Development of Human-Robot Interaction Systems for Humanoid Robots

Bruce A. Maxwell, Brian Leighton, Andrew Ramsay
Colby College
{*bmaxwell,bmleight,acramsay*}*@colby.edu*

Abstract - Effective human-robot interaction is one of the primary challenges for humanoid robots. Sources of uncertainty, such as robot motion, perception, and people challenge interaction systems that are not adaptive or robust to failure. Developing and testing robust and effective interaction algorithms requires a test system that incorporates real robots, perception and human components. Simulation of any of these three components does not allow realistic evaluation. We present a low-cost system that will enable testing and development of interaction systems for humanoid robots. The system uses a Robonova humanoid robot, an external camera, a vision system designed for social interaction, and a host computer. As a demonstration of a working system, we also present an initial framework for object-centered human-robot interaction that has the potential to be scalable, robust, and efficient in enabling the robot to achieve goals in a social situation.

Keywords - human-robot interaction, vision, planning, evaluation

1. INTRODUCTION

Many future uses of robotics involve robots working directly with people in the same environment. Examples include household robots, tourguides, and workplace assistants. A challenging aspect of these robot systems is making the human-robot interface natural, efficient, robust, and scalable. The interaction system should be natural, in the sense that the robot's actions do not surprise the person with whom it is interacting. It should be efficient so that the robot's assistance is a net gain in time spent on task. It should be robust to failures in communication or action by either party, and it should scale to many domains, objects, and people.

When developing an interaction system, it is necessary to undertake extensive testing and evaluation in real-world situations. The world introduces complexity into an interaction system in three different ways: the robot's actions, the robot's perceptions, and human actions. When robots move, their actions are not deterministic and can modify the world state in unexpected ways. Perception systems are not perfect, and perceptions of the world state can be incomplete, incorrect, or inconsistent. People introduce additional complexity by often acting in unexpected ways. The only way to incorporate all of these sources of uncertainty is to build a real-world system.

Our ultimate goal is to build a human-robot interaction system for a full-scale humanoid robot, HUBO [1]. HUBO is currently in development, and even when fully functional will be expensive to operate and unavailable for extensive testing. Therefore, we need a way to evaluate and test interaction systems using a low-cost system with high up-time that provides sufficient similarity to HUBO for meaningful testing.

To satisfy the need for a test platform, we have designed and built a low-cost end-to-end system that includes all of the sources of uncertainty given above: robots, perception, and people. In this paper, we present the test system. As a proof of concept, we also present an initial framework for interaction that focuses on efficiency, robustness, and scalability, with the hope that the resulting interaction is close to natural.

1.1. Related Work

Robot systems, to date, have relied largely on state machines or pre-programmed scripts to guide the interaction [2], [3], [4]. Well-designed state machines provide robustness to failure and efficiency, but are difficult to scale. Each type of interaction requires a different state machine, or script, and the interaction is only as complex as the state machine itself.

More recently, systems are beginning to appear that attempt to model the person with whom the robot is interacting and use the model to predict the results of requests for interaction [5], [6]. A planning system takes into account the current world state and a goal and tries to identify a reasonable path from the current state to the goal given expectations and costs. Moving beyond state machines and incorporating a planning system is necessary in order to provide scalability and efficiency.

A number of researchers have developed systems for testing and evaluating human-robot interaction. Breazeal developed the Kismet platform to explore facial expression as a facet of human-robot communication [7]. Kismet has appendages, but was not designed to be mobile.

Gockley et. al developed Valerie as a long-term social robot experiment with an animated face [8]. Valerie was able to evaluate different methods of interaction using speech and computer graphics, but had no appendages and generally did not move.

Nourbakhsh put together a long-term study of interactive robot museum guides [9]. The guides could move around and interact with people. As with Valerie, the robots were large wheeled platforms without appendages or humanoid appearance.

We expect to learn from these longer term interaction scenarios, but humanoid robots present new possibilities in terms of interaction, including the ability to use gestures not available to prior generations of robots. The testbed system we have built permits us to explore the possibilities of humanoid systems in a controlled, but realistic environment with levels of uncertainty that match real-world situations.

1.2. Interaction System Overview

Given a sufficiently robust planning system, the challenge is to design the rules that define the space of possible actions and their resulting world states. To address this challenge, we are implementing an object-centered world representation that attaches actions and outcomes to objects in the robot's world state. The sensing system identifies objects currently within the robot's domain and passes the information on to the world state. The planner then assesses the world state and identifies a potential path to a goal state. As the world changes, the planner can re-asses the current strategy or note that a step along the current plan is complete.

