Using Mixed Reality to Facilitate Education in Robotics and AI

John Anderson and Jacky Baltes
Department of Computer Science
University of Manitoba
Winnipeg, Canada R3T2N2
andersj,jacky@cs.umanitoba.ca

Abstract

Using robots as part of any curriculum requires careful management of the significant complexity that physical embodiment introduces. Students need to be made aware of this complexity without being overwhelmed by it, and navigating students through this complexity is the biggest challenge faced by an instructor. Achieving this requires a framework that allows complexity to be introduced in stages, as students' abilities improve. Such a framework should also be flexible enough to provide a range of application environments that can grow with student sophistication, and be able to quickly change between applications. It should be portable and maintainable, and require a minimum of overhead to manage in a classroom. Finally, the framework should provide repeatability and control for evaluating the students' work, as well as for performing research. In this paper, we discuss the advantages of a mixed reality approach to applying robotics to education in order to accomplish these challenges. We introduce a framework for managing mixed reality in the classroom, and discuss our experiences with using this framework for teaching robotics and AI.

Introduction

Until recent years, robotics was largely relegated to advanced specialty courses in engineering and computer science: the subject was covered as a field of its own, usually from a control perspective, and with some overlap to other fields such as computer vision. More recently, robotics began to be used successfully as a tool for teaching artificial intelligence (AI): as a grounded framework for illustrating search and representation, for example, and as a means of illustrating how AI has been influenced by embodied perspectives. Today, robotics is being touted as an embodied approach and a motivational tool for many levels of computer science and engineering, ranging from educating students in first-year programming (Sklar et al. 2006; Burhans 2007; Chilton and Gini 2007) to delivering physics and math curricula to high school students (Goldman, Eguchi, and Sklar 2004).

While the demands on students differ significantly between these levels, in our experience the core problems in using robotics to introduce any curriculum are very similar, and place corresponding demands on a platform for educational robotics. First and foremost, robotics must be recognized as a highly complex domain, and this complexity must be managed appropriately, or students will simply be overwhelmed. As an example, consider the issue of perception. While it is certainly possible to develop interesting robotics projects using simple perceptual devices (e.g. a single sonar for obstacle avoidance), the sophistication of resulting applications will always be limited without vision. The problem with vision is that it generates an enormous amount of data and requires sophisticated algorithms to deal with even relatively simple issues such as recognizing basic shapes, let alone tracking distances, dealing with noise, and tracking objects over time. It is crucial that vision be abstracted in order to provide manageable but vision-rich domains while not overwhelming students. Whether that abstraction must remain in place over the course depends on the level of the students, and certainly any abstraction used should be able to be removed if desired.

In a similar fashion, the curriculum must also be organized so that a simple core can be built upon, or abstractions removed, allowing a focus on one aspect of robotics at a time, without necessarily requiring the complexity of many other areas to be considered simultaneously. A platform for supporting educational robotics must cater to this approach.

Maintainability and portability are also important issues. Educators can be skeptical about using robotics in their curricula simply because of the perception of additional work maintaining this infrastructure. This maintenance includes setting up and dismantling a platform many times, since classroom or lab space may not be available for dedicated use, as well as calibrating vision, initializing control software, and dealing with uncooperative equipment. With a good platform this can really be just a perception issue, since this work should be minimized through design for portability and maintainability. It should also decrease work on the educator by physically distributing some of it: ideally, students should be able to perform much of the set-up and maintenance of the platform themselves as they become more sophisticated.

Keeping students motivated is also crucial. Robots are always motivating at the start, because of the hands-on nature of the equipment (and to some degree the natural anthropomorphism that occurs when people deal with robots). As

---

things become more complex, however, any part of a platform for educational robotics that introduces a greater sense of creativity or fun can make the difference between students moving forward with a project or completing it at only a minimal level.

