 INTRODUCTION

Throughout the development cycle of a robot system, experiments are conducted under many different environments and validated using a multitude of scenarios and inputs in order to gather reliable results before deploying the robot in the actual operation. Moreover, high risk robot operations, such as underwater, aerial, and space applications, typically require substantial resources and human support to ensure safety. The considerable cost and time for creating well-designed tests and for meeting safety requirements often presents a barrier to rapid development of expensive robot systems.

Motivated by concerns regarding safety, cost, and time required for real world experimentations of complex systems, Hybrid Simulation and Mixed Reality (MR) Simulation are proposed to simulate robot operations involving real and virtual components. This makes it possible to replace expensive or dangerous simulation components with virtual objects and allows testing of prototypes before the robot system is fully developed.

The simulator operates based on the Hardware-In-the-Loop (HIL) simulation paradigm which combines physical hardware experiments with virtual numerical simulations. In particular, MR simulation is founded on the concept of mixed reality (Milgram and Colquhoun, 1999) and varies the level of reality of simulation based on a continuum that spans from the virtual world to the real world. MR simulation incorporates real time visualisation techniques which have been proven to benefit robot developers in understanding complex robot data during development (Collett and MacDonald, 2006).

This paper presents the conceptual framework which our MR robot simulator is built upon. The framework formalises MR simulation systems in a generic manner that will be useful to other MR system researchers and designers, and will enable clearer comparisons and more standardised implementations of MR systems. We propose a novel method for facilitating interactions in an MR environment. In comparison to previous work, the interaction methods enables various entities (real or virtual, robot or the environment) to physically participate in simulation, thus

allowing complex test scenarios to be created for evaluating robot systems.

Section 2 describes related work. Section 3 presents our MR framework. Section 4 illustrates how we apply our concepts to building MR robot simulation. Section 5 demonstrates our MR robot simulator.

2 RELATED WORK

Applications of Augmented Reality (AR) and MR in robotics have provided developers a safe environment for testing robot systems. The work by *metaio GmbH* and the *Volkswagen Group* (Pentenrieder et al., 2007) demonstrate the application of AR in the industrial factory planning process and have reduced construction costs and improved planning reliability. Nishiwaki *et al.* (Nishiwaki et al., 2008) propose a mixed reality environment that aids testing of robot subsystems by visualising robot, sensory information, and results from planning and vision components over the real world for a more intuitive visual feedback. However, the focus of work has been placed on visualisation of robot information with limited information on creating MR environments for simulation.

MR environment construction requires combining real and virtual representations, which have been studied in the fields of AR and MR (Barakonyi and Schmalstieg, 2006; O'Hare et al., 2003). These concepts can be generalised to simulation of robot systems. However, there exist few formal frameworks for describing interactions between real and virtual objects. Many AR/MR applications support physical interaction with virtual information using tangible interfaces (Kato et al., 2000; Nguyen et al., 2005). These interaction techniques are limited in the sense that the designs are mostly constrained by the capabilities of human users, which makes it difficult to extend current MR interaction techniques to account for sensor capabilities of robot systems. Interactions between the robot and the environment are a requisite in simulation of most robot tasks. In comparison to previous work, we present a generic framework that formalises interactions between objects from across the real and the virtual dimensions, and demonstrate its application to robot simulation.

3 MIXED REALITY FRAMEWORK

The core of our MR Framework is designed to be generic and can be considered as an extension to gen-

eral concepts in the field of computer graphics and object-oriented design. We extend and apply these concepts to modelling a world composed of objects from different reality dimensions.

3.1 MR Entity

In order for virtual objects to interact with real objects, it is necessary to create representations of real objects in the virtual world. However, unlike virtual objects that are digitised and completely modelled in the computer, we may not always have prior knowledge of all objects in the physical environment. The completeness and accuracy of the model of the real world depends on the amount of information available, either measured *a priori* or collected during operation of the MR system, and they also affect the degree to which the interaction can be achieved. To create a model capable of representing an object that exists within the real-virtual continuum, we introduce an abstract *MR entity*.

In the MR world, an entity can be physical, digital or anywhere in between the two extremes. We model the MR entity with an attribute called *level of physical virtualisation* which describes the degree to which an entity's physical characteristics (primarily physical features that are measurable, such as shape, colour, and weight) are virtualised with respect to an object in the real world. An entity with an intermediate level of physical virtualisation is possible, as will be described in Section 3.2.

