Classification and Prediction of Human Behaviors By a Mobile Robot

D. Paul Benjamin[1,*], Hong Yue[1], Damian Lyons[2],

[1] Pace University, 1 Pace Plaza, New York, NY, 10038, USA
[2] Fordham University, 340 JMH, 441 E. Fordham Rd. Bronx, NY, 10458, USA
dbenjamin@pace.edu, yh19243n@pace.edu, dlyons@fordham.edu

Abstract. Robots interacting and collaborating with people need to comprehend and predict their movements. We present an approach to perceiving and modeling behaviors using a 3D virtual world. The robot's visual data is registered with the virtual world to construct a model of the dynamics of the behavior and to predict future motions using a physics engine. This enables the robot to visualize alternative evolutions of the dynamics and to classify them. The goal of this work is to use this ability to interact more naturally with humans and to avoid potentially disastrous mistakes.

Keywords: virtual world · Soar Cognitive Architecture

1 Introduction

The ADAPT project (**A**daptive **D**ynamics and **A**ctive **P**erception for **T**hought) is a collaboration of three university research groups at Pace University, Brigham Young University, and Fordham University to produce a robot cognitive architecture that integrates the structures designed by linguists and cognitive scientists with those developed by robotics researchers for real-time perception and control. ADAPT is under development on Pioneer robots in the Pace University Robotics Lab and the Fordham University Robotics Lab. Publications describing ADAPT are [1,2,3,4].

2 The ADAPT Architecture

Our approach is fundamentally different from other projects, which typically attempt to build a comprehensive system by connecting modules for each different capability: learning, vision, natural language, etc. Instead, we are building a complete cognitive robotic architecture by merging RS [6,8,9], which provides a model for building and reasoning about sensory-motor schemas, with Soar [5], a cognitive architecture that is under development at a number of universities. RS possesses a sophisticated formal language for reasoning about networks of port automata and has been successfully applied to robot planning. Soar is a unified cognitive architecture [7] that has been successfully applied to a wide range of tasks.

Soar's model of problem solving utilizes a single mechanism of subgoaling and chunking to explain human problem solving performance; utilizing Soar as the basis of ADAPT permits us to unify the mechanisms underlying perception, language and planning. Furthermore, it permits us to explore possible interrelationships between learning in these areas, e.g. how learning language and learning perception may be related. Finally, it permits us to test our architecture on robotic versions of well-known cognitive tasks and explore how robot learning might be related to human learning.

RS provides a powerful representational language for the system's dynamics, language and percepts; however, RS does not provide a mechanism for synthesizing the dynamics. Furthermore, RS lacks demonstrated cognitive plausibility, and in particular lacks a learning method.

We have implemented RS in Soar to take advantage of Soar's cognitively plausible problem-solving and learning mechanisms. Soar uses universal subgoaling to organize its problem solving process into a hierarchy of subgoals, and uses chunking to speed and generalize that process. Universal subgoaling permits Soar to bring all its knowledge to bear on each subgoal. Chunking stores generalized preconditions for search control decisions, so that in future tasks similar search control decisions are made in a single step.

3 Visualization and Prediction of Human Behaviors

We believe that the comprehension of human movements and intentions requires the ability to visualize human movements and predict possible future movements. We view visualization as consisting of both a perceptual component and a reasoning component. The perceptual component is performed using the same perceptual mechanism that the robot uses to perceive its environment; the difference is that visualization perceives a simulation of the environment. Visual reasoning manipulates and superimposes representations that consist of a combination of symbolic knowledge and 3D animations. This approach to comprehension requires the robot to be able to create different situations in which it can generate behaviors of robots, people and physical systems, and perceive the results of these behaviors. This requires implementing a virtual world that the robot can control.