One of the elements that is commonly used by many educators to add motivation and a feeling of play are competitive frameworks. While there are many opportunities for students to compete nationally and internationally in robotics competitions (which also serves to emphasize the importance of portability in a platform), small competitions in a classroom setting can be ideal for motivation. In such settings, a platform should support fair comparison of one student or group's work against another. This entails a robustness requirement, since failures at crucial times invalidate comparisons. There are also related issues in using such a platform for research, in that experimental trials should guarantee repeatability as much as possible, within the boundaries of possible equipment failure. This is a very difficult thing to achieve in practice. Even when performing a task as simple as navigating through a set of moving obstacles, these must move *equally* randomly in each trial, for example.

This set of problems creates a particularly challenging environment for an educational robotics platform. In previous work (Anderson et al. 2003), we described a framework for using robotic soccer with global vision as a basis for introducing undergraduates to robotics, which was employed in RoboCup. Subsequently, we described how we use that facility to teach robotics (Anderson and Baltes 2006). While this approach has been used in and outside of RoboCup by ourselves and others (e.g. (Imberman, Barkan, and Sklar 2007)), in recent years we have worked to extend the educational approach to incorporate mixed reality, whereby robots perceive both the physical environment they inhabit as well as a virtual reality provided for them. The interaction between the physical and virtual allows for many more exciting environments to motivate students than would usually be possible in a pure robotics environment, such as a mixture of video games and robotics. It also provides superb support for repeatability in student evaluation and experimental control. The hardware and the particular pieces of software with which we implement this approach allow it to be easily maintained and inexpensively put together, and the manner in which we employ the platform allows for abstractions to be placed to limit students' exposure to complexity. This paper describes the mixed reality approach and how we make use of it in an educational setting.

Overview of Approach

A high-level overview of our approach to mixed reality robotics for education is shown in Fig. 1. The environment a robot interacts with consists of both a virtual layer (provided by a flat-panel LCD) and a physical layer (placed on the LCD). At an abstract level, the same approach is used in the RoboCup Mixed Reality Competition (which itself was adapted from the prior E-League (Anderson et al. 2003)). Our use of this approach in education pre-dates the first Mixed Reality competition, and in our work, all of the components of Fig. 1 are our own (and are available via our laboratory web site). The design of the platform has emerged from the necessity of supporting both these perceptual elements as well as the specific needs for education outlined earlier.

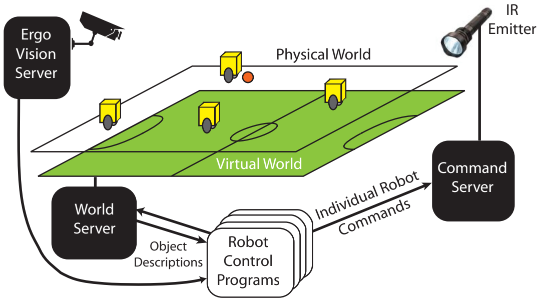

Figure 1: A mixed reality platform using global vision.

Mixed reality is provided through visual perception of both the physical and virtual layers simultaneously. While this does not preclude the use of local vision, we employ global vision as a key design decision to abstract complexity for students. Global vision allows a student to work with abstracted perception (e.g. the X and Y location, orientation and velocity of all items on the playing field) rather than dealing with the complexities of low-level vision. A vision system must still be maintained in terms of setup and calibration, however, which serves as an excellent means to introduce the students to the complexities of vision without requiring them to implement sophisticated algorithms for visual processing. The key in using a global vision approach in this manner is to have a system that is simple enough that it is possible for students to calibrate it themselves.

The virtual layer is presented on a horizontally-mounted LCD. The size of this component affects the other elements of the system, partly from a budgetary standpoint (since not every high school school can afford a 40" LCD screen, for example), but also from the standpoint of physical constraints. A smaller field size, for example, requires smaller robots or fewer of them to leave room to move, while the size of the field and the size of the robots determine the resolution required for the camera providing vision.

