Action and Perception in Man-Made Environments*

Daniel D. Fu and Kristian J. Hammond and Michael J. Swain
Department of Computer Science
The University of Chicago
1100 East 58th Street, Chicago, Illinois 60637
U. S. A.

Abstract

We discuss the types of functional knowledge about an environment an agent can use in order to act effectively. We demonstrate (1) the use of structural regularities for acting efficiently, and (2) the use of physical regularities for designing effective sensors. These ideas are described in the context of an everyday task: grocery store shopping. We discuss how SHOPPER, a program, uses regularities of grocery stores in order to act appropriately and sense efficiently in GROCERYWORLD, a simulated grocery store.

1 Introduction

Much of the useful knowledge people employ in everyday life is often implicitly understood to be common knowledge that everyone possesses. For example, getting a drink of water in someone else's home involves many assumptions: the kitchen is located on the first floor, there are no major obstructions to getting there, a faucet over a sink will be in the kitchen, clean glasses will be in the cupboards, the glasses will be near the faucet, etc. People use these social rules within a shared culture to function effectively. Without them, unlikely possibilities would be admitted (e.g., the storage of clean glasses in the shed out back). None of these possibilities could be dismissed.

To a large extent, researchers building autonomous agents to work within a culturally rich environment have implicitly built these assumptions into their agents. For example, many robots don't explicitly avoid holes in the floor. That's because there aren't any. As pointed out in [Agre, 1988], cultures go out of their way to make environments safe for people.

With the SHOPPER project, we are studying the ways in which an agent can use the structure of the environment to its own advantage. In particular, we are identifying the useful types of knowledge an agent can use to shop in a grocery store. Grocery stores are completely man-made environments in which practically everyone has to look around for the items they need. Stores in general are a good domain since their organization is intended to be simple enough for everyone to comprehend. This organization makes it easy for a person to go into any grocery store and shop effectively.

In addition to the way environments are structured, we can also simplify sensor design by taking into account the information needed from the sensor. Our primary sensing is in the form of color images the agent sees in GROCERYWORLD, a simulated store. For the primary task of finding an item in an image, we consider the task in the context of a grocery store, as opposed to attempting something more general. In the section on computer vision, we describe methods SHOPPER uses for gathering information for recognizing objects quickly and effectively.

2 Related work

Tasks within the context of a highly structured environment have been studied before. Hammond & Converse [1991] have noted that our environments are designed to aid rather than hinder activity. Regularities, actively maintained, can greatly simplify a person's interactions with the world. They demonstrate the efficacy of this approach for the task of making coffee in a simulated kitchen. Agre & Horswill [1992] investigate the influence of culture in TOAST, a program which cooks food in a simulated kitchen. They demonstrate how activity in the midst of cultural artifacts can be improvised to produce nontrivial behavior. They do this by characterizing regularities, or constraints, on cooking tools and materials. Because all the necessary tools are nearby and the materials undergo straightforward transformations, they show how cooking tasks become much simpler.

Much of the vision work described here is based on recent work done in active and purposive vision [Ballard, 1991; Aloimonos, 1990]. For simplifying sensor design, Horswill [1993] has noted that environments have computational properties which allow a designer to build simpler sensors. By starting with a complex mechanism to compute a piece of information, successively simpler, less general mechanisms can be substituted by noting regularities an environment supports. These regularities can be used for analyzing under what conditions sensors will and won't work. Horswill describes POLLY - a robot designed to give guided tours. Polly's visual mech-

This work was supported in part by Office of Naval Research grant number N00014-91-J-1185.

anisms are very simple (optimized) since they're tailored to a specific environment.

Because conditions in the environment change, Pinhanez and Bobick [1995] maintain an approximate world model for selecting appropriate visual computations. These computations have *applicability conditions* which must hold in the current context in order for their results to be acceptable. For example, by knowing the location of a chef who is wrapping a chicken, the appropriate hand tracking routines can be selected and instantiated.

