Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

The Use of Perceptual Cues in Multi-Robot Box-Pushing

C. Ronald Kube and Hong Zhang
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
kube@cs.ualberta.ca zhang@cs.ualberta.ca

Abstract

In this paper we present an approach to controlling transitions in multi-robot tasks which have been modelled as a linear series of steps. A box-pushing task is described as a sequence of sub-tasks with a separate controller designed for each step using finite state automata theory. Perceptual cues are formed by concatenating binary variables which represent locally sensed stimuli into boolean vectors used to specify transitions between sub-task steps. The approach is designed for a redundant set of homogeneous mobile robots equipped with simple sensors and stimulus-response behaviours. A set of perceptual cues used in box-pushing are designed and tested on 10 physical mobile robots. It is argued that perceptual cues and finite state automata offers a new approach to environment-specific task modelling in collective robotics.

1 Introduction

Recent interest in accomplishing tasks with multiple mobile robots has led to a number of canonical tasks on which to test theories in multi-robot control[1]. Examples of such tasks include formation marching, foraging and box-pushing, each of which requires the task to be specified in the group architecture under study, but few studies have considered the role local sensing plays in controlling task execution.

For tasks requiring a synchronization of efforts from a group of asynchronous robots, as found in tasks composed of a series of ordered steps, explicit communication between robots is often the mechanism used to ensure coordinated control [8, 6]. An alternative approach, which scales better as the number of robots increase, uses implicit communication by passive sensing through the task's environment, a common mechanism thought to be at the core of synergistic systems of social insects [11, 3]. In this paper, we present an approach to controlling transitions, based on locally sensed stimuli, between steps in a multi-robot box-pushing task.

Our model assumes a homogeneous set of physical robots using decentralized control without the use of explicit communication. Rather, all information about a robot's environment is obtained by local perception through a set of simple sensors arranged spatially to map sensing to actuation in a reactive manner. The approach attempts to solve perceptual decisions at the sensory level by considering the environment and its stimulus as part of the task specific robot solution to a given problem.

2 Perceptual Cue Framework

Our approach in designing task-specific multi-robot systems is to decompose the given task into ordered steps or sub-tasks, and then build a controller that accomplishes each step. The transition between steps is specified as a locally sensed *perceptual cue* which is necessarily highly dependent on both the selected sensors and the environment for which the cue was designed. This method of control is also becoming popular in other areas within robotics such as dextrous manipulation where it is referred to as *phase-based* control [9].

Paramount in this approach is the reliable detection of events, which mark transitions in task execution, through some form of feature extraction from the robot's sensor suite. Working in a known environment in which sensory stimulus can be partially defined helps alleviate the designer's burden of sensor selection, integration and fusion. We simplify the feature extraction problem by constructing perceptual cues from the data of sensors which are *spatially* and *modally* orthogonal. Sensors are defined to be orthogonal if their spatial workspace is non-overlapping, as

[1] See [2] for a recent survey of cooperative mobile robotics.

0-7803-2988-4/96 $4.00 © 1996 IEEE

in diametrically placed range sensors, or if their sensor output is incomparable such as touch and light intensity.

Sensor output is processed to yield a boolean value and then used to construct the perceptual cue. The perceptual cue is thus simply a boolean vector of variable length specified by concatenating the binary outputs of select sensory systems on the robot. We have tested this approach by creating perceptual cues for a box-pushing task described in Section 3.

2.1 Task Modelling

Our approach to task modelling assumes the task can be decomposed into a linear series of steps with a finite state machine (FSM) designed for each step[5]. Transitions between FSMs are specified as perceptual cues, and indicated as directed arcs in Figure 1, with each FSM having a forward, backward, and repeat link. The controller for each step, or subtask, is a FSM with each state specified as a primitive actuation behaviour, a reactive behaviour that maps sensor output to actuator variables. In each step an action or sequence of actions is performed, defined in terms of the robot's actuators, and is repeated until an event that is locally perceived by the robot's sensors is detected marking the step's completion.

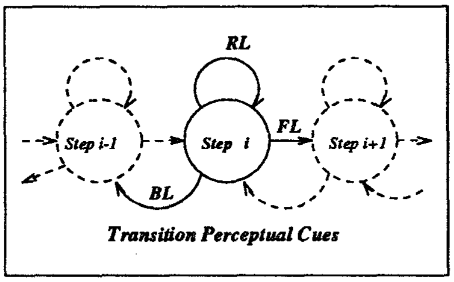

Figure 1: Each step in the task is modelled as a finite state machine with perceptual cues used to govern transition between steps. The perceptual cue causing a forward transition (FL) is simply a concatenation of another boolean variable to the previous step's forward perceptual cue.

