

Mixed Reality Simulation for Mobile Robots

Ian Yen-Hung Chen, Bruce MacDonald
Dept. of Electrical and Computer Engineering
University of Auckland
New Zealand
{i.chen, b.macdonald}@auckland.ac.nz

Burkhard Wünsche
Dept. of Computer Science
University of Auckland
New Zealand
burkhard@cs.auckland.ac.nz

Abstract—Mobile robots are increasingly entering the real and complex world of humans in ways that necessitate a high degree of interaction and cooperation between human and robot. Complex simulation models, expensive hardware setup, and a highly controlled environment are often required during various stages of robot development. There is a need for robot developers to have a more flexible approach for conducting experiments and to obtain a better understanding of how robots perceive the world. Mixed Reality (MR) presents a world where real and virtual elements co-exist. By merging the real and the virtual in the creation of an MR simulation environment, more insight into the robot behaviour can be gained, e.g. internal robot information can be visualised, and cheaper and safer testing scenarios can be created by making interactions between physical and virtual objects possible. Robot developers are free to introduce virtual objects in an MR simulation environment for evaluating their systems and obtain a coherent display of visual feedback and realistic simulation results. We illustrate our ideas using an MR simulation tool constructed based on the 3D robot simulator Gazebo.

I. INTRODUCTION

The tasks expected of robots have grown more complicated, and are situated in complex and unpredictable environments shared with humans. In many cases, the high accuracy and speed required permits little room for errors in robot design. Various offline robot simulation tools have emerged that support high graphics and physics simulation fidelity. They offer valuable insights to problems that are likely to occur in the real world, however, inconsistencies to practical experimentations are unavoidable. As offline robot simulation tools become more accurate, the demand for computational resources increases. It is a challenge for a standard desktop computer to incorporate all sources of variations from the real world for realistic modelling of robot sensory input and motion characteristics, which may require simulation of lighting, noise, fluid and thermal dynamics, as well as physics of soil, sand, and grass as encountered in nature.

On the other hand, real world experiments help to obtain realistic results in later stages of robot development. Nevertheless, some experiments require substantial human resources, equipment and technical support to produce reliable results while ensuring safety. There may be high risk and uncertainty during the transfer of results from offline simulation to the real world, especially for expensive robotic systems.

Mixed Reality (MR) merges real and virtual worlds in a

registered, coherent manner. Elements from the two worlds reside in a uniform space and interact in real time. We present an MR simulation framework that gives robot developers a more flexible approach for performing simulations, enabling them to design a variety of scenarios for evaluating robot systems involving real and virtual components. Our MR robot simulation tool includes a real robot in an online simulation process. The tool simulates virtual resources such as robots, sensors and other objects, and has the ability to observe the effect on the real robot's behaviour. Robot developers may choose components to be simulated and objects to be included from the real physical world.

Consider a simulation of a robot navigation task in agriculture. Vision identifies targets and range sensor data is used to navigate in a dynamic environment. A real target object, such as an apple to be picked or a cow to be tracked, could be placed in a physical environment filled with virtual crops and cattle. Realistic image processing results could be achieved since the primary component is real, and harm to any agricultural objects or to the robot itself could be prevented. Robot developers can evaluate the overall task performance and observe interactions between different robot subsystems, e.g. vision and motion. This simulation can not be achieved by processing pre-recorded video images alone.

MR simulation relieves offline robot simulators from recreating a complete replica of the real environment, since simulation occurs in a partially real world where certain properties, such as noise and complex physics, do not have to be modeled. MR simulation is however not intended to replace existing simulation methods. It is a complementary, additional step for validating the robotic software's robustness before it is deployed. As robotic software is tested using simulation methods closer to the actual real world operation, the risk and cost normally grow larger. However, in an MR simulation, physical robots are exposed to a real world environment, nevertheless, certain interactions can be limited to virtual objects. The robot navigation example mentioned above demonstrates a safe, controlled interaction between the robot and the environment.

During development, there are limitations to real world views available to humans, who cannot sense, reason, or act like mobile robots. Additional textual, graphical and virtual displays are commonly used to help humans understand robots. However, the human may find it difficult to relate the additional information to the real world. MR helps by

presenting physical and digital data in a single coherent display. Virtual information such as maps, sensor data, and internal robot states can be mixed with information gathered from the physical environment and visualised in geometric registration with relevant physical components.