Just as a robot can perceive the physical and virtual realities in this approach, it can affect objects in either domain. We assume that robots this small cannot carry significant local computation (though this is evolving), and thus will be controlled by external programs. While this control can be effected in any fashion, we employ infrared communication in our work. An infrared communication device is robust, unaffected by electromagnetic interference from other forms of wireless communication, and in the case of supporting multiple fields, can be easily isolated from the interference of other fields by curtains. The serial infrared communication mechanism we currently employ is IRDA-based, in order to supply enough bandwidth to support larger numbers of robots (Anderson et al. 2008), and is the only custom-

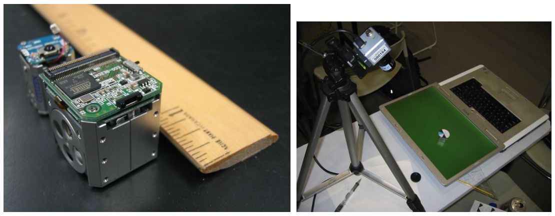

Figure 2: Micro-Robots for Mixed reality: left, Citizen Eco-Be, v1 and v2; right: using a 17” laptop as a playing field;.

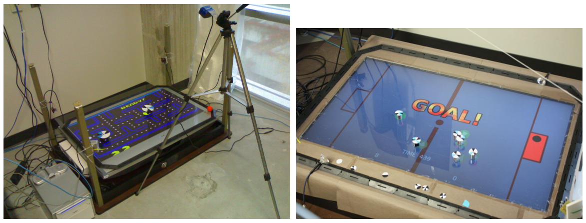

Figure 3: Mixed Reality Applications: left, Pac Man; right: Hockey.

built hardware we use in our platform.

The robots themselves may be very elaborate or extremely inexpensive. Fig. 2 (left) shows the former alternative: the Citizen Eco-Be (version 1 and 2, left to right) Version 2 has an improved microcontroller and a separate controller board that can be expanded with custom components such as local vision. While these are not expected to be available to any high school, we have also employed much simpler robots. Figure Fig. 3 (left) illustrates an undergraduate-built application (Pac Man) using simple IR-controlled toy tanks, which are available cheaply from any department store. The larger size requires a much larger playing area (a 40” LCD, while the Citizen robots can be used on anything as small as a laptop (Fig. 2, right). Vision can be provided by anything from a $20 webcam or security camera, up to high resolution cameras.

Software Support

Our application of this approach to education relies on a number of pieces of software that make the approach manageable for students, also illustrated in Fig. 1 (each of these is downloadable from our laboratory website). These pieces of software have been used by the authors for educational application for the last five years. Many originated in the implementation of the RoboCup U-League (Anderson et al. 2003) (later renamed the E-League, an entry-level league designed to be accessible to high schools and undergraduates). The intent is that students can write individual agent control programs (software agents that control individual robots), which take perception at a high level and produce high-level commands for robots to execute. This permits (but is not limited to) a focus completely on robot control, with all other issues abstracted away, which in turn allows the approach to be used even by the most novice students.

The ability to make this abstraction relies on a sophisticated global vision server, *Ergo* (Anderson and Baltes 2007). Ergo has a number of features that make it ideal for an educational environment: because visual frames are interpolated to an overhead image, the camera can be set at any convenient angle, and because the system relies on background differentiation as the major means of recognizing objects, it operates under varying lighting conditions and requires little recalibration. The system also requires no predefined colors, further enhancing robustness under lighting variation compared to other vision systems, and requiring little set-up time. The instructor or lab supervisor is thus freed to spend more time assisting students and far less keeping the system operating. If vision is not completely ignored in the curriculum, it is our experience that even high school students can easily learn to calibrate the system themselves, allowing greater student independence. The calibration process is also an excellent means of covering some of the basics of vision if this is desired.