3 Shopping in GroceryWorld

While we would like SHOPPER to eventually function in a real store, we have opted to work with a simulator first. There are two reasons for this: (1) A real grocery store is unavailable for frequent testing, and (2) For now we want to avoid problems with real robots, like fixing broken hardware, writing motor driver code, having to transport the robot, etc. We are also able to ignore problems such as noise in sonar readings and wheel slippage; however, we intend to incorporate similar problems in the simulator.

The simulator we have built is called GROCERYWORLD. This "world" is a novel simulator in that it integrates some real visual data along with simulated sonar information. While being a simulation, we still wanted to address some real sensing problems: GROCERYWoRLD is a videodisc-based reproduction of a local grocery store. The store has nine aisles of food items. For each aisle, four film clips were taken to provide views of each side as well as up and down the aisle. In total, the simulator provides access to over 75,000 images by merely moving around the store.

GROCERYWORLD also provides simulated range information on the relative proximity of objects to the agent. Sign information is also given. When an agent is at the end of an aisle and looking down that aisle, it automatically receives the symbolic text of those signs. The signs in GROCERYWORLD are a faithful reproduction of the signs in the real grocery store.

In the next few sections we will illustrate the types of structural regularities SHOPPER can use in a grocery store. Next we discuss the control and visual routines which make use of structural and physical regularities. Then we step through an example of SHOPPER finding an item. In the last section we end with a discussion.

4 Regularities in a grocery store

Any customer shopping for groceries in a store can find the necessary items in reasonable time whether or not they've been to the store before or not. Yet the average store stocks around 10,000 items. In order to sift through all the food items available, customers are able to exploit their knowledge about the structure of the store. Stores organize food items consistently so that customers can find items without too much difficulty. Below we illustrate the different types of knowledge that can be used for finding goods.

- **Type:** Stores group together items of the same type or that serve the same function. This is a most basic organization principle under which many items fall under; e.g. Mcintosh apples are near Rome apples; Gerber baby food will be found with other baby foods; tomatoes clustered with other vegetables; an apple placed with other fruits; coffee is near tea.
- **Brand:** Within a section of a specific type, foods made by the same company will be clustered together. For example, in a typical grocery store aisle, soups of the same brand (e.g. Campbell's, Progresso) will be clustered with each other.
- **Counterparts:** Items that complement each other are sometimes grouped together. For example, salad and salad dressing, pancakes and maple syrup, pasta and tomato sauce, etc.
- Physical Constraints: Perishable items requiring refrigeration or bulky items requiring larger storage space. For example, orange juice, eggs, frozen entrees, charcoal, etc.
- **Specialty foods:** Stores often have sections devoted to foods related to certain cultures, countries, dietary foods: e.g. soy sauce, curry, matzah, water cress, refried beans.
- Packaging: Bulk items such as bags of oranges, apples, and potatoes will be placed separate from their individual versions.

These regularities are also principles under which store designers build stores [Peak and Peak, 1977]. One natural way of segmenting the space in the store is by the use of aisles. Items of similar nature are placed together in the same aisle. Sometimes signs are placed at the end of an aisle to indicate some contents of that aisle.

A person looking for an item can use regularities along with knowledge of how a store is organized to select promising areas. So if we wanted to find a box of Froot-Loops cereal, a sign above an aisle that says "cereal" can serve as a pointer to the location of the cereal using the regularity of "type." A sign saying "syrups" could be useful in finding Mrs. Butterworths pancake mix according to the "counterparts" relation. See [Fu *et* a/., 1994] for an example of SHOPPER finding a box of pancake mix.

5 Action and Perception

In this section we discuss the control and visual routines which make use of the regularities described earlier.