In addition, an entity may not be, or does not always fully represent the functional character of the intended object in the actual operation. For example, a photograph or a 3D model of a cow can be used in represent a real cow but it does not have the capability to move and act like a living animal. Consequently it is necessary to model the MR entity with another attribute called *level of functional virtualisation* which describes the degree to which an entity's functional characteristics (quantitative performance measures) are simulated with respect to the intended object.

The degree of virtualisation of the entity is determined by the combined result of physical virtualisation and functional virtualisation.

3.2 Entity Model

It is common to consider an object in the real world as a high level representation of a combination of several smaller objects. For example, a simple table is composed of a flat surface and four supporting legs. Modelling of composite objects has been thoroughly

addressed in the area of computer graphics, such as with the use of scene graphs to store a collection of nodes. In the MR world, we also treat a high level MR entity as a composition of multiple individual entities and other composite entities. We refer to the group entities that form a high level MR entity as an *entity model*. However, in comparison to traditional scene graphs, each node in the tree can be chosen to be real or virtual. The resulting entity model can be composed of a mixture of real and virtual entities and possesses an intermediate level of virtualisation. An entity with an intermediate level of virtualisation is referred to as an *augmented entity*.

3.3 Mixed Reality Interaction

While interactions between real objects occur naturally in the real physical world, interactions between real and synthetic (augmented and virtual) objects are more complex and require interventions from the system to model and realise the process. To facilitate interaction between real and synthetic entities, we need to model the process of transforming actions into effects (Rogers et al., 2002). It is necessary to know how a participating entity reacts to given stimuli, behaves under certain constraints, and also how its response can be generated. We treat this as a behaviour-based interaction problem.

When two agents interact, they exhibit different behaviours. Their behaviours describe the way they act and respond during the interaction. A typical expression of behaviour is shown in Fig. 1 (Arkin, 1998).

Figure 1: An expression of behaviour.

In an MR environment, an agent participating in the interaction can be any single real, augmented, or virtual entity, or an entity model consisting of group of entities, e.g. it can be a robot, a single sensor device, or an environmental object.

Simply put, if we know the interaction is possible between two objects in the real world, we can also try to reproduce the results for interactions with synthetic entities. To achieve this, it is required to model and generate the expected behaviours of the participating agents as if the interaction had happened. We first derive different stages of an interaction based on the expression in Fig. 1 then analyse the requirements for a successful interaction between two agents.

The behaviour expression suggests three stages that must be executed for an interaction to occur. 1)

Stimuli Processing, 2) Behaviour Modelling, and 3) Response Generation. The three interaction stages map to three sets of constraints that must be satisfied for each stage to be completed, see Fig. 2.
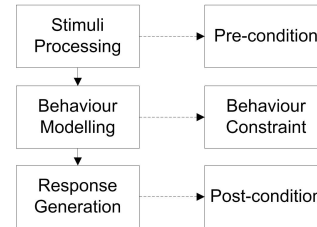

Figure 2: Three stages of behaviour-based interaction mapped to three sets of constraints.
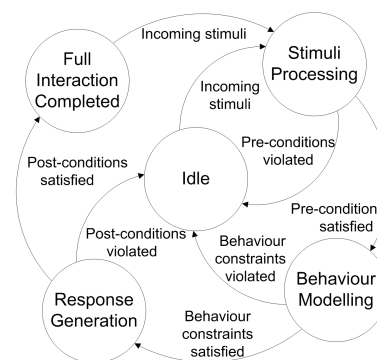

Figure 3: A condensed statechart of MR interaction.

Stimuli Processing – As two agents engage in interaction, input stimuli are processed with pre-conditions checking whether all necessary inputs to the modelling process are valid and sufficient information is available to initiate the interaction. Stimuli must be measurable and digitised for the modelling stage. Inputs can be visual, tactile, and audio in order to support most interaction means between humans, computer devices, and the environment.

Behaviour Modelling – Once the input data has been processed and interaction has been confirmed to proceed, the behaviours of the agents are modelled with the given inputs. Behaviour constraints are rules that govern the modelling process. They are typically mathematical equations or laws of physics which the agents' behaviours are bounded by.