In summary, the contribution of our work is an MR simulation framework that:

- 1) enables integration of virtual resources in the real world for constructing a safe simulation environment.
- 2) provides real time visual feedback of robot and task relevant information using MR visualisation techniques.
- 3) facilitates interaction between robots and virtual objects during simulation.

Section II describes related work. Section III presents our MR Simulation framework. Section IV details our implementation. Section V gives results obtained from experiments. Section VI discusses future improvements.

II. RELATED WORK

MR can be illustrated using a Reality-Virtuality (RV) continuum [1], [2]. The real and virtual environments sit on the opposite ends of the continuum which includes Augmented Reality (AR) and Augmented Virtuality (AV). In this section, a review of the literature on the application of MR in various fields of robotics will be presented.

Existing AR systems overlay visualisations of complex data onto a real world view, for example in teleoperation and monitoring of robots [3], [4], [5], [6]. A view of the robot and environment is synthesized graphically from onboard sensory data, such as camera images, and presented to remote operators, increasing their situation awareness.

AR may also convey robot state information to improve human robot interaction. For example, virtual arrows are overlaid on top of robots to show the robot heading [7]. Bubblegrams help interaction between collocated humans and robots by displaying robot states and communications [8]. Animated virtual characters can express robot states using natural interactions such as emotions and gestures [9].

While AR displays virtual data in a real world, AV places real data in a virtual environment. AV can visualise spatial information of robots in a dynamically constructed virtual environment based on distributed sensory readings [4]. Real time robot sensory data can also be visualised in a pre-constructed virtual environment to detect newly appeared objects [10]. A more advanced AV based MR environment is presented by Nielsen *et al.* [11] for improving users' situation awareness during robot teleoperation. They combine video, map, robot and sensor information to create an integrated AV interface based on Gibson's ecological theory of visual perception [12]. Disparate sets of information are presented in a single display and their spatial relationship with the environment can be easily determined.

Interactions between real robots and virtual objects can be seen in an educational robotics framework [13]. MR is used for presenting robotics concepts to students in MR games such as robot Pac Man and robot soccer. The MR

game takes place over a table-like display where small robots interact with virtual objects displayed on the table in real time. Given available geometric knowledge of all real and virtual objects, interactions such as collision between a robot and a virtual ball can be achieved using simulated physics. A similar technology is used in the Mixed Reality Sub-league of the Simulation League in RoboCup Soccer [14] which involves teams of physical thumb-size robots engaging in soccer matches on a virtual simulated soccer field.

Very few MR visualisation tools are specifically designed for robotic debugging and evaluation. Collett and MacDonald [15] present an AR visualisation system for robot developers. Robot data, such as laser and sonar scans, can be viewed in context with the real world. Inconsistencies between the robot's world view and the real world can be highlighted during the debugging process. Similarly, Stilman *et al.* [16] and Nishiwaki *et al.* [17] create an MR environment for testing of robot subsystems. The environment provides robot developers an AR visualisation of robot states, sensory data, and results from planning and recognition algorithms.

In comparison to previous work on MR for robot development, we treat the construction of the MR environment as a separate problem from visualisation. In addition to visual augmentations of virtual information, we also augment the real physical environment with simulated components which real robots can interact with. Currently there is limited work on MR interaction in robotics. We explore this field and describe a new method for rich interactions between the robot and the MR environment by augmenting the robot's sensing. We avoid environment modifications or the use of expensive equipment, thus making our system scalable to different robot platforms, tasks, and environments.

III. MIXED REALITY SIMULATION

The MR simulation framework includes: 1) client, 2) MR simulation server, 3) virtual world, and 4) real world.

The client program is the application being developed and to be tested in simulation. The MR simulation server handles requests and commands from the client while keeping track of data produced by the two worlds. The data includes geometric information of virtual objects, data sensed by a robot while operating in the real world, and any other available data measured in the physical environment prior to simulation. The real world is essentially the physical environment where the experimentation takes places. The virtual world is a replica of the real world environment but in addition users are able to introduce additional virtual objects to create different scenarios for testing the client. The MR environment is created by the MR simulation server after mixing the real and virtual world.