An object-based system should be easily scalable. As we give the robot the ability to identify new objects, the object definition contains within it all the actions the robot can take with the object. For example, a block can be picked up, placed, or moved by a robot. If the world state includes a person, the person object defines the possible actions the robot can take with respect to the person. The person's actions can be conditioned on other objects in the world state, such as a block; if a block is in the world, the robot can ask the person to move it. As objects move in or out of the world state, the set of possible actions for the robot change according to the world state, and each object keeps track of its current possible actions. For planning purposes, each action can also have a cost so that the robot can make efficient plans.

Object-based systems have been used in remote robot operation interfaces, with pop-up menus attached to objects telling the user what options are available for each object in view [10]. We are proposing that such an approach is equally valid for human-robot interaction. If a person is in the world state, they are an object that enables certain actions by the robot– such as a greeting. The detection of a block enables more actions with respect to the person, and as more objects enter the scene, more actions become enabled.

In order to build and evaluate a planning system for human-robot interaction we have put together a complete end-to-end system. While testing, we want to avoid simulation of any single component in order to develop, evaluate, and test systems that are robust enough to use in the real world. The system incorporates a small humanoid robot and an external camera that views the robot workspace.

This paper presents the system design and our initial tests of an object-centered interaction planning system. Herein we describe our overall setup, the vision system, the initial interaction system design, and the robot control system. Then we provide examples of simple interaction scenarios and show how the current system responds.

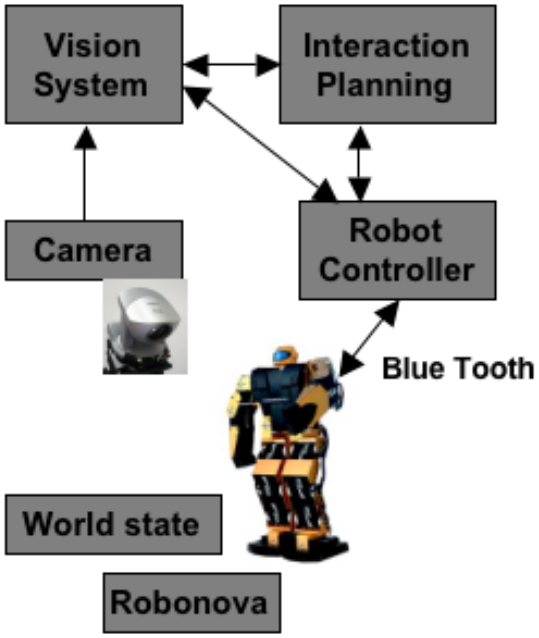


Fig. 1. Diagram of the robot system and communication paths between modules.

2. Experimental Setup

The experimental setup uses a Robonova platform, a 25cm tall humanoid robot with 16 degrees of freedom. The robot has an onboard microcontroller with a Basic interpreter that can execute simple programs. We have added a BlueSmirf Blue Tooth serial adapter that allows for data and commands to be sent to the robot from a host computer. The Robonova provides sufficient complexity that we can model many of the actions we would expect a full-size humanoid robot to execute. With the Blue Tooth adapter we avoid the need for a tether while still enabling significant processing power for the perception and interaction systems.

Visual feedback for the robot is provided by a VC-C4 Canon PTZ camera placed 1m above the robot's work area. The work area is approximately 0.5m x 0.5m. The camera is attached to a host computer running a vision system that can detect the robot and objects in its work area. The host computer is also executing our interaction and reasoning system and building plans based on the world state detected by the vision system. The system comprises a complete feedback loop so the robot's actions are reflected in changes in the perceived world state. A diagram of the system is given in figure 1.

The robonova's workspace is a 50cm x 100cm workspace with 12cm walls. A two-tiered rack above the workspace permits mounts for a downward facing camera to view the robot and a second camera at head-height to view someone interacting with the robot. The entire setup sits on a table and provides a self-contained demonstration area where a person can easily interact with the robot and objects within its workspace.

3. System Design

The overall system design, shown in figure 1 is based on the design of prior systems for social robots [2][11][12]. Three independent modules–the vision system, the interaction planner, and the robot controller–communicate via the Inter-Process Communication [IPC] package [13]. IPC is a message-passing system implemented over TCP/IP, permitting the modules to run on different computers, as necessary.

