Tekkotsu: A Framework for AIBO Cognitive Robotics

David S. Touretzky and Ethan J. Tira-Thompson
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213-3891
dst@cs.cmu.edu, ejt@andrew.cmu.edu

Abstract

Cognitive robotics is an approach to robot programming that draws inspiration from ideas in cognitive science, such as visual routines (Ullman 1984), dual-coding representations (Paivio 1986), and perceivable affordances (Gibson 1977; 1979). We have implemented primitives based on these ideas as part of Tekkotsu, an open source application development framework for the Sony AIBO.

Introduction

Tekkotsu (the name means "framework", literally "iron bones" in Japanese) is an application development framework for the Sony AIBO robot dog (Tira-Thompson 2004). It provides a layer of abstraction above the Sony OPEN-R software interface and offers a variety of services, including an efficient event routing architecture, the ability to share C++ objects across processes, a hierarchical state machine formalism for constructing behaviors, and an extensive collection of wireless remote monitoring and teleoperation tools. The latter are written in Java for portability. Tekkotsu is an open source project and builds on the work of several other open source developers: it provides forward and inverse kinematics solvers based on ROBOOP (Gordeau 2005), simple object detection using CMVision (Bruce, Balch, & Veloso 2000), and two walking engines, one from CMPack-02 (Veloso *et al.* 2002) and one from the University of Pennsylvania (Cohen *et al.* 2004). Tekkotsu is currently in use at over 20 universities around the world, either in introductory robotics courses or for robosoccer. It is available at `www.Tekkotsu.org`.

Over the last two years we have been developing a new layer of Tekkotsu to support an approach to robot programming that we call "cognitive robotics". The idea is to provide a set of higher level primitives for perception and action, inspired by ideas from cognitive science, so that programmers can construct intelligent behaviors at a much more abstract level. Three components of our approach are described here: visual routines, dual-coding representations, and perceivable affordances.

Visual Routines

Ullman proposed that low-level vision might be implemented as a set of composable parallel operators he called visual routines (Ullman 1984). There is some evidence that such operations are performed in primary visual cortex (Roelfsema, Lamme, & Spekreijse 2000). Tekkotsu provides a set of visual routines that operate on 2D "sketches," starting with a color-segmented camera image (Halelamien 2004; Tira-Thompson *et al.* 2004). We use simple, uniform-colored objects (Figure 1) so that object segmentation can be done based on color alone. Tekkotsu's visual routine operators include color masking, connected components labeling, flood-fill, boundary distance, skeletonization, and neighbor sum, along with basic pixel-wise arithmetic, comparison, and boolean functions.

Sketches are automatically organized into a derivation tree, i.e., the result of applying an operator to a sketch is a new sketch that references the original sketch as its parent (Figure 2). A remote viewing tool allows the programmer to "look inside the dog's head" and examine the derivation tree and any of its component sketches.

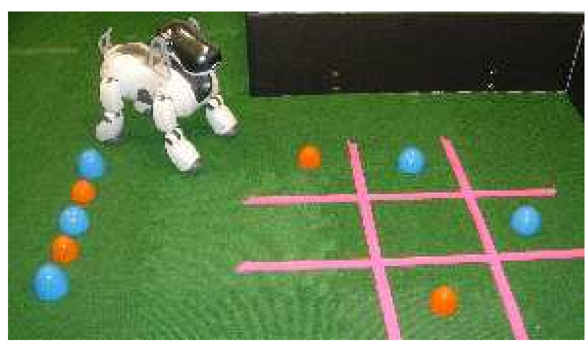

Figure 1: AIBO examining a tic-tac-toe board.

Dual-Coding Representations

Paivio's "dual coding theory" of representations posits parallel verbal and non-verbal (imagistic) systems with extensive referential connections between them (Paivio 1986). In Tekkotsu, sketches provide the imagistic representation, and "shapes" provide a complementary symbolic representation.

