Behavior-based Approaches to Vision in Habile Robots

by

Thomas E. Horton

Abstract. The development of artificial cognitive mechanisms that support the intelligent use of tools is an necessary step in the design of autonomous habile (i.e. tool using) robots, as well as a potential opportunity to further our understanding of similar biological mechanisms in humans and other tool-using animals. For habile agents operating in environments designed for humans, visual mechanisms will be particularly important. We begin with a survey of the existing research in the areas of agent design and computer vision, focusing on theories of behavior-based control and animate vision that could provide a basis for the design of habile agents. We suggest ways in which such theories could contribute to the development of autonomous habile robots and conclude with suggestions as to how such a system could be implemented based on the Sony Aibo robot platform.

1 Introduction

The evolution of increasingly complex tool-using behaviors conferred a significant advantage upon the hominid lineage. Today, humans fill their environments with tools; from wrenches, to tennis rackets, to computers[1]. If we wish to create autonomous robots that can function in real human environments at anything near our level of skill, support for tool-use must be an integral part of their design. Thus, the development of such "habile" (i.e. tool-using) agents would represent a significant advance in robotics. At the same time, the development of artificial habile agents has the potential to provide insight into the biological mechanisms that support tool-use in humans and other animals, as well as providing a test bed for exploring theories of biological cognition from psychology, neurobiology, and related fields.

More than simply creating specialized, task-specific machines that can employ tools, our ultimate goal is the development of highly robust and generalized tool-using agents. Though in some sense, machines such as those that might be found on an assembly line may be able to "use" tools by following an explicit sequence of motions, they rely upon completely known and predictable environments. "Fully" habile agents should be able to handle novel situations, such as when a new type of task is presented, or when the most appropriate tool is not available and a substitute must be found or manufactured.

How then, should we initially approach this task? We begin with the following observation: in terms of an agent and its environment, tool-use (at least among humans and other

[1] Computers can be considered a type of cognitive (rather than physical) tool; enhancing an agent's innate mental abilities by increasing memory, computation speed, etc. While an interesting area of study, we will limit ourselves to consideration of physical tools within this paper.

high-order tool users) tends to be both a highly interactive and a highly visual endeavor. In a typical tool-using task, a habile agent (whether artificial or biological), must:

- recognize an external target in the environment on which it needs to act to achieve its current goal;
- realize that on its own, the agent is incapable of, or poorly suited to the required action;
- determine what kind of tool will enhance its effectivity;
- locate (or manufacture) an appropriate tool from the environment (possibly choosing from among several candidate artifacts);
- acquire the tool (possibly requiring navigation through the environment to the tool and back);
- position and orient the tool and target; and finally,
- apply the tool to the target in an effective manner, readjusting the tool as needed during the course of the action, and possibly repeating the action until the desired effect is observed.

Thus, a habile agent is constantly relying on input from its visual sensors to guide the manipulation of artifacts in the environment with its effectors, ideally in real-time. This suggests that a situated, behavior-based, and visually-oriented approach to design will be well suited to habile agents.

Behavior-based approaches attempt to exploit the interaction of agent and environment to simplify the task of cognition, and are founded on an ecological view of the agent as an embodied entity in a world rich with observable cues (affordances) that can help to guide the agent's behavior. Such approaches rely heavily on simple, efficient perceptual mechanisms (as opposed to focusing primarily on complex internal mental constructs), particularly visual mechanisms. A strong dependence on vision is a trait in common with human tool-using behavior, making this a reasonable approach to the construction of agents that operate in human environments and that use tools designed for humans.

Section 2 explores behavior-based approaches in more detail: beginning with a general overview of behavior-based design, and then looking more specifically at behavior-based

approaches to computer vision. Section 3 looks at related research in the areas of computer vision and habile robotics. Section 4 summarizes the beginnings of a design for a general habile agent implemented on the Sony Aibo robot platform, and section 5 concludes with some open questions and the current direction of our research.

2 Behavior-based design

In this section, we begin with an overview of common approaches to robotic design, with emphasis on behavior-based approaches. We then focus more specifically on the application of behavior-based approaches in the context of computer vision.

2.1 Approaches to agent design

The complexity, predictability, and stability of an environment strongly influence the design of an agent; the more complex and uncertain the environment, and the less time the agent has in which to act, the greater the demands placed on the design[14]. Because they must operate in real world environments (which tend to be both noisy and fast paced), the design of agents that control robots is often particularly difficult.