ADAPT's virtual world is a multimedia simulation platform capable of realistic simulations of physical phenomena. It combines the various forms of map information found in most robots: topological, metric and conceptual information. ADAPT completely controls this virtual world, and can create arbitrary objects and behaviors in it, including nonexistent objects and behaviors that were not actually observed. Central to ADAPT's use of its virtual world is its ability to view these constructions from any point. This enables ADAPT to create visual representations with desired properties.

In the current implementation, ADAPT's world model is PhysX. PhysX gives the robot the ability to create a detailed and dynamic virtual model of its environment, by providing sophisticated graphics and rendering capabilities together with a physics engine based on the PhysX physics engine. PhysX models a wide variety of dynamic environments, including modeling other agents moving and acting in those environments.

This world model is used to represent the important entities and behaviors in the environment. The built-in physics capability of PhysX is then used to predict what is going to happen in the immediate future. Let's examine how ADAPT uses the virtual world together with its vision system to model and predict the environment. ADAPT's vision system consists of two main components, a bottom-up component that is always on, and a top-down goal-directed component controlled by Soar.

The bottom-up component is simple and fast. It does this by not producing much detail. The idea is for it to produce a basic stereo disparity map, a coarse-grained image flow, and color segmentation in real time. It runs on the robot's onboard computer using Intel's open vision library, and segments the visual data from the robot's two frame-grabbers. These "blobs" are transmitted together with stereo disparity data and optical flow to the off-board PC that is running Soar, where it is placed into working memory. This component is always on, and its output is task-independent.

The top-down component executes the more expensive image processing functions, such as object recognition, sophisticated image flow analysis, and application of particular filters to the data. These functions are called in a task-dependent and goal-dependent manner by Soar operators. This greatly reduces their frequency of application and speeds the operation of the vision system significantly. These two components are not connected to each other; instead, the output of the bottom-up component is used by Soar to determine when to call the top-down operations. Soar compares the bottom-up output to the visual data predicted by the virtual world.

The virtual world can display the view that the virtual copy of the robot "sees" in the virtual environment. The output of this graphics "camera" in PhysX is segmented and sent to the MMD (Match-Mediated Difference), together with distance information and motion information. The MMD tests for significant differences between the expected view and the actual view, e.g. the appearance of a large new blob or a large change in optical flow. It aligns the real and virtual images with an affine map, then finds a set of matched key points and place a normalized Gaussian at each of them. The normalized match quality is the inverse of the distance between matched points divided by the sum of all match errors. We use this as a coefficient of the Gaussians to create the MMD measure. Any significant difference is placed in Soar's working memory, where it can cause an operator to be proposed to attend to this difference.

Soar controls all major aspects of perceptual processing, with the goal of constructing a virtual model of the environment that can be used in task planning. These aspects include focusing on regions of interest, choosing the depth of field, and deciding on the degree of detail for each part of the virtual model. For example, if a new blob appears, an abstract operator will be proposed to focus on this blob and try to recognize it. If this operator is selected (if there is no more important operator to do at the moment) then RS/Soar will instruct the robot to turn its cameras towards this blob, focus at the appropriate depth, and obtain a point cloud from the visual input. Keypoints are extracted from the point cloud and used both in object recognition and to create a mesh that is registered with surrounding meshes.

Figure 1 shows two point clouds of a kitchen counter by this vision system. These are transformed into meshes. By joining small meshes together, the system can create larger meshes when necessary. Portions of the world that are not relevant to task goals are not rendered, greatly increasing efficiency.

