

MIT Open Access Articles

Eye-In-Hand Visual Servoing Curriculum for Young Students

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Rus, D., M. Vona, and K. Quigley. "Eye-In-Hand Visual Servoing Curriculum for Young Students [Education]." *Robotics & Automation Magazine*, IEEE 17.1 (2010) : 116-117. ©2010 IEEE.

As Published: <http://dx.doi.org/10.1109/mra.2010.935801>

Publisher: Institute of Electrical and Electronics Engineers

Persistent URL: <http://hdl.handle.net/1721.1/62179>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Eye-In-Hand Visual Servoing Curriculum for Young Students

Daniela Rus, Marsette A. Vona, and Kevin Quigley

It is never too early to introduce students to robotics. Inspired by courses that use Lego Mindstorms as platform, by the huge success of the FIRST competition, and by the interest and enthusiasm of many local robotics clubs, we have developed a short robotics curriculum aimed at introducing middle- and high-school students to robotics.

Our curriculum introduces manipulation with an articulated robot. In contrast, many common curricula for beginners start with mobile robots. Focusing on manipulation allows us to introduce some of the basic mathematical aspects of robotics, including kinematics and differential control. We have found that even very young students who have not previously learned trigonometry can make progress in this exercise because the robot itself serves as a tangible, graphical instantiation of the math. Table-top manipulators also have practical and logistical advantages over mobile robots rolling around on the floor: they encourage students to stay focused and to work together in one place.

Overview

The curriculum is a hands-on project in which students assemble a four-joint robot arm with an eye-in-hand camera and program it to visually track a ball and ultimately grasp it. The project is structured using basic algebra, trigonometry, and geometry. Students learn how these lead to the derivation of the forward and inverse kinematics of the arm from first principles. They also learn how mathematics and geometry enable visual servoing control to track and grasp a ball. This curriculum package includes specifications for a hardware kit, software for interacting with and controlling the robot, and an instructional handout that walks the students through both mathematical derivations and experimental exercises.

We have offered this curriculum as a one-day robotics workshop for the Johns Hopkins University's Center for Talented Youth (CTY). The workshop was attended by 75 students from across the country. The students were organized into groups of three to four. Each group was able to succeed at creating an eye-in-hand robot manipulator capable of tracking and grasping balls. This curriculum is organized in four parts.

Introduction and Safety Procedures

The first part introduces the students to the hardware platform and components, including the definition of all named joints and quantities on the robot, the software package, and safety procedures. The hardware platform is a CrustCrawler SG5

arm with four revolute joints and a gripper, an SSC-32 servo interface board, an acrylic base for the robot, a camera, a camera-mounting bracket, and tools. Students are also given a movable foam board held in a vertical pose by a second acrylic base, graph paper, a graspable pencil, and a ruler. Software is provided that presents a customized graphical user interface for entering robot commands and for evaluating empirical observations. It is especially important to discuss safety to prevent harm both by the equipment to the students and vice versa. The students are given instructions on parking, homing, and stopping the robot.

Forward and Inverse Kinematics

The next part introduces the students to controlling the arm and learning about its kinematics. Students analytically model the forward kinematics and empirically measure a Jacobian using the vertical board, graph paper, graspable pencil, and ruler.

First, they derive constraint equations for moving the wrist based on the other joints to keep the wrist horizontal. Forward kinematics is then introduced as a function taking the three remaining robot joint angles to the gripper height, radial distance, and facing angle. The arm is considered as a physical calculator that computes the formulas relating these quantities. Students are asked to derive the functions that compute height, radius, and angle in terms of link lengths and joint angles using basic trigonometry and geometry.

By holding a pencil in the gripper, the students can measure quantitatively the effects of moving the joints using traces on the graph paper mounted on the vertical board. Each joint is independently commanded to move a small amount. By measuring the distance between the end points of the motion marked on the graph paper and dividing by the commanded angle, the students estimate the Jacobian at a given configuration. This part of the curriculum gives an empirical intuition for the Jacobian of the robot.

Using simple algebraic formulas for the inverse of a small matrix, differential control using the Jacobian inverse is then developed and empirically demonstrated. For this introductory exercise, the students focus only on the core elements of the method; some details, such as singularity robustness, are implemented in the software but not directly exposed.

Image Processing

Detecting the Ball

The third part of the curriculum guides the students to attach the camera to the robot hand and to develop a procedure for detecting a ball using video images. The approach has two

stages. First, the raw image is altered to select some pixels based on the color of the ball and reject the others. This produces a binary image where a pixel is black if it matches the ball color and white otherwise. Second, a blob within an appropriate size range is identified and used to derive estimates of the ball centroid and distance in camera frame.

