Sensible Planning: Focusing Perceptual Attention

Lonnie Chrisman and Reid Simmons
School of Computer Science
Carnegie Mellon University
chrisman@cs.cmu.edu, simmons@cs.cmu.edu

Abstract

A primary problem facing real-world robots is the question of which sensing actions should be performed at any given time. It is important that an agent be economical with its allocation of sensing when sensing is expensive or when there are many possible sensing operations available. Sensing is rational when the expected utility from the information obtained outweighs the execution cost of the sensing operation itself. This paper outlines an approach to the efficient construction of plans containing explicit sensing operations with the objective of finding nearly optimal cost effective plans with respect to both action and sensing. The scheduling of sensing operations, in addition to the usual scheduling of physical actions, potentially results in an enornous increase in the computational complexity of planning. Our approach avoids this pitfall through strict adherence to a *static sensing policy*. The approach, based upon the Markov Decision Process paradigm, handles a significant amount of uncertainty in the outcomes of actions.

Selective Attention

One of the most important decisions in the design of a mobile robot is the choice of which aspects of the environment should be sensed at any given time. The most common approach is to always observe *all* relevant aspects of the environment. In the classical planning framework (eg. [Sacerdoti, 1975], [Fikes and Nilsson, 1971]) this takes the form of assuming that a complete state description is obtained at the onset of planning. In the reactive paradigm (eg. [Brooks, 1986], [Kaelbling, 1986]) it involves adding additional sensors and processors as necessary to continually monitor all relevant features of the world.

We are interested in building robots that perform a large number of diverse and complex tasks. As the robot's tasks become more complex and more numerous, the number of potentially relevant features of the environment quickly exceeds the sensing and processing resources that are feasible to supply to a robot [Simmons, 1990]. Thus, it is very important that the agent intelligently manage the control of its sensors and actions — in particular it is necessary to directly address the issue of what should be perceived at any given moment. The relevance of most features tends to be highly task dependent, and thus the choice of which sensing operations to perform varies considerably from task to task. For each task, a robot must explicitly decide which sensing operations to perform.

In some cases where resources disallow the simultaneous sensing of two or more aspects of the environment, selective sensing is absolutely necessary. This can occur with physical resources, for example when a given observation requires pointing the sonar in a certain direction, or as the result of computational resources, for example when the vision processor has available to it thousands of potential visual routines [Ullman, 1984] but only a limited number of CPU cycles.

Selectively sensing the environment introduces the additional complication that at any given time the agent has only a partial state description. This implies that the robot must deal with the possibility that some desired information is simply not known. In addition, when actions may have uncertain or non-deterministic outcomes, closed-loop execution can be important and the selection of a few items to observe can be critical to the efficient detection of a failing plan of action.

This paper presents some initial results in a long term research program aimed at understanding the interactions between sensing and action and at developing techniques for the effective usage and scheduling of focused perceptual attention. Our initial goals include the development of a useful normative theory describing both the utility of sensing versus action and efficient methods for finding optimal plans of action. The centerpiece of our approach is the adherence to a *static sensing policy*, that is, the adoption of a single fixed sensing procedure for the duration of a given task. The static sensing policy is the fundamental component that makes planning for selective perception computationally feasible.

We begin by introducing the notion of a sensing procedure. We use the Robot and Cup domain through-

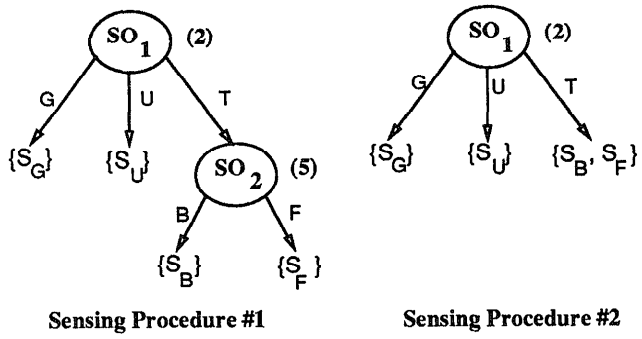

Figure 1: Sensing Procedures

out the paper as an example domain to demonstrate the basic ideas. Next, we present the concept of static sensing policies and describe planning algorithms that take advantage of static sensing policies. Finally, we discuss methods for introducing hierarchical layering of statically sensed plans.