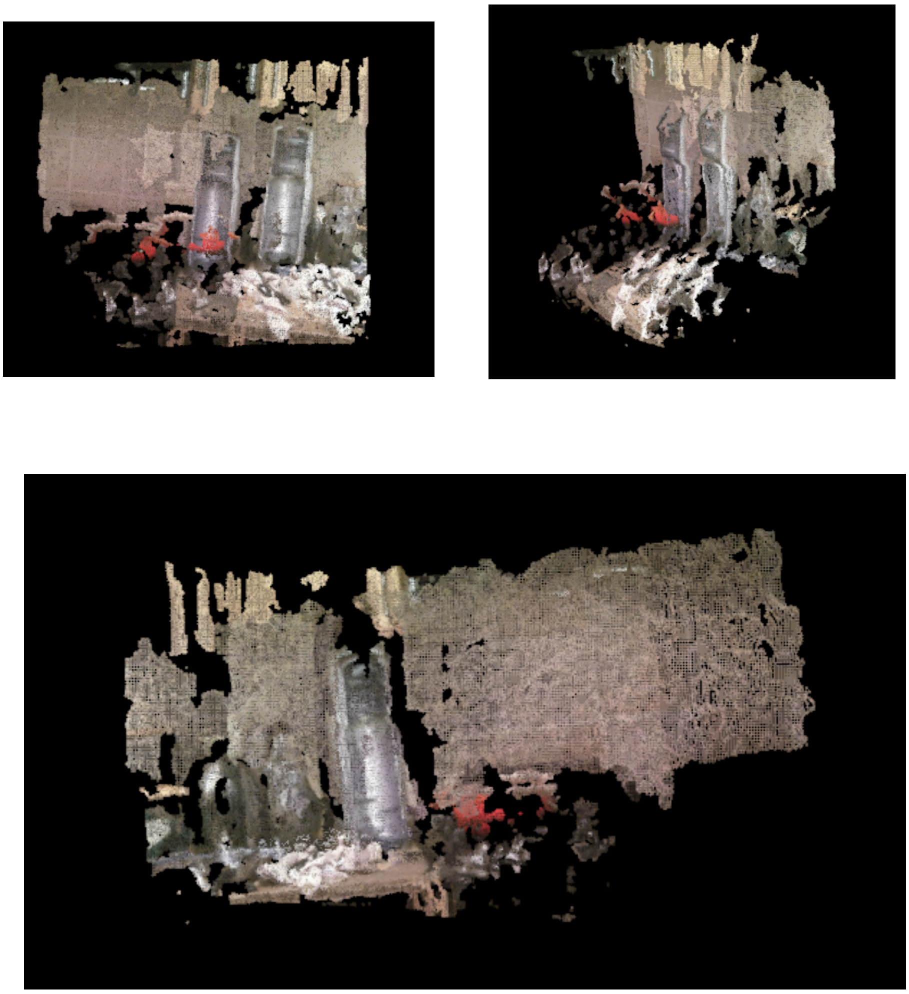

Fig. 1. The two point clouds at top are registered and joined to create the bottom cloud, which is transformed into a PhysX mesh.

This approach also works for people, as shown in Figure 2. People are skeletonized in a manner similar to that used by the Kinect, and the skeleton is covered with mesh. Our initial work was with the Kinect, but now uses only stereo vision.