Visual Servoing to Follow the Ball

The final part of the exercise uses the centroid and distance estimates computed by the ball-detection procedure to drive the necessary motion of the gripper to track the ball. Jacobian inverse kinematics is used to map the desired gripper differential motion into incremental joint commands that move the arm. Students calculate the error between the current and the desired position of the ball in the camera frame, they learn how to clamp the error, and then they map the clamped error through an automatically computed inverse Jacobian to get the change in each joint angle. Finally, they test the full procedure for grasping the ball.

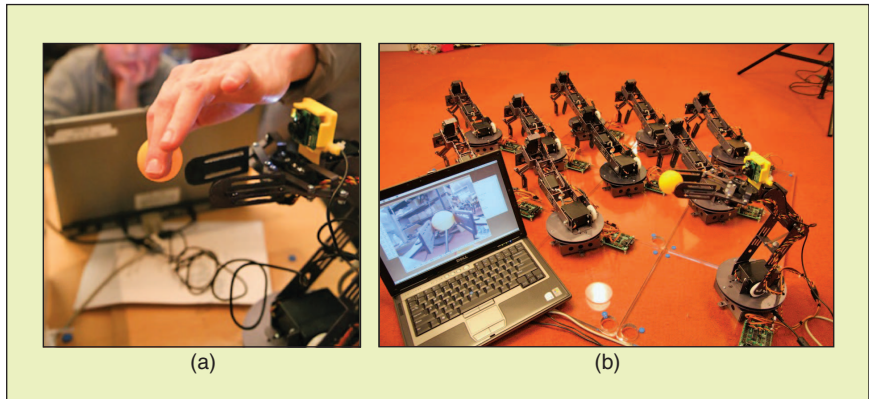
Workshop Results

During our CTY workshop, 75 students were divided into 20 groups to follow this curriculum for building and controlling the robot arm. We allocated one tutor for every two groups of students. The tutors helped the students with the project, answered questions, and explained the intuition behind the new mathematical concepts. All groups successfully completed the exercise. Students were uniformly excited about their working robot hand and were ready to go beyond the scope of the exercise. Some of the groups were able to program additional skills such as throwing the ball.

Outlook and Lessons Learned

This introductory robotics curriculum demonstrates that it is possible to teach mathematical robotics concepts, including forward and inverse kinematics, differential and closed-loop control, and even some basic machine vision, to middle- and high-school students with only basic knowledge of geometry and algebra. Some students in our workshop had not even studied trigonometry, but we found that it was easily introduced and graphically motivated in the context of the exercise.

The CrustCrawler SG5 kits performed adequately only at slow speeds, mainly because of the sloppy behavior of their low-cost hobby servo actuators. This type of servo is not even able to report current position, which is sensed but only used in a relatively low-fidelity internal position control loop. This constraint was a primary factor in our choice to implement the eye-in-hand configuration specifically, since the visual servo



(a) Robot hand grasping an orange ball (photo by Jason Dorfman). (b) Robot hardware and software visualization for the eye-in-hand visual servoing curriculum (photo by Luliu Vasilescu).

loop can then correct for some slop in the end effector. However, effects such as mechanical oscillation can still degrade performance. Crustcrawler has now introduced a “Smart Robotic Arm” based on more advanced servos that do support external position feedback.

This robotics curriculum, the hardware specifications, software, handout, and supporting information can be downloaded from <http://groups.csail.mit.edu/drl/highschool-robotics>.

Acknowledgments

This work was developed with support from Computer Science and Artificial Intelligence Lab (CSAIL) and the EECS Department at MIT.

Daniela Rus earned her Ph.D. degree in computer science from Cornell University. She is a professor of electrical engineering and computer science. She is an associate director of MIT’s CSAIL and codirects the MIT Center for Robotics at CSAIL. Her research interests include distributed robotics and mobile computing, and her application focus includes transportation, security, environmental modeling and monitoring, underwater exploration, and agriculture. She is a Class of 2002 MacArthur Fellow and a Fellow of AAAI and IEEE.

Marsette A. Vona is an assistant professor in the College of Computer and Information Science at Northeastern University and a recent graduate of MIT. From 2001 to 2003, he worked at NASA’s Jet Propulsion Lab on the science operations interface software for the Spirit and Opportunity Mars rovers, for which he shared the NASA Software of the Year Award in 2004.

Kevin Quigley is a software engineer at Charles Stark Draper Laboratory. He is an architect of, and team leader for, a real-time software- and hardware-in-the-loop simulation framework.