A. Mixed Reality (MR) Environment

Robot tasks are varied and robot environments are unpredictable; there is no "best" approach to the design of a simulation environment using MR. Robot developers should be given the flexibility of choosing the level of reality for constructing the simulation, depending on the application

and requirements. In some applications a virtual environment saves cost because the consequences of malfunction are too severe, whereas for other applications involving a complex but low risk environment, modelling is unnecessary costly.

We allow robot developers to introduce rich representations of various virtual objects into a real physical environment. These virtual objects include robots, simulated sensors, and other environmental objects. By augmenting the real world with varying level of virtual components, effectively the level of realism is altered. From another perspective, the developer can introduce a complete 3D virtual model of the environment that is overlaid onto the real physical world leaving certain real world objects unmodelled. This gives the impression that real objects are placed in a virtual environment. The level of realism is influenced by the level of augmentation of virtual components to some extent. Certain virtual information will have no effect on the simulation. We allow elements that do not necessarily possess physical forms, such as sensor readings, robot states, way points, trajectories, etc. to be added within the simulation environment. These mainly serve as visual aids that help to improve the user's perception of robot behaviour.

An important design issue is the visual display of the simulation environment. We integrate existing AR and AV techniques while preserving the advantages of both. The ability to present information in context with the real physical environment is a strong benefit of AR. Contributions of AR in robotics have been shown in Section II. Nevertheless, there are limitations when relying on a single AR visual interface. Development of some robot applications must allow users to observe the simulation environment from different perspectives. AR relies on the use of a physical camera to provide images on which visual augmentation takes place, but only from a single view. This is infeasible in large unprepared environments, especially outdoors. This weakness can be compensated using AV techniques. We adopt the ecological interface paradigm proposed by Nielsen *et al.* [11] to create an integrated display of real and virtual information. The AR view of the environment becomes immersed within a virtual environment at a location which spatially corresponds to the physical environment. This enhances the user's global awareness of the entire simulation.

An example simulation display is shown in Fig. 3. Any changes to the simulation environment will be reflected in both the AR and AV view.

B. Mixed Reality Interaction

Our method facilitates interaction between a real robot and virtual objects in the MR environment. The goal is for the robot to perceive virtual objects as if they are part of the physical environment. We first consider the different stages of robot perception: Raw data measurement, Information extraction, and Interpretation. Robots perceive the environment by taking sensor measurements then extracting useful information for mapping, planning, and control. Thus, to enrich a robot's interaction with the environment, we interfere and modify the robot's perception to reflect the

changes we have made to the environment, by *augmenting the robot's sensing* in the very first stage of perception. There are three steps:

- 1) **Intercept** the raw data produced by the real robot sensors and the raw data from the virtual world.
- 2) **Mix** the two data sets of the same type.
- 3) **Publish** the new MR data to the client programs.

Consider a simple obstacle avoidance algorithm. The robot randomly navigates around the environment and avoids obstacles using its laser sensor readings. The sensor readings describe range to the nearest objects and the algorithm commands the robot to turn away if the reading indicates an object is within a maximum allowable distance. Suppose a virtual object is introduced. The laser sensor readings are modified according to the known robot and object poses before publishing the data to the client applications. The robot will now move around the environment as if there is a real obstacle. Robot application developers can observe realistic robot behaviour as the robot interacts with objects that are virtual, controllable, and safe.

IV. SYSTEM DESIGN

It is desirable to exploit and extend existing robot simulation tools, instead of taking the time-consuming process of building a robot simulator from the ground up. An examination of the literature reveals a number of popular robot simulation tools available for research use.

Amongst many popular 3D robot simulators such as USARSim [18], Webots [19], and the Microsoft Robotics Studio simulator [20], we chose to build our MR robot simulator using Gazebo [21], developed by the Player Project [22]. Gazebo is a 3D robot simulation tool widely supported and used by many research organisations. It is open source, modular, highly modifiable, and has independent rendering and physics subsystems which facilitate the integration of MR technology.

