Just Add Wheels: Leveraging Commodity Laptop Hardware for Robotics and AI Education

Jonathan Kelly, Jonathan Binney, Arvind Pereira, Omair Khan and Gaurav S. Sukhatme

Department of Computer Science
University of Southern California
Los Angeles, California, USA 90089-0781
{jonathsk, binney, apereira, omairkha, gaurav}@usc.edu

Abstract

Along with steady gains in processing power, commodity laptops are increasingly becoming sensor-rich devices. This trend, driven by consumer demand and enabled by improvements in solid-state sensor technology, offers an ideal opportunity to integrate robotics into K–12 and undergraduate education. By adding wheels, motors and a motor control board, a modern laptop can be transformed into a capable robot platform, for relatively little additional cost. We propose designing software and curricula around such platforms, leveraging hardware that many students already have in hand.

In this paper, we motivate our laptop-centric approach, and demonstrate a proof-of-concept laptop robot based on an Apple MacBook laptop and an iRobot Create mobile base. The MacBook is equipped with a built-in camera and a three-axis accelerometer unit – we use the camera for monocular simultaneous localization and mapping (SLAM), and the accelerometer for 360 degree collision detection. The paper closes with some suggestions for ways in which to foster more work in this direction.

Introduction

Robotics projects are an exciting way to learn about many aspects of engineering, from software systems to mechanical design. The University of Southern California, our institution, has acknowledged this potential with the recent introduction of a new multidisciplinary robotics course for engineering freshman (Matarić 2007). Further, educators are now recognizing that robotics can be used to motivate students who may not otherwise choose computer science or engineering as their primary college degree program (Blank 2006). Our opinion is that robotics can have an even greater positive impact, if more students have access to the appropriate hardware. Specifically, it is important for students to 'get their hands dirty' with real robots.

We see several main requirements for a robot to be a useful educational tool. The robot must be mobile, and it must have sensors which are capable enough to be used for common robotic tasks such as localization, mapping, and collision detection. It also must have sufficient processing power for these activities, and must be reliable enough to allow students and teachers to focus on algorithms and experimentation rather than on hacking the hardware. We believe that it is possible to leverage the ubiquity of laptop computers to fulfill the above, by *using a student's own laptop as part of a capable robot*. With the addition of servo motors, a motor control board, and a pair of wheels, a laptop can become a high-performance experiment testbed.

We have two immediate goals in this work: first, to show that a useful, laptop-centric robot system can be built, and second to assess the performance of such a hardware/software platform. Our discussion is focused on a software package that we are developing for Apple MacBook line of laptop computers. The MacBook is an Intel-based machine that is able to run Microsoft Windows, Mac OS X, and Linux. As such, it represents a flexible choice for our preliminary demonstration. For 'wheels', we use the iRobot

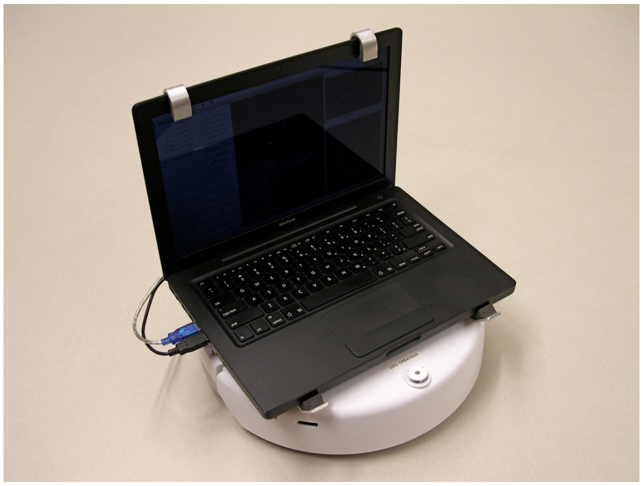

Figure 1: Apple MacBook laptop mounted on top of an iRobot Create. The MacBook provides exteroceptive and interoceptive sensors (a camera and an accelerometer, respectively), while the Create provides several proprioceptive sensors and the drive hardware for mobility. Together, they form a very capable robot platform.

Create, a low-cost mobile base available for $130 US. Together the MacBook and Create provide a fully-functional robot platform.