Response Generation – The response generation stage is concerned with realising the modelled behaviour. To achieve a full interaction, the response needs to be executed by propagating the results from the behaviour modelling stage to both the real and the virtual world.

During response execution, the level of physical virtualisation of an entity indicates the class of reality it belongs to (completely real, augmented, or com-

pletely virtual) and thus determines whether an action to be performed on an entity model, e.g. translation or rotation, can be executed in the same manner for all successor nodes in the tree. The level of functional virtualisation indicates the entity's capability of generating accurate responses by executing the actions expected of it from the behaviour modelling stage, regardless of its class of reality. It is important to note that a full interaction may not always be achieved. Instead, a *partial interaction* may occur if the response can not be executed due to resource limitations or safety concerns. For example, it is not always safe to alter the path of a car during its travel after a simulated collision. In this case, the response can be generated by reporting the resulting behaviours to the user using alternative methods, such as textual or graphical output. Post-conditions are used to verify the completeness of the interaction. A full interaction is achieved only if mechanisms are available for executing the response, and the outputs from the executed behaviour match those from the behaviour modelling stage.
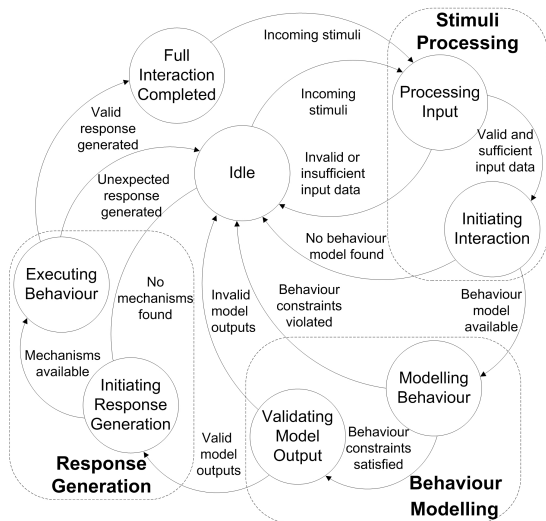

Figure 4: An expanded statechart of MR interaction.

Fig. 3 illustrates the state transitions within an MR interaction. The interaction statechart can be decomposed to the one shown in Fig. 4.

4 ROBOT SIMULATION

This section illustrates how we apply our framework to building an MR robot simulator.

4.1 Simulation Environment

MR entities allow users to create test scenarios involving real and simulated components. Objects that we wish to simulate can be built with different levels of virtualisation. Robots, environmental objects (including humans), as well as objects that do not possess a physical form can be extended from the MR entity, as shown in Fig. 6.
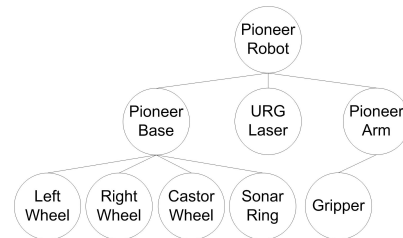

Figure 5: An example entity model of a Pioneer robot.

An entity model gives the flexibility of choosing what parts of a robot or an environmental object to be virtualised. Consider a Pioneer robot as an example of an entity model (see Fig. 5). Each node in the tree can be real or virtual. For instance, when a Pioneer arm is unavailable, a virtual counterpart can be used instead. However, we also place constraints between the parent and the child nodes depending on the application and the resources available. Sometimes it may not seem sensible to use a real gripper in simulation when a virtual Pioneer arm is chosen, whilst at other times, the use of a real gripper on a virtual arm is possible by employing surrogate devices for moving the gripper according to the calculated motions of the virtual robot arm.

4.2 Interaction Types

We demonstrate our MR interaction concept with two types of interactions, 1) Sensor-based interaction, and 2) Contact interaction. Note that these interactions are not limited to robotics. Sensors are increasingly used in technologies around us, such as in mobile phones which are becoming a popular platform to deploy MR systems, while contacts/collisions between real and virtual objects are important in tangible interaction studies. The concepts can be extended to common MR applications, such as the entertainment industry.