Matarić[14] groups robot control methodologies into four general classes. Which approach is most appropriate for a given design depends on the complexity of the environment, the nature of the task, and the physical and computational capabilities of the robot's hardware:

- *Reactive control* emphasizes speed of response over deliberative planning, allowing a robot to respond quickly to dynamic and unpredictable environments. This approach borrows heavily from stimulus-response theories of simple animal behavior (e.g. running from a light source). Such systems can be powerful, but are limited by a lack of internal models, memory, and learning capacity.

- *Deliberative control* utilizes internally stored knowledge in addition to sensory input to reason about a course of action before making a move. This allows the agent to optimize its behavior to a given situation, however the necessary planning can be computationally expensive and is often dependent on an accurate, detailed, and up-to-date internal model of the world. Such agents may not function well in noisy or

rapidly changing environments, where by the time a decision is made, circumstances have already changed.

- *Hybrid control* attempts to combine the strengths of the reactive and deliberative approaches. Typically, a reactive component and a deliberative component interact via an intermediate layer that oversees the two. The reactive component makes time critical actions, while the deliberative component computes a long-term strategy. Coordinating the (possibly conflicting) outputs of these two components can be difficult.

- *Behavior-based control*, like reactive control, is inspired by biological theories of animal behavior, but goes beyond simple stimulus-response pairs. The basic units of such control systems are "behaviors" - patterns of activity in the interaction between a robot and its environment. Beginning with a basic set of simple behaviors (e.g. collision avoidance), new, increasingly complex, behaviors are added until their overall interaction results in the desired performance (e.g. a high-level goal of navigating to a food source might depend on the behavior for collision avoidance). Although the selection and design of such a set of behaviors may be a complex task, behavior-based control produces agents that are robust, flexible, and responsive.

The behavior-based approach is modular, like reactive systems, but allows for the use of representations when appropriate, as in deliberative systems. Unlike deliberative systems however, individual behaviors may maintain their own state, and representations and planning are distributed across a network of parallel behavioral modules rather than centralized, allowing for greater flexibility and responsiveness. Components may be layered as in a hybrid architecture, but all layers operate on similar time-scales and use similar representations, simplifying coordination. Reasoning, sensing, and acting all share the same mechanisms, creating a tight coupling between thinking and execution. (For a far more thorough introduction to behavior-based robotics, see Arkin[1].)

In deliberative systems, planning is a discrete phase, performed prior to execution. In a behavior-based design, reasoning can be a continuous effort, with the agent constantly recomputing the best course of action at each moment, simultaneous with the execution of the current task[7]. This avoids the need to maintain a control state that keeps track of the agent's progress in a (fixed) sequence of actions and relies on potentially out-of-date information. Such an agent can demonstrate flexibility in the face of changing conditions,

while still performing complex behaviors: the agent can abort a routine that is no longer appropriate; reattempt a failed action; temporarily suspend one task in favor of another; interleave tasks; and combine tasks to simultaneously achieve multiple goals (see for example, Chapman's video game playing agent, "Sonja"[7]).

The terminology in the literature on behavior-based methodology is often a bit fuzzy. Some authors prefer to refer to such methods as "situated" approaches. Others may use both terms, but with slightly different interpretations. For example, Matarić points out that since robotic agents are embodied (interacting with the environment through a physical body), they are inherently "situated," regardless of the specific design emphasis. Whatever the terminology, these approaches generally have the following in common:

- a view of the agent as an embodied entity, closely integrated with its environment;
- mechanisms inspired by biology;
- use of the same cognitive structures for perception, action, and reasoning; and
- support for real-time responses in dynamic and noisy environments.

2.2 Computer vision

There are a variety of means by which an robot can perceive its environment. For example, range maps can be constructed with sonar or laser scans, the weight of a tool can be judged by lifting it, or the elasticity of an object might be determined by applying force to its surface and measuring the deformation. In humans, tool-use is a highly visual enterprise[2], and vision provides a rich and continuous stream of information about an agent's environment. Thus we expect that to be successful, a habile agent operating in a human environment should rely heavily on vision. Here we briefly contrast traditional and behavior-based[3] approaches to vision.

Traditionally, vision has been viewed as the task of reconstructing a three-dimensional scene from a two-dimensional projection of the environment onto the retina (or electronic

[2]Haptic and proprioceptive feedback is of course also vital, but their implementation is relatively straightforward.

[3]Here again, terminology in the literature varies. Different authors refer to ecological and interactive approaches, active and animate vision, etc., which, while differing in their details, all generally refer to biologically-inspired approaches that focus on the agent as an embodied and active participant in its environment. For consistency, we continue to use the term "behavior-based" to describe such approaches here, though this term is not widespread in the vision literature.

equivalent)[17]. However, automatically constructing a complete and detailed 3-d representation is often extremely difficult and computationally expensive. Many algorithms have been developed to break an image down into primitive elements such as edges and regions, but reconstituting the scene from these elements is almost always an under-constrained task, complicated by noise such as shadows and occlusion[4]. Behavior-based mechanisms provide a partial alternative to this computational approach to vision.