As robots move across the environment, the vision server picks up both physical and virtual elements in the camera’s field of view. Those elements that Ergo has been informed are of interest (e.g. patterned paper hats that identify robots and their orientation, and other uniquely-marked objects such as a ball, which are described as items of particular sizes and shapes in a configuration file) are tracked. The vision server transmits the position, orientation and velocity of all objects that were found over Ethernet, which is the basis for providing visual information to students’ control programs. The messages are transmitted in ASCII via UDP broadcast in a single package. The camera and computational hardware currently provide approximately 30 frames per second of data in this format, allowing the students’ control programs to formulate a picture of the world they affect. Note that items visually tracked by the vision server may be physical (i.e. a real ball) or virtual (a picture of a ball on the screen) - the visual image is the same to the vision server.

At the same time this physical perception occurs, a world server describes the state of the virtual world to the students’ control programs, allowing objects to be tracked outside of physical vision (e.g. the location of the soccer goals, game power-ups, or other elements that exist only virtually). This allows the potential to have difficult-to-perceive items tracked outside of the vision server if this is desirable.

Collectively, the students’ robot control programs take this information and individually make choices for action (the set of these potential choices is obviously dependent on the particular robots employed). Control programs transmit their action choices asynchronously via ethernet, and these commands are batched by a command server into packets that are transmitted via IR to all robots on the field. To affect the virtual world, these same commands are also communicated to the World Server so that the effects of the robot on the virtual world can be calculated and displayed. Note that because information to and from control programs are entirely ethernet-based, student’s work can be spread over as many machines as is convenient, or run on a single machine of sufficient power.

Setting up this platform for a given environment involves providing Ergo with descriptions of the objects to track, implementing the physics necessary in the world server for altering the virtual world and its display, and developing agent control programs. In our coursework, only the latter is performed by students, but developing robot environments could also be used as a creative element in any course.

A Mixed Reality AI-Based Robotics Course

While some of the advantages of mixed reality, such as student motivation, portability, and ease of maintenance can be seen from the platform itself, the method in which the platform is used is just as important in terms of making robotics manageable to students. As an illustration of how we employ mixed reality in education, this section describes our experiences with applying this technology in a small class size (15-20 students) fourth-year robotics course, taught from an AI perspective. The elements of this course, however, could easily be used to illustrate the basic principles of AI in third year, or more basic principles of computer science at lower levels. When the approach is properly applied, students can be doing hands-on robotic work from the very beginning, and remain motivated while gradually being exposed to the complexities involved in robotics.

We organize our course work around a series of groupwork assignments that culminate in a large application. The choice of subject matter is obviously one area in which student motivation can be greatly enhanced. The mixed reality approach is nicely amenable to many games, for example, simply because there is a usually a physical reality to any game as well as issues where a pure physical implementation would be difficult for students. We are currently working with hockey, for example (Fig 3, right), where physical players manage a virtual stick and puck. This allows students to implement the game using simple wheeled robots while ignoring the complexities of physical arms that would be required for stick-handling. The virtual stick can shoot a virtual puck in such a way that it slides on the playing surface, or can be raised above it (simulating a third dimension in the virtual reality), and cannot be hit by opposing players when raised.

Many video games are also well-suited to the mixed reality approach, because they have elements that are difficult to manage physically. The Pac Man game illustrated in Fig. 3 (constructed in one offering of this course) is an example of this type of problem, since consuming the dots is an element of the game which would be difficult to manage physically. When considering an application, the work it incorporates should easily be broken into manageable pieces, or possibly be partly pre-constructed for younger students. We find that the previous exposure to video games that is common among students serves as a great motivator in these applications. Once students create a basic Pac Man controlled by a robot, for example, they quickly wish to improve the behavior of the system so that it better approximates the game they are used to. The positive motivational potential of introducing video games to robotics has been noted previously (Dickenson et al. 2007), but this is far easier to achieve in a mixed reality environment.