Sensing Procedures

For a given task, the choice of what to sense can be viewed as the choice of which distinctions in the world the agent should perceive. Perceiving more distinctions usually results in a better basis for choosing appropriate actions, but also usually results in higher sensing costs.

Distinctions are made by executing a sequence of sensing operations. Each individual sensing operation discriminates between possible subsets of world states, and the choice of which sensing operation to perform may depend upon results of previous sensing operations in the sequence.

This process is captured with the introduction of *sensing procedures*. The two sensing procedures of Figure 1 represent miniature programs where execution begins at the root and terminates at a leaf. The path followed depends upon the result of the previous sensing operation. A unique value for the sensing procedure is assigned to each leaf.

The two fundamental properties of sensing procedures are that the probability of reaching a given leaf is a function of the current world state, and that the cost of executing a sensing procedure is a function of the leaf reached. When sensing procedures are represented in tree form as in Figure 1, the cost of execution for a given leaf can be calculated by summing the costs of each sensing operation on the path from the given leaf to the root. However, since the sequence of sensing operations depends on the world state, it is not possible to know in advance what the actual cost of running the sensing procedure will be.

When sensing operations are noiseless, such that their output value is uniquely determined by the world state, the leaves of the sensing procedure correspond to mutually exclusive and exhaustive subsets of the possible world states. Thus, the leaves correspond to the set of distinctions the sensing procedure is able to make. It is also possible to handle any time-invariant distribution of sensor noise by specifying for each leaf l and world state s_i the values $\eta_{l,i} = Pr\{\text{SP reaches } l | state = s_i\}$. For simplicity, the discussion in this paper is limited only to the case of noiseless sensing operations (where all $\eta_{l,i}$ values are either 0 or 1). However, the approach and all of the techniques reported in this paper have been developed to handle general sensor noise.

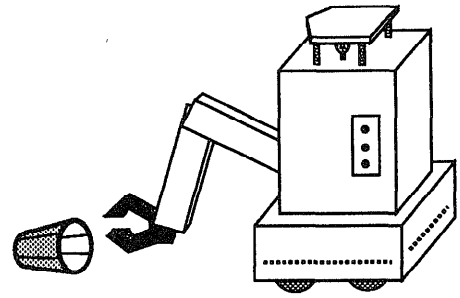

Figure 2: The Robot and Cup Domain.

Example Domain

In this paper we will adopt the Robot and Cup domain shown in Figure 2 as a simple working example. The robot's task is to grasp the cup using its three actions A1:UPRIGHT-GRASP, A2:SPIN, and A3:SIDE-GRASP. The UPRIGHT-GRASP action succeeds 80% of the time in grasping the cup when the cup is in the upright state. However, when it fails, it always tips the cup over. When the cup is tipped forward with the mouth of the cup facing the robot, the SIDE-GRASP action (shown in Figure 2) succeeds 90% of the time in grasping the cup; however, if it is tipped backward with the mouth away, the SIDE-GRASP action is ineffective. In this case, it is best to SPIN the cup giving a 50% chance that after the action, the mouth of the cup will face the robot. The effects of all these actions are summarized in Figure 3 as a *Markov Decision Process (MDP)* [Howard, 1960]. We label the four possible states as

s_G=Cup is Grasped (the goal).
s_U=Cup is Upright.
s_F=Cup is Tipped Forward.
s_B=Cup is Tipped Backward.

Our primary concern is the robot's perceptual capabilities. The two sensing operations SO_1 and SO_2 are available for execution. Loosely speaking, SO_1 discriminates between upright, tipped over, and grasped states, and SO_2 senses the orientation of the mouth of the cup. With this example, we focus primarily on whether sensing operation SO_2 is worthwhile by comparing the utility of the two sensing procedures shown

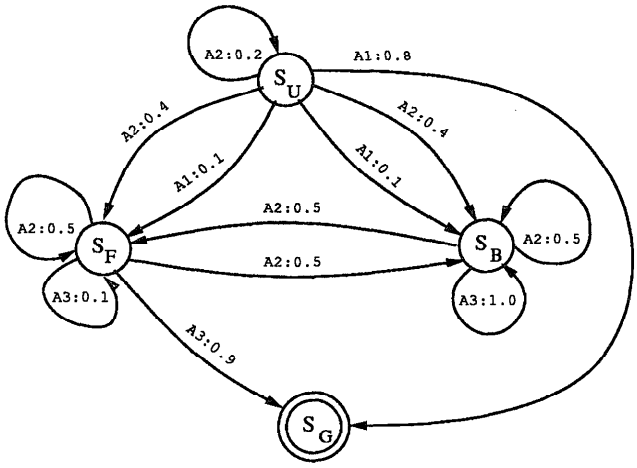

Figure 3: Transition Probabilities

in Figure 1.

