Object Capture with a Camera-Mobile Robot System

BY ROBERT J. WEBSTER III

This article describes a mobile robot/camera system that is both simple and inexpensive enough for an undergraduate engineering student to construct. It provides an excellent first introduction to hands-on robotics, enabling the capture of small objects with the robot. It is scalable and can lead the interested student further into many diverse areas of robotics research. We have carefully designed and arranged components so that the system can work with straightforward algorithms, off-the-shelf hardware, and minimal programming. Creating a working robot system not only introduces the beginning roboticist to many interesting problems in robotics but also creates a testbed to begin exploring them.

Why Build A Mobile Robot System?

Mobile robots are able to do more today than ever before, thanks to recent technical advances and cost reductions. They are often used in situations that are dangerous for humans, such as bomb disposal and detonation [4], rescuing disaster victims (e.g., 2001 World Trade Center) [3], decontaminating highly radioactive environments (e.g., Chernobyl) [1], or even vacuuming your rug [9]. The potential of robots to accomplish such tasks depends on how well they can locate and interact with objects in their environments.

If one has never before attempted a robotics project, building and using a mobile robot may seem prohibitively complex and difficult. However, armed with this article, the beginning roboticist can create a straightforward and inexpensive mobile robot/vision system able to locate and capture objects near it.

System Components and Overview

Figure 1 shows a complete mobile robot/vision system able to identify and capture objects (e.g., tennis balls) in its environment. It is built from the the following components:

- *Camera:* We use the inexpensive HomeConnect webcam from 3Com.
- *Computer:* Any desktop or laptop will work. We used a 400 MHz Pentium II.
- *Robot Vision CAD Software:* A freeware program downloadable from http://www.ccs.neu.edu/home/psksvp/.
- *Small wheeled robot:* We used Mitsubishi's "Sumo-Robot," but alternatives include a small LEGO robot or any other wheeled vehicle available.
- *Capture device:* Our object capture device (ideal for tennis balls) is made from cardboard, a pack of #2 pencils, two metal rods, and some small metal scraps (instructions in "Catching the Ball" section).

The camera is mounted above the tennis court and communicates with the computer using USB (although a firewire camera can also be used). The Robot Vision CAD (RvCAD) software running on the PC automatically recognizes the camera and imports its images. It then transmits commands to the mobile robot via the COM port, and the mobile robot moves its wheels accordingly. The robot captures tennis balls by bringing the capture device into contact with them.

The Robot's Eye: Where Should It Go?

Attaching the camera near the ceiling above the robot greatly simplifies the task of extracting information about the robot and its environment. The camera images now contain both the robot and all objects near it, and camera perspective does not change as the robot moves. Also, if we point the camera straight down, we can bypass camera calibration and make the approximation that that pixels correspond to physical coordinates. Alternate placements of the camera are discussed later and provide an avenue for further investigation once the system is constructed.

A color camera is useful because color is a convenient way to differentiate objects in the environment with minimal programming effort. As shown in Figure 1, we cover our robot with a circular shape (blue in color) that is easy to differentiate from the background (dark green) and the objects to capture (in our case, light green tennis balls).

Now that our system has been set up in this manner, each camera image contains a circle representing the robot, one or more target objects to be captured, and possibly some objects that are neither of these (obstacles). To make the system "see," we need to determine where each of these items is as well as what they are.

1070-9932/06/$20.00©2006 IEEE

Making the Robot See

Machine vision and image processing are very broad areas of research, and there is an ever-growing number of creative and useful methods for retrieving information from images. Many of these are quite complex, but what we desire here is a very simple way to make the mobile robot see its environment. Our mobile robot needs to know 1) where it is located, 2) what direction it is facing, 3) where the tennis ball target object(s) is, and 4) the location of any obstacles present.

The RvCAD software will automatically import the video stream from the Webcam, and it provides a simple graphical interface where one can "wire" together blocks that perform different image processing functions. We first use a color threshold (a function built into RvCAD) to decide which pixels contain robot, tennis ball, background, or something else (an obstacle). We slightly modify the color threshold function by specifying ranges for each object in terms of the red, green, and blue (RGB) content of its pixels (these RGB levels are numbers ranging from 0–255). Note that the main purpose of this thresholding step is so that the person using RvCAD can easily see the classification given to each pixel. This helps the user tune the RGB color ranges to the appropriate values for the given background and lighting conditions. One obvious choice is a red background, a green tennis ball, and a blue robot, but the specific colors are not important. Any given color can be described by a range in each component of RGB (a range instead of one specific value is needed to account for noise in the camera image). Figure 2 shows an example original and filtered image using a dark green background. Once the appropriate range is determined, this color filter block may be omitted if desired, and the RGB ranges programmed directly into the findblobs function (described next), or it may be retained and its output wired into the input of findblobs.