Though he focused on animals rather than computers, Gibson's work on ecological perception[9] is often cited as providing the theoretical basis for the current biologically-inspired approaches to machine vision[8, 15]. In Gibson's view, animal and environment form a mutually dependent "inseparable pair." Through this relationship, an animal can directly perceive all the information needed to act without the need for high-level mediating processes or representations; this information being inherent and readily accessible in the environment[5]. Perceptual cues in the environment (affordances) suggest courses of action to the agent.

Behavior-based approaches to computer vision aim to avoid the need to construct detailed 3-d representations of a scene. Creating and maintaining such representations is too expensive. Further, only a small subset of the information in a scene is usually relevant to the agent's current task, and resources spent analyzing the rest of the scene are simply wasted[3, 7]. Chapman also suggests that the information in a scene can only be usefully interpreted in terms of the goals the agent is trying to accomplish[7]. For example, objects in the real world are not discrete entities, but are composed of multiple smaller components, and are often themselves parts of larger wholes, making completely domain-independent object recognition difficult, if not impossible; one needs to know what level of abstraction is required for the task at hand in order to know how to interpret the scene.

Chapman cites several advantages of avoiding the use of internal representations by substituting continuous, direct perception of the environment via sensory input:

· If information about the world is needed to make a decision, it is usually cheaper (though not free) to "look and see" than to deduce, and the information can be relied upon as up-to-date (though potentially incomplete, if, for example, an object blocks

[4] A great deal of work has been accomplished in this area, far more than can be easily summarized here - see, for example, Marr[13] for a detailed treatment.

[5] Various perceptual mechanisms suggested by Gibson, for example the concept of optic flow, have subsequently been supported by animal studies [8].

the agent's view).

- In many cases, the next step the agent should follow is obvious from the configuration of the materials in front of it. And once that step is complete, the next is likely to follow from the resulting configuration. Thus, complex behaviors can arise without the need for complex control structures to keep track of the agent's progress through a sequential task (similar to concepts in Kirsh's"intelligent use of space"[12]).

- A situated agent does not necessarily have to perform simulations to know the outcome of an action. Rather, the action can be tested in the real world, simplifying computation and removing the uncertainty of an abstract simulation.

- Instead of solving a general case prior to action, skills can be learned and improved through exploration and observation.

- Real world environments are often simply too complex to be represented in detail by an internal model or accurately predicted in simulation.

2.2.1 Animate vision

An example of a behavior-based approach to machine vision is Ballard's "animate vision"[3]. Closely related to work in "active perception"[2], animate vision systems integrate vision with behavior, with particular emphasis on mobility. This distinguishes animate vision systems from passive vision systems, such as Marr's[13], which focus on the analysis of static images independent of their environmental context. In this way, animate vision avoids the standard vision problem of reconstructing a 3-d scene, by substituting behaviors that do not depend on an elaborate three dimensional representation of the environment, to begin with.

Ballard's main argument in favor of an animate approach is a significant reduction in computational complexity, supporting robots functioning in real-time[6]. In the philosophy of animate vision, behaviors may not always be successful, but they should always be fast.

Under the direction of a high-level controller specifying goals and instructions, animate vision systems utilize strategies borrowed from animal (particularly primate) visual systems

[6]While the steady increase in computing power has perhaps weakened this argument, the computational efficiency of animate systems can still be an important advantage, especially when using consumer-level hardware.

such as stereo-vision, foveae, and attention. Fundamental to animate vision are high-speed gaze control mechanisms that can rapidly reposition the camera coordinate system in response to stimuli. Combined with a fovea that provides high resolution vision over a very small viewing angle, gaze control allows for computationally cheap, low resolution imaging over a large field of view, while still providing imaging with high acuity at the area of interest.

Animate vision systems may utilize additional strategies to further reduce computation costs, such as exploiting environmental cues (e.g. colors) that are readily apparent in the local context. For example, Rabie and Terzopoulos developed a virtual fish employing an animate vision approach with directable, foveated eyes, and utilizing stereo and color cues, capable of target following, obstacle avoidance, and predator avoidance in a simulated aquatic environment[25]. Animate systems can also employ physical search strategies by moving the cameras or the entire robot to get a better look instead of running a computationally complex search over a single (and possibly incomplete) image. Further, the act of fixating the robot's gaze on a target facilitates the use of object-centered (rather than egocentric) coordinates, which can simplify calculations and support object representations that are independent of the robot's (possibly changing) perspective.