The basis for determining whether a given sensing operation is worthwhile eventually reduces to the relative costs of actions and sensing. We will arbitrarily assume that each of the three physical actions are of unit cost, $Cost(A_i) = 1$, and that perception is slightly more expensive with $Cost(SO_1) = 2$ and $Cost(SO_2) = 5$.

While the Robot and Cup domain was chosen for expository purposes due to its simplicity, the methods presented in the paper are applicable to any fixed arbitrary set of markov actions and sensing procedures.

Static Sensing Policies

The introduction of sensing operations into planning problems increases the difficulty of the planning task enormously. Because sensing operations do not actually change the physical world, the agent cannot reason solely about world states but must instead reason within the space of possible *beliefs* about the world. This move to belief space results in an explosion of complexity in the planning space (In the most general case, belief space consists of all possible state mass distributions). If a senseless planner reasons within a space of n possible discrete world states, after sensing operations are added the corresponding sensible planner would reason within an n-dimensional continuous space. This general case has been studied in detail in the area of *Partially Observed Markov Decision Processes* (POMDPs) [Monahan, 1982], [Koenig, 1991].

Any effective integration of planning and intelligent sensing will require restricting generality in order to obtain computational feasibility. To be feasible, the result should preserve the tractability and scalability of the basic (senseless) planning process. We feel it is important for any theory of intelligent sensing to be cast within a framework that at least roughly preserves this basic planning complexity.

The adoption of a *static sensing policy* addresses these concerns and forms the basis of our approach. There are several advantages to adopting a static sensing policy, the most important being that it is possible to perform a decision-theoretic analysis of the utility of sensing versus action by solving the corresponding decision process in terms of the basic state space, thus avoiding the move to belief space. This is possible because the static sensing costs can be directly folded into action models, preserving the decision process over the original state space.

Before elaborating further, consider the basic execution cycle of an agent. Execution can be viewed as a repeating *sense–act* cycle. At the beginning of each cycle, the agent selects a sensing procedure to perform. At one extreme, the agent may choose a null procedure corresponding to no sensing at all (cf. [Erdmann and Mason, 1988]). At the opposite extreme the agent perceives every distinction possible. Between these extremes exists a continuum of possible sensing procedures with varying characteristics.

When, for a given task or subtask, an agent always chooses the same sensing procedure for the *sense* step of the sense–act cycle, we say the agent is using a *static sensing policy.* Note that the use of a static sensing policy does not imply that the same sequence of sensing operations is performed at every cycle, since sensing procedures are highly conditionalized upon the results of the previous sensing operations. In different world states the precise sequence of sensing operations will differ. Another important point is that an agent commits to a specific sensing procedure only for the duration of a single task. By selecting different sensing procedures, the agent varies what it perceives for different tasks.

The disadvantage with pure static sensing policies is simply that the potential sensing behavior is restricted. The limitations are significant in real applications. There are two primary techniques for overcoming these limitations while preserving the static structure and tractability.

The first and simplest technique is the use of *virtual sensing operations.* A node that predicts the outcome of a sensing operation without actually performing the operation may be inserted into a sensing procedure. The sensing efficiency gained from the use of virtual sensing operations comes with a tradeoff of robustness. When physical sensing operations are used, unexpected events or sensor noise is quickly detected and reacted to on the following sense–act cycle. These can go unnoticed when virtual sensing operations are used. For this reason, a virtual sensing operation is appropriate when it has a very high probability of correctly predicting the true sensor value.

Another technique for overcoming the limitations of static sensing policies is the introduction of hierarchy. Hierarchy results from substituting complete plans that encapsulate lower levels in place of the *act* step of the

sense-act cycle. Each hierarchical level in a plan uses a static sensing policy, but the policy may change between levels. Each level treats the nested plans simply as "*mega-actions*" which occupy the act step of the sense–act cycle.

Action Selection

This section considers the problem of finding an optimal plan given a particular static sensing policy. The comparison between two competing sensing procedures is performed by directly comparing the utility of the two resulting optimal plans. We begin with the case of a given sensing policy and a given collection of basic actions, where the objective is to find an optimal stationary action policy. In the following section we will generalize this process to the hierarchical case where the set of actions includes mega-actions.