2.2 Perceptual Cues

Perceptual Cues (PC) come in two flavours: *stimulus cues* used to extract features by using threshold logic and Boolean operators, and *transition cues* which manage control between subtask controllers and are formed by concatenating stimulus cues into boolean vectors.

The specification of stimulus cues are simplified if created for an environment with known and limited stimulus output. By thresholding an analog sensor's output a binary value is obtained which may be combined with other binary sensor outputs, forming a crude feature extraction. For example, contact with the side of our brightly lit box[2] is detected by a stimulus cue formed by taking the output of a photocell sensor set with a high threshold and combining it with the binary output of a forward pointing microswitch using a logical AND. In our environment this combination of sensor outputs can only occur on the side of the box to be pushed.

Transition cues are used to govern task execution. Control system processing is handled in discrete steps, with control either passing on to the next, previous or current subtask controller as specified in the task model digraph using forward (FL), backward (BL), or repeat (RL) links (see Figure 1). The transition cue for each subtask controller, or step i, are related to its predecessor by:

$$FL_i = FL_{i-1} \wedge SC_i \tag{1}$$

for $i = 1, 2, \ldots, n$ where n = subtasks and SC_i is a new stimulus cue.

$$RL_i = FL_{i-1} \wedge \overline{SC_i} \tag{2}$$

$$BL_i = \overline{FL_{i-1}} \tag{3}$$

The approach is dependent on reliable event detection, a problem which can be mitigated by considering and controlling the task environment's stimulus output coupled with an orthogonal sensing strategy.

2.3 Orthogonal Sensing

In order to simplify sensor processing, stimulus cues are binary and integrated by employing either spatially or modally orthogonal sensing strategies. This removes the sensor processing requirement from the behaviour allowing the cue to simply "trip" the stimulus-response behaviour, a mechanism found commonly in insects [7].

Sensors of the same type can be made *spatially orthogonal* either by geometric arrangement, with non-overlapping fields of view, or by partitioning the sensor's range discretely using thresholds.

[2] The robots push a 16" square box which contains a 100 Watt bulb.

A stimulus cue that is *modally orthogonal* is specified by taking the binary outputs of sensors with incompatible outputs like temperature and contact sensing, or range and odometry data. Specifying which sensor modes to combine is a nontrivial problem.

2.4 Additive Cue Construction

Transition cues that guide task execution are specified by concatenating select stimulus cues into boolean vectors. By selecting only those stimuli that are relevant to the current stage in a task's execution, the set of possible cues formed by considering all onboard sensors is reduced, thus simplifying the control decision. For example, in the box-pushing task, described in the sequel, once a robot has found a position on a box-side from which to push, obstacle sensing is not relevant and the pushing controller's transition cues do not make use of these sensors. The advantage of creating boolean transition cues, whose computation is disjoint from the operation of the subtask controller, lies in the ease of sensor replacement as feature extraction techniques improve.

3 Box-Pushing Task Model

In collective box-pushing the objective is to locate a brightly lit box and push it towards a final destination. The box-pushing task is modelled by first dividing the task into subtasks (steps), then specifying the perceptual cues for each step transition, and finally designing the subtask controllers to be implemented from a set of primitive actuation (PA) behaviours.

3.1 Sub-Task Controllers as FSA

Each of the 10 robots runs an identical algorithm which implements the box-pushing task as a linear series of four subtasks:

1. **FIND-BOX** - a three state machine that searches for the lit box by executing a random walk while avoiding obstacles.
2. **MOVE-TO-BOX** - a three state machine that moves the robot towards a brightly lit box while avoiding obstacles.
3. **PUSH** - a single state machine that increases motor speed until motion is achieved.
4. **PUSH-TO-GOAL** - a three state machine that pushes the box towards a goal destination if in the forward direction.

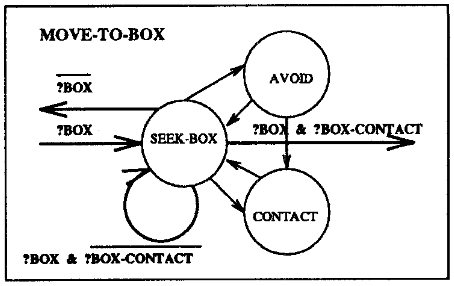

Figure 2: An example of a subtask controller as a finite state automata. The MOVE-TO-BOX automata moves the robot towards the lit box while avoiding obstacles. FIND-BOX is similar with the *seek-box* goal-seeking behaviour replaced with *random-walk*. The heavy arcs indicate the transition cues that pass control between FSMs.

