Createing Range from Texture

Zeke Koziol, Anatole Paine, Sabreen Lakhani, and Zachary Dodds

Harvey Mudd College Computer Science Department
301 Platt Boulevard
Claremont, CA 91711
zkoziol, apaine, slakhani, dodds@hmc.edu

Abstract

This synopsis presents the vision and capabilities of Harvey Mudd College's entry into the 2009 IJCAI robotics program. At the event we demonstrated a low-cost platform built atop an iRobot Create. The platform supports not only an accessible Wiimote-based interface, but also sensing sufficient to build an autonomous range-from-vision system. Here, we contrast the ability of two machine learning algorithms to segment ground plane from obstacles. Our results demonstrate that multilayer perceptrons more precisely classify image texture as navigable or not; restricted Bolzmann machines, on the other hand, better generalize to environments unseen during training.

Motivation

Harvey Mudd College's entry into the 2009 IJCAI robot program provides a template for supporting robot vision research atop the inexpensive iRobot Create. Although most autonomous platforms use sensors that directly compute range from time-of-flight, e.g, from laser range finders, low-cost platforms are unlikely to be able to use those sensors. Monocular vision, on the other hand, offers at least the *promise* of an advantageous alternative to laser range-finding across several axes: cameras are less power-hungry, less heavy, less bulky, less range-limited, and, perhaps most importantly, less expensive.

Less is more, however, when it comes to computation. Extracting range from pixel intensities requires far more algorithmic and computational lifting than extracting range from time-of-flight. Usual range-from-vision approaches use temporal feature correspondence across a monocular image stream to deduce distance from pixels (Kanade et al. 2001). This corpus of work is mature, but it is worth noting that these techniques are most successful when significant spatial context is used across the image stream. Large patches of pixels facilitate accurate and precise correspondence.

Recent approaches have boldly asked, "What can we deduce from *only* those patches, and not the correspondence at all!?" For instance, Hoiem et al.'s Photo pop-out (Hoiem et al. 2007) software and Saxena et al.'s Make3d system (Saxena et al. 2008) yield range at each of a single image's pixels. Spatial grouping, e.g., into the edges between ground and vertical planes enable the compelling visualizations those groups have produced.

Such work has seen robot applications: in (Hoiem et al. 2007) Hoiem et al. show confidence levels in terrain navagability, in (Saxena et al. 2008) Saxena et al. drive an RC car safely and quickly through rough, natural terrain. Those projects emphasized machine-learning contributions over the resulting range accuracy. More recently, Plagemann quantified the range accuracy of the learned mapping from *columns* of pixels from omnidirectional images to range (Plagemann et al. 2008). That work used Gaussian Processes to achieve ~1m precision sufficient to support off-the-shelf SLAM algorithms under assumptions common to indoor environments. Such results underscore the power of range-from-texture approaches, pioneered in Horswill's Polly (Horswill 1995), and used in many systems since.

Background

In this work we seek to learn indoor range-to-obstacle directly from visual texture. We contrasted two machine learning approaches, multilayer perceptrons (MLP) using backpropagation (Rosenblatt, 1958) and restricted Boltzmann machines (RBM) (Hinton and Brown, 2000).

Networks' Architecture

The MLP and RBM both have 64 visible inputs: the raw pixel values from an 8x8 patch of one of the images. The MLP had a single output that was reinforced toward 1.0 for obstacle patches and 0.0 for floor patches. The RBM had two distinct labels: one for floor and one for obstacles. RBMs do not output a traditional binary classification signal; rather, they reconstruct the input using both the learned models, floor and obstacle. The fidelity of each reconstruction allows our system to decide how to classify each patch. In our implementation, we weighted each component equally with opposite signs. Both networks were 5 layers with 3 20-neuron hidden layers.