The remainder of the paper is organized as follows. We discuss our motivation for the project below. We then describe our preliminary laptop software system for visual navigation and collision detection. Next, we present results from several experiments which demonstrate the capabilities of the platform. Finally, we close by offering some important directions for future work in this area.

Motivation

Students learn best by doing, and hands-on robotics assignments are an ideal way to apply the theoretical knowledge gained in a typical undergraduate engineering program. Likewise, we believe that incorporating robotics into the standard K–12 curriculum will encourage students to pursue math and science programs in college. However, there are numerous barriers to the widespread adoption of a robotics curriculum, at both the K–12 and undergraduate levels. Among these are a lack of teacher training, suitable educational resources, and affordable robot platforms (Matarić, Koenig, and Feil-Seifer 2007). We address the last issue in this paper.

Maximizing the Potential of Robotics for Students

Recent work has suggested that the sensor suite available on a robot is likely to have the most significant curricular and financial impact on an undergraduate's experience with robotics (Dodds et al. 2006). If robotics is to have its greatest pedagogical effect, then it is critical to make capable and reliable sensors available to students at a reasonable cost.

One way to do this is to leverage the sensors available on hardware that a student likely already owns; laptop computers are increasingly prevalent in education settings such as high schools and college campuses. For example, a 2007 ECAR survey of undergraduates at 103 two-year and four-year colleges and universities in the United States found that 73.7 percent of students now own laptops. Further, 64.0 percent of entering freshmen at the four-year institutions own laptops which are less than one year old (Salaway, Caruso, and Nelson 2007).[1] These statistics indicate that laptops have reached a very high level of market penetration among college students, and that new models are adopted rapidly by students at the start of their degree programs. If this trend continues, it is very likely that, in the next several years, the majority of senior high school and beginning undergraduate students will own a recent laptop, complete with several on-board sensors.

A second way to ensure that robotics has the greatest possible impact is to provide a free and open source software suite that is able to take advantage of available sensors. We describe an early implementation of this idea in the next section. Although we focus on robotics here, robotics projects can very effectively serve as a foundation for introducing many broader problems in artificial intelligence, and thus the approach we propose is also applicable in the context of AI.

Leveraging Laptop Hardware

From a robotics perspective, the modern laptop is rapidly becoming a well-equipped *sensor platform*. Many off-the-shelf laptops already provide a built-in color camera and microphone. At least three different manufacturers now offer models with built-in accelerometers.[2] Tomorrow's machines may incorporate GPS receivers, touch interfaces, or other novel sensing devices. Further, the processing power available on board is comparable to or better than the most advanced desktop machines from only two to three years ago. This means that many laptops are capable of running advanced vision and artificial intelligence algorithms. The remaining components needed to build a complete robot, i.e. a mobile base and motor controller, can be purchased for a combined retail cost of less than $150 US.

Using this basic platform, students can learn about sensing, filtering, estimation, control theory, computer vision, and many other topics. By collaborating with their peers, they have the opportunity to work on multirobot systems, as part of cooperative or competitive games, for example. Finally, the use of real hardware allows students to observe the effects of both systematic and random noise on the performance of their robots. Depending on student experience, interest and expertise, these topics may be presented at a non-technical level (with fully-functional software provided), as assignments (where students write small pieces of code or configure existing modules), or as research projects (where students implement a complete piece of the system).

In one scenario we envision, students would develop and test software in simulation on their individual laptops. Upon arriving at school, each student would sign out an available mobile base for use during the class period. If a school was unable to provide a sufficient number of base units, they could be assigned in a rotating manner or by team. Alternatively, if adequate resources were available, students could borrow the hardware for the term or semester, or purchase the base themselves if desired. In this way, we imagine a future in which initiatives such as the One Laptop Per Child (OLPC) (Perry 2007) program could become One Robot Per Child.

There have been several recent efforts to develop low-cost personal robot systems for education. One such project, the result of a collaboration between the Georgia Institute of Technology and Microsoft Research, is Gyro (Blank 2006), a personal educational robot that uses a laptop to run its main control program. The Gyro robot itself is a small, wheeled platform that communicates with the laptop over a wireless link. Another alternative is the ER1, manufactured by Evolution Robotics, which includes an external camera as part of the product package. The ER1 is a mobile chassis designed to be used with an on-board laptop. Both of these platforms fail to maximize the utility of a laptop's built-in sensors, however.