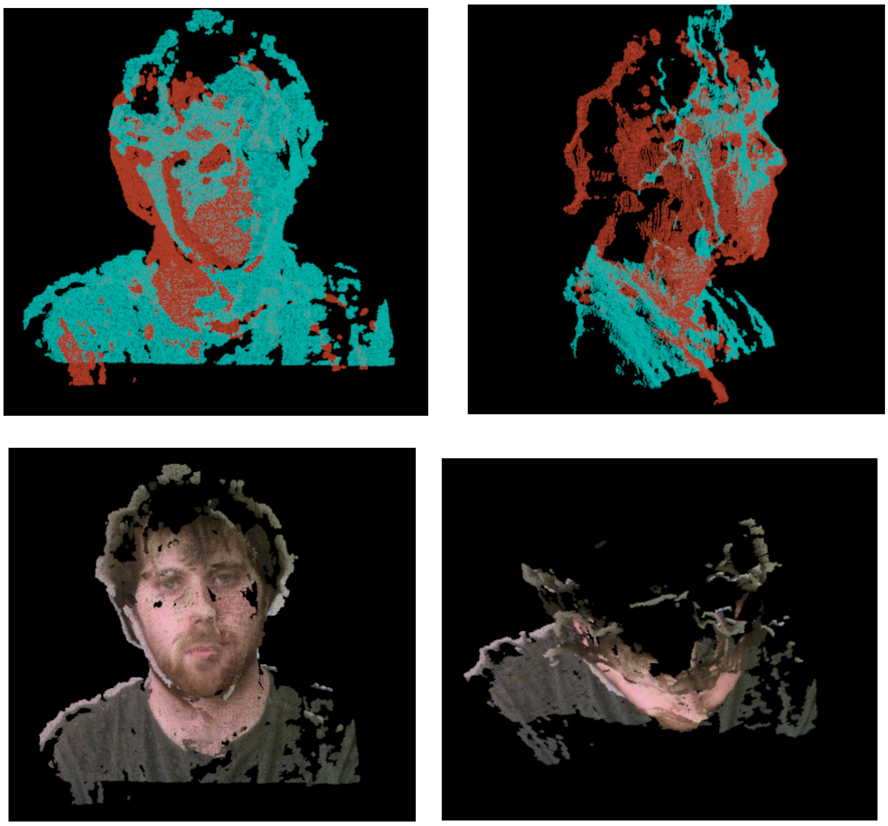

Fig. 2. Point clouds for a face, registered and joined to create a 3D mesh for a skeleton. The two point clouds at top are joined to create the cloud seen below, frontal view at left and from above at right.

Once an object is recognized or rendered, a virtual copy is created in PhysX. The object does not need to be recognized again; as long as the blobs from the object approximately match the expected blobs from PhysX, ADAPT assumes it is the same object. Recognition becomes an explicitly goal-directed process that is much cheaper than continually recognizing everything in the environment. The frequency with which these expensive operations are called is reduced, and they are called on small regions in the visual field rather than on the whole visual field.

Thus, ADAPT's vision system spends most of its time verifying hypotheses about its environment, instead of creating them. The percentage of its time that it must spend attending to environmental changes depends on the dynamic nature of the environment; in a relatively static environment (or one that the robot knows well from experience) there are very few unexpected visual events to be processed, so visual processing operators occupy very little of the robot's time.

Figure 3 shows the overall flow of control of our vision system [11,12]. The real and synthetic images of the scene as viewed by the robot are compared. If the scenes are considered the same but from different viewpoints, then the viewpoint of the camera in the simulation is changed, and the simulation generates an image taken by the camera at the new location. If an unexpected object is seen in the real image, an object is introduced at the corresponding position in the simulated scene. The region of the real image responsible for the difference is used as video texture on the object and a new synthetic image generated. The information on whether there is no difference, an unexpected object, or an object missing between the image pairs is made available to action planning.