5.1 Control of action and perception

SHOPPER uses hierarchical plans to control all decision-making and actions. The plan representation is a version of that used in RUNNER [Hammond *et* al., 1990] and include ideas taken from RAPs [Firby, 1987]. Initially, a plan is given a *permission* to activate. An active plan first checks to see if its objectives (its *success* clause) are met. If so, it finishes. If not, it selects a method based on current context (sensor and state) information. Each method will have a sequence of plans or actions. These plans and actions will then be permitted (retrieved and activated) in sequence, as successive plans succeed.

Execution of this control mechanism behaves in a very "depth-first search" manner by permitting abstract plans which become more and more concrete depending on sensor/state conditions. The resulting "leaves" are either physical, visual, or mental actions. For example "(align-body-to-head)" is a physical action which orients the direction of travel to the direction the head is facing.

```
(defplan (move-down-aisle-looking-for ?item)
  (success (or (see-verified ?item)
               (not (clear-space forward))))
  (method (context (and (see-sign ?type)
                        (isa ?item ?type)))
          (serial (align-body-to-head)
                  (move-out-of-intersection)
                  (look-for ?item)))
  (method (context (and (isa ?item ?type)
                        (counterpart ?type ?other-type)
                        (see-sign ?other-type)))
          (serial (align-body-to-head)
                  (move-out-of-intersection)
                  (disable all) ;; deactivate all histograms
                  ;; activate histograms related to signs
                  (sign-enable)
                  (look-for-type ?type ?other-type ?item)
                  (search-vicinity ?item))))
```

Figure 1: A plan to select and execute a method for finding an item in an aisle after a relevant sign is encountered.

In Figure 1, the plan is satisfied if either the item sought has been spotted, or the end of the aisle has been reached. If not, a method is chosen. The two methods listed in Figure 1 represent two different search strategies: looking for a specific item, or looking for an item's related types and then searching a local vicinity.

5.2 Vision in GroceryWorld

The vision operations rely on the regularities discussed in the previous section as well as some simple assumptions we make about the domain:

- The lighting comes from the ceiling.
- Items usually sit directly on shelves.
- Food items are displayed on shelves in a consistent manner, e.g. cereal boxes are upright with the front of the box facing outward.

Basing vision routines on these assumptions allows us to build a very effective ensemble which, while being very simple and easy to understand, combine to execute nontrivial visual tasks.

SHOPPER uses three basic vision routines for obtaining information from the images. The routines (in order of increasing complexity) are: shelf detection, histogram intersection, and comparison of edge images using Hausdorff distance. Figure 2 shows the three routines in intermediate states.

The first routine is a shelf detector. This helps to constrain the relevant regions in an image. Given that the agent is looking at a side of an aisle, we locate the shelves by assuming that (1) light comes from above, and (2) the shelves are light in color. From these assumptions, we build a simple filter sensitive to changes from light to dark since shadows are cast beneath shelves. The detector histograms the responses and then finds maxima by partitioning the ID histogram. The maxima correspond to shelf locations in the image.

The second routine is a histogram intersection routine [Swain and Ballard, 1991]. Histogram intersection involves discretizing the pixels of a food item image into a color space histogram. Intersection matches are determined by intersecting two color spaces. Given a model histogram M and a sample histogram I with n color bins each, the intersection is computed as $\sum_{k=1}^{n} \min(I_k, M_k)$, which indicates the number of color pixels in the model which also appear in the sample. In order to obtain a fractional value, the result is normalized by the size of the model. Sample histograms are taken successively across a shelf area. The sample size is exactly the same size as the original model.

The third routine we use is a comparison function using Hausdorff distance [Rucklidge, 1994] to compare two edge images. Hausdorff distance is a measure of how close a set of model points are to a set of image points, and vice versa.

Because each routine's speed is related to the size of the image, we sought to successively limit the size of regions of interest. So, we first constrain search to be on shelves. Then, on each shelf we further bound the region using color. Finally, we compute the Hausdorff distance over the smallest region possible. Computing the Hausdorff distance is the most expensive operation. By taking into account the safe assumptions available to us, we are able to constrain areas in the image for processing.