3.2 Perceptual Cues for Box-Pushing

To construct the transition cues for box-pushing the four stimulus cues listed below are concatenated to form a boolean vector for each transition illustrated in Figure 3.

?box True when either the left or right box photocell thresholds are triggered.

?box-contact True when either the left or right box-side photocell thresholds are triggered AND the push sensor is on.

?motion True when the idle wheel sensor is on.

?goal True when at least one of the 4 upward pointing goal-direction sensor threshold is triggered.

The stimulus cues for box-pushing are formed by thresholding analog sensor outputs and applying a boolean operator to the result.

3.3 Primitive Actuation Behaviours

Primitive actuation behaviours map their sensor inputs, preprocessed by an associated threshold, to one of 8 motion commands from the set { *idle, right-turn, left-turn, right-rotate, left-rotate, backward, stop, forward* }. The set of PA behaviours for box-pushing include two for obstacle avoidance, *AVOID* and *CONTACT*, a default motion, *RANDOM-WALK*, a box-tracking behaviour called *SEEK-BOX*, and a directional behaviour, *SEEK-GOAL*, which indicates the direction of the box destination. Behaviour arbitration between PA behaviours is handled using a fixed priority

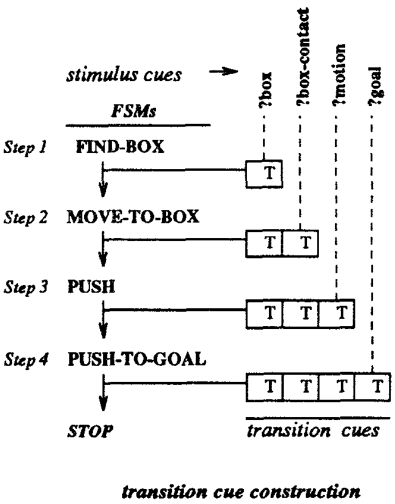

Figure 3: Transition cues are created by concatenating stimulus cues to form a boolean vector. A transition to the next step occurs when each element in the vector is true. Illustrated are the forward transitions. Repeat and backward transitions are formed in a similar manner.

scheme, and implemented by having each motion command write to actuator variables with the last command taking effect.

4 Multi-Robot Experiments

The objective in the box-pushing task is to find a brightly lit box, randomly placed in the robot's environment, and to push it towards a designated location. The box is weighted such that the task requires at least three robots to complete it. In the sequel we present the results of testing the perceptual cues on the subtask controllers.

4.1 Robot Hardware

The box-pushing perceptual cues have been tested using a set of 10 identically designed micro-robots shown in Figure 4. Sensor topology can be changed and is geometrically configured to suit the intended task. Thus both the number and placement is task dependent.

4.2 Sensor to Actuator Mapping

All motion of a box-pushing robot is accomplished through left and right wheel motors. As a result of this division, behaviours that produce motion have their perceptual systems divided into left and right sides. The result is a stimulus-response sensor-to-actuator mapping reminiscent of Braitenberg's vehicles [1]. Motion is accomplished using discrete commands with a 100ms duration. The increment for linear and angular movement is 1/8 inch and 2° respectively. The sensor-to-actuator mappings for the motion producing behaviours are listed in Table 1.

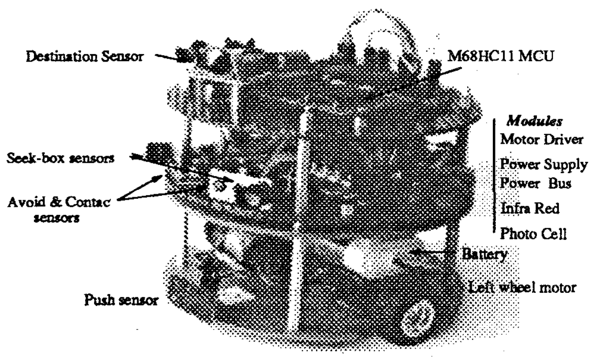

Figure 4: Each of the 10 robots are equipped with 2 forward pointing infrared obstacle sensors, one touch sensor, 2 CdS box-tracking photocells, and a destination sensor, all mounted on a differentially steered base.