We also find games to be a good choice because they adapt well to competitions, which we find to be another strong motivational factor for students. In any course employing robotics, no matter what the level, we find some element that can be run in a competitive manner. This provides a quantitative means of evaluating students work, but also adds an atmosphere of excitement, and drives students much more than simply turning in working code. Competitions also provide a good basis for learning the concept of parsimony, since students quickly see that overly-elaborate solutions may not be worth the time taken to construct them in terms of delivering greater functionality.

In terms of this particular course, we begin with ideas that will scale up well to projects (e.g. the Pac Man game), and choose elements that can be done in a relatively isolated manner, allowing a layered approach. We try to use more than one sample domain, chosen so that code can easily be ported to the final project so the students see how interconnected problems in robotics are. For example, a path planner for soccer is easily employed in Pac Man, even though elements of the game are very different. For a course at a much lower level, some of the student work we describe below can be provided directly, or in the form of a skeleton. Simpler examples that require the same techniques in robot control could also be employed (e.g. the more basic movements used by (Burhans 2007; Martin 2007; Chilton and Gini 2007; Baltes and Anderson 2005) that are typically used to show sequence, decision-making, and iteration control structures). Even something as simple as turtle graphics-based applications using robots can be used to bring the motivation of mixed reality to young students.

To present a course in robotics from an AI perspective, we begin by covering elements of computer vision, although as the previous section has illustrated, this issue can be removed completely while still being able to use this platform. The goal at this level is for students to understand some of the basics of computer vision (e.g. color models, edge detection, perspective geometry) while they learn to use the environment described above in a laboratory setting. They then write an interface to control the robots manually, while learning basic control algorithms (e.g. PID controllers, Egerstedt's look-ahead controller (Egerstedt, Hu, and Stotsky 1998)) in class. To embody these in an agent architecture, students are introduced to the basic elements of reactive agents and the ideas behind behavior-based approaches (Arkin 1998). The students' first major assignment, which will ultimately form the core of their complete project, is a path tracking agent that will allow the robot to follow a pre-specified path. In order to represent a path, points on the playing field are marked (by spots on the virtual plane, which are then perceived by the vision server), and the agents are responsible for internally representing a path between these markers and issuing appropriate motor control commands to follow that path, using their own position as reported by the video server for feedback. To evaluate this work, we have a racing competition, performing multiple timed laps of an irregularly shaped virtual track, as shown in the left side of Fig. 4. The concept of obstacle avoidance is then added by considering poten-

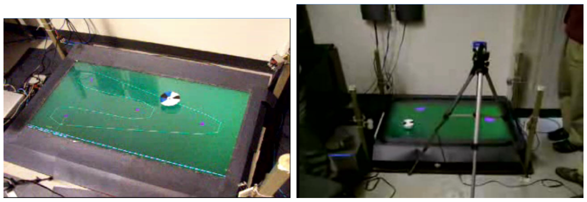

Figure 4: Left: path following competition; Right: avoiding moving obstacles while path planning.

tial fields (Arkin 1998), and following a path while avoiding fixed obstacles that are altered with every new lap.

During this time, we cover sophisticated path planning methods and representations in class, such as skeletonization methods (e.g. Voronoi diagrams (Latombe 1991)) and cell decomposition methods (e.g. quad-tree decomposition, binary space partitioning (Zelinsky 1992; Baltes and Anderson 2003; de Berg et al. 2008)). The students' second major assignment, a treasure hunt where spots on the field must be visited in a planned order, directly subsumes their previous work, in that paths must be planned and then followed through visual feedback. A competition-based evaluation of student programs here is also much easier to manage (and fairer to the teams) because of the mixed reality environment: since the spots to visit are part of the virtual plane, they can be made to disappear as soon as they are touched, and there is never a question as to whether a robot barely brushed against or missed one of the spots during its travel.