From these basic routines, we create more sophisticated routines for processing images in GROCERY - WORLD. Rather than presenting them now, we will describe them in the context of an example in the next section.

6 Example

In this section we illustrate an example of finding a box of FrootLoops cereal. These regularities apply:

- Type - FrootLoops is a cereal.
- Counterparts - Milk is often used with cereal.
- Physical constraints - Milk and cereal have different physical constraints, so milk is not likely to be very near cereal.

Milk is not a good indicator for the presence of FrootLoops because of physical constraints. However, the "type" relation still holds when looking at an aisle sign. As we will demonstrate, this is correct for this particular example in GROCERYWORLD.

The following is an edited trace of SHOPPER finding a box of FrootLoops. Of the 121 primitive actions done, only illustrative ones are reported here.

```
Permitting (retrieve-item frootloops)
 Permitting (align-head)
  [Action: (aligning head to body)]
```

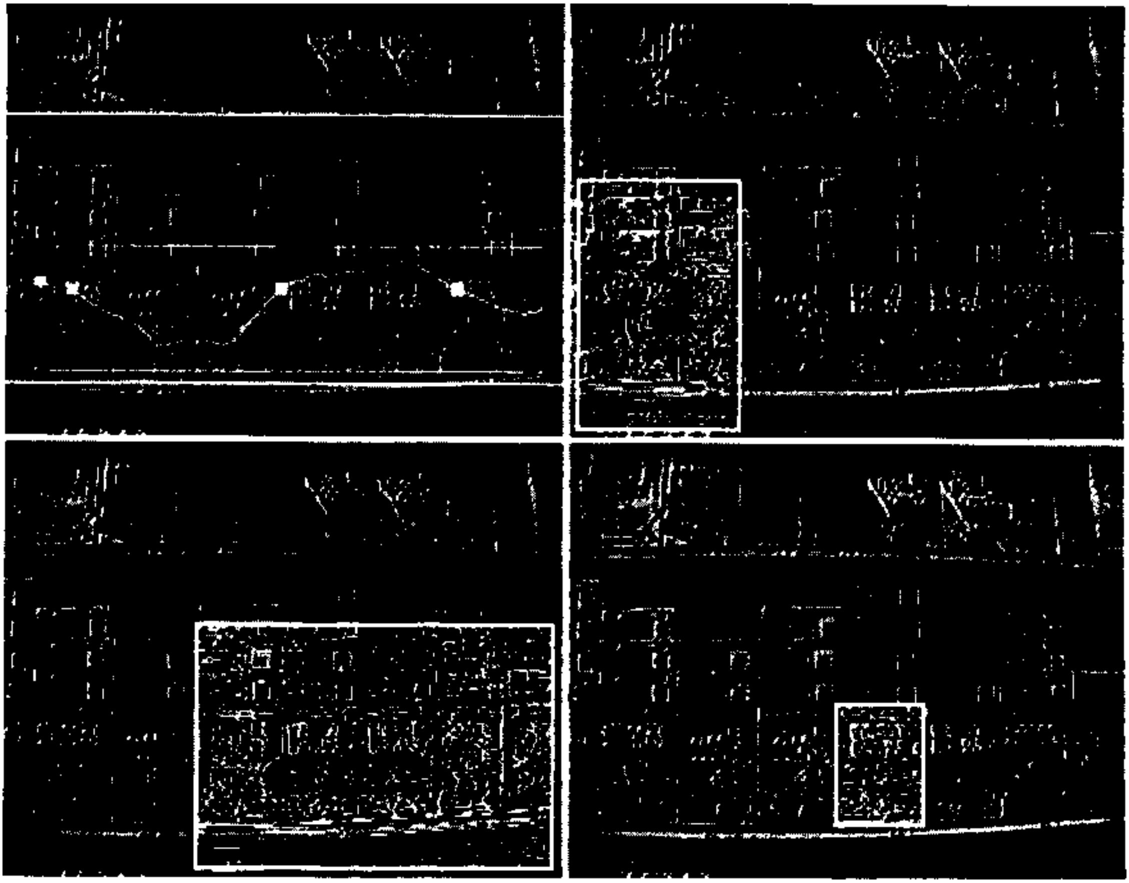

Figure 2: From top left: (a) Image of cereal boxes with shelf positions drawn as solid lines. The jagged lines are made of match values of histograms taken above a shelf against a histogram of FrootLoops. The higher the point, the better the match. Regions of interest are bounded as seen by the bigger dots on the jagged line, (b) Shows an edge image taken around a cereal box of Smacks for subsequent comparison with FrootLoops using Hausdorff distance. (c) Same as in b, but with FrootLoops. (d) FrootLoops' edge image is superimposed; it's found.

```
 Permitting (find-sign)
  [Action: (turning body left)]
```

At this point, SHOPPER is looking down the first aisle at the entrance to the store. Sign information is passed from the simulator: "aisle-1 bread cracker cookie meat frozen-entree baked-good."

From here, SHOPPER executes a plan to move across aisles by first picking an open direction to move and then turning the head back toward the aisle. This way, SHOPPER can read signs while moving across aisles.

```
  [Action: (turning body left)]
  [Action: (turning head to look right)]
 Permitting (move-across-aisles-looking-for
                frootloops)
  [Action: (moving forward)]
  [Action: (moving forward)]
```

At this point, SHOPPER keeps moving forward until a relevant sign is seen in the fourth aisle: cereal. Because the sign is relevant according to the "type" relation, SHOPPER commits to exploring this aisle. See the first method in Figure 1.

```