The choice of state and action model representations have a major effect upon possible planning algorithms. Because one of our goals is to develop a normative theory of sensing and action, we adopt a complete markov model of the effects of actions. This has the advantage of being at least as general as any action model that would be of pragmatic interest to A.I. (a good property for a normative theory); unfortunately, it requires the state space to be small enough to explicitly enumerate for the algorithms to be practical.

We begin by incorporating the costs of sensing and acting into the MDP. The cost of a given transition is the sum of the sensing cost for the target state and the action cost. The desirability of a state can be encoded as a reward, where goals are assigned high positive rewards. The *net reward* of a transition is the difference between the state reward and the total cost. For example, using the first sensing procedure in Figure 1 for the Robot and Cup example, we have

$$\begin{aligned} r^2_{U,F} &= Reward(s_F) - [ACost(A_2) + SCost_1(s_F)] \\ &= 0 - [1+7] = -8 \\ r^3_{F,G} &= Reward(s_G) - [ACost(A_3) + SCost_1(s_G)] \\ &= 10 - [1+2] = 7 \end{aligned}$$

where $r^k_{i,j}$ is the net reward of transitioning from s_i to s_j using action A_k, $ACost(A_k)$ is the cost of executing A_k, and $SCost_1(s_i)$ is the cost of sensing using SP_1 from state s_i.

Next, a utility criterion must be adopted. We use the average expected net reward per transition over the entire problem solving episode, commonly referred to as *gain*. This can be conceptualized as the average utility per action if the robot were to repeatedly solve the present problem *ad infinitum* starting from the initial state distribution. For the example, we will assume that the initial state distribution is $\langle s_U, s_F, s_B, s_G \rangle = \langle .6, .2, .2, 0 \rangle$.

To efficiently solve the present problem we can apply Howard's policy iteration algorithm [Howard, 1960].

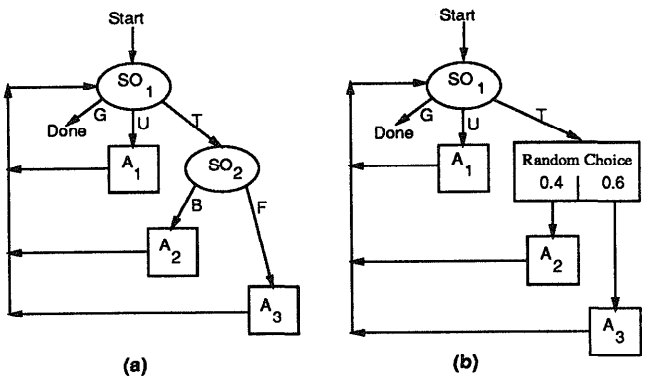

Figure 4: Reactive Plans

Applying the algorithm results in the policy in Figure 4a, shown in plan form, and a gain of -0.96. This represents the optimal behavior possible when sensing procedure 1 is used statically.

Next, apply this same analysis to the second sensing procedure in Figure 1. The same steps as above are repeated and new net transition rewards result. However, one critical difference exists — sensing procedure 2 does not discriminate between s_B and s_F, and therefore the resulting network is not a pure MDP, but rather a MDP with the constraint that the action selected from s_B must be the same as the action selected from s_F. To solve the MDP with constraints, we use a variation on Nafeh's algorithm [Nafeh, 1976] based on a highly directed branch-and-bound technique which guarantees an optimal non-randomized stationary policy. In the current example, the resulting policy has a gain of -3. This corresponds to the optimal reactive behavior based on static sensing procedure 2 when the robot is not allowed to base its choice of action on the flip of a coin.

If we allow the robot to randomly choose its action, the plan in Figure 4b with gain -0.72 is the optimal reactive plan. At the random choice in the figure, action A_3 is chosen with probability 0.6. For a given coin bias, the above techniques can be used by introducing a new action as the weighted sum of the options. Choosing the optimal bias efficiently is an open problem.

The example so far has shown that for non-random action policies, it is better to discriminate between the two tipped states using SP_1. If the agent can choose some of its actions randomly from certain states, then the utility gained from the extra sensing operation SO_2 does not outweigh its cost and the plan in Figure 4b is superior. The next section will show that the use of hierarchy can provide an even further improvement.