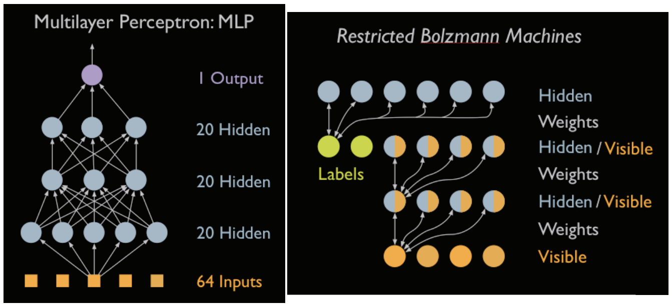

Figure 1. Diagrams of our multilayer perceptron (MLP) and restricted Boltzmann machine (RBM) architectures appear at left and right, respectively. Each network used 64 inputs taking in the raw grayscale values of an eight-by-eight pixel patch of an image. Through backpropagation, the MLP learned a mapping of floor to the output 0.0, while the edge between floor and obstacle mapped to 1.0. The RBM learned a low-dimensional representation for floor and edge patches, respectively.

Training and training data

Identical training data was fed into the MLP and RBM networks. To obtain that data, we drove the robot through a hallway-dominated environment for several minutes and collected images. A subset of 300 of the tens of thousands of images collected were then segmented by hand using a custom-built tablet application. One such hand-segmentation appears as the pink line in Figure 2.

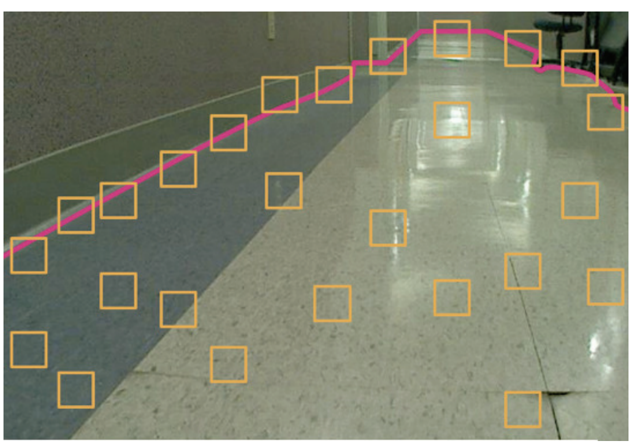

Figure 2. In contrast to traditional approaches' feature correspondence and camera calibration, our range-from-texture approach learns from labeled images to distinguish the source of 8x8 pixel patches: floor vs. wall. The pink line is human-labeled ground truth; yellow squares indicate training patches. Although the training is off-line, the resulting classifier extracts groundplane from a live image stream, as shown in Figure 3.

From those hand-segmented images many 8x8 patches of pixels were extracted: for each one within the floor plane (*below* the segmentation contour) a corresponding patch from the facet between floor and obstacle was used, as well. These patches, labeled as "floor" or "edge" became the training set for both the MLP and the RBM.

Results

The fundamental differences between multilayer perceptrons and restricted Boltzmann machines are apparent in Figure 6, below. The MLP carefully hews to the environment in which the classifier was trained: as a result it is more accurate in segmenting ground plane from obstacles in those environments. For example, the MLP segments the alternating bicycle wheels and wall segments as well or better than a human observer can.

On the other hand, because the RBM seeks parsimony in its constructed representation of the floor vs. wall patches, it does not classify known environments as well as the MLP. However, it generalizes far better when patches very different from training data appear. It is the parsimony of representation that the RBM seeks that makes this generalization possible. In Figure 3, an observer looks into the camera of the robot: such texture had never appeared in the training set, and the RBM segments it better than the MLP.

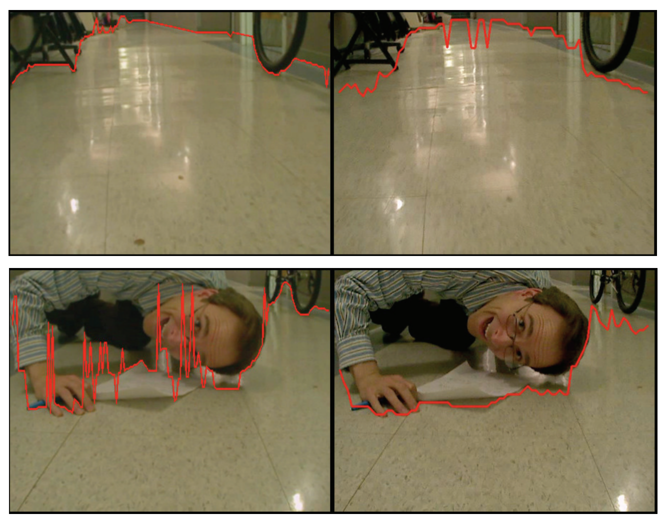

Figure 3. The results of our ground-plane segmentation based on both MLP and RBM networks. The images to the left in each case have been segmented by an MLP. Those on the right are segmented by an RBM. The greater precision of the MLP in familiar environments is evident in the top two images, whereas and the greater ability of RBMs to generalize is apparent in the bottom two images: images containing a person did ***not*** appear in the training data.