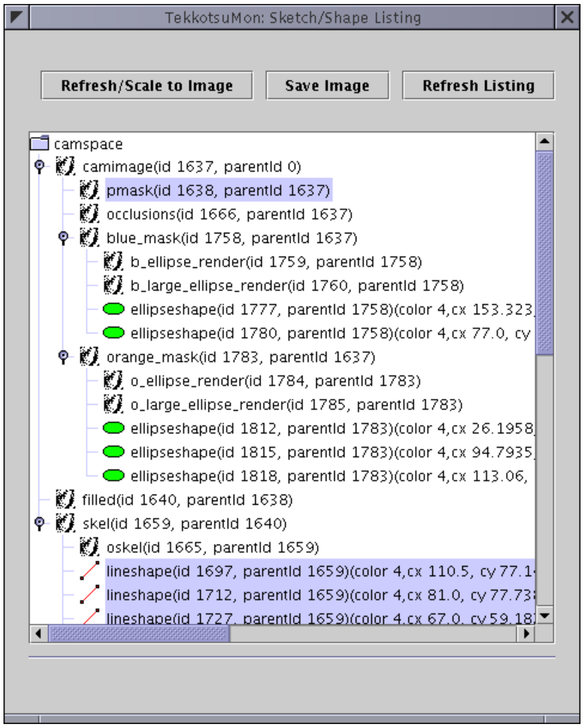

Figure 2: Derivation tree showing sketches and shapes extracted from a tic-tac-toe board image.

Shape data structures such as lines and ellipses can be extracted from images using visual routines operators.

Lines are described by parameters (r, θ) defining the distance and angle to the perpendicular, as in the Hough transform, although our line extraction process utilizes a different algorithm. A line's endpoints may also be specified, but are not required, since one or both endpoints may not be visible in a given image. Ellipses are described by their centroid coordinates, semimajor and semiminor axis lengths, and orientation of the major axis.

Built-in rendering operators automatically convert from a symbolic representation back to an iconic one, yielding a sketch of the shape. Thus, computations can be carried out using a mix of iconic and symbolic operations. Some types of computations are more easily performed in the symbolic space, e.g., constructing a line perpendicular to another, while others are essentially iconic. Shapes are recorded in the same derivation tree as sketches. Figure 2 shows a derivation tree representing a tic-tac-toe board that has been parsed into a collection of sketches and shapes. Figure 3 shows a sketch containing the pixels making up the game board, with four shapes (the extracted lines) superimposed on top of it.

Map Construction

The AIBO's camera has a limited field of view, so a map of its environment must be constructed incrementally, from multiple camera images (Tira-Thompson *et al.* 2004). This is more easily done in symbolic space. The camera image,

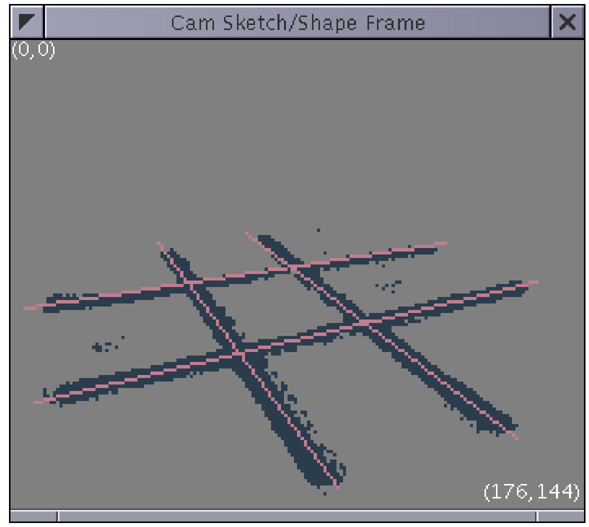

Figure 3: Display of selected objects from the derivation tree. Pink pixels are rendered here in blue; extracted line shapes are superimposed over the underlying sketch.

imported as a sketch, forms the root of the "camera space" derivation tree. As shown in Figure 4, shapes are extracted from the camera sketch, yielding a set of objects in camera shape space. Then, information about the camera's position and orientation, computed from the robot's joint angles, is used to project the camera shapes onto the ground plane in front of the robot. (We make the simplifying assumption that all lines and ellipses lie in the ground plane, that spheres rest on the ground, etc.) The ground space shapes are then matched against and imported into the "local map," an egocentric map with the robot's body at the origin. As multiple camera images are processed, the local map develops into a description of the nearby environment. The shape matching and update process allows the robot to construct accurate representations of lines in the local map shape space even if an entire line cannot fit into any one camera image.