Hierarchy

We now consider the introduction of hierarchy in the form of *mega-actions* which encapsulate lower hierar-

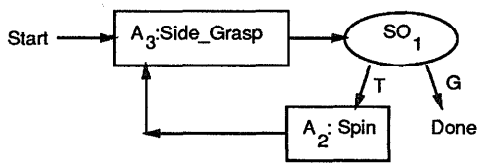

Figure 5: A Mega-Action

chical levels. Basically, a mega-action is any arbitrary plan complete with sensing, action sequences, iteration, and conditionals. While the hierarchical planning process is beyond the scope of this paper, here we consider the process of solving for optimal plans given a sensing policy and a set of actions and mega-actions. Currently, mega-actions can be assumed to be supplied by the system designer or cached as the result of previous planning episodes.

Once invoked, a mega-action retains complete control until it terminates. We select between mega-actions by assigning their invocation to a particular state in the same way we assigned basic actions to states in a stationary policy.

To solve the MDP with equality constraints in the presence of mega-actions, we modify the basic policy-iteration process in Nafeh's algorithm. For each instantiation of a mega-action, we introduce new temporary nodes into the markov process graph corresponding to virtual world states that occur at the internal nodes of the mega-action. We then replicate each action in the mega-action as a transition arc in the graph, and fold the sensing operations into the transition costs between the temporary nodes. The result is an enlarged state graph that can be treated using variants on the standard methods from Nafeh's algorithm.

We demonstrate the result of this process for the robot and cup example when the mega-action of Figure 5 is introduced. In the example, since it is not possible to occupy state s_U directly after executing A_3, the U branch of SO_1 is irrelevant and has been omitted from the figure. When sensing procedure 2 is used, the algorithm assigns A_1 to s_U and the mega-action to s_F and s_B. The resulting action policy is shown in plan form in Figure 6 and has a gain of 0.619. This plan can no longer be viewed as a reactive plan — it is distinctly procedural. It also has a higher utility than the plan in Figure 4a and thus leads to the conclusion that it is best to not discriminate between s_F and s_B when using the mega-action.

Related Work

The explicit consideration of sensing operations and sensing costs has been rare in A.I. and is usually dominated by the concern of action scheduling. Nevertheless, there is other work that is relevant to the current research. Space permits only a brief mention of the most relevant.

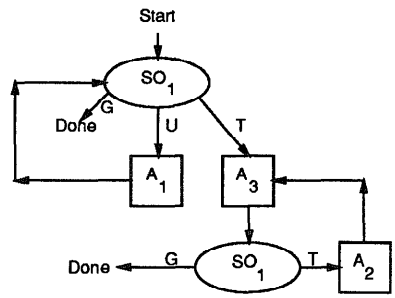

Figure 6: Procedural Plan

Work by [Whitehead and Ballard, 1990] in the area of Reinforcement Learning shares many similar aspects to our own work and considers the problem of learning perceptually unambiguous mappings from world states to internal states.

Tan [Tan, 1990] uses an ID3 style algorithm to inductively learn sensing procedures for selecting an appropriate grasping strategy.

Work in execution monitoring has dealt with inserting explicit, selective sensing operations into plans for the purposes of detecting stray execution paths and to mark the reinvocation of a planner [Doyle *et al.*, 1986], [Miller, 1989], [Van Baalen, 1984]. [Brooks, 1982] considered explicit sensing for the purposes of reducing numerical uncertainty.

Dean [Dean *et al.*, 1989] adopts, like us, a decision-theoretic approach to planning and control where sensing operations and physical actions are considered at the same level. Dean's group has concentrated on the efficient evaluation of a given plan using the more concise bayesian influence-diagram based representations for states and action.

Koenig [Koenig, 1991] discusses many of the issues involved in using MDPs and POMDPs for modelling and solving planning problems.

An alternative methodology has been advanced recently of deferring planning until the acquisition of requisite sensory data [Hsu, 1990], [Gervasio, 1990], [Olawsky and Gini, 1990]. Some issues relevant to this comparison are discussed in [Olawsky and Gini, 1990].

Discussion

As the number and complexity of tasks that a robot performs becomes large and the required sensing capabilities grows, it becomes more and more the case that *a plan without sensing is a senseless plan.* Just as the effective selection and scheduling of physical action constitutes the foremost concern of today's robots, the effective selection and scheduling of sensing operations will represent a primary concern in the more competent robots of the future.