A Mixed Reality Robot Simulation toolkit, MRSim, has been developed and integrated into the Player/Gazebo simulation framework to demonstrate our concept of MR robot simulation.

A. Player/Gazebo Overview

Player [23] is a socket based device server that provides abstraction to robot hardware devices. It enables distributed access to robot sensors and actuators and allows concurrent connections from multiple client programs.

Gazebo is a multi robot, high fidelity 3D simulator for outdoor environments. Gazebo is independent of Player but interface to Player is also supported using a Player driver (GazeboPlugin) to allow simulation of Player client programs without any modifications to the code. Physics is governed by the open source dynamics engine ODE [24] and high quality rendering is provided by the open source graphics rendering engine OGRE [25]. Controllers in Gazebo are responsible for publishing data associated to simulated devices and Player client programs can subscribe to these devices the same way as they would to Player servers on real robots.

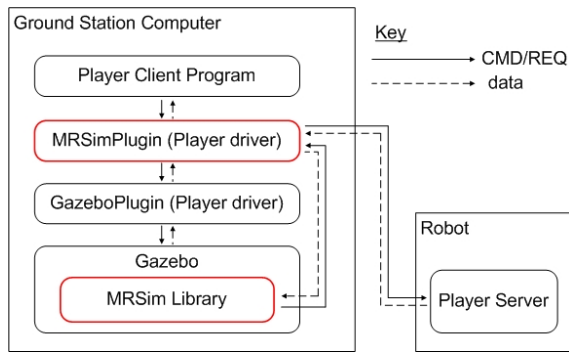


Fig. 1. Integration of MRSim into Player/Gazebo simulation.

B. MR Robot Simulation Toolkit

MRSim is a toolkit that is independent of Gazebo and uses its own XML file for configuring properties of robots and their devices. It integrates well into the Gazebo framework to provide MR robot simulation.

In the case of MR robot simulation, MRSim plays the role of the MR simulation server responsible for tracking the states of the two worlds. The physical environment where the real robot performs its tasks is the real world, and the virtual world is created by Gazebo. Modifications to the components and dataflow in the Player/Gazebo simulation process are shown in Fig. 1. Two new components from the MRSim toolkit are added in the overall simulation process. MRSim consists of 1) MRSimPlugin - a Player driver and 2) the main MRSim library which has been integrated into Gazebo. The client program now connects to MRSimPlugin for controlling and requesting data from robot devices.

MRSimPlugin is responsible for combining real world and simulation data to achieve MR interaction. MRSimPlugin performs the three steps: Intercept, Mix, and Publish. For example, real laser sensor readings are augmented to reflect the added virtual objects. First, MRSimPlugin intercepts messages sent by the client program and dispatches them to Gazebo and the real robot. The readings returned can be mixed by taking the minimum between the real and virtual range values for each point in the laser scan. The resulting data is then published. Fig. 2 shows an example of MR laser sensor readings. The MR laser data is displayed using Player's built-in utility, PlayerViewer. It is essentially a Player client which connects to MRSimPlugin and requests sensor readings. The augmented laser sensor data are also visualised in Gazebo using the MRSim library which requests MR data from our own MRSimPlugin. We have applied the same concept to laser, vision, sonar, and contact sensors and the implementations have been tested using Gazebo and other Player compatible simulation tools.

The MRSim library constructs the MR environment and handles MR visualisations. It monitors Gazebo's rendering and physics subsystems and directly makes changes to the virtual world created by Gazebo. The AR interface and AV interface are provided using the MRSim library.