3.1. Vision System

The social vision module [SVM] is an extension of the robot vision system described in [14]. The vision system is designed to permit many different vision operators to function effectively in realistic time under cpu cycle constraints.

Each operator in the vision system executes a different function. For example, operators exist for identifying color blobs, faces, motion, and text in an image. If the system ran all of the operators on every frame, the frame rate would degrade and cpu performance would suffer. If the CPU is actually on a robot, this situation becomes dangerous, as the robot control system no longer has sufficient computing resources.

In a social situation, however, the robot rarely needs to know that a face is in the image at 30Hz. To balance out the needs of the operators and the CPU, two operators are selected stochastically for each frame, with probability proportional to their weight. This guarantees that the CPU is not overloaded and that the vision system can maintain 30fps. The user or control system can set the weight for each operator independently, so that critical operators run more often. The user can also specify how often an operator needs to update its information, so if the control system does not need to know certain information more often than 1Hz, it only receives updates on that schedule.

For the current system, the face detection and colored blob detection algorithms provide the world state information. The face detector is based on the OpenCV implementation of the Viola-Jones algorithm [15]. The colored blob detection uses histograms of the colors to segment the image and identify sufficiently large connected regions. Both the face and colored blob detectors use a Kalman filter to reduce jitter in the reported image locations. Figure 2 shows positive detections of three blocks in the robot workspace.

To improve the estimation of the positions of objects and the robot within the work space, the system is designed to work with a camera calibration. Using a standard calibration procedure, a user can build an internal and external calibration for the camera relative to the robot's ground plane. The SVM then reads in the calibration and provides estimated ground plane locations of object and robot detections by converting pixels into 3-D rays that intersect the ground plane. The calibration permits SVM to provide accurate ground plane locations regardless of the camera orientation.

Overall, the vision system provides robust location estimates for the color blobs, faces, and the robot within the robot's workspace. The system sends messages via IPC to the interaction module when its active operators find objects in the world.

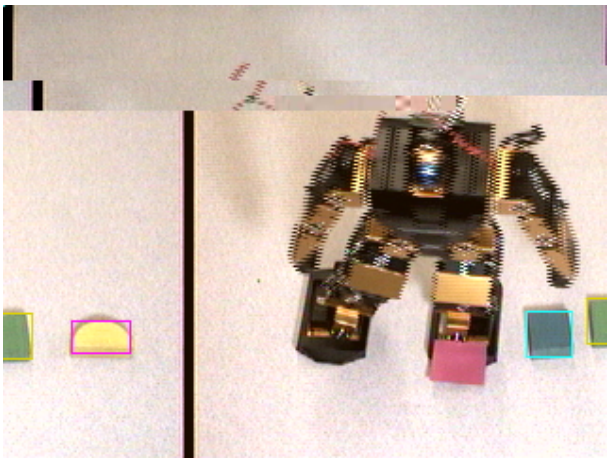


Fig. 2. Robot workspace showing three blocks identified by the vision system.

3.2. Interaction System

The goals for the interaction system are to make it robust, easily scalable, and to make the interactions rule-based rather than state-based or script-based. For our first interaction system design we are moving to a rule-based planning system, which also provides a means for making the interaction efficient in terms of costs, however those costs are determined.

The fundamental principle of our interaction concept is that, from the robot's point of view, objects or actors in the world have certain actions, or mannerisms attached to them that are available to the robot. Objects and actors in the scene determine what mannerisms are available to the robot by scanning the world state and activating or deactivating actions as appropriate.

For example, consider a world state consisting of a block and a robot. The block evaluates the world and provides a set of potential mannerisms to the robot for planning purposes. If the block is far away from the robot, the only mannerism that is active is for the robot to approach the block. If the robot is close to the block, the robot can move away from the block or move the block by kicking it. The active mannerisms are determined by a rule set within the block object. All simple objects can be defined by a state and a set of rules for activating or deactivating mannerisms. The preconditions for each mannerism define when it is active, and each mannerism provides a post-condition estimate for the purposes of planning.

The world state can also include composite objects that consist of two or more simple objects in a specific relationship. For example, a Cluster object consists of one or more blocks that are close together in the workspace. The interaction system continuously evaluates its view of the world, and if two blocks get close enough, it forms a Cluster object with additional mannerisms that become available to the robot, such as breaking apart the cluster by moving one of the blocks.