Figure 4 contrasts the segmentation performance of the RBM and the MLP at the IJCAI robotics workshop in Pasadena in July 2009. Although no images of that venue had been used in the training, the RBM still segments floor from obstacles remarkably well.

They key difference, we feel, is that the RBM builds two continua: one along which "floor" is measured and one along which "edge" patches are compared. The multilayer perceptron, in contrast, projects each patch (nonlinearly) onto a single continuum that spans from floor to edge. As a result, when completely novel patches appear, they are arbitrarily assigned by the MLP; the RBM detects them as new and (conservatively) classifies them as untraversable.

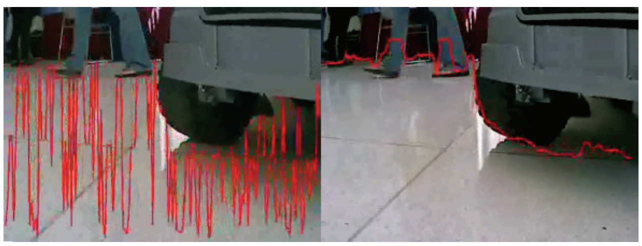

Figure 4. The ability of the RBMs to generalize to new environments was underscored in its performance at the Pasadena conference center at the IJCAI robotics workshop. No images from that venue had been used in the training data, but the RBM results (right) were quite good. MLP results (left) showed that most patches were simply unrecognized.

Platform

The range-from-texture capabilities made possible by the modified iRobot Creates take an important step toward a new generation of service and entertainment robots. The solid foothold of today's commodity robots has come *without* the capability of *spatial deliberation*. Although ~$250 platforms will likely never use laser range finders, in indoor environments, cameras offer a possible alternative. Dense range estimates, in turn, will enable such platforms to leverage off-the-shelf algorithms for mapping, localization, and navigation. Such maintenance and use of environmental maps will open up delivery, surveillance, and other service capabilities to a broad audience.

To investigate the feasibility of low-cost robots that can reason about more than their local environment, we presented at IJCAI 2009 several prototype extensions of an iRobot Create. The key design choice is to manage all sensor processing off-board via wireless router. An ethernet camera and short-range IR sensors provide excellent data, complementing the Create's, for map maintenance. Currently the platform uses pre-provided maps in order to complete point-to-point navigation tasks, e.g., for pickup and delivery. In addition, it serves as an accessible basis for research and experiments in vision-based navigation.

Hardware and design

On top of the proprioceptive and actuation capabilities provided by the Create (iRobot, 2007) a suite of six Sharp GP2D12 IR sensors and an Axis207 ethernet camera send a stream of data via an off-the-shelf wireless router. Any router will do; we have used Linksys WRT320N and 546L models. Router-to-robot control occurs through an Arduino microcontroller and its accompanying ethernet interface. Voltage regulators and their heat sinks are the only additional wiring required. In total, the cost beyond the Create is less than $400: $200 for the camera and $200 for the other components. These reflect prototype costs; commercial production would yield substantial savings. Figure 5 shows our integrated platforms and their components.

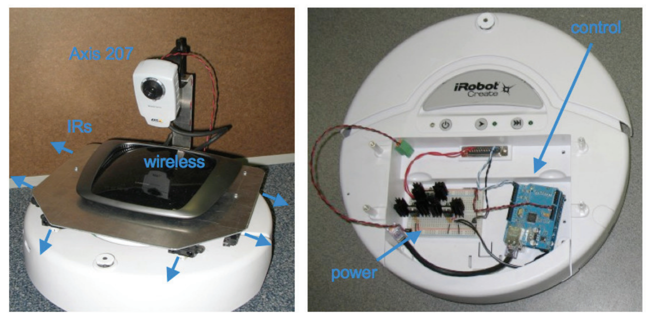

Figure 5 - (**left**) At left is the assembled platform and its external sensors, six infrared range sensors and an Ethernet camera. Sensor processing and control is off-board; lag is not an issue; the limiting factors are the Create's serial bandwidth and the image-processing time, not data- or image-transfer costs. (**right**) With the shelf removed, the simplicity and robustness of the design is apparent: a power-distribution circuit of three voltage regulators and an Arduino control interface to connect the sensors and base.