While the students complete this assignment, they learn about methods for making path planning more dynamic (e.g. re-planning). Students apply this knowledge by adding re-planning to their previous robot controller, along with balancing when to re-plan vs. when to rely on the potential-field based obstacle-avoidance already implemented. This is embodied in an assignment that requires them to plan a path across the field multiple times, while avoiding randomly-moving obstacles on the virtual plane. This assignment also illustrates the power of mixed reality for consistent quantitative evaluation and repeatability. Since the obstacles are virtual, they can be started at precisely the same point and move with an identical degree of randomness for each trial, ensuring fairness in evaluation. This is extremely difficult to do with purely physical robots without having to repeat trials due to failure or inconsistency in the moving objects used as obstacles, and attempting to deal with issues of battery drain over these necessary repetitions. This also shows the potential for this approach in supporting repeatability in experimentation. While we are not focussing on research issues in this paper, it can be seen that many of the same strengths the mixed reality approach brings to education are research strengths as well.

During this time, larger agent architectures are covered in class (e.g. subsumption, BDI, hybrid architectures). Once these have been covered, students can build more complete sophisticated agents that can participate in applications, such as Pong or Pac Man. Implementing a game as sophisticated as the latter, or other applications involving multiple agents (soccer, hockey) is typically the capstone of the course. Different groups of students may choose different capstones, and supporting these choices during development and evaluation also serves to illustrate the flexibility of the mixed reality approach in terms of education. The entire system is easily altered in only a few minutes time to move from working with one application to working with another, as opposed to the time it would take to reset a new completely physical field, recalibrate vision based on robot markers, etc. This is extremely important both in terms of managing instructor time, allowing students more options in terms of choosing work they would most enjoy, and dealing with limited space and limited equipment.

While this is only one particular course, the principles of employing mixed reality apply irrespective of the level. For an introductory CS class (or even a high school class) introducing programming concepts, vision can be left as a black box, and a number of simple behaviors for basic motion and obstacle avoidance could be provided, to be expanded upon by the students. Previous work has illustrated that even twelve year olds can easily grasp the concepts of control structures using robots, and more interestingly, ideas such as parallelism are very naturally seen through these mechanisms (Baltes and Anderson 2005). For an earlier undergrad course in AI, more constrained examples such as the robotic maze-solving advocated by (Duke, Carlson, and Thorpe 2007) (which has elements of Pac Man but removes dealing with antagonist agents and the strategy required for traversing the entire field) could be easily adapted to a mixed-reality setting. Here, the addition of mixed reality would bring the same flexibility, motivation, ease-of-maintenance, and consistency-of-evaluation we have described above.

Moreover, even within the choice of application, there is a great deal of flexibility in terms of adapting to different levels of student sophistication. The boundary between physical and virtual can be adjusted in any domain, for example. In soccer, students can play with a ball on the virtual field with simplified simulated physics, or a real ball for greater unpredictability in perception and physics. The latter would require significantly greater sophistication in the control agent for a successful performance.

Discussion

In this paper, we have illustrated the use of mixed reality for the purposes of applying robots to student education. While the specific course described in detail was a higher-level robotics course, we have attempted to show that the difficulties associated with employing robotic technology and the advantages of using mixed reality are similar in any course applying robotic technology. The approach we use has the ability to pre-define partial solutions, allowing students at lower levels to use the same technology without the same implementation abilities.

From a student's standpoint, our approach to mixed reality allows for abstraction (e.g. in vision) to hide complexities that are currently beyond the student's abilities. The gradual increase in sophistication leads to a layered approach for assignments, which in turn leads to choices made on previous assignments impacting later code. This helps students focus on the overall design process, and ultimately serves as an excellent lesson in practical software engineering that is difficult to bring to students in any other way.

The platform itself, and most significantly the vision server, is also student-maintainable after a short introduction, allowing students greater independence. Students can work on different projects and switch between environments quickly, allowing greater variation in the types of projects that can be supported with the same equipment. Finally, the system allows fast-moving, vision-rich environments that are exciting and motivating to students.