---

[1] The survey involved a total of 27,864 students at 99 four-year colleges and universities and four two-year colleges.

[2] The Apple Sudden Motion Sensor, IBM HD Active Protection System and Acer GraviSense.

(a) (b)

Figure 2: (a) Rear view of the MacBook mounted on the Create. (b) Create chassis with the MacBook removed, showing the aluminum mounting frame.

The LapBot: An Educational Hardware/Software Platform

As stated above, one of our goals is to demonstrate that a capable robot can be built using a commodity laptop. If the on-board sensors are of low fidelity, or are unable to reasonably provide the level of functionality (e.g. bandwidth) required for robotics applications, then this approach will be less compelling for educational use.

As a proof of concept, we have built a prototype laptop robot, or *LapBot*, using an Apple MacBook. The MacBook is a popular laptop that is widely used by students; it comes equipped with an internal 'iSight' color camera and a three-axis accelerometer unit. To carry the MacBook, we use an iRobot Create as a mobile base (shown in Figure 1). The Create shares the majority of its parts with the very successful Roomba robotic vacuum, and is an example of how mass production and economies of scale can put capable robotics hardware within reach of the average consumer. The only additional equipment needed is an inexpensive USB-to-serial converter, which allows the MacBook to talk to the Create's microcontroller. We selected the Create primarily because previous work has shown that its sibling, the Roomba, is robust and reliable (Dickenson et al. 2007; Tribelhorn and Dodds 2007b). It is important to emphasize, however, that the Create is just *one* possible platform choice – other options exist, ranging from custom-built, low-cost solutions to (sometimes significantly) more expensive and capable self-contained mobile robots.

The MacBook is held in place on top of the Create by a lightweight aluminum support frame (Figure 2(b)). This frame holds the display portion of the laptop clamshell, with the iSight camera at the top, rigidly upright and stable while the robot is moving. Without this support, the display hinge is not stiff enough to prevent the display from oscillating, which makes visual feature tracking more difficult.

Our main control application uses Player, the well-known open source robot device server (Gerkey et al. 2001; Collett, MacDonald, and Gerkey 2005), to interface with both the laptop sensors and the Create's microcontroller. Player is available free of charge for Linux[3], and can be used with Gazebo, a companion 3D robot simulator, allowing students to perform experiments without the Create chassis. A block diagram for our software architecture is shown in Figure 3.

The proof-of-concept demonstration involves two tasks: visual simultaneous localization and mapping using an open source monocular SLAM package, and collision (bump) detection. We describe both tasks in more detail below.

Visual SLAM with a Built-In Camera

One of the most useful and important capabilities for a robot is to be able to build a map of its surroundings, and also to determine and track its own position within that map. This process is called simultaneous localization and mapping (SLAM), and is an extensively studied and very active area of robotics research. When a robot detects known landmarks with its sensors, it can use those landmarks to calculate its own position. If the robot does not have a map of the environment, then it must determine both the positions of the landmarks and its own position, which is a challenging estimation problem.

Many solutions for SLAM are probabilistic in nature: given an error distribution for the data from its sensors, as well as a model of its own motion, the robot computes the most likely map of its environment, and its most likely position within that map. Understanding such SLAM algorithms requires knowledge of probability, statistics, and linear algebra, among other subjects. Although these topics go beyond K–12 math, we believe that students can still acquire a strong intuitive understanding of what SLAM does by seeing it in action.