Permitting (move-down-aisle-looking-for frootloops)
 Permitting (align-body)
  [Action: (aligning body to head)]
 Permitting (move-out-of-intersection)
  Permitting (move-forward)
   [Action: (moving forward)]
   [Action: (moving forward)]
 Permitting (look-for frootloops)
  Permitting (look-head-left)
   [Action: (turning head to look left)]
```

At this point SHOPPER is actively searching for a box of FrootLoops by moving down the aisle and looking left and right.

For each image processed, SHOPPER first looks for shelf locations. By taking color histograms across and above a shelf location, it can quickly tell if the box is not present if all resulting intersections are low in value.

In contrast, if the intersection values are high, we bound the regions of high response and use Hausdorff distance comparison by first using a precomputed edge image of FrootLoops and computing an edge image of the region of high response. If the edge images match well, we have verified the location of the item. If not, we consider the item to be absent from the image and continue on.

```
 Permitting (process-shelves frootloops)
  Permitting (detect-shelves)
   [Action: (checking for shelves)]
  Permitting (item-boundary frootloops 59 0.8)
   [Action: (checking for frootloops at
               height 59)]
  Permitting (item-boundary frootloops 253 0.8)
   [Action: (checking for frootloops at
               height 253)]
  [Action: (moving forward)]
  [Action: (turning head to look right)]
  Permitting (detect-shelves)
```

The information passed to plan "item-boundary" are the FrootLoops color histogram, the shelf height in the image (512x484), and the lowest histogram intersection threshold for finding boundaries for Hausdorff verification.

SHOPPER continues its search in this fashion until it eventually encounters FrootLoops:

```
 Permitting (item-boundary frootloops 315 0.8)
  [Action: (checking for frootloops at
              height 315)]
  Permitting (check-obj-boundaries frootloops)
   Permitting (hausdorff-boundary frootloops
                165 206 256 -forward 141 .86
                -reverse 141 .89
                -scale .7 .7 1.02)
    [Action: (hausdorff boundaries 165 to 206
               at height 256 for frootloops
               -f 141 .86 -r 141 .89
               -scale .7 .7 1.02)]

>> Asserting (see-verified frootloops 124 228
               0.76 0.78)
```

SHOPPER found a shelf at height 315 in the image, found a high intersection region (165 - 206), and verified its presence at coordinates (124, 228) in the local region.