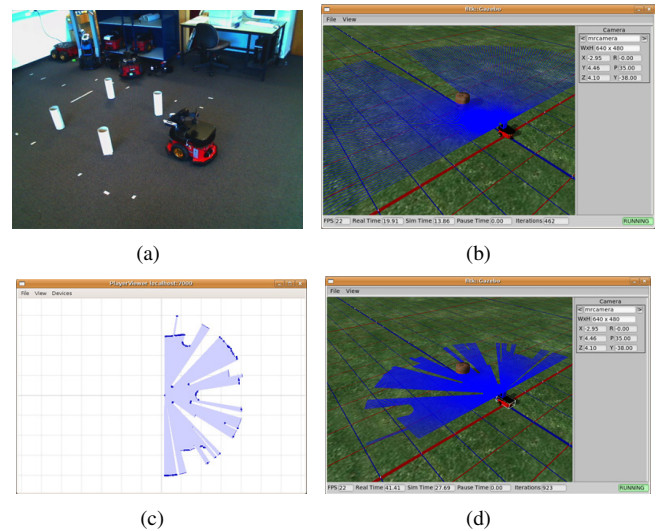


Fig. 2. (a) a real Pioneer robot sensing cylindrical objects in a lab set-up, (b) a virtual robot with its laser sensor readings displayed in Gazebo, (c) the resulting MR laser range readings visualised using PlayerViewer, (d) MR laser visualised in Gazebo.

1) *AR Interface*: One of the main challenges for effective AR is to accurately register virtual objects onto the real world background images so that the virtual objects appear to be part of the physical environment. Moreover, markerless AR techniques are preferred for the computation of the camera pose in order to apply AR in unprepared robot environments.

Our markerless AR system combines feature tracking and object detection algorithms for creating AR that has the ability to recover from tracking failures due to erratic motion and occlusion of the camera [26]. In summary, four co-planar points are tracked in real time and used to derive the camera translation and rotation parameters. During tracking failures, a planar object detection algorithm is applied to detect the planar region previously defined by the four co-planar points and recover the lost planar feature points. At any time the planar region reappears and is detected, tracking can continue and AR is resumed.

2) *AV Interface*: In the AV interface, we augment the virtual world with sensor data captured from the devices mounted on the physical robot. Currently, the AR interface represents a form of camera sensor data, synthesized with virtual information. We place the AR interface a certain distance from the virtual robot, representing the view seen by its real counterpart. The position and orientation of the AR interface is dependent on the pose and offset of the real camera on the robot, which can be pre-configured or adjusted during simulation. A combination of nominal viewpoints is provided through the use of different camera modes to enable users to observe the MR simulation from different perspectives, see Fig. 3.

V. EXPERIMENTS

A. Preliminary Experiment

In MR simulation, any source of variation in the real world that affects the behaviour of the real robot must be

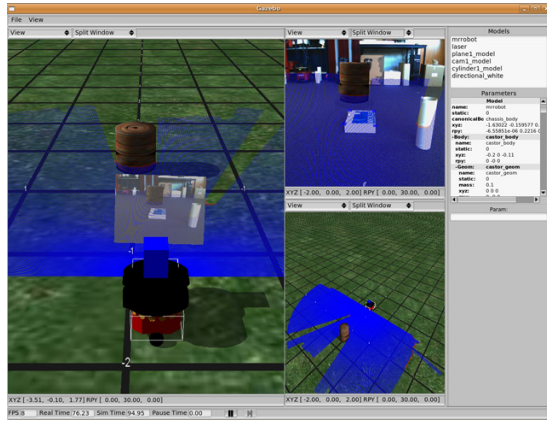


Fig. 3. A typical display of the MR simulator with multiple camera modes. Left: tethered camera mode, Top Right: First person perspective using the AR interface, Bottom Right: fixed camera mode.

correctly reflected in the virtual world. The virtual robot must be an accurate representation of the real one for realistic experimental results and more accurate MR interaction.

To keep the state of the real and virtual robot consistent, we implement a pose correction algorithm that corrects the pose of the virtual robot when the pose difference becomes too large. The algorithm uses the pose estimation output from the markerless AR system to deduce the pose of the real robot then updates the virtual robot pose accordingly. Assuming the offset of the camera to the robot center is known and accurately measured, the error in the pose correction algorithm is narrowed to the error produced by the markerless AR tracking system. The residual error between the actual robot positions and the estimated robot positions was measured to be approximately 0.012 metres.

B. Functional System Validation

In this experiment, we simulate a robot search in a hazardous environment to investigate the new capabilities offered by MR simulation. In real robot exploration tasks, such as robot search and rescue, robots maneuver in unknown environments while exposed to various threats. Most often, extensive testing and experimentation in highly controlled environments and the use of expensive resources are required. MR simulation aims to relieve some of these requirements by using virtual simulated components.