4.2.1 Sensor-based Interaction

Sensor-based interaction is concerned with interactions between a sensor device and other entities in the environment. Sensor devices are essential components of a robot system for robots to interact with and
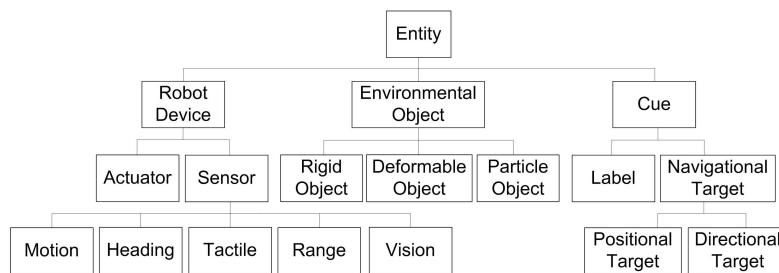
Figure 6: An inheritance diagram of an entity in an MR robot simulation.

gain understanding of the environment. In particular, we are interested in robot interactions through extero-ceptive sensors, both active and passive. This includes range, vision, thermal, sound, and tactile sensors.

Consider an interaction between a real range sensor device and a virtual wall. Pre-conditions check the validity of the input stimuli data before modelling the range sensor behaviour. The necessary data for this interaction include configurations of the range sensor device, positive sensor values, and known position, orientation, and dimension of the virtual wall in the environment. The behaviour constraint for a range sensor device requires that the range values are to be modified according to certain mathematical model in order to reflect a new object in the range output, e.g. a boolean operation to subtract the expected obstructed volume from the original range readings. The resulting range values must also be checked if they are valid, e.g. no negative range values. Once the expected behaviour has been computed, it is necessary to propagate the results to the real world, i.e. making alterations to the real range output. We also assume that during a sensor-based interaction, physical effects of the sensor on the environment are neglectable and thus, behaviour responses from the other party in interaction do not need to be generated. Once the response is executed, post-conditions verify the new range readings against the expected output values computed during the behaviour modelling stage to ensure a full interaction.

4.2.2 Contact Interaction

Contact interaction is concerned with physical interactions between entities, regardless of whether the entities are robotic devices or environmental objects. Contact interaction is common in robotics, e.g. collisions between moving objects, and is often necessary, e.g. in manipulation tasks.

Consider the example of a real robot colliding with a large virtual ball. Before initiating the interaction, pre-conditions check whether the dynamics properties of the virtual robot and the real ball are known, e.g. mass, and acceleration. An example be-

haviour constraint would require a physics engine to be available for calculating the resulting motion characteristics of the two entities in interaction. Lastly, to execute the response, checks are needed to identify whether mechanisms are available for safely interrupting the motion of the real robot or deflecting its direction of travel to result in a new motion that resembles the outcome of the behaviour model. Post-conditions are optional to ensure the virtual ball is moved in a similar manner to the output of the physics model for a full interaction.

5 SYSTEM IMPLEMENTATION

A Mixed Reality Simulation toolkit, MRSim (Chen et al., 2009b), has been developed based on the concepts introduced in our MR framework. We integrate MRSim with robot simulation frameworks to provide MR simulation of a variety of robot platforms and environments. The initial MR robot simulator is created by integrating MRSim with the Player/Gazebo robot simulation framework (Player/Stage, 2008). Since the design of our MR framework is general and independent from the underlying simulation framework, MRSim is sufficiently flexible that we have also ported MRSim onto another robot simulation framework, namely the OpenRTM robot simulation environment (Chen et al., 2009a).

To create a world where real and virtual objects co-exist and interact in real time, we implement an MR server that keeps track of the states of the real and the virtual world and seamlessly fuses information from the two worlds to create a single coherent MR world. The real world is essentially the physical environment where experimentation takes place. We use a graphics rendering engine, OGRE (OGRE, 2009), and a physics engine, ODE (Smith, 2008), to create the virtual world.