This particular example was used on a second version of GROCERYWoRLD using a second videodisc. Because the second version was filmed with a different lens, we also search across scale as well as translation for a given model. The models we use were taken from the first videodisc. FrootLoops was found by scaling the model's width and height.

In this example, we used the type regularity in order to design more complicated routines. This merging of simpler visual routines into more sophisticated routines results in more robust performance at a smaller cost. The color histogram intersection routine could be scanned across the entire image and produce many possible locations for an object. However, by itself, it is not enough to reliably verify the existence of the object. The Hausdorff distance between a model edge image and an entire image could yield the same results, but at a prohibitive time cost. Since we can localize regions using shelf detection and color histograms, the area of processing is substantially reduced. By combining routines of shelf detection, color histogramming, and Hausdorff distance we are able to lessen computation time without compromising reliability.

7 Status

SHOPPER currently uses four out of the six regularities outlined earlier: type, counterpart, physical constraint, and specialty foods. Tests were conducted on two versions of GROCERYWORLD.

Many of the items we tested initially were relatively small in size - about 40x50 pixels. The size is relevant to both the color intersection and Hausdorff distance routines. For color intersection with small items, the shelf placement is critical since a vertical ten-pixel error could seriously affect the histogram intersection value when histograms are taken across a shelf. Bigger items such as laundry detergent are less affected since their histograms are based on a larger set of pixels. For Hausdorff distance, the edge image is computed from a subsampled greyscale image. The video is NTSC interlaced (odd scan lines are recorded, then even lines) and was filmed while the camera was moving. This results in jagged vertical edges. The problem was alleviated by sampling every other line. However, this makes the model edge image twice as small.

In picking items for testing, we restricted items to be of larger than normal sizes (cereals, laundry detergents, etc.). Also we did not pick items whose shape is cylindrical. Because cans and bottles can be rotated, the current Hausdorff verification method will not suffice. It might be possible to isolate the outside shape of an object a *priori* and use its label in various rotations to later find the object again. However, we have not attempted this method.

Of the twenty-five items tested on the first videodisc twenty were found (80% found), one was missed by color histogramming (false negative), one wrong item was picked (false positive) and the other three didn't match correctly using our set thresholds for Hausdorff matching (false negatives). The results for two videodiscs are summarized in Table 1.

	Videodisc 1	Videodisc 2
Number Items Tested	25	11
Correctly Found	20	6
Color Errors	1	2
Hausdorff Errors	3	1
Number False Positives	1	2

Table 1: Summary of tests for two versions of GROCERYWORLD.

The second version of GROCERY Wo RLD differed from the first in a few ways. First, the transfer process of videotape to videodisc was different since the two discs were pressed at least a year apart. Models appearing in the second version are more yellow. Second, a different lens was used so most items appear different in size.[1]

[1] The width and height of each item appearing is approximately .72 of the original model.

Also, distances to shelves while filming were not kept constant, so there is significant variation up to the size of the original model.

Correcting the color is necessary since the yellowness of images from the second videodisc directly impinges on the effectiveness of color histogram intersection. Assuming constant lighting in the store, we corrected colors in the second disc using models of known color from the first. The scale differences for edge matching were handled by loosening scale parameters in the Hausdorff distance routine. We allow scale matches from 70% to 100% of the original width and heights of the models.

Currently we only use sonar information for navigation. If the distance to an object were known, better color histogram samples (we use the original model's size) and scale parameters can be chosen since we know at what distance the original models were filmed.