Sensor to Actuator Mapping					
Stimulus		Primitive Actuation Behaviours			
L	R	AVOID	CONTACT	SEEK-BOX	SEEK-GOAL
0	0	*idle*	*idle*	*idle*	*idle*
0	1	*left-turn*	*left-rotate*	*right-turn*	*right-turn*
1	0	*right-turn*	*right-rotate*	*left-turn*	*left-turn*
1	1	*right-turn*	*right-rotate*	*forward*	*forward*

Table 1: The stimulus-response mappings for motion producing PA behaviours. Stimulus cues with boolean outputs are divided into Left and Right stimuli to match the left and right division of the actuator model.

4.2.1 Obstacle Avoidance

The box-pushing task makes use of two PA behaviours for obstacle avoidance, *AVOID* and *CONTACT*. The output of a left and right obstacle sensor is mapped to left and right motion commands. The possible outputs of *AVOID* are { *idle, left-turn, right-turn* } while *CONTACT* maps sensor output to { *idle, left-rotate, right-rotate* }. Ambient infrared light is accounted for by taking a detector reading while the emitter is off. Currently both behaviours make use of the same infrared obstacle sensors, but with different threshold functions on the outputs. Although in this example both behaviours are using the same sensor output this separation allows for sensor replacement [3].

4.2.2 Box Tracking

The box to be pushed (versus boxes that are obstacles) is equipped with a bright light which shines in the horizontal plane at the same level as each robot's left and right box sensors[4] The stimulus cue used to trigger the *SEEK-BOX* behaviour is set to an intensity threshold that is four times greater than the room's ambient light. The *SEEK-BOX* has four possible outputs { *idle, forward, left-turn, right-turn* }. To test the behaviour the lit box is moved throughout the environment with the robot in pursuit. Since box sensing is not currently omni-directional *SEEK-BOX*, with its limited field-of-view, can lose sight of its target and requires the random-walk behaviour to ensure spatial sensing coverage.

Phototaxis is also used to sense the final box destination. A bright spot light placed vertically at eight feet indicates the box destination direction and is tracked in a similar manner by taking pairs of the four upwardly pointing photocells, arranged in a square, giving a somewhat omni-directional field-of-view. An alternate sensor design we are currently considering, rotates a single light intensity sensor instead of the current fixed sensor configuration.

4.3 Step Controllers

Each of the *stimulus* and *transition* perceptual cues listed in Section 3 was first tested individually by allowing the robots to be controlled by a single FSM for steps 1 to 3. The **FIND-BOX** controller generated a motion pattern sufficient for a single robot to cover a 16 by 16 foot room in under 3 minutes. Next the **MOVE-TO-BOX** FSM was added and tested for its ability to find and track the box. The box light was then turned off to test the correct transitional return to the previous step, at which point **FIND-BOX** resumes execution. In the same environment this two step controller located the lit box, randomly placed, while avoiding obstacles. With the **PUSH** controller added pushing force increased stepwise with each robot/box contact condition detected. With the box light off control resumes with **FIND-BOX** [5].

In order to test the transition perceptual cue between **FIND-BOX** and **MOVE-TO-BOX** two experiments were run in two separate environments. In the first, 10 robots were placed in one corner of a 16 foot by 10 foot room with the box located at the opposite end. The initial and final configurations were then compared against the simulated version of the same experiment. In both versions the initial configuration of robots expanded spatially avoiding obstacles while executing the random-walk PA behaviour. Once their forward pointing light sensors detected the box they converged and attempted to occupy a spot on a boxside. Additional robots converged until obstacle sensors detected a robot already present on a boxside, forcing the additional robot to find a free spot. This resulted in a distribution of robots around the box and marked the end of the second task step (see Figure 5).

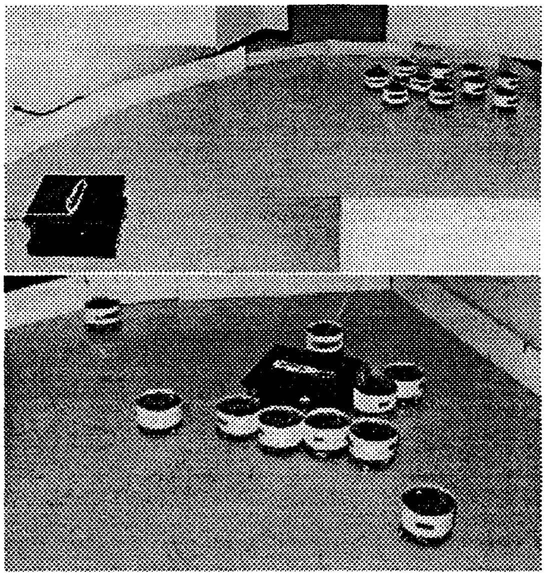

Figure 5: The results of physical tests on locating a lit box. The distribution results from avoidance behaviour until a free spot against a boxside is found.