To construct the MR simulation environment, we first create a virtual model of the physical environment setup. MR entities representing real world objects that we wish to include in simulation are created
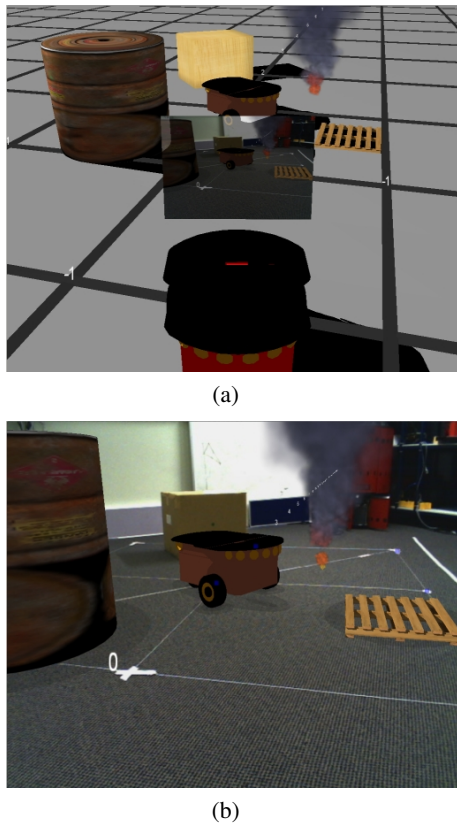
(a)

(b)

Figure 7: MR simulation of a hazardous environment. a) The virtual robot (red) is an avatar of the real robot in the real world. The robot is equipped with an onboard vision sensor which provides video imagery of its surroundings for the remote operator. Virtual MR entities are created and introduced into the environment along with other virtual representations of real world objects. b) A close up on the AR view of the MR world from the robot's perspective.

using available data measured *a priori*. In the ideal situation, a complete model is built and the virtual world constructed is a replica of the physical environment. To introduce additional virtual objects into the simulation environment, we let users create virtual MR entities and specify their geometric locations corresponding to the real world, as well as constraints and their relationships with real world objects (e.g. parent-child) using a XML configuration file.

An essential component of the MR server is a markerless AR system (Chen et al., 2008) which tracks planar features in the environment to provide pose information for merging the virtual world in geometric registration with the physical environment. Fig. 7 shows an example MR simulation environment constructed using a mixture of real and virtual MR entities. The screenshots illustrate a scenario where a real robot is teleoperated to carry out tasks within a simulated hazardous environment. Disparate sets

of information, e.g. robot and environment data, are visualised within a single integrated display based on the ecological interface paradigm (Nielsen et al., 2007).

During MR robot simulation, the robot will carry out tasks while interacting with objects from all reality dimensions. The MR server is responsible for this phenomenon. A simplified UML class diagram capturing the core components of our MR server and the implementation of the proposed behaviour-based interaction scheme is shown in Fig. 8. The world monitors the states of all entities and detects all possible interactions at each iteration. Associated with each entity is an interaction callback function which carries out the three stages of interaction described by the attached behaviour. During the interaction process, modelling of different entity behaviours and execution of their responses may require software- and/or hardware-specific implementations. The design supports custom behaviours to be extended and integrated into the framework. In our implementation, we create custom behaviours that utilise the underlying robot development tool for controlling robot devices to achieve MR interaction.
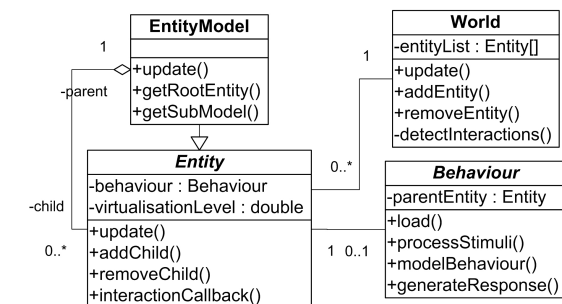

Figure 8: A simplified UML class diagram showing the core components of the MR server.

6 RESULTS

The AR interface shown in Fig. 7(b) is treated as a sensor-based interaction between a real camera sensor and virtual objects in the simulation environment. The interface is created using a behaviour attached to a camera sensor entity that registers virtual objects onto the input image. The resulting image generated correctly reflects the locations of the virtual objects which are introduced into the scene.