We plan to implement interaction with a person using exactly the same structure. A person is defined as an object in the scene and the current world state defines which mannerisms

are available to the robot for human-robot interaction. If a block exists in the world, for example, the robot can ask the person to move the block.

The interaction system currently uses an A* search to plan a series of actions to achieve a goal. During the search, each action evaluated by the search process provides an expected new world state which the search system uses to identify what the set of next possible actions will be. The system currently re-plans whenever the world state changes. While this provides robustness to unexpected results or new objects in the scene, future work will involve studying different planning systems that integrate uncertainty more directly.

3.3. Robot Control System

The Robonova has an on-board MR-C3024 control board with a Basic interpreter. The limitations on code side and complexity mean that decision-making must be implemented on a different processor. Collaborators on our project have successfully added an on-board ARM processor with a camera and Blue-tooth connection, which provides one possible future development path [16].

We have taken the alternative approach of creating a basic vocabulary of actions that can be activated by sending characters across a serial connection implemented using a Blue Tooth adapter. The vocabulary of actions is small enough to be implemented using the Basic microcontroller. The on-board program polls the serial port waiting for a command and executes a gesture or action based on single characters sent by the host computer.

The current vocabulary of actions implemented by the onboard program include the following.

- Step forward
- Step backward
- Step left
- Step right
- Rotate left
- Rotate right
- Bow
- Wave left arm
- Wave right arm

The above vocabulary is sufficient to test out a variety of common human gestures as well as move the robot around the workspace.

The robot control system currently accepts commands directly from a user who enters a command by typing the appropriate letter on a keyboard. When system integration is complete, the robot control system will accept high level commands from the interaction system. We also hope to implement simple feedback control by using the vision system to track the robot when it is asked to move to a particular location in the workspace.

4. Experiments and Results

As an initial experiment of feasibility, we implemented the following example using the overall system.

4.1. Example: Robot and Blocks

Initial World State: The robot and two blocks, each separated from the other by some minimum distance, are detected within the workspace, as shown in 3a.

Goal: Create a Cluster of blocks by moving one block close to the other.

Process

- Vision system detects the robot and two blocks with their associated positions (3b)
- The only available actions in this state are for the robot to move itself.
- Planning system identifies a path to the goal.
  - Robot must move itself into position to move a block (blue block).
  - Moving itself involves a multi-action process to get to the right of the blue block.
  - Moving to the right and close to the blue block enables the kick mannerism for that block.
  - Robot can move the block by kicking it towards the other block.
  - When the blocks are close enough, a Cluster object is created
- Robot implements first step in the plan (3c)
- Robot implements second step in the plan (3d)
- Robot implements third step in the plan (3e)
- Proximity of robot and a block enables a kick action
  - Robot can move the block by kicking it towards the other block.
  - When the blocks are close enough, a Cluster object is created
- Robot implements a move block action (3f)
- End of the current plan causes the robot to re-plan.
- Robot implements a move block action (3g)
- End of the current plan causes the robot to re-plan.
- Robot implements a move block action (3h)
- Final goal state is achieved as the Cluster object is created when the two blocks are close enough together..

Other than selection of the goal, the above example was completely automated within the interaction testbed, demonstrating that all aspects of the system are functional. Including a person in the interaction would enable new types of mannerisms, such as enabling the robot to ask the person to move one of the blocks. The robot would still need to move out of the way, in that case.

4.2. Discussion

We have described an interaction testbed that should allow us to test systems prior to installation on a full-size (3.4m) robot. Some of the issues that may arise because of the hardware setup include the following.

- The scale of the testbed is approximately 1/4 of the full size system. We can simulate the full-size robot's camera using the camera at head-height above the workspace, but there may be other differences due to scale. In particular,

we have no way to test realistic physical contact with people, if that becomes part of the interaction.
- The testbed does not have robot-centered vision, although we may able to provide that in the near future by putting a blue tooth capable camera on the robot itself.
- The environment in the testbed is only as complex as we make it. Care must be taken to avoid building systems that are tuned to clean environments.

Many issues arise in the design of an interaction system, some of which we will be able to test and evaluate using the setup.