Software for spatially aware interaction

To highlight the capabilities of these platforms, we have created several interfaces:

(1) a remote-control interface using a game controller
(2) autonomous exploration routines for indoor use
(3) spatial-reasoning software supporting navigation
(4) an OpenCV wrapper for vision-based capabilities

At the low level, the robot presents itself as a network device to which ASCII strings are passed back and forth. Although any language can be used, to date we have built a Python interface that exposes all of the motor and sensory capabilities except vision. That web-accessible image stream, 640x480 at 20fps, is available to whatever tools a developer may wish to use. We use an OpenCV-based interface written in C++.

Inspired by work at Brown University (Lapping-Carr et al., 2008), our first software interface leverages pygame (**www.pygame.org**) to provide direct control of the robot through a Wiimote game controller. The result is engaging and challenging interactions appropriate for elementary-school audiences – and enjoyable for any age!

Figure 6. The resulting platform is seven-year-old-proof. Inexpensive enough to use in outreach activities, the software and Wiimote-control allows elementary school students to navigate the robots using only sensor-provided data.

As highlighted in Figure 6, we have developed pedagogical activities that allow young students to experience hands-on how it is different to "be" a robot. Navigating to a destination with only the video feed or, more difficult, following the walls with only the IRs deepens appreciation of the challenges inherent in bridging sensing and control. The addition of a Nerf launcher has been a wonderful ice-breaker, and is equally effective among researchers attending IJCAI and elementary school children.

Figure 7. Having an OS onboard makes the addition of peripherals straightforward. Indeed, the actuation itself *is* a peripheral. The robots, at home in hallways (**left**) were also exhibited at IJCAI '09 (**center**); in addition, a Nerf missile launcher can act as a universal ice-breaker (**right**).

A second design possibility we have pursued (Figure 7) replaces the router and camera with a netbook computer, following the lead of CMU's Tekkotsu-on-the-Create (Nickens et al. 2009) and designs for telepresence by researchers at Southern Illinois University, Edwardsville. The OS-on-board offers an easy interface to additional devices, but camera quality and software development suffer atop a netbook: those conditions will likely improve in the future.

Fast, inexpensive autonomy via IR sensing

The second interface, one for autonomous exploration, mimics the Creates' ability to quickly cover a large environment by wall-following. Using the IRs increases efficiency, however, leading to an average indoor wall-following speed of 39.1 centimeters per second, including concave and convex corners. The Create's built-in wall-following behavior runs at 24.3 cm/sec. As Figures 7 and 8 show, the sensor suite is well-suited to navigating within indoor environments.

Atop this autonomous wandering we have added a spatial-reasoning module. Similar in spirit to the pioneering platforms of (Simmons et al. 1997; Thrun et al. 2000) but at a cost orders of magnitude lower, the software allows a user to click on a goal location within the robot's known map. The system then autonomously plans a path to that goal and executes it. Figure 8 contrasts the odometric estimate (in purple) and map-corrected locations (in red along straightaways, green when turning, and blue upon task completion) of the robot throughout a short run. This "delivery" capability will be tested in November 2009's robotics innovation competition and conference (RICC).

Our software is freely available at **https://svn.cs.hmc.edu/svn/robotics/Summer09/roombaLib** (Koziol et al. 2009).

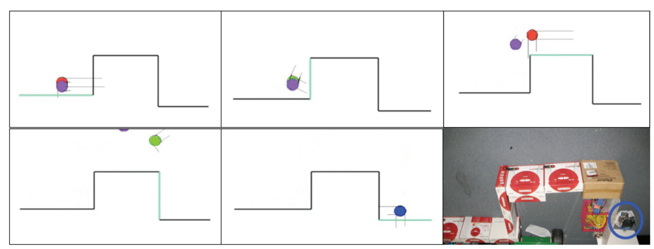

Figure 8. Snapshots from the robot's odometric position (purple) and actual position through a multi-segment wall-following task. The walls themselves enable correction of the robot's location, culminating in a correct estimate of its pose even though the odometric estimate has long left the field of view. Here, color represents task state: red for wall-following, green for turning corners, and blue at the desired destination. IR values are also visible. The wall that the platform believes it is currently following is highlighted in cyan. The final snapshot shows the location of the robot at the run's end, very near its estimated pose.