Fig. 10 shows an example of sensor-based interaction between a laser sensor and physical objects in the environment. The same algorithm was run on a real and a virtual robot, with a real and a virtual laser sensor respectively, to control the robot to avoid obstacles based on the laser range readings. The robot
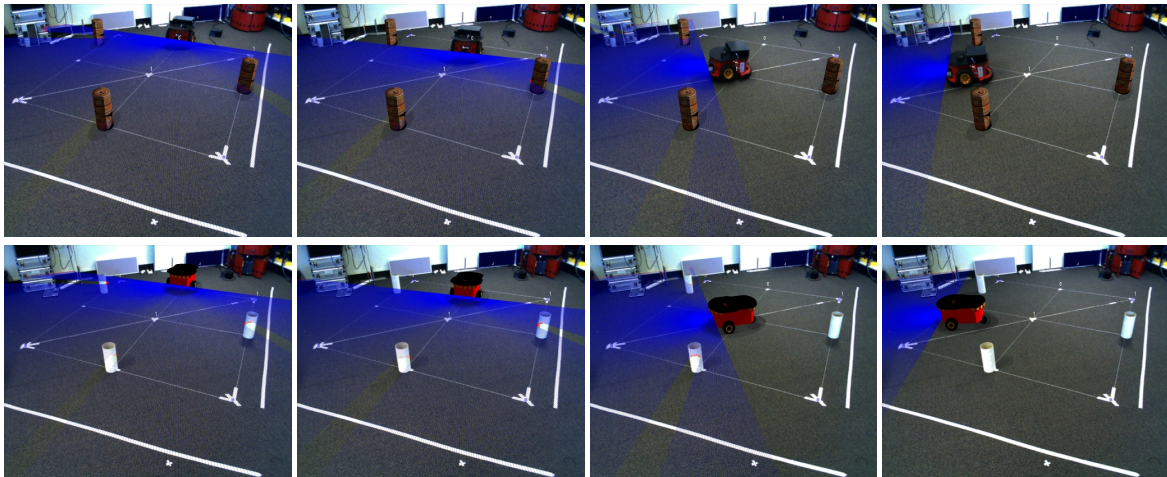
Figure 10: A sequence of screenshots illustrating a sensor-based interaction between a laser sensor device and three cylindrical objects in the simulation environment. Top Row: A real robot detects and avoids virtual obstacles. Bottom Row: A virtual robot detects and avoids real obstacles.
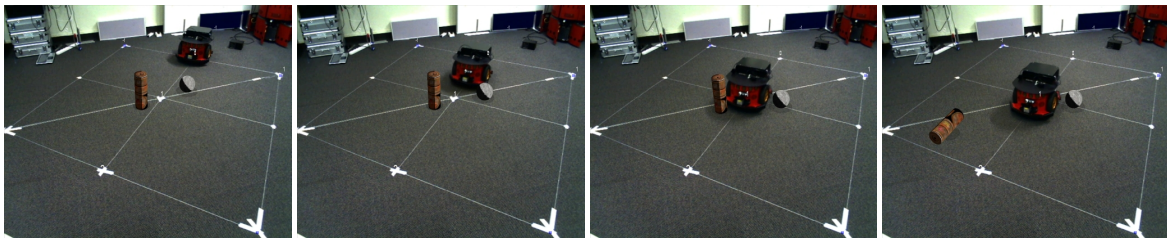

Figure 11: A sequence of screenshots illustrating a contact interaction between a real robot and two virtual objects.
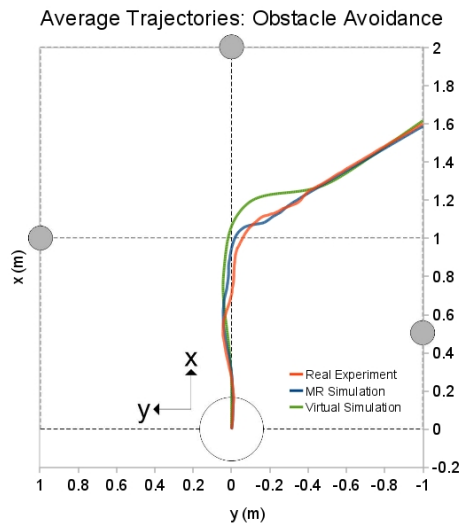

Figure 9: Average trajectories travelled by the robot while avoiding three cylindrical obstacles (gray circles).

successfully detected the presence of the obstacles in both cases and navigated towards the open space.

We also show an example of a contact interaction in Fig. 11. A real robot was operated to move until it collides with virtual objects in its path. The ODE physics engine was used to model the physics behaviour of the two parties in interaction, replacing the real robot with a virtual representation in physics simulation. The response of the virtual objects were successfully generated and visualised, however, mechanisms were not available in this example to alter the motion of the real robot according to the physics simulation, thus only a partial interaction was achieved.