- When planning actions, how do we generate the relative costs of different actions? What should form the basis for action costs? Costs, for example, may change over the course of an interaction. If a person intentionally ignores requests for action, or intentionally takes the wrong actions, the robot may want to change the costs of interaction with respect to future planning. Different responses can be tested with our setup.
- Interaction involves uncertainty with respect to people and how they react. In particular, when dealing with people, they will not always take the desired or optimal action, perhaps leading to greater costs than having the robot undertake an action itself. These probabilities or costs could be learned in a testbed where people are interacting with the robot over an extended time period.
- The interaction system must keep track of how the world changes during an interaction and constantly re-estimate the world state, mapping the changes to expectations (i.e. did the person move the red block, as asked, or the green block?). Methods for dealing with world state evaluation can be tested with our setup.

5. SUMMARY

Overall, we have presented a testbed for evaluating humanoid robot interaction systems along with a prototype interaction methodology as proof of concept. The testbed is low-cost and provides a sufficiently controlled environment that the effects of changes in the interaction system should be measurable.

Our prototype interaction system is object-based, with the estimated world state, and the objects therein, controlling what actions are available to the robot. We have an end-to-end demonstration of the system in a simple blocks world situation, with plans to expand it to incorporate people and other objects. The real-world nature of the system–sensing, motion, people–makes it a challenging task. The testbed should enable us to develop a robust, scalable interaction system for HUBO.

6. ACKNOWLEDGEMENTS

This work was supported in part by Colby Collage, and by the National Science Foundation Partnership for International Research and Education (PIRE) award OISE 0730206. Special thanks to Rob Ellenberg at Drexel for assistance with the Robonova software and hardware.

REFERENCES

[1] J.-H. Oh, D. Hanson, W.-S. kim, I.-Y. Han, J.-Y. Kim, and I.-W. Park, "Design of android type humanoid robot albert hubo," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1428 – 1433, October 2006.

[2] B. A. Maxwell, "Building robot systems to interact with people in real environments," *Autonomous Robots*, vol. 22, no. 4, pp. 353–367, 2007.

[3] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita, "Interactive humanoid robots for a science museum," *Intelligent Systems*, vol. 22, pp. 25–32, March/April 2007.

[4] T. Fong and I. Nourbakhsh, "Socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, pp. 143–166, March 2003.

[5] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Task planning for human-robot interaction," in *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, (New York, NY, USA), pp. 81–85, ACM, 2005.

[6] J. Allen and G. Ferguson, "Human-machine collaborative planning," in *Int'l NASA Workshop on Planning and Scheduling for Space*, October 2002.

[7] C. Breazeal and B. Scassellati, "How to build robots that make friends and influence people," in *Proc. of the Int'l Conf. on Intelligent Robots and Systems (IROS99)*, 1999.

[8] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A. Schultz, and J. Wang, "Designing robots for long-term social interaction," in *Proc. of the Int'l Conf. on Intelligent Robots and Systems (IROS05)*, August 2005.

[9] I. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: A five year experiment," in *Proc. of the Int'l Conf. on Intelligent Robots and Systems (IROS03)*, 2003.

[10] H. Jones and M. Snyder, "Supervisory control of multiple robots based on a real-time strategy game interaction paradigm," in *IEEE Int'l Conf. on Systems Man and Cybernetics*, 2001.

[11] K. R. Thorisson, H. Benko, A. Arnold, D. Abramov, S. Maskey, and A. Vaseekaran, "Constructionist design methodology for interactive intelligences," *AI Magazine*, vol. 25, no. 4, pp. 77–90, 2004.