2.2.2 Vision and tool use

There are many ways in which behavior-based vision techniques could be applied to support tool-use. For example, by determining the basic size or shape of the tool the agent needs, a visual search for likely objects can be simplified - if the agent needs to extend its reach by a meter, objects significantly less than a meter in length can be excluded from consideration as soon as their size is estimated, at which point, no further analysis is needed. Similarly, an agent's reasoning about what kind of tool to use might be influenced by first performing a survey of the environment to determine what sorts of tools are available, thus letting the environment constrain the internal search space.

By altering the agent's effectivity, possession of a tool may also influence a visual search. For example, if an agent's goal is to acquire objects of distinct kinds, each of which requires a different tool (e.g. hooks of different size), then as long as the agent is holding the tool for one type of target, it can ignore the other objects until it has finished gathering all of the current targets and swapped tools.

3 Previous work

In this section, we summarize a number of cases in which researchers have implemented systems that demonstrate examples of visually-aided tool-use. This survey is not exhaustive, but is hopefully representative enough to provide an overview of the state-of-the-art in the area.

3.1 Bogoni and Bajcsy

Bogoni and Bajcsy examined ways of representing tools in terms of function[5, 6, 4]. Believing that object representations should go beyond the description of intrinsic properties such as shape and material, their goal was to include empirically determined descriptions of an object's functional features. To this end, they developed a system for automatically discovering the functional properties of objects through task-oriented interactions using active perception.

Initial experiments studied tools for piercing, while later experiments also explored the use of chopping tools. The setup used two robotic manipulators, one of which gripped the tool and was equipped with a force sensor, while the other was equipped with a camera for visual feedback.

The system maintained an event-based state description of the task under analysis. A supervisory controller stepped the manipulators through the task and selected the appropriate observation strategy at each stage. The experiments varied the material properties of the tool and target objects (e.g. shape and hardness), as well as the mechanics of the interaction, such as the speed and the angle of the tool's approach.

Vision was used both prior to and during the interaction phase of the experiments. To begin with, the visual system was used to segment the outlines of the tools. This allowed for shape-based comparisons of the different tools tested by the system. It also allowed the system to monitor the tools for deformation during their application to the target. During this tool-application phase, vision was used in combination with the force and position sensors to observe the progress of the robot by measuring the distance between the tool and the target. Visual feedback was also used to interpret the results from the other sensors. For example, during a trial, the force and position sensors might indicate a successful piercing operation with the tool, while a visual comparison between the tool and its stored

representation showed that the force and position readings were actually due to the tool bending, rather than penetrating the target material.

3.2 GRUFF (Sutton, Stark, Bowyer, et. al.)

GRUFF (Generic Recognition Using Form and Function) is a system developed and expanded over several years by Sutton et. al. for the visual recognition of novel objects[23, 24]. Rather than using explicit geometrical models of specific objects, GRUFF tries to assign objects to generalized categories based on their functional characteristics (it is assumed that these functional characteristics can be extracted from an object's geometry). This functional analysis involves determining whether a given object satisfies the constraints of the object category (for example, a container must be graspable to be a cup). The functional characteristics GRUFF looks for are similar in concept to Gibson's affordances.

Though early work focused exclusively on the object category of chairs and used pre-built 3D models of objects, later research added automatic generation of the object models and expanded the knowledge base to include various types of objects including hand tools. Recent work has also looked at object recognition using environmental context to offset the uncertainty found in real world scenes (for example, the problem of object occlusion).

In operation, an object is placed in an observation area, where a camera system is used to acquire a range map, which is then used to construct a 3D model of the object. Next, concepts of physics and causation are applied to the model to label the object's potential functionalities (for example, in order to allow sitting, the object must rest stably on the ground). The system checks the object geometry for the existence of required surfaces, the proper relationships of the surfaces, and the appropriate accessibility or clearance of the functional elements. Finally, a plan for confirming the object's functionality through interaction is created and enacted using a robotic arm and feedback from visual and haptic sensors.

3.3 Kemp and Edsinger

Kemp and Edsinger observe that for many tools and tasks, a basic, yet essential, component of tool-use is the identification and control of the tool's endpoint (this ignores tools such as saws, where effectivity is not localized to a single point)[11]. To take advantage

of this observation, they designed a robotic system for automatically detecting a tool's endpoint and estimating its position in space relative to the robot's hand. With this information, the robot has control over the position of the endpoint, which might enable the use of very simple tools (a hook, for example) as well as support further automated exploration and testing of more complex tools (even in the case of basic tools such as hammers and screwdrivers, knowledge of the point of effectivity alone is insufficient for effective use).