8 Discussion

Regularities are general rules of thumb - not hard and fast rules. However, they provide reference points from which we can base the search for an item, as opposed to doing exhaustive search or constructing elaborate reasons for looking in certain areas. Undoubtedly, regularities will be wrong in instances. SHOPPER works within the structure of a store maintained by managers who wish to maximize their profits [Dipman, 1931; Peak and Peak, 1977] while still providing a pleasant shopping experience. This can lead to mistaken beliefs about the locations of objects. However, when a failure occurs, the regularities which pointed to a mistaken location can be identified and then repaired incrementally. Eventually, SHOPPER can learn and optimize plans of action over several visits. When new grocery stores are encountered, the agent can be better prepared since its knowledge of particular grocery stores serves as a field from which it can reap the benefits of past experience.

Note that we don't need explain why, for example, most stores have their produce section on the right perimeter next to the entrance of the store; we just need to know the tendency for produce sections to be located near the entrance. Explaining *why* produce is near the entrance does little for the typical customer while knowing *where* to find produce is most useful. As another example, toothpaste and nail clippers tend to be located near the front of the store. This is to reduce pilferage since cashiers (a form of store security) are also near the front.

From the earlier example discussed in this paper, we have demonstrated that the physical search space can be drastically reduced using functional knowledge of the domain. Prom the same example we have also illustrated visual routines which speeded computation by restricting regions of interest. These optimized the basic recognition routines we had available to us, without losing effectiveness. Certainly, the physical search mechanism depends on the environment, but everyday life has the same restraints. Any agent working in an everyday man-made domain can use its knowledge to help facilitate its own activity. In this paper we have shown the effectiveness of such knowledge.

References

[Agre and Horswill, 1992] Philip E. Agre and Ian Horswill. Cultural support for improvisation. In *The Proceedings of the Tenth National Conference on Artificial Intelligence,* pages 363-368, 1992.

[Agre, 1988] Philip E. Agre. The dynamic structure of everyday life. Technical Report 1085, MIT, October 1988.

[Aloimonos, 1990] John Aloimonos. Purposive and qualitative active vision. In *International Conference on Pattern Recognition,* pages 346-360, 1990.

[Ballard, 199l] Dana H. Ballard. Animate vision. *Artificial Intelligence,* 48:57-86, 1991.

[Dipman, 1931] Carl William Dipman. *The Modern Grocery Store.* Williams Press, Albany, New York, 1931.

[Firby, 1987] R. James Firby. An investigation into reactive planning in complex domains. In *The Proceedings of the Sixth National Conference on Artificial Intelligence,* pages 202-206, 1987.

[Fu *et al.*, 1994] Daniel D. Fu, Kristian J. Hammond, and Michael J. Swain. Vision and navigation in man-made environments: Looking for syrup in all the right places. In *Proceedings of the Workshop on Visual Behaviors,* pages 20 26. IEEE Press, 1994.

[Hammond and Converse, 199l] Kristian J. Hammond and Timothy M. Converse. Stabilizing environments to facilitate planning and activity: An engineering argument. In *The Proceedings of the Ninth National Conference on Artificial Intelligence,* pages 787-793, 1991.

[Hammond *et al*, 1990] Kristian Hammond, Timothy Converse, and Charles Martin. Integrating planning and acting in a case-based framework. In *The Proceedings of the Eighth National Conference on Artificial Intelligence,* pages 292-297, 1990.

[Horswill, 1993] I. Horswill. *Specialization of Perceptual Processes.* PhD thesis, MIT Dept. of Electrical Engineering and Computer Science, 1993.

[Peak and Peak, 1977] Hugh S. Peak and Ellen F. Peak. *Supermarket Merchandising and Management.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

[Pinhanez and Bobick, 1995] Claudio S. Pinhanez and Aaron F. Bobick. Using approximate models as source of contextual information for vision processing. In *Proceedings of the Workshop on Context-Based Vision.* IEEE Press, 1995.

[Rucklidge, 1994] William Rucklidge. *Efficient Computation of the Minimum Hausdorff Distance for Visual Recognition.* PhD thesis, Cornell University Department of Computer Science, 1994. Technical Report TR94-1454.

[Swain and Ballard, 199l] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision,* 7:11-32, 1991.