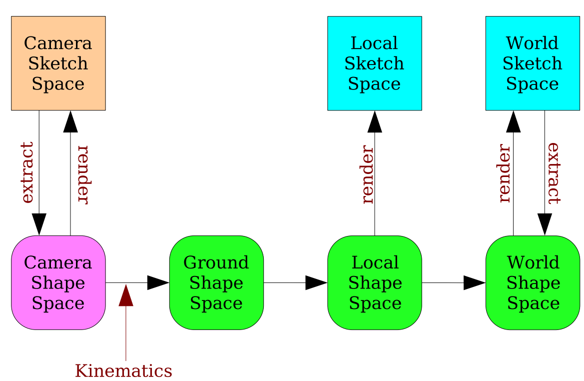

Figure 4: Dual-coding representations in various coordinate systems are used to derive a world map from a series of camera iamges.

The local map is in turn matched against an allocentric world map using a particle filter, and new local map shapes are imported into the world map shape space as appropriate. The robot's own location and orientation on the world map are represented using a special "agent" shape. The dual-coding nature of the maps means that an iconic representation of world space is also available; this can be useful for path planning operations.

Perceivable Affordances

A high-level approach to robot programming should relieve the programmer from having to perform complex perceptual or kinematics calculations to manipulate objects in the world. A perceptual system based on J. J. Gibson's theory of "affordances" (Gibson 1977; 1979) can meet this goal, by representing objects in terms of the actions that can be performed on them.

On Gibson's view. affordances describe relationships between an organism and its environment: they are what the environment "offers, provides, or furnishes [the organism], either for good or ill" (Gibson 1979). Don Norman popularized this idea in his book *The Psychology of Everyday Things* (Norman 1988). In a later essay, Norman summarized Gibson's affordances as "the actionable properties between the world and an actor," and strongly emphasized the distinction between actual affordances, which exist naturally in the world, and *perceived affordances* arising in the actor's mental representations (Norman 1999).

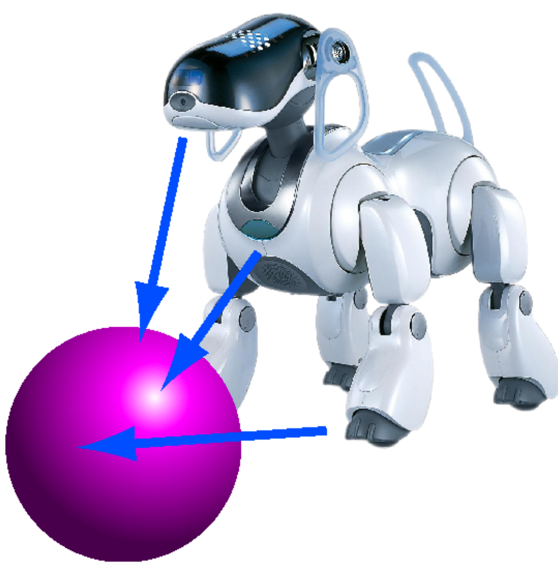

Figure 5: A ball affords pushing with the paw, chest, or head.

Figure 6: Small balls afford pushing with a paw; large balls afford pushing with the chest.

Figure 7: Joint limits constrain motion. In this position, the head can not turn far enough to the left to see both endpoints of the line.

Edirisinghe and Touretzky implemented a first version of an affordance recognizer for the AIBO that uses visual routines plus knowledge of the body's physical constraints to determine the feasibility of actions (Edirisinghe 2005). For example, a ball affords pushing, and there are several ways to accomplish this (Figure 5). Which type of pushing is appropriate depends on the size of the ball relative to size of the robot's body (Figure 6). Small balls cannot be pushed with the chest; the robot would simply walk over them. Large balls should be pushed with the chest; use of a paw is inadvisable due to greater weight and awkward geometry. The criteria for assessing which actions are feasible for which objects must be determined empirically. Presently we do this by experimentation, but it should also be possible for the robot to learn this for itself by trying actions and noting successes and failures, as Gibson (1979) suggests children do.