While many other robotic systems either require a pre-existing model of a tool or must construct a model using complex perceptual processing, Kemp and Edsinger use a relatively simple optical flow technique to rapidly model the tool's endpoint, even under noisy conditions simulating a real world work environment. While rotating the tool, the robot monitors the optical flow to detect the points in the image that move the fastest. After studying the tool from several different angles, the system estimates the 3-d position of the endpoint. This use of optical flow simplifies the problem of perception and allows for real-time detection of the end point without the need for detailed modeling of the tool. The technique has been tested with a variety of tools, including a pen, hammer, and pliers, as well as with other objects such as a bottle and the robot's own finger.

3.4 Wood

Recent work in our lab by Wood[19, 26], led to the development a simple habile agent based on a Sony Aibo robot dog. The task set for the Aibo was the acquisition of its toy ball. However, as part of the task, the ball was initially placed on a stand, putting it out of the robot's reach. Thus the agent had to first determine that the ball was out of reach, then locate and acquire a suitable tool (in this case, a "stick" of sufficient length and with suitable dimensions to allow the Aibo to grasp it in its mouth was placed nearby), and finally use the tool to push the ball off of the stand and onto the floor where it was within the Aibo's unaided reach.

The agent's tool-using strategy was implemented using a simple finite state machine, with the Aibo first performing a visual search for the ball[7]. Once the ball was located, a test was performed against an internal body schema to determine whether the ball was within reach. If not, a second visual search was performed to locate the stick tool's "handle". This

[7]The ball was a distinct color from the rest of the environment, so the visual search was based on color segmentation algorithms provided by the Tekkotsu Aibo development framework from Carnegie Melon.

handle had a uniquely colored visual affordance and was shaped to fit in the Aibo's mouth. The Aibo then moved to the stick tool, oriented itself using its body schema and picked up the tool. A third visual search reacquired the ball and the Aibo moved within range. Using the body schema, the Aibo then aimed the tool and leaned forward, poking the ball with the stick and causing it to roll off of the platform.

While much of the agent's behavior was hard coded into the state machine, strictly limiting the tasks that it could handle, the agent did demonstrate several important features:

- recognition of a problem that can benefit from application of a tool, using vision and an internal body schema representation,
- acquisition of a tool in an effective orientation based on a visual affordance,
- extension of the body schema to accommodate the altered effectivity resulting from tool acquisition, and
- use of an acquired tool to accomplish an otherwise unachievable goal.

3.5 Stoytchev

One of the most impressive habile agents yet developed is Stoytchev's[22]. His work concentrated on the autonomous learning of tool affordances by a robot, in this case a 5 d.o.f. manipulator arm and gripper mounted on a mobile base that allowed the robot to move from side-to-side. The robot's tasks consisted of basic extension-of-reach activities.

In the experiments (based on similar studies done with chimps[16]), a hockey puck was placed on a table in front of the robot. The robot's ultimate objective was to move the puck using a provided tool. Five different tools were used in the experiments: a straight stick, an L-shaped stick, a T-shaped stick, and versions of the L and T-sticks with extra (serif-like) extensions to form hooks. The robot's wrist, the tools and the puck were color coded to simplify visual tracking using a camera, which was mounted above the workspace. The robot had five basic behaviors: extend arm, contract arm, slide left, slide right, and position wrist. Only the effective affordances of the tools were learned (i.e. the robot already knew where the tool's handle was).

The design was based on research that shows many animal species use stereotyped exploratory behaviors to experiment with an unfamiliar object. Accordingly, when presented

with a new tool, the robot engaged in a "behavioral babbling" stage, in which behaviors were randomly selected and the effects on the environment were observed. Once a tool's affordances were learned, the behavioral primitives could be strung together based on their expected outcomes to solve a given task. Similarly, the accuracy of the learned representation could be tested by the agent and the representation updated if needed - for example, the agent could recognize that a broken tool no longer performed as expected.

In the basic extension-of-reach tasks, the robot was given one of the tools and had to push/pull the puck over a colored goal region on the table. The robot was usually successful, and the most common cause of failure was pushing the puck out of reach. In a second set of experiments, the robot was given a "broken" tool - an L-hook, colored such that the robot recognized it as a T-hook. In all trials, the robot was able to recognize that affordances associated with the "missing" part were no longer valid and adjust its behavior accordingly.

As Stoytchev notes, there are some short-comings to relying on a preexisting set of exploratory behaviors - in particular, there may be tool affordances that will not be discovered because the required exploratory behavior is not within the agent's behavioral repertoire (an issue also observed in animal studies). He suggests that this can be resolved by the addition of new mechanisms that support the learning of novel exploratory behaviors.