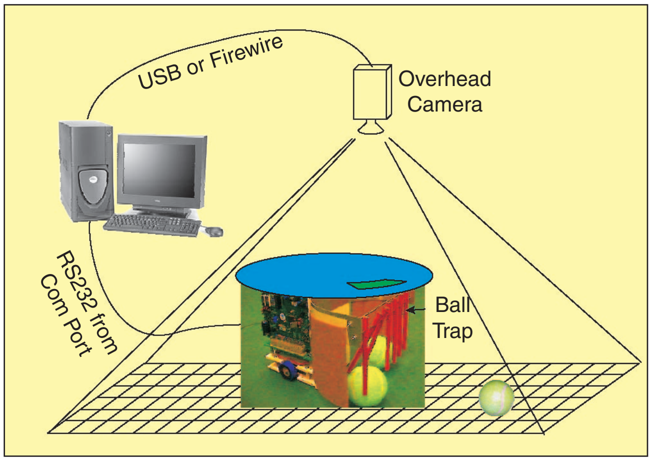

Figure 1. *The complete mobile robot/camera system consists of an overhead Webcam, a computer, and a mobile robot vehicle equipped with a device to capture tennis balls. The overhead camera simplifies the system; pixels approximately correspond to physical coordinates.*

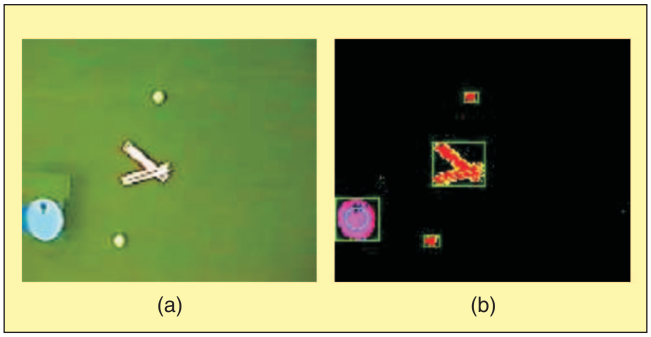

Figure 2. *Image (a) is obtained directly from the camera. Image (b) has been thresholded to classify each pixel as belonging to the robot (purple), tennis ball (red), and obstacle (yellow) for visualization. Note that the obstacle is nearly the same color as the tennis balls, and a size check helps determine which is which. The rectangular bounding boxes in the image on the right show patches of similarly colored pixels (or "blobs") that have been identified.*

The findblobs function is the next (and last) block we use in the image processing pipeline we construct in RvCAD. It locates various areas or "blobs" of similarly colored pixels in the image. The source code of this block is provided with RvCAD. Perhaps the simplest way to figure out what is in each blob is to test the color of the pixel at the center of the blob (by making a small modification to the source code to check which color range it belongs to). The blob that is blue is the robot, blobs that are light green are tennis balls, and any blobs with other colors are obstacles. Now that we know what is in the image and where each object is (the center of each blob is taken to be the object location), the only thing left to determine is the direction the robot is facing.

The blue marking placed on top of the robot serves two purposes. First, it makes the center of the blue blob equivalent to the center of the robot. Second, the small hole cut in the circle (see Figure 1) provides a way to determine the direction the robot is facing. This is done by testing pixels one by one in a circle around the robot center and looking for pixels that are not blue. While it is possible to use other markings on top of the robot, other shapes may necessitate more programming to ascertain not only the location of the robot's center but also the direction it faces.

The vision system has now accomplished all objectives. One small enhancement is to check blob size (since the blobs associated with the robot and the tennis balls are of known size), so that we can differentiate even blue or green obstacles from the robot and tennis balls (as in Figure 2). Now that the robot can "see" its environment and understand all it needs to know, we are ready to command the robot to move.

Telling the Robot Where to Go

A simple way to move the robot to capture a tennis ball is to 1) turn to face the ball, 2) roll forward for a short time, and 3) begin again at Step 1. This cycle repeats until the ball is captured in the ball trap (and thus disappears from the camera view).