6.1 Comparative Evaluation

An experiment is conducted to quantitatively compare the behaviour of the robot from running an obstacle avoidance algorithm in a completely real experiment, MR simulation, and virtual simulation. The experiment setup uses the same layout as the sensor-based interaction example in Fig. 10. In MR simulation, the real robot equipped with a real laser will avoid three virtual cylindrical objects. Each experiment is run 5 times and the average trajectories taken by the robot are plotted in Fig. 9

The results show the average trajecotry travelled by the robot in MR simulation closely resembles the one in the real experiment, with mean trajectory error 0.02m and standard deviation 0.01m. Comparing

with virtual simulation, MR simulation was found to yield results closer to the real experiment.

7 CONCLUSIONS

A generic MR framework has been presented in this paper for creating MR applications. The contribution of our work is a novel behaviour-based interaction scheme that enables real and virtual entities to physically participate in interactions. We have demonstrated the use of our MR framework in building robot simulations, giving the developers the flexibility of virtualising robot and environmental components for cost and safety reasons. Sensor-based interaction and contact interactions between entities of varying level of virtualisation have been successfully achieved based on the stimulus-behaviour-response interaction approach. In comparison to previous work where the application of AR or MR to robot development has mostly been limited to visualisation, our framework enables virtual robots, sensors devices, and environmental objects to physically take part in simulation.

The diverse field of MR requires any MR frameworks to be general and extendable to suit different operating contexts. Future work will investigate the scalability of our framework for achieving other forms of interactions, e.g. social interactions such as speech and gesture, that help to create a broad range of MR applications.

REFERENCES

Arkin, R. (1998). *Behavior-based robotics*. MIT press.

Barakonyi, I. and Schmalstieg, D. (2006). Ubiquitous animated agents for augmented reality. In *IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006. ISMAR 2006.*, pages 145–154.

Chen, I., MacDonald, B., Wünsche, B., Biggs, G., and Kotoku, T. (2009a). A simulation environment for OpenRTM-aist. In *Proceedings of the IEEE International Symposium on System Integration*.

Chen, I. Y.-H., MacDonald, B., and Wünsche, B. (2008). Markerless augmented reality for robots in unprepared environments. In *Australasian Conference on Robotics and Automation. ACRA08*.

Chen, I. Y.-H., MacDonald, B., and Wünsche, B. (2009b). Mixed reality simulation for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2009. ICRA'09.*, pages 232–237, Kobe, Japan.

Collett, T. and MacDonald, B. (2006). Augmented reality visualisation for player. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3954–3959.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. (2000). Virtual object manipulation on a table-top ar environment. In *Proceedings. IEEE and ACM International Symposium on Augmented Reality, 2000. (ISAR 2000)*, pages 111–119.

Milgram, P. and Colquhoun, H. (1999). A taxonomy of real and virtual world display integration. *Mixed Reality-Merging Real and Virtual Worlds*.

Nguyen, T. H. D., Qui, T. C. T., Xu, K., Cheok, A. D., Teo, S. L., Zhou, Z., Mallawaarachchi, A., Lee, S. P., Liu, W., Teo, H. S., Thang, L. N., Li, Y., and Kato, H. (2005). Real-time 3d human capture system for mixed-reality art and entertainment. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):706–721.

Nielsen, C., Goodrich, M., and Ricks, R. (2007). Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics*, 23(5):927–941.

Nishiwaki, K., Kobayashi, K., Uchiyama, S., Yamamoto, H., and Kagami, S. (2008). Mixed reality environment for autonomous robot development. In *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA.

OGRE (2009). OGRE 3D : Object-oriented graphics rendering engine. http://www.ogre3d.org.

O'Hare, G., Duffy, B., Bradley, J., and Martin, A. (2003). Agent chameleons: Moving minds from robots to digital information spaces. pages 18–21.

Pentenrieder, K., Bade, C., Doil, F., and Meier, P. (2007). Augmented reality-based factory planning - an application tailored to industrial needs. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007. ISMAR 2007.*, pages 31–42.

Player/Stage (2008). The player/stage project. http://playerstage.sf.net/.

Rogers, Y., Scaife, M., Gabrielli, S., Smith, H., and Harris, E. (2002). A Conceptual Framework for Mixed Reality Environments: Designing Novel Learning Activities for Young Children. *Presence: Teleoperators & Virtual Environments*, 11(6):677–686.

Smith, R. (2008). Open dynamics engine. http://www.ode.org/.