4 Designing a habile agent

Tool-use can be described as the effective application of an external object to enhance the innate abilities of the user. We have previously divided physical tools and tool-using behaviors into four categories[18]:

Effective - The use of tools such as hammers, screwdrivers, and paint brushes produce a persistent effect on the environment.

Instrumental - Tools like tape measures and magnifying glasses provide the user with enhanced information about the environment.

Constraining - Stabilizing or constraining the movement of environmental objects or other tools. An example is the use of a straight edge to draw a line.

Demarcating - Imposing structure on the environment. This category includes using a

carpenter's pencil for marking where to cut a board, or using push pins of different colors to denote different types of landmarks on a map.

These behavioral categories encompass a wide variety of tasks, but all require an agent to follow the same basic pattern. In a typical tool-using scenario, a habile agent performs the following sequence of tasks:

- recognize an external target in the environment on which it must act to achieve its current goal (for example, a hungry agent might perform a visual search for food, eventually locating a bunch of bananas hanging from a hook in the ceiling)
- realize that on its own, the agent is poorly suited to, or incapable of the required action (maybe the bananas are too high to reach);
- determine what kind of tool will enhance its performance (by extending its reach, a long object might let the agent knock the bananas down to the floor[8]);
- locate (or manufacture) an appropriate tool from the environment (hey, there's a stick lying in the corner);
- acquire the tool (fetch!) ;
- position and orient the tool (holding the stick by the middle or distal end won't be effective, so the stick should be held by the proximal end); and finally,
- apply the tool to the target in an effective manner, readjusting the tool as needed (push the bananas up and off the hook).

Tool-using tasks can easily become very complex, so for now, we aim to limit our considerations to a small representative set of basic tasks. Such tasks might include reaching or pushing, pulling, lifting (e.g. with a lever), picking up (e.g. with a scoop), probing, marking, and blocking (e.g. propping open a door). Our goal is the development of a flexible and robust habile robot, capable of autonomously performing these kinds of tasks (in real-time if possible), incorporating behavior-based elements in its design.

[8] Bruised bananas - ick!

4.1 Design constraints

There are numerous constraints that guide our thinking when trying to design a habile robot. Some of these arise from assumptions we've made about how a habile agent should function. Other constraints are imposed by the choice of hardware platform for our agent.

4.1.1 Assumptions

We make the following general assumptions about the design of visually-oriented habile robots.

Independence of high-level cognition from low-level processing

It should be possible to fix the low-level algorithms for handling the basic visual and motor processing of a habile agent, while allowing for flexibility in the higher level cognitive representations. Thus, we should be able to retain the basic support framework, while swapping various cognitive modules. This would allow for the development of increasingly sophisticated representations supporting more sophisticated tool-use, without the need to completely redesign the agent. Additionally, while one of our goals is the development of generalized agents able to handle a wide variety of tasks and environments, there could be times where the ability to swap in a specialized cognitive module for a particular situation might be advantageous (one can also conceive of an architecture that includes many such modules, with a mechanism for automatically selecting the one best suited for the task at hand).

Kinematic body schema

Recent research in neurobiology has found significant evidence that an extensible kinematic body schema is employed by some primates when interacting with objects in the environment[10]. This suggests that as part of its basic low-level architecture, a habile agent will benefit from the incorporation of a similar type of representation. Earlier work by Stoytchev[21] and by Wood[26] already incorporates extensible body schemas into habile agents.

A body schema gives an agent, whether artificial or biological, an internal representation of the agent's body in physical relation to its immediate environment. Such a representation can be useful for avoiding, acquiring, and manipulating objects in the environment.

An *extensible* body schema is a representation that can be temporarily adjusted to accommodate an external object that is treated as part of the agent's own body. Thus, when an agent acquires a tool for the purposes of tool-use (i.e. not simply holding an object, as during transport), the tool can be considered an extension of the agent's body without the need for tool-specific mechanisms. When the tool is released, or is handled not as a tool but as simply an object, the body schema returns to its normal bounds. It will be necessary to develop visual mechanisms to support and utilize such a representation.

Symmetry

We also believe that geometrical notions of symmetry and proportionality will play an important role in the recognition and utilization of visual affordances, though we still lack a formalized description of these concepts. For example, when presented with a hammer, a human might note the symmetry between the curvature of the handle and the shape of the palm, thus perceiving that the handle affords grasping in a particular orientation. In addition to shape, size is also key to the proper recognition of affordances; consider a handle-shaped object that is too large or too small for a manipulator to grasp securely, or a screwdriver a size too big to fit a screw's head. Further, symmetry and proportionality might provide useful constraints on visual search; for example, when searching for candidate tools, ignoring all objects above a certain size, such that they would be difficult for the robot to move.