How far and in which direction the robot should turn to face the ball can be determined by taking the cross product of two vectors based at the center of the robot. Vector A points toward the front of the robot, and vector B points toward the tennis ball. Solving the cross product formula for angle $(\theta = \sin^{-1}(|A \times B|/|A||B|))$ yields the angle the robot should turn. However, some ambiguity remains because the absolute values in the above formula mean θ will always be in the first or fourth quadrants, even if the actual angle the robot should turn is greater than 90° in either direction. The correct angle can be found by taking the dot product between the A and B. If the sign of the dot product is positive, the robot should turn θ degrees. If it is negative, then the robot should turn $(180 - \theta)$ degrees.

These calculations can be performed either on the PC or on the mobile robot's microcontroller. We chose the latter because the Sumo-Robot has sufficient onboard processing power. Whichever strategy is selected, information can be transmitted to the mobile robot by making a minor modification to the code in "findblobs" allowing it to write to the COM port. If one is using an alternative to the Sumo-Robot, it may also be useful to consider using the computer's parallel port to transmit data.

The mobile robot receives information from the PC and turns its wheels (one forward and one backward) so that it pivots about its center until it faces the tennis ball. As mentioned previously, it then drives forward a short distance before updating its path. This closed-loop method of driving the robot to the ball is robust in that it works even in the presence of robot wheel slippage, tennis ball motion, or multiple tennis balls (provided the location of the ball closest to the robot is the one selected as the target).

Catching the Ball

Figure 3 shows a simple, reliable, and inexpensive device for capturing tennis balls, consisting of a one-way gate. The gate allows tennis balls in but prevents them from escaping. To make this ball trap, we bent a thin metal strip into a half-circle and placed the two thin rods through holes drilled in it. The ends of the rods are threaded and held in place with nuts on the outside of the strip. The pencils serve as the gate for the trap. Drilling holes through them, they can be suspended from the lower metal rod in front. The other rod, placed slightly behind and above the first, serves as a stop to prevent the pencils from swinging outward. This creates a one-way gate that allows tennis balls in but not out, enabling the robot to transport them as desired. The tennis balls can be released by driving the robot up a ramp so that the ball trap extends over the edge. The tennis balls will then fall into a collection basket as shown in Figure 3.

What Else Can You Do with It?

The completed mobile robot/vision system project described here can serve as a launching point for a wide variety of further explorations of robotics research topics. Some of the the possible avenues of study have been hinted at previously. For example, one can consider alternate camera placement options. Some mobile robotics applications require the camera to be placed at an angle rather than directly overhead. This requires the vision and image processing algorithms to deal with perspective as well as camera calibration. There are many sophisticated algorithms for camera calibration and distortion correction (including many useful functions in the MATLAB camera calibration toolbox [2]) that can be investigated by the interested student.

There is an ever-growing number of creative and useful methods for retrieving information from images.

Other mobile robotics applications call for a camera onboard the mobile vehicle. This placement causes the camera view to depend on the robot's position and orientation. To catch a tennis ball with an onboard camera, one can turn the robot's wheels to maintain the tennis ball as closely as possible to the center of the image. Then, as long as the tennis ball image is always growing in size, the robot will eventually catch it. This is an example of visual servoing, and a tutorial on this topic can be found in [5]. There are many further research issues to explore with onboard placement, such as mapping the environment based on camera images and the recognition of objects whose size and perspective can change.

With any choice of camera placement, one can also investigate obstacle avoidance and navigation. The vision system described previously is capable of determining the location of each obstacle in the environment. There are a wide variety of possible techniques to use this information to plan paths from the initial robot position and orientation to a final goal without contacting obstacles [6].

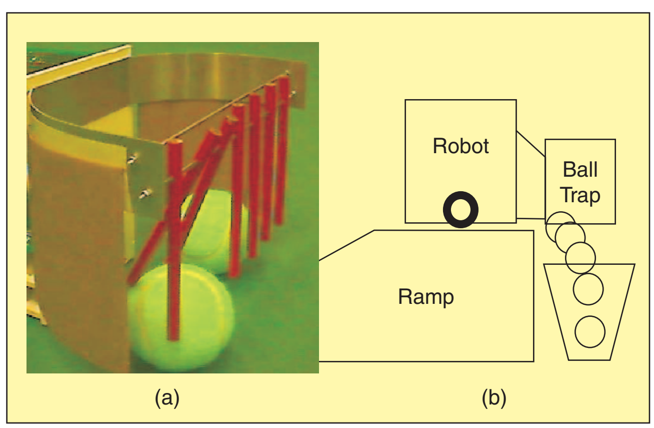

Figure 3. *(a) The ball capture device is essentially a one-way gate made from #2 pencil "teeth." (b) The robot may drive up a ramp to release tennis balls.*

Some mobile robotics applications require the camera to be placed at an angle.