Another source of physical constraints is the limited range of travel of joints. Suppose the robot is looking at a segment of a line drawn on the ground. The Trace Line action moves the robot's gaze along the line in order to locate the endpoints and determine the line length. Any line potentially "affords" such a tracing action, but if one of the robot's head or neck joints has reached a turning limit, the tracing action may not be feasible in the current context (Figure 7). The affordance recognizer uses Tekkotsu's kinematics solver to detect this situation, and report that a potential trace action is blocked due to a positioning problem. Fortunately, objects also afford locomotion actions that can move the robot to a position where other afforded actions become feasible.

Our approach potentially allows the robot's behavior to be determined by specifying a policy for selecting from the set of affordances the robot currently perceives. When integrated with visual routines and a world map, this will provide a powerful high level approach to robot programming.

The pushing action afforded by a ball is very different than that afforded by a wooden block. Blocks can be either translated across the groundplane, or rotated about their center of mass, depending on where pressure is applied. These considerations don't exist for balls. We are planning on adding block recognition to the visual routines code in the near future.

The idea of perceived affordances could be further extended by introducing gestalt perception mechanisms, e.g., a set of lines on the ground suggesting a bounded region might be perceived as a "container" that affords moving an object into it. In the case of the tic-tac-toe board, only the central square is bounded on all four sides. The other eight board positions have only two or three explicit bounding lines; the remaining boundaries are implied by the convex hull of the board. Seeing board positions as enclosed rectangles therefore entails a kind of gestalt perception. We view this as an important component of affordance recognition which should be pursued further.

A drawback to our current approach is that affordances are tied to specific objects that the robot must be able to recognize. To allow the robot to deal with novel objects, perception must operate at a more primitive level, where size and shape information give rise to affordances even when the robot does not know what the object is for. Consider an AIBO learning about pushing a banana: a complex, unfamiliar shape. Regions of maximal and minimal curvature along the object boundary might provide interest points where a push action could be applied. Experimenting with various pushing actions could then lead to knowledge about the best place to push a banana depending on whether the goal is to translate or rotate it.

Another difficulty with our current approach is that our affordances describe an operation on a single object, such as tracing a line. Often one wishes to alter the relationship of an object to something else, e.g., "push this game piece to that board position." To avoid a combinatorial explosion, we need a way to independently describe the roles each object can play in an action, and then compose those roles to test the viability of a specific action. For example, the game piece affords pushing, while an open board square affords receiving an object. Moving a piece onto a board square can thus be seen as the composition of two mutually reinforcing affordances. In richer domains, domain knowledge might be employed to filter the set of perceivable affordances so that the most common actions are the most salient.

Our current implementation of affordances resembles Millikan's notion of a "pushmi-pullyu" representation (Millikan 2004). The "pushmi" component is *descriptive*: a representation of a perceived potential for action, while the "pullyu" component is *directive*: its function is to "guide the mechanisms that use it so that they produce its satisfaction condition" (Millikan 1996). Our directive components are hand-coded state machines that perform the actions described by the affordance.

As Millikan notes, affordances provide direct connections between perception and action. To proponents of situated cognition, affordances exemplify the specialized nature of perception, i.e., perception does not produce neutral representations of the world, but rather is purposely tailored for action (Clark 1997). This may be true for insects or lower vertebrates, but we share Clark's skepticism that neutral mental representations can be dispensed with in higher organisms. Our robots do not perceive objects *only* in terms of the actions they afford; they also construct maps of the world, where objects are represented in other ways. An extensive literature survey on affordances and their use in robotics is given in (Ugur 2005).

The next major class of primitives to be developed for our cognitive robotics framework are manipulation operations on objects. We have hand-coded state machines for a few manipulation actions, such as pushing balls across lines, but this is only a crude first step. Ideally we would like to have geometry-based primitives such as "apply a translational force in direction ϕ to the estimated center of mass of this object." Implementing such primitives on a quadruped robot will require a more detailed physical model of the robot's body that takes balance, pose, and foot placement constraints into account.