4.1.2 Platform constraints

The hardware platform we have chosen as the basis for our habile robot is the Sony Aibo ERS-7 robotic dog (continuing the earlier research by Wood[26]). The Aibo provides a tight integration of processor, sensors, actuators, and wireless communication in a single compact and affordable package[9]. Significant development support is available through existing frameworks and documentation created by both Sony and the Tekkotsu project at Carnegie Mellon. Additionally, as its form is that of a stylized canine, the Aibo also offers an opportunity to study tool-use as separate from the anthropomorphic features (e.g. opposable thumb) with which it is usually associated. This choice of platform does however impose significant limitations.

[9]Now, unfortunately discontinued.

Real-time execution

While the Aibo has significant processing power on-board, computer vision algorithms tend to be computationally expensive. It may be necessary to offload some or all of the visual processing to a networked computer (in which case data transfer rates may become an issue). Such problems could prevent the robot from functioning in real time, or force us to use lower-resolution images. Low-level preprocessing performed on-board might help reduce such constraints. For example, using a foveated visual system would mean that only a relatively small portion of the Aibo's view need be transmitted at a high resolution, while peripheral vision could be transmitted at a much lower resolution.

Trajectory deviation

Depending on the walking surface and the walking gait, the Aibo's locomotion becomes error-prone, leading to significant deviations from the desired path, even over short distances. This means that navigation will depend heavily on continuous closed-loop control to perform corrections to the Aibo's trajectory and maintain course. Consequently, we will need to process visual input quickly or else limit the walking speed or pause frequently for adjustment.

Object manipulation

The Aibo lacks a traditional gripper-style manipulator, and we are presently considering various different designs for Aibo-compatible tools. For example, in place of a handle, a tool might be equipped with a sleeve, into which the Aibo can fit a paw. Another approach might be to design tools that can be held between the Aibo's forelimbs. Most likely, we will continue to use the Aibo's mouth to grasp tools as in Wood's work (though this has the disadvantage that the tool can interfere with the robot's nose-mounted vision). There is an upside to the limited range of shapes that the Aibo can pick up - it significantly constrains the search space when identifying potential tools in the environment. Any object that fails to meet the narrowly defined graspable affordance can be ignored[10].

While the location of the camera in the nose may sometimes mean that range of vision is reduced when a tool is held in the Aibo's mouth, it also simplifies the problem of guiding

[10]The search space can also be artificially constrained when appropriate (for example by using color to denote a graspable component).

the tool's endpoint. Assuming that the tool's point of effectivity is not offset to the the side (i.e. the endpoint is in-line with the Aibo's camera), the Aibo can guide the tool simply by maintaining the alignment between camera, tool, and target; essentially sighting along the tool's length.

Depth estimation

The Aibo is equipped with three forward-facing infra-red depth sensors (each optimized for a different range of distances). In our experience however, while these are generally sufficient for obstacle avoidance, they are ill-suited to precision tasks, such as picking up objects[11] and shape-based object recognition. Especially given our conceptual emphasis on vision, we would like to be able to precisely and accurately gauge distances using the Aibo's visual hardware. Unfortunately, the Aibo has only a single camera, ruling out real-time stereo-vision (a difficult computational problem in itself). One possible solution when dealing with objects of known dimensions is to estimate depth based on the area the object covers in the Aibo's view.

Another possibility is related to the "peering" behaviors exhibited by some insects and birds, in which the head is moved from side-to-side in order to gain better estimates of depth from parallax or from optical flow techniques[12]. We are experimenting with a behavior in which the Aibo leans its whole body from side-to-side (the neck not allowing this type of movement), while recording images at two or more points along the camera's path. This allows a kind of pseudo-stereoscopic vision, but requires the relaxation of our real-time goal and depends on a reasonably stable environment. It may also be possible to take advantage of camera displacement during walking to obtain depth information - thus, instability in the image during walking, usually just a source of noise, could potentially become a useful tool.

5 Conclusions

There remain a number of open issues in the design of habile agents which we have not yet mentioned. In focusing on late and intermediate vision, we have thus far ignored

[11] This sort of precision control seems to be a rather difficult problem - even Sony's own commercial Aibo "mind" often fails at this sort of task.

[12] Nearer objects appear to move more rapidly than those further way, proportional to distance.

the implementation of the early visual mechanisms on which higher level visual processes depend (e.g. edge detection, color segmentation). In biological organisms, early vision is performed by dedicated hardware which we would rather emulate in software. As a great deal of work has already been done on such mechanisms, we are currently looking into the use of existing software to perform these functions.