The same results were obtained in a second environment in which six robots were placed in a 20 by 18 foot room with the box located 19 feet from the robots. In each of the three trials the robots located the box after an initial random search in under two minutes.

The next transition from **MOVE-TO-BOX** to **PUSH** has been successfully tested with three robots and the result is being replicated to the remaining group. The results of the final box-pushing test should be forth-

[3] *CONTACT*'s sensors could be replaced with left and right tactile switches

[4] Cadmium Sulfide photocells pointing forward.

[5] The last controller PUSH-TO-GOAL has not been tested.

coming.

5 Conclusions

The lack of a more general framework in collective robotics has resulted in a wide variety of methodologies used to investigate multi-robot control. Our approach to the control of a homogeneous set of mobile robots is based on a subtask decomposition in which FSM controllers are built for each step and transitions are controlled by *perceptual cues*, a concatenation of boolean outputs derived from locally sensed stimuli. The approach is motivated by entomological evidence that suggests local perception can be used to guide stimulus-response bahaviours in social insects which we conjecture can also be applied in collective robotics [4].

Our goal in examining the multi-robot box-pushing task was to create an abstraction of the robot's perceptual system that could be used to control a multi-step task, and then to find a minimal set of sensors that would demonstrate the feasibility of the approach. Our initial results are promising in that they show for an environment-specific system, the perceptual cue abstraction, which simplifies sensor integration by using spatially and modally orthogonal sensors, makes sensing computationally disjoint from the operation of the subtask controllers. This has the advantage of being able to replace a robot's sensor system with more powerful feature extraction techniques as they become available, since the underlying controller is not sensor dependent.

This raises the question of whether a complex decision making process on a robot with multiple sensors can be reduced to simple sensor preprocessing? Insects are often cited as prime examples that solve perceptual problems through the spatial arrangement of receptors to match some environmentally specific stimulus [10]. We conjecture that this mechanism may also serve the artificially created systems found in collective robotics.

A question we are currently exploring is the relationship between the number of robots and task performance improvements. The number of robots used in box-pushing is dependent on both the size of the box and robot, as well as the number of boxes in the environment. We have considered the problem in simulation [4] and are comparing the results with physical experiments.

References

[1] Valentino Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, 1984.

[2] Y. Uny Cao, Alex S. Fukunaga, Andrew B. Kahng, and Frank Meng. Cooperative mobile robotics: Antecedents and directions. In *1995 IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 226–234, 1995.

[3] H. A. Downing and R. L. Jeanne. Nest construction by the paper wasp, Polistes: a test of stigmergy theory. *Animal Behaviour*, 36:1729–1739, 1988.

[4] C. Ronald Kube and Hong Zhang. Collective robotics: From social insects to robots. *Adaptive Behavior*, 2(2):189–219, 1993.

[5] C. Ronald Kube and Hong Zhang. Task modelling in collective robotics. submited to Autonomous Robots special issue on Robot Colonies, June 1995.

[6] Maja J. Mataric, Martin Nilsson, and Kristian T. Simsarian. Cooperative multi-robot box-pushing. In *1995 IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 556–561, 1995.

[7] J.C. Moser. Pheromones of social insects. In D. Wood, R. Silverstein, and M. Nakajima, editors, *Control of Insect Behavior by Natural Products*, pages 161–178. Academic Press, 1970.

[8] Lynne E. Parker. ALLIANCE: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In *1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, 1994. to appear.

[9] Marc R. Tremblay and Mark R. Cutkosky. Using sensor fusion and contextual information to perform event dete ction during a phase-based manipulation task. In *1995 IEEE International Conference on Intelligent Robots and Systems*, pages 262–267, 1995.

[10] Rudiger Wehner. Matched filters - neural models of the external world. *Journal of Comparative Physiology A*, 161:511–531, 1987.

[11] E.O. Wilson. *The Insect Societies.* The Belkap Press of Harvard University Press, 1971.