In order to perform SLAM, robots used by the students must have a software implementation of a particular SLAM algorithm, as well as sensors which provide the necessary input. Past approaches have used laser rangefinders or stereo vision. Neither of these modalities is entirely suitable for a low-cost educational platform. Laser rangefinders cost on the order of several thousand dollars, and can be quite large

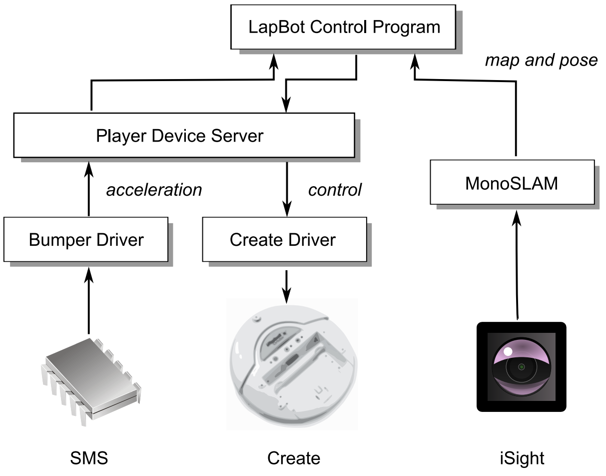

Figure 3: System block diagram.

---

[3] Player will also run under Mac OS X, although changes to OS-specific portions of the driver code are sometimes required.

and heavy. Precision stereo camera rigs are also expensive, and require careful calibration before use.

Fortunately, there has been recent research on single-camera SLAM (called MonoSLAM) by Davison (Davison 2003). This work involves performing SLAM using a monocular camera and a simple motion model. As the camera moves, features are detected and matched between frames. By enforcing the geometry of the camera model and the constraints of the motion model, the camera poses can be determined for each point in time (assuming that enough features are in view). The features used are square patches of pixels, chosen using a saliency operator. The motion model is very simple, and says only that the angular and translational velocities are expected to be approximately constant, except for accelerations which are Gaussian and zero mean. Nothing is assumed about the direction of the accelerations because the algorithm does not know where the camera is moving. There are two improvements which could be made to this model in our specific case. First, we have some idea of where the camera is moving because its movement is (at least partially) a result of the control signals being sent to the robot base. Second, the MacBook has a built-in 3D accelerometer which, if added as an input to the motion model, could help to identify translational movements.

The SLAM algorithm assumes that the calibration parameters of the camera are known, but in practice it works qualitatively well if reasonable default parameters are assumed.

Using the Laptop Accelerometer

The MacBook includes a three-axis accelerometer unit, called the Sudden Motion Sensor (SMS) by Apple, which detects large accelerations and strong vibrations (Apple Inc. 2006). If the laptop is dropped, for example, the SMS immediately parks the hard disk drive heads to prevent a head crash from damaging the drive. Although Apple does not provide an official Application Programming Interface (API) for the SMS, it is relatively straightforward to access the device directly and read the accelerometer data. The sensor has a resolution of approximately 250 counts per gravity, and can be sampled at more than 300 Hz.

We repurpose the SMS unit as a bump sensor. An accelerometer-based bump sensor has two advantages over the standard Create bumper: it is an entirely solid-state device, with no moving parts to break or wear out, and it provides 360 degree bump sensing (unlike the mechanical bumper on the Create, which only covers the front of the unit).

Our software module (Bumper) issues an auditory warning when the laptop's acceleration exceeds a predefined threshold value; this usually occurs when the robot runs into an obstacle or is bumped horizontally (by a person or another robot). We typically low-pass filter the accelerometer data (using a simple Gaussian filter) to remove noise before the performing the threshold check. We also subtract the vertical gravity vector from every sensor measurement. A Player driver for the SMS is available from http://robotics.usc.edu/ jonathsk/software.php.

The mounting frame for the Create, shown in Figure 2, places the MacBook at a slight incline relative to the laptop's normal (flat) desktop position. Since the SMS x and y axes are aligned with the plane of the keyboard, it is necessary to determine the laptop to chassis transform to process the accelerometer values correctly (i.e. to subtract the component of gravity acting along each axis). Calibration involves making several measurements with the MacBook attached to the Create, while the platform is not moving. For students, this procedure can serve as a simple introduction to more complicated calibration techniques for other sensors, e.g. camera calibration.