From an instructor's standpoint, the platform is quick to set up and modify, allowing time to be better spent on instruction. The system is portable (and thus also amenable to regional and national competitions), is based entirely on open-source software, and can be assembled with a limited budget (while at the same time not being restricted to cheap equipment). In the same fashion that abstractions prevent students from being overwhelmed with complexity, the nature of this approach also limits complexity from the instructor's standpoint, and allows the instructor's efforts to be more strongly focussed on one topic at a time.

Because switching worlds is simple, different environments can be used in the same assignment, providing the instructor with a means of getting students to think generally. This is especially valuable in robotics, where the temptation to students is to create solutions that would only ever work for a single problem.

Issues of control and repeatability, important in using this platform in research, also assist in providing consistency in evaluating students' work. Finally, the motivation that students experience is also infectious to the instructor. When the chores involved in maintaining a system are minimized, employing robotics becomes a much more reasonable alternative for delivering many possible curricula.

Evidence of the success of this approach is mainly anecdotal at this point. In the two years we have been using this approach, students consistently comment that the use of mixed reality is exciting and motivates them to develop more sophisticated applications. Students' general comments imply that they appreciate the greater focus on interesting higher-level issues that mixed reality allows, compared to prior years where there had to be a much stronger focus on low-level programming for motors.

References

Anderson, J., and Baltes, J. 2006. An agent-based approach to introductory robotics using robotic soccer. *International Journal of Robotics and Automation* 21(2).

Anderson, J., and Baltes, J. 2007. A pragmatic global vision system for educational robotics. In *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Anderson, J.; Baltes, J.; Livingston, D.; Sklar, E.; and Tower, J. 2003. Toward an undergraduate league for RoboCup. In *Proceedings of the RoboCup 2003 Symposium*.

Anderson, J.; Baltes, J.; de Denus, M.; Allen, J.; and Troniak, D. 2008. Keystone mixed reality 2008. In *Proceedings of RoboCup-2008 (Team Description Papers)*, 8 pp.

Arkin, R. C. 1998. *Behavior-Based Robotics*. Cambridge, MA: Cambridge: MIT Press.

Baltes, J., and Anderson, J. 2003. Flexible binary space partitioning for robotic rescue. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*.

Baltes, J., and Anderson, J. 2005. Introductory programming workshop for children using robotics. *International Journal of Human-Friendly Welfare Robotic Systems* 6(2):17–26.

Burhans, D. 2007. A robotics introduction to computer science. In *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Chilton, J., and Gini, M. 2007. Using the AIBO in a CS1 course. In *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

de Berg, M.; Cheong, O.; van Kreveld, M.; and Overmars, M. 2008. *Computational Geometry: Algorithms and Applications*. Heidelberg: Springer-Verlag.

Dickenson, B.; Jenkins, O. C.; Mosely, M.; Bloom, D.; and Hartmann, D. 2007. Roomba pac-man: Teaching autonomous robotics through embodied gaming. In *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Duke, D.; Carlson, J.; and Thorpe, C. 2007. Robotics in early undergraduate education. In *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Egerstedt, M.; Hu, X.; and Stotsky, A. 1998. Control of a car-like robot using a dynamic model. In *Proceedings of the IEEE Conf. on Robotics and Automation*, 3273–3278.

Goldman, R.; Eguchi, A.; and Sklar, E. 2004. Using educational robotics to engage inner-city students with technology. In *Proceedings of the Sixth International Conference of the Learning Sciences*, 214–221.

Imberman, S.; Barkan, A.; and Sklar, E. 2007. Entry-level soccer for undergraduates. In *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Latombe, J.-C. 1991. *Robot Motion Planning*. Boston: Kluwer Academic Publishers.

Martin, F. 2007. Real robots don't drive straight. In *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Sklar, E.; Parsons, S.; Azhar, M. Q.; and Andrewlevich, V. 2006. Educational robotics in Brooklyn. In *Proceedings of the AAAI-06 Mobile Robot Workshop*.

Zelinsky, A. 1992. A mobile robot exploration algorithm. *IEEE Trans. on Robotics and Automation* 8(6):707–717.