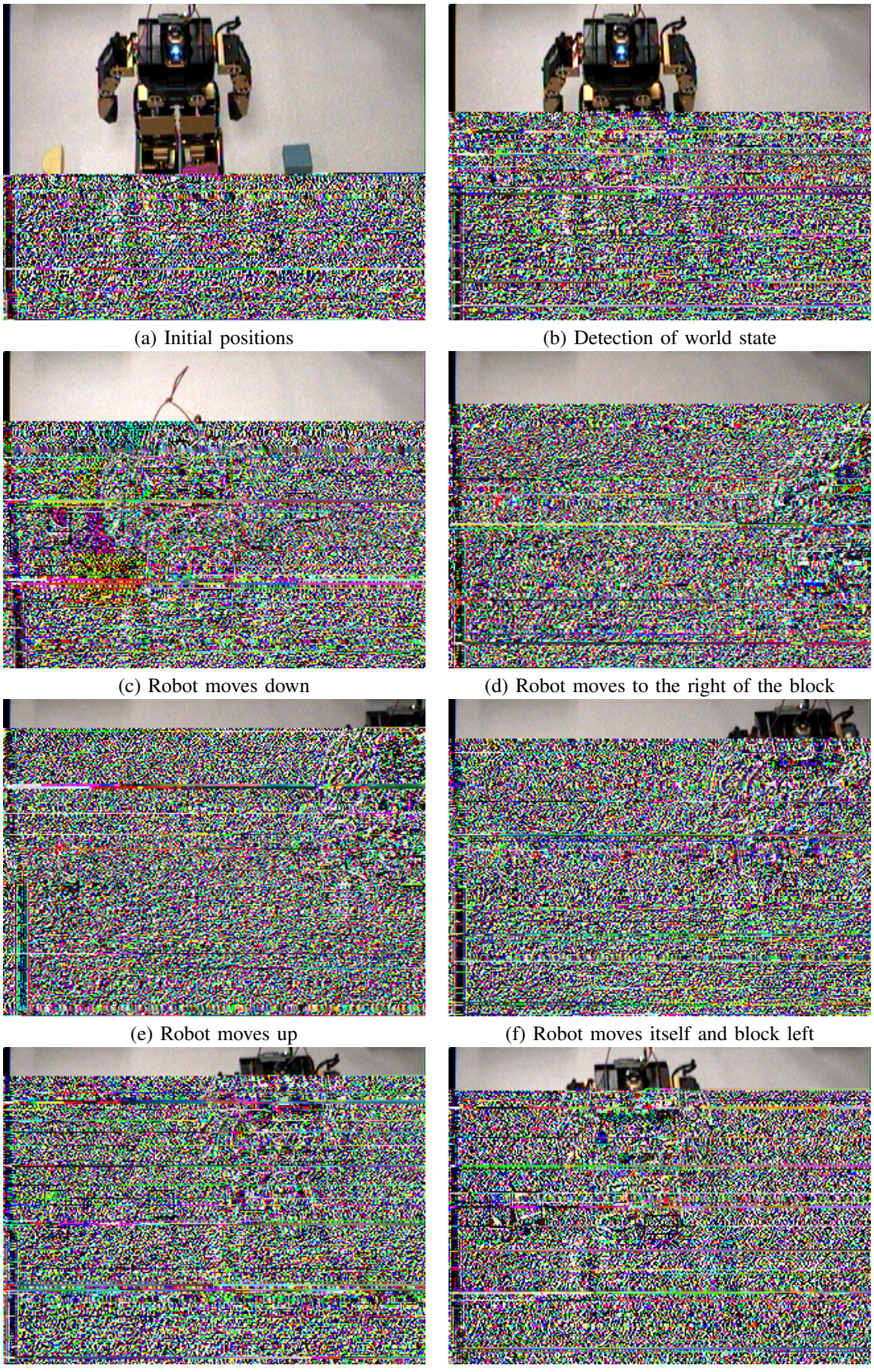
[12] B. A. Maxwell, L. A. Meeden, N. S. Addo, P. Dickson, N. Fairfield, N. Johnson, E. G. Jones, S. Kim, P. Malla, M. Murphy, B. Rutter, and E. Silk, "Reaper: A reflexive architecture for perceptive agents," *AI Magazine*, vol. 22, no. 1, pp. 53–66, 2001.

[13] R. Simmons and D. James, *Inter-Process Communication: A Reference Manual*. Carnegie Mellon University, March 2001.

[14] B. A. Maxwell, N. Fairfield, N. Johnson, P. Malla, P. Dickson, S. Kim, S. Wojtkowski, and T. Stepleton, "A real-time vision module for interactive perceptual agents," *Machine Vision and Applications*, vol. 14, pp. 72–82, 2003.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of Computer Vision and Pattern Recognition*, vol. 1, pp. 511–519, June 2001.

[16] D. Blank, T. Thomas, T. Stewart, and K. O'Hara, "Humanoids in the classroom," in *Int'l Conf. on Ubiquitous Robots and Ambient Intelligence*, 2008.



(a) Initial positions

(b) Detection of world state

(c) Robot moves down

(d) Robot moves to the right of the block

(e) Robot moves up

(f) Robot moves itself and block left

(g) Robot moves itself and block left

(h) Robot moves blocks together

Fig. 3. Robot moving through a series of actions to move the blocks close together. The boxes show automatic detection of the robot and blocks using the vision system.