Another broad research area that can be investigated with this system is nonholonomic robotics. If, instead of simply catching the tennis ball, we want the robot to approach it from a specific direction, a simple straight line path will no longer be sufficient. The robot cannot slide sideways around the ball as it approaches because its wheels impose a nonholonomic constraint that prevents sideways motion. However, it is intuitively clear that the robot can perform additional maneuvers (such as those needed when parallel parking an automobile) to capture the ball from a specific desired direction. A thorough treatment of nonholonomy can be found in [8], and an example directly applicable to the Sumo-Robot can be found in [7, Ch. 2]. The unicycle example presented there is directly analogous to the two-wheeled Sumo-Robot. When operated as described here (so that it moves either in a straight line or pivots about its center), the Sumo-Robot has the same kinematic constraints as a unicycle.

The issues mentioned above are only a sampling of the many possible avenues of future research for the interested student. The challenges and interesting features of such endeavors have the potential to inspire a student toward continued exploration in robotics.

Conclusion

The system described in here, known as the "Electronic Ball Boy," was developed at the University of Newcastle, Australia. Videos and additional information can be found in [10] and [11]. The system was constructed by two undergraduates with minimal experience in robotics and no prior computer vision training. The amount of time required was one semester, as a senior design project. As an introduction to robotics research, this project provides exposure to the fields of mobile robotics, image processing, hardware development, and system integration. After the initial system is functional, it serves as a testbed for further investigation and can be taken as far as the student desires. Constructing this system can be an excellent first introduction to hands-on robotics research.

Acknowledgments

This work could not have accomplished without Alan Brannon, who was involved in all phases of the Electronic Ball Boy project. Rick Middleton and Ian Walker both contributed ideas, and both Clemson University and the University of Newcastle in Australia provided financial support for the project.

Keywords

Mobile robotics, educational robotics, visual servoing.

References

[1] J. Abouaf, "Trial by fire: Teleoperated robot targets Chernobyl," *IEEE Comput Graph. Appl.*, vol. 18, no. 4, pp. 10–14, 1998.

[2] J.-Y. Bouguet, "Camera calibration toolbox for Matlab," [Online]. Available: http://www.vision. caltech.edu/bouguetj/calib_doc/index.html

[3] J. Casper and R.R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Trans. Syst., Man, Cybern.*, vol. 33, no. 3, pp. 367–385, 2003.

[4] B.G. DeRoos, J.D. Price, and J.J. Reidy, "Law enforcement robot technology assessment," *Proc. SPIE*, vol. 4232, 2001.

[5] S. Hutchinson, G.D. Hager, and P.I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, 1996.

[6] S.M. LaValle, *Planning Algorithms*. Cambridge Univ. Press, to be published.

[7] M.T. Mason, *Mechanics of Robotic Manipulation*. Cambridge, MA: MIT Press, 2001.

[8] R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Ann Arbor MI: CRC Press, 1994.

[9] G. Musser, "Robots that suck. Have they finally come out with a robot for the rest of us?" *Sci. Amer.*, vol. 288, no. 2, pp. 84–86, 2003.

[10] R.J. Webster III and A.S. Brannon," The electronic ballboy Mark III" [Online]. Available: http://murray.newcastle.edu.au/users/students/2001/c2100098/ebb.html

[11] R.J. Webster, III and A.S. Brannon, "The electronic ball boy: A reactive visually guided mobile robot for the tennis court," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)* 2002, pp. 2054–2059.

Robert J. Webster III is pursuing his Ph.D. in mechanical engineering at the Johns Hopkins University, where he received his M.S. in 2004. He received his B.S. in electrical engineering from Clemson University in 2002. He has held research positions in mobile and bioinspired robotics at the University of Newcastle in Australia and at the Savanna River Site, respectively. His current dissertation research focuses on the design of miniature flexible medical robots. This includes steerable needles and active cannulas to reduce trauma and improve the accuracy of surgery.

Address for Correspondence: Robert Webster, Department of Mechanical Engineering, The Johns Hopkins University, 223 Latrobe Hall, 3400 North Charles Street, Baltimore, MD 21218-2681 USA. Phone: +1 410 516 4184. Fax: +1 410 516 7254. E-mail: robert.webster@jhu.edu