In our simulation, the target object is represented by ARToolkitPlus markers and placed in a lab environment. The robot must navigate using an onboard laser rangefinder and slowly approach the target object when found. The MR environment consists of virtual hazards that are potential threats to the real robot. These include a virtual robot, fire, a barrel, and a small wood pallet. Real objects, such as boxes of different sizes, are also placed in the MR environment to represent obstacles. Fig. 4 shows the layout of the simulation environment.

To register the virtual objects into the real world, a planar object on the back wall is tracked to determine the camera pose. Once the tracking is initialised, the client program then

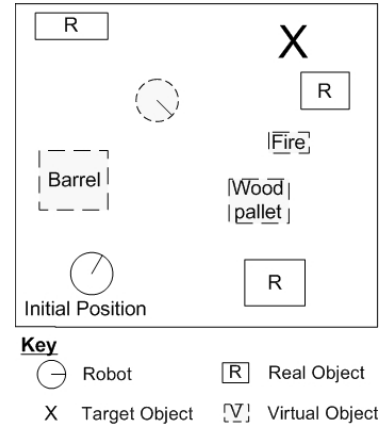


Fig. 4. Layout of the MR environment for simulation of a robot search operation.

connects to the MR simulation server to begin simulation. Screenshots from the experiment are shown in Fig. 5

VI. RESULTS AND DISCUSSIONS

Interaction between the real robot and the virtual objects was successful and the robot navigated in the environment while avoiding real and virtual obstacles sensed by the laser sensor. The use of MR simulation effectively highlighted different causes of damage to the real robot in our experiments, particularly collisions with small virtual objects which can not be detected by the laser sensor. Introduction of virtual objects in a real physical environment allowed rich simulation of resources, which some of these objects can also be very difficult to emulate or recreate in real world experiments, e.g. smoke produced from fire.

The combination of AR and AV views provided effective visualisation of robot information and simulated objects. However, without an external view of the real physical environment for AR visualisation, it is still difficult to relate virtual and real information. This compromise was made in order to scale the system to encompass simulations of robot tasks in large outdoor environments in the future.

The main limitation of our MR robot simulation is the markerless AR component. Currently, visual augmentation of virtual objects will temporarily be lost when the planar object leaves the camera view and resumed when it reappears. During the loss of augmentation, MR interaction still operates but returns less accurate results since the virtual robot pose is not constantly corrected.

VII. CONCLUSIONS AND FUTURE WORK

We have presented a new approach for performing robot simulations based on the concept of Mixed Reality. Robot developers can create scenarios for evaluating robot tasks by mixing virtual objects into a real physical environment to create an MR simulation with varying level of realism. The simulation environment can be displayed to users in both an AR and an AV view. We have demonstrated our ideas using an MR robot simulation tool built on top of Gazebo



Fig. 5. Screenshots of an MR simulation in a robot search scenario. (a) An AV view of the MR simulation environment, (b) Switching to the AR interface; AR is initialised by tracking four feature points (in blue) corresponding to the four corners of the notice board on the back wall; robot starts moving while avoiding real and virtual obstacles, (c) & (d) Robot slowly approaches the target object in the corner; despite partial occlusion of the tracked planar region in (d), AR continues with small jitter.

and facilitated interaction between a real robot and virtual objects.

A thorough comparative evaluation of the MR Simulation needs to be conducted to fully identify its benefits and limitations with respect to common practices in robot simulation, e.g. pure virtual simulation, and real world experiments. The working area of the AR system also needs to be extended in order to apply AR in a wider range of robot applications.

In the near future, we also plan to investigate the use of MR robot simulation to minimise the costs and risks for aerial robot tasks, which demand significant resource and safety requirements.