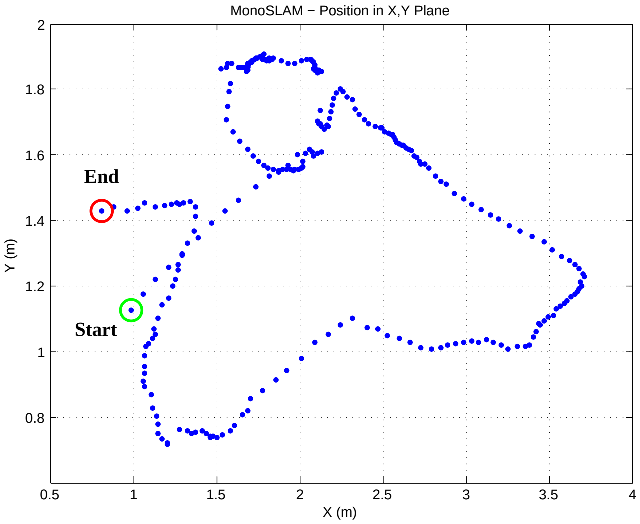

Figure 4: Robot position, estimated by the MonoSLAM algorithm, during an experimental trial in our laboratory. The total length of the trajectory was approximately 9 meters.

Experiments

To demonstrate the capabilities of our LapBot prototype, we performed a series of mapping and collision detection experiments in our laboratory. During the experiments, we drove the platform manually using a wireless joystick, while running Davison's open source MonoSLAM implementation and our Bumper program on the MacBook. The Create has relatively small wheels, and as such, the LapBot operates best indoors on reasonably flat floors.

Laptop SLAM

We ran several tests with MonoSLAM, using the built-in iSight camera to acquire images in real time. To initialize the algorithm, we identified four known image patches that we knew the 3D locations of. These features were at the corners of the USC banner shown in Figure 5. Once running, the MonoSLAM software quickly found these initial features, and within a few seconds added several more features on its own.

Although ground truth was not available, we observed good agreement between our hand measurements of the start and end positions of the robot and the values reported by MonoSLAM. The total length of the trajectory, shown in Figure 4, was approximately 9 meters. We also noted that the MonoSLAM software correctly re-identified old features when they came back into view, indicating that the LapBot ‘knew where it was’.

Bump Sensing

We tested the accelerometer-based bump sensor by driving forward, backward and in a series of arcs of varying radii, while purposely causing the robot to collide with obstacles placed in its path (e.g. a person’s foot). Bumping the LapBot while it is moving typically results in a horizontal acceleration of 0.4 to 0.6 g; we use 0.4 g as our auditory warning threshold. We have found that this value is small enough to detect most bump and collision events, while avoiding false triggers due to uneven floors etc. Figure 6 is a plot of accelerometer data acquired during an experimental trial in which a ‘bump’ was detected.

Discussion and Conclusions

In this paper, we proposed leveraging the sensors and processing power of commodity laptops to build capable robot platforms. Our idea is motivated by the desire to improve the accessibility of robotics, across a range of educational levels. We emphasize that, increasingly, it should be possible for students to take advantage of the hardware they already own, and by so doing maximize the ‘bang for their buck’.

We presented a proof of concept LapBot prototype, built using an Apple MacBook and an iRobot Create. With this prototype, we demonstrated real-time monocular SLAM and accelerometer-based collision detection. An important benefit using off-the-shelf hardware is that the separate pieces of the system have undergone significant development and testing in their own right. Compared with the typical effort required to assemble and test a hobbyist or low-cost robot platform, our approach offers significantly enhanced reliability and faster implementation.

There are many opportunities for future work in this area. Our immediate goal is to develop a standardized software package, available as an out-of-the-box solution for students and educators, that is already customized for specific laptop models. We would also like to evaluate the LapBot in a classroom environment, to gauge reaction from students.

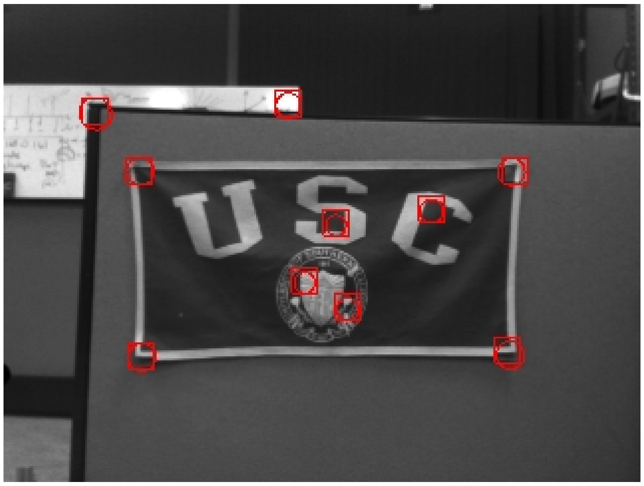

Figure 5: Image frame captured by the MacBook’s built-in iSight camera, showing features identified by MonoSLAM.