Perspective

We believe that commodity robots are the foundation for a coming generation of capable autonomous agents. Certainly corporate start-ups such as Heartland, Willow Garage, and The Droid Works - among many others - will "change the game" with novel platforms. Yet this work shows that even modest resources in educational and research settings can augment available hardware to create accessible and capable systems capable of autonomous spatial reasoning – and the tasks that build upon it.

Acknowledgments

The authors recognize and thank the generous support of funding from NSF REU #0451293, NSF CCLI #0536173, The Rose Hills Foundation, and Harvey Mudd College.

References

Esposito, Joel M. and Barton, O. Matlab toolbox for the iRobot Create, at URL **www.usna.edu/Users/weapsys/esposito/roomba.matlab/** 2008.

Hinton, G. E. and Brown, A. Spiking Boltzmann Machines. *Advances in Neural Information Processing Systems* 12, MIT Press, Cambridge, MA, 2000.

Hoiem, D., Efros, A. A., and Hebert, M. Recovering Surface Layout from an Image. *IJCV*, Vol. 75(1) Oct. 2007.

Horswill, Ian. Analysis of Adaptation and Environment. *Artificial Intelligence*, Vol. 73, pp. 1-30. 1995

iRobot, The iRobot Create Open Interface, 2007, from **www.irobot.com/filelibrary/pdfs/hrd/create/Create%20Open%20Interface_v2.pdf** .

Kanade, T., Kanade, B. Y., and Morris, Daniel D. Factorization methods for structure from motion. *Phil. Trans. of the Royal Society of London*, *Series A*. 356, pp.1153-1173, 2001.

Koziol, Z., Paine, A., Lakhani, S, and Dodds, Z. Software repository with public access to the interface: **https://svn.cs.hmc.edu/svn/robotics/Summer09/roombaLib**

Lapping-Carr, M., Jenkins, O., Grollman, D., Schwertfeger, J., and Hinkle, T. (2008) Wiimote interfaces for lifelong robot learning. In *AAAI Symposium on Using AI to Motivate Greater Participation in Computer Science*, Palo Alto, CA, USA, Mar 2008, AAAI Press, pp. 61-66.

Nickens, Glenn V., Tira-Thompson, Ethan J., Humphries, Thorna, and Touretzky, David (2009) An inexpensive hand-eye system for undergraduate robotics instruction *SIGCSE Bulletin* 41(1) ACM Press, pp. 423-427. Online guide available at **chiara-robot.org/Create/** .

Plagemann, C., Enres, F., Hess, J., Stachniss, C., and Burgard, W. (2008) Monocular Range Sensing: A non-parametric learning approach. *Proceedings, ICRA 2008* May 19-23, Pasadena, CA IEEE Press, pp. 929-934.

Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Cornell Aeronautical Laboratory, Psychological Review*, v65, No. 6, pp. 386-408, 1958.

Saxena, Ashutosh, Chung, Sung H., and Ng, Andrew (2008) 3-D Depth Reconstruction from a Single Still Image *International Journal of Computer Vision* 76(1) Kluwer Academic Publishers, pp. 53-69.

Simmons, Reid G., Goodwin, Richard, Haigh, Karen Zita, Koenig, Sven, O'Sullivan, Joseph, and Veloso, Manuela M. (1997) Xavier: experience with a layered robot architecture *SIGART Bulletin* 8(1-4) ACM Press, pp. 22-33.

Thrun, Sebastian, Beetz, Michael, Bennewitz, Maren, Burgard, Wolfram, Creemers, A.B., Dellaert, Frank, Fox, Dieter, Hahnel, Dirk, Roy, Nicholas, Schulte, Jamieson, and Schulz, Dirk (2000) Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva *International Journal of Robotics Research*, 19(11) Sage Science Press, pp. 972-999.