We have presented selective sensing as the process of choosing which discriminations to perceive and weighing the cost of sensing against the utility of the extra information obtained. The key to planning for perception while preserving the basic complexity of the models of action and world state comes from the adoption of a *static sensing policy*. When the static restriction is too strong, additional flexibility can be obtained using techniques such as virtual sensing and hierarchy that preserve the static structure across any given level.

The current work represents initial results in a long term research program, and as such, a number of items are on our agenda for future research. While the general markov action models are appropriate for the initial development of a normative theory, pragmatically they are quite limiting in domains with many world states [Ginsberg, 1989]. Thus, we plan to study other, more concise action model and state representations. Also important are hierarchical planning methods and directed strategies for searching through the space of possible sensing procedures. We are also interested the interplay of learning with these methods, especially with respect to learning action models and caching mega-actions, and in developing a more precise understanding for the role of virtual sensing operations.

Acknowledgements

We would like to thank Sven Koenig and Tom Mitchell for helpful and influential discussions about the topics in this paper. Sven Koenig directed us to the relevant literature in MDPs and POMDPs that formed the foundations for this research. This research was sponsored by NASA under contract number NAGW-1175.

References

[Brooks, 1982] R. A. Brooks. Symbolic error analysis and robot planning. *International Journal of Robotics Research*, 1(4):29–68, 1982.

[Brooks, 1986] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, March 1986.

[Dean *et al.*, 1989] T. Dean, K. Basye, R. Chekaluk, S. Hyun, M. Lejter, and M. Randazza. Coping with uncertainty in a control system for navigation and exploration. In *AAAI*, Boston, MA, 1989.

[Doyle *et al.*, 1986] R. J. Doyle, D. J. Atkinson, and R. S. Doshi. Generating perception requests and expectations to verify the execution of plans. In *AAAI*, pages 81–87, August 1986.

[Erdmann and Mason, 1988] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4(4):369–379, August 1988.

[Fikes and Nilsson, 1971] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[Gervasio, 1990] M. T. Gervasio. Learning general completable reactive plans. In *AAAI*, 1990.

[Ginsberg, 1989] M. L. Ginsberg. Universal planning: An (almost) universally bad idea. *AI Magazine*, 10(4):40–44, Winter 1989.

[Howard, 1960] R. A. Howard. *Dynamic Programming and Markov Processes*. The M.I.T. Press, 1960.

[Hsu, 1990] J. Y. Hsu. Partial planning with incomplete information. In *AAAI Spring Symposium on Planning in Uncertain, Unpredictable, or Changing Environments*, Mar 1990.

[Kaelbling, 1986] L. Kaelbling. An architecture for intelligent reactive systems. In *Proc. of the Workshop on Planning and Reasoning about Action*, 1986.

[Koenig, 1991] S. Koenig. Probabalistic and decision-theoretic planning using markov decision theory. Master's thesis, U.C. Berkeley, 1991. In Preparation.

[Miller, 1989] D. P. Miller. Execution monitoring for a mobile robot system. In *SPIE Conference on Intelligent Control*, 1989.

[Monahan, 1982] G. E. Monahan. A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science*, 28:1–16, January 1982.

[Nafeh, 1976] J. Nafeh. *Markov Decision Proceses with Policy Constraints*. PhD thesis, Stanford University, Dept. of Engineering-Economic Systems, April 1976.

[Olawsky and Gini, 1990] D. Olawsky and M. Gini. Deferred planning and sensor use. In *Proc. DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, Nov 1990.

[Sacerdoti, 1975] E. D. Sacerdoti. The nonlinear nature of plans. In *IJCAI*, 1975.

[Simmons, 1990] R. Simmons. Robust behavior with limited resources. In *AAAI Stanford Spring Symposium*, 1990.

[Tan, 1990] M. Tan. CSL: A cost-sensitive learning system for sensing and grasping objects. In *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, 1990.

[Ullman, 1984] S. Ullman. Visual routines. *Cognition*, 18:97–156, 1984.

[Van Baalen, 1984] J. Van Baalen. Exception handling in a robot planning system. In *Conference on Principles of Knowledge-Based Systems*, December 1984.

[Whitehead and Ballard, 1990] S. D. Whitehead and D. H. Ballard. Active perception and reinforcement learning. In *Proc. of the Seventh International Conference on Machine Learning*, 1990.