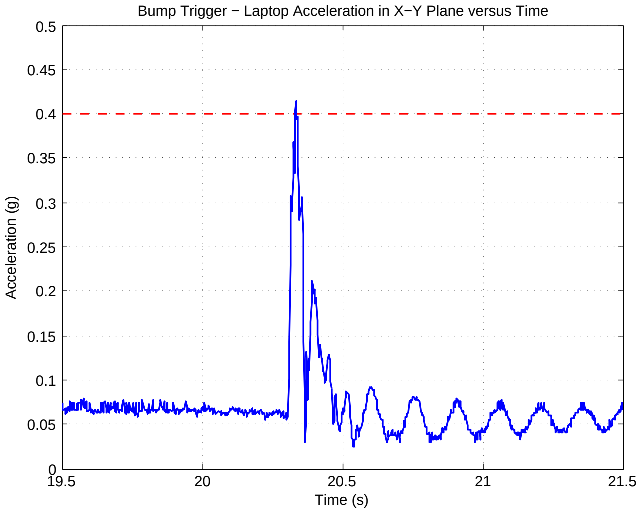

Figure 6: SMS accelerometer data captured over three seconds during an experimental trial. The dashed line indicates the ‘bump’ detection threshold of 0.4 g. In this case, a bump was detected at $t = 20.3$ seconds. The data above has not been low-pass filtered.

As a final thought, we note that the success of the project depends on having open access to laptop sensor specifications. It is important to encourage vendors to publicly release detailed information about the sensors inside their machines. This will save time spent reverse engineering the devices, which can instead be used more productively to develop educational tools.

Acknowledgements

This work was funded in part by a gift from the Okawa Foundation. Jonathan Kelly was supported by an Annenberg Fellowship from the University of Southern California and by an NSERC postgraduate scholarship (PGS-D) from the Government of Canada. Jonathan Binney was supported by a Viterbi School of Engineering Doctoral Fellowship from the University of Southern California.

References

Apple Inc. 2006. Knowledge Base Article 300781: About the Sudden Motion Sensor. [Online] Available: http://docs.info.apple.com/article.html?artnum=300781.

Blank, D. 2006. Robots make computer science personal. *Communications of the ACM* 49(12):25–27.

Collett, T. H.; MacDonald, B. A.; and Gerkey, B. P. 2005. Player 2.0: Toward a practical robot programming framework. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005).*

Davison, A. J. 2003. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'03)*, volume 2, 1403–1410.

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

Dodds, Z.; Greenwald, L.; Howard, A.; Tejada, S.; and Weinberg, J. 2006. Components, curriculum and community: Robots and robotics in undergraduate AI education. *AI Magazine* 27(1):11–22.

Gerkey, B. P.; Vaughan, R. T.; Stoy, K.; Howard, A.; Sukhatme, G. S.; and Matarić, M. J. 2001. Most valuable player: A robot device server for distributed control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*, volume 2, 1226–1231.

Matarić, M. J.; Koenig, N.; and Feil-Seifer, D. J. 2007. Materials for enabling hands-on robotics and STEM education. In *AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Matarić, M. J. 2007. USC ENGR 499 - Engineering Science and Systems: From Humans to Robots. [Online] Available: http://robotics.usc.edu/ maja/teaching/.

Perry, T. S. 2007. The laptop crusade. *IEEE Spectrum Magazine* 44(4):28–33.

Salaway, G.; Caruso, J. B.; and Nelson, M. R. 2007. *The ECAR Study of Undergraduate Students and Information Technology*, volume 6. Boulder, Colorado, USA: EDUCAUSE Center for Applied Research.

Tribelhorn, B., and Dodds, Z. 2007a. Envisioning the Roomba as AI resource: A classroom and laboratory evaluation. In *AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Tribelhorn, B., and Dodds, Z. 2007b. Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'07)*, 1393–1399.