We have also said little about low-level motor control. Possibilities here include hard-wired controllers such as the MMC Net[20] used by Wood[26], as well as learning approaches such as those used by Stoytchev[21] that learn motor control during an exploratory babbling phase. Similarly, we will need a high-level controller to coordinate and guide the perceptual and motor systems. Finally, we will need to implement some form of learning so that the agent can build upon past experiences and expand its abilities. Stoytchev's work in the learning of affordances in extension-of-reach tasks[22] might serve as a good starting point.

Along with developing the basic architecture for our Aibo-based system, our current goal is the creation of a formal description for the kinds of visual affordances the agent should be able to recognize (e.g. for grapsing, reaching, etc.). This should help guide the development of the visual and reasoning systems, as well as the design of the tasks we will eventually use to test the robot.

In summary, to effectively perform human-like tasks in environments designed for humans, artificial agents will need the ability to use tools. The development of habile agents also affords an opportunity for exploring and testing cognitive theories of tool-use in humans and other animals. Finally, the highly interactive and visual nature of tool-use suggests that developing behavior-based vision systems is a promising direction for research into the design of autonomous habile robots.

Bibliography

[1] R. C. Arkin. *Behavior-based Robotics*. MIT Press, Cambridge, MA, USA, 1998.

[2] R. Bajcsy. Active perception. In *Proceedings of the IEEE*, volume 76, pages 996–1006, 1988.

[3] D. H. Ballard. Animate vision. *Artificial Intelligence*, 48(1):57–86, 1991.

[4] L. Bogoni. More than just shape: a representation for functionality. *Artificial Intelligence in Engineering*, 12:337–354, 1998.

[5] L. Bogoni and R. Bajcsy. Active investigation of functionality. In *Proc. Workshop on the Role of Functionality in Object Recognition, IEEE Computer Vision and Pattern Recognition Conference, Seattle, WA*, 1994.

[6] L. Bogoni and R. Bajcsy. Interactive recognition and representation of functionality. *Comput. Vis. Image Underst.*, 62(2):194–214, 1995.

[7] D. Chapman. *Vision, instruction, and action*. MIT Press, Cambridge, MA, USA, 1991.

[8] A. Duchon and W. Warren. Robot navigation from a gibsonian viewpoint. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (IEEE, Piscataway, NJ, 1994)*, pages 2272–2277, 1994.

[9] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

[10] A. Iriki, M. Tanaka, and Y. Iwamura. Coding of modified body schema during tool use by macaque postcentral neurons. *Neuroreport*, 7:2325–2330, 1996.

[11] C. C. Kemp and A. Edsinger. Visual tool tip detection and position estimation for robotic manipulation of unknown human tools. Computer Science and Artificial Intelligence Laboratory Technical Report 83, Massachusetts Institute of Technology, December 2005.

[12] D. Kirsh. The intelligent use of space. *Artificial Intelligence*, 73(31–68), 1995.

[13] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982.

[14] M. J. Matarić. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 9:323–336, 1997.

[15] R. Murphy. Case studies of applying gibson's ecological approach to mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 29(1):105–111, April 1999.

[16] D. Povinelli, editor. *Folk Physics for Apes*. Oxford University Press, NY, 2000.

[17] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[18] R. St. Amant and T. E. Horton. Characterizing tool use in an interactive drawing environment. In *Proceedings of the 2nd international symposium on Smart graphics*, pages 86–93. ACM Press, 2002.

[19] R. St. Amant and A. B. Wood. Tool use for autonomous agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 184–189, 2005.

[20] U. Steinkühler and H. Cruse. A holistic model for an internal representation to control the movement of a manipulator with redundant degrees of freedom. *Biological Cybernetics*, 79:457–466, 1998.

[21] A. Stoytchev. Computational model for an extendable robot body schema. College of Computing Technical Report 44, Georgia Institute of Technology, October 2003.

[22] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings fo IEEE International Conference on Robotics and Automation*, 2005.

[23] M. Sutton, L. Stark, and K. Bowyer. Function from visual analysis and physical interaction: a methodology for recognition of generic classes of objects. *Image and Vision Computing*, 16:745–763, 1999.

[24] M. A. Sutton, L. Stark, and K. Hughes. Exploiting context in function-based reasoning. In G. H. et.al., editor, *Sensor Based Intelligent Robots*. Springer-Verlag, 2002.

[25] D. Terzopoulos and T. F. Rabie. Animat vision: Active vision in artificial animals. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, pages 801–808, Washington, DC, USA, 1995. IEEE Computer Society.

[26] A. B. Wood. Effective tool use in a habile agent. M.S. thesis, Department of Computer Science, North Carolina State University, 2005.