Conclusions

High-level robot programming systems provide abstract primitives for perceiving and manipulating objects. This idea can be traced all the way back to the block pushing primitives of Shakey the robot (Nilsson 1984). There are many other examples in the literature. One of the more recent ones is the Personal Rover system of Illah Nourbakhsh and colleagues (Falcone *et al.* 2003), where users program missions in terms of high-level primitives such as "locate a landmark" or "follow a hallway," without having to know how these operations are implemented.

Acknowledgments

The initial development of Tekkotsu was funded by a grant from the Sony Corporation. Matt Carson, Dilan Edirisinghe, Anoopum Gupta, Neil Halelamien, Alok Ladsariya, Kei Usui, and Jordan Wales contributed to the develpment of the Tekkotsu components described here.

References

Bruce, J.; Balch, T.; and Veloso, M. 2000. Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, volume 3, 2061 – 2066.

Clark, A. 1997. *Being There: Putting Brain, Body, and World Together Again*. Cambridge, MA: The MIT Press.

Cohen, D.; Ooi, Y. H.; Vernaza, P.; and Lee, D. D. 2004. The University of Pennsylvania Robocup legged soccer team. Technical report, University of Pennsylvania. Available at http://www.cis.upenn.edu/robocup/UPenn04.pdf.

Edirisinghe, D. C. 2005. Perceivable affordances and mobile robots. Senior honors thesis, Computer Science Department, Carnegie Mellon University.

Falcone, E.; Gockley, R.; Porter, E.; and Nourbakhsh, I. 2003. The personal rover project: The comprehensive design of a domestic personal robot. *Robotics and Autonomous Systems* 42:245–258.

Gibson, J. J. 1977. The theory of affordances. In Shaw, R. E., and Bransford, J., eds., *Perceiving, Acting, and Knowing*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gibson, J. J. 1979. *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin.

Gordeau, R. 2005. Roboop – a robotics object oriented package in C++. Available at http://www.cours.polymtl.ca/roboop/.

Halelamien, N. S. 2004. Visual routines for spatial cognition on a mobile robot. Senior honors thesis, Carnegie Mellon University, Pittsburgh, PA. Available at www.Tekkotsu.org/media/thesis_neilh.pdf.

Millikan, R. G. 1996. Pushmi-pullyu representations. In Tomberlin, J., ed., *Philosophical Perspectives*, volume IX. Atascadero, CA: Ridgeview Publishing. 185–200.

Millikan, R. G. 2004. *The Varieties of Meaning: the Jean-Nicod Lectures 2002*. The MIT Press.

Nilsson, N. J. 1984. Shakey the robot. Technical Report 323, SRI International, Menlo Park, CA.

Norman, D. A. 1988. *The Psychology of Everyday Things*. New York: Basic Books.

Norman, D. A. 1999. Affordance, conventions, and design. *Interactions*, May, 38–43. Available at www.jnd.org/dn.mss/affordance_conventi.html.

Paivio, A. 1986. *Mental Representations: A Dual-Coding Approach*. New York: Oxford University Press.

Roelfsema, P. R.; Lamme, V. A. F.; and Spekreijse, H. 2000. The implementation of visual routines. *Vision Research* 40:1385–1411.

Tira-Thompson, E. J.; Halelamien, N. S.; Wales, J. J.; and Touretzky, D. S. 2004. Cognitive primitives for mobile robots. In Schultz, A., ed., *The Intersection of Cognitive Science and Robotics: From Interfaces to Intelligence*, 110–111. AAAI Fall Symposium, report FS-04-05.

Tira-Thompson, E. J. 2004. Tekkotsu: A rapid development framework for robotics. Master's thesis, Carnegie Mellon University, Pittsburgh, PA. Available at www.Tekkotsu.org/media/thesis_ejt.pdf.

Ugur, E. 2005. Literature survey on affordances. Available at <http://www.kovan.ceng.metu.edu.tr/~emre/literature/Literature_On_+Affordances.html>.

Ullman, S. 1984. Visual routines. *Cognition* 18:97–159.

Veloso, M.; Lenser, S.; Vail, D.; Roth, M.; Stroupe, A.; and Chernova, S. 2002. CMPack-02: CMU's legged robot soccer team. In Kaminka; Lima; and Rojas., eds., *RoboCup 2002: Robot Soccer World Cup VI*. Springer-Verlag.