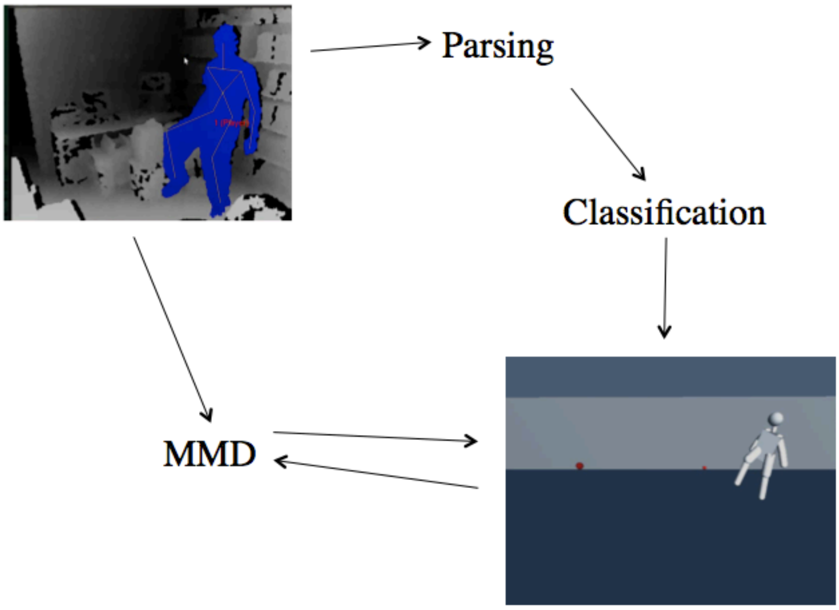

Figure 3. The input from the physical world (left above) is parsed, classified and rendered into the virtual world (below right). The MMD compares both inputs, detects differences and updates the virtual world.

This loop of difference detection and simulation modification is used to keep the simulation synchronized to the observed environment. For prediction purposes, the simulation can be allowed to 'fast forward' in time, so that the expected position, for example, of a target can be calculated and then compared to observations.

4 Summary

We have sketched the overall design of a new approach to the comprehension of behavior that is part of a robotic cognitive architecture. A powerful 3D multimedia world model is used to render the behaviors in the environment. This gives the robot the ability to visualize alternative evolutions of the dynamics and to classify them. The implementation of the basic components is complete.

Further information on this work, including video clips showing the robot moving under the control of schemas and the use of the world model, can be downloaded from the website for the Pace University Robotics Lab: http://csis.pace.edu/robotlab

References

1. Benjamin, D. Paul, Damian Lyons and Deryle Lonsdale, “Embodying a Cognitive Model in a Mobile Robot”, Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision, Boston, October, (2006)
2. Benjamin, D. Paul, Damian Lyons and Thomas Achtemichuk, “Obstacle Avoidance using Predictive Vision based on a Dynamic 3D World Model”, Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision, Boston, October, (2006)
3. Benjamin, D. Paul, Damian Lyons and Deryle Lonsdale, "Designing a Robot Cognitive Architecture with Concurrency and Active Perception", Proceedings of the AAAI Fall Symposium on the Intersection of Cognitive Science and Robotics, Washington, D.C., October, (2004)
4. Benjamin, D. Paul, Damian Lyons and Deryle Lonsdale, " Cognitive Robots: Integrating Perception, Action and Problem Solving in Behavior-Based Robots", AAMAS-2004 Proceedings, pp. 1308-1309, (2004)
5. Laird, J.E., Newell, A. and Rosenbloom, P.S., “Soar: An Architecture for General Intelligence”, *Artificial Intelligence* **33**, pp.1-64, (1987)
6. Lyons, D.M. and Hendriks, A., “Exploiting Patterns of Interaction to Select Reactions”, Special Issue on Computational Theories of Interaction, Artificial Intelligence **73**, pp.117-148, (1995)
7. Newell, Allen, *Unified Theories of Cognition*, Harvard University Press, Cambridge, Massachusetts, (1990)
8. Lyons, D.M., “Representing and Analysing Action Plans as Networks of Concurrent Processes”, IEEE Transactions on Robotics and Automation, June (1993)
9. Lyons, D.M. and Arbib, M.A., “A Formal Model of Computation for Sensory-based Robotics”, IEEE Transactions on Robotics and Automation **5**(3), Jun. (1989)
10. Martha Steenstrup, Michael A. Arbib, Ernest G. Manes, *Port Automata and the Algebra of Concurrent Processes*. JCSS 27(1): 29-50, (1983)
11. Damian M. Lyons, Mohamed Chaudhry, D. Paul Benjamin, Marius Agica, John Vincent Monaco, "Integrating Perception and Problem Solving to Predict Complex Object Behaviors", Conference on Multisensor, Multisource Information Fusion, SPIE, (April 2010)
12. Damian M. Lyons and D. Paul Benjamin, "Robot Video Tracking by Comparing Real and Simulated Video Scenes", Conference on Intelligent Robots and Computer Vision, SPIE, San Jose, Calif., January (2009)