REFERENCES

- [1] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual display," in *IEICE Transactions on Information Systems*, vol. E77-D, no. 12, December 1994, pp. 1321–1329.
- [2] P. Milgram and H. Colquhoun, *A Taxonomy of Real and Virtual World Display Integration*, 1999.
- [3] P. Milgram, A. Rastogi, and J. Grodski, "Telerobotic control using augmented reality," in *Proceedings of the 4th IEEE International Workshop on Robot and Human Communication, 1995. RO-MAN'95 TOKYO*, 1995, pp. 21–29.
- [4] P. Amstutz and A. Fagg, "Real time visualization of robot state with mobile virtual reality," in *Proceedings of the IEEE International Conference on Robotics and Automation, 2002. ICRA '02.*, vol. 1, 2002, pp. 241–247.
- [5] V. Brujic-Okretic, J.-Y. Guillemaut, L. Hitchin, M. Michielsen, and G. Parker, "Remote vehicle manoeuvring using augmented reality," in *International Conference on Visual Information Engineering, 2003. VIE 2003.*, 2003, pp. 186–189.
- [6] M. Sugimoto, G. Kagotani, H. Nii, N. Shiroma, M. Inami, and F. Matsuno, "Time follower's vision: a teleoperation interface with past images," *IEEE Computer Graphics and Applications*, vol. 25, no. 1, pp. 54–63, January–February 2005.
- [7] M. Daily, Y. Cho, K. Martin, and D. Payton, "World embedded interfaces for human-robot interaction," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003.*, Y. Cho, Ed., 2003, p. 6.
- [8] J. Young, E. Sharlin, and J. Boyd, "Implementing bubblegrams: The use of haar-like features for human-robot interaction," in *IEEE International Conference on Automation Science and Engineering, 2006. CASE '06.*, E. Sharlin, Ed., 2006, pp. 298–303.
- [9] M. Dragone, T. Holz, and G. O'Hare, "Using mixed reality agents as social interfaces for robots," in *The 16th IEEE International Symposium on Robot and Human interactive Communication, 2007. RO-MAN 2007.*, T. Holz, Ed., 2007, pp. 1161–1166.
- [10] H. Chen, O. Wulf, and B. Wagner, "Object detection for a mobile robot using mixed reality," in *Interactive Technologies and Sociotechnical Systems*, 2006, pp. 466–475.
- [11] C. Nielsen, M. Goodrich, and R. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 927–941, 2007.
- [12] J. J. Gibson, *The Ecological Approach to Visual Perception*. Boston: MA: Houghton Mifflin, 1979.
- [13] J. Anderson and J. Baltes, "A mixed reality approach to undergraduate robotics education," in *Proceedings of AAAI-07 (Robot Exhibition Papers)*, R. Holte and A. Howe, Eds. Vancouver, Canada: AAAI Press, July 2007.
- [14] The RoboCup Federation, "Robocup," February 2008, <http://www.robocup.org/>.
- [15] T. Collett and B. MacDonald, "Augmented reality visualisation for player," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 3954–3959.
- [16] M. Stilman, P. Michel, J. Chestnutt, K. Nishiwaki, S. Kagami, and J. Kuffner, "Augmented reality for robot development and experimentation," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-55, November 2005.
- [17] K. Nishiwaki, K. Kobayashi, S. Uchiyama, H. Yamamoto, and S. Kagami, "Mixed reality environment for autonomous robot development," in *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, May 19–23 2008.
- [18] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "Usarsim: a robot simulator for research and education," in *IEEE International Conference on Robotics and Automation, 2007*, Roma, April 10–14 2007, pp. 1400–1405.
- [19] Cyberbotics, "Webots," January 2008, <http://www.cyberbotics.com/products/webots/index.html>.
- [20] Microsoft, "Microsoft robotics studio," January 2008, <http://msdn2.microsoft.com/en-us/robotics/default.aspx>.
- [21] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004).*, vol. 3, September 28–October 2 2004, pp. 2149–2154.
- [22] Player/Stage, "The player/stage project," January 2008, <http://playerstage.sf.net/>.
- [23] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, June 30–July 3 2003, pp. 317–323.
- [24] R. Smith, "Open dynamics engine," January 2008, <http://www.ode.org/>.
- [25] OGRE, "Ogre 3d : Object-oriented graphics rendering engine," 2008, <http://www.ogre3d.org>.
- [26] I. Y.-H. Chen, B. MacDonald, and B. Wünsche, "Markerless augmented reality for robots in unprepared environments," in *Australasian Conference on Robotics and Automation. ACRA08*, December 3–5 2008.