Where to Look Next using a Bayes Net: An Overview

Raymond D. Rimey*
The University of Rochester
Computer Science Department
Rochester, New York 14627

1 Introduction

Our goal is to formalize and implement mechanisms for true task-oriented (not task-specific) vision. In our model of a task-oriented vision system, a question is first asked about the scene. The system determines what scene information would be sufficient to answer the question. The level of detail sufficient to answer the question may vary among the different pieces of information extracted from the scene. Specific vision modules are sequentially brought to bear on selective areas of the image, the selection and exact processing of each vision module depending on the results of previously executed vision modules. Each module produces a partial representation of the (minimal) information needed to answer the question, and these results are combined to produce the answer. Table 1 summarizes the key differences between task-oriented vision and the classical, passive approach to computer vision.

A "where to look next" capability can act as the cognitive executive for active vision. If an active computational agent is subject to an information load that can overwhelm its resources, the executive can allow it to ignore irrelevant stimuli, choose its tasks wisely, survive, and achieve its goals. Alternatively, "where to look next" can simply save time and effort in doing visual jobs that in humans require attentional shifts, such as radiograph and CAT scan interpretation, photo interpretation, traffic monitoring, etc. Last, the task-oriented approach should make for more dependable vision performance built from more general (less domain-specific) vision tools. The claim is that current vision modules, even relatively simple ones, can become useful and robust when they are carefully applied in a specific context.

Our approach has as background a large amount of research into visual attention, classical work in eye movements, and recent advances in active vision, including camera movements and foveal - peripheral sensors. Specifically, our tools are decision theory, utility theory, and Bayesian probabilistic models [3, 4, 6, 7]. Two recent key developments are Bayes nets [9], and influence diagrams [9, 11]. Applications using these new techniques are beginning to appear. The first large experimental system that applied Bayes nets to computer vision is by Levitt [8]. The formulation of that system using influence diagram techniques is discussed in [2]. A sensor/control problem involving a real milling machine is solved using influence diagram techniques in [1]. A special kind of influence diagram, called a temporal belief network, is discussed in [5], and is being studied for an application in sensor based mobile robot control.

In what follows we present the basic framework of a task-oriented computer vision system, called TEA, that uses Bayes nets and a maximum expected utility decision rule. Knowledge about the scene and about the nature of the specific task given to the system are represented in the Bayes net. The decision of what vision modules to run is made using a value/cost utility measure, where value is based on mutual information measured between nodes in the Bayes net that correspond to actions and to the goal of the task. We introduce a new method for incorporating relational knowledge both into the Bayes net and into the util-

Passive vision	**Task-oriented vision**
use all vision modules	use only some vision modules
process entire image	process areas of the image
maximal detail	sufficient detail
extract representation first	ask question first
answer question from representation data	answer question from scene data
unlimited resources	resource limitations

Table 1: Key differences between passive vision and task-oriented vision.

*This material is based on work supported by the National Science Foundation under Grants numbered IRI-8920771 and IRI-8903582. The Government has certain rights in this material.

ity measure. The decision of what areas of the scene to run a vision module on can be made using this relational knowledge. TEA models camera movements and distinguishes between vision modules that operate either on foveal or on peripheral image data.

Experimental results are presented from the TEA-0 system, our initial implementation of the general TEA framework. We also outline TEA-1, which uses a richer knowledge representation to support more complex visual tasks. The TEA systems solve the "where to look next" problem, enabling us to study a problem we call "how to look", and by combining our solutions to these problems we expect to build a true task-oriented vision system. This paper is intended to be mainly an overview of our work. A fuller treatment appears in [10].

2 Preliminaries

The TEA system runs by iteratively selecting the evidence gathering action that maximizes an expected utility criterion:

1. List all the executable actions.
2. Select the action with highest utility.
3. Execute that action.
4. Attach the resulting evidence to the Bayes net and propagate its influence.
5. Repeat, until the task is solved.

The following section expands on aspects of the above algorithm. But first we present the application domain that we are currently using.

The approach is applicable whenever the scenes obey regularities or have structure that can be captured in the semantic data structures supporting Bayesian inference. One example is biomedical images that reflect known relationships and properties of anatomy. Another is aerial views of certain cultural areas such as industrial sites or airports. There is nothing inherently 2-D in the method. With the TEA system we currently use table settings. Figure 1 shows a typical table top scene.

We assume a spatially-varying sensor, which makes the "where to look next" question even more central. In the TEA system the peripheral image is a low-resolution image of the entire field of view from one camera angle, and the fovea is a small high-resolution image (*i.e.* window) that can be selectively moved within the field of view.

We assume the system can not view the entire scene at once. Often a camera movement must be made to an area of the scene that has not been been viewed before. The target location of such a camera movement must be determined via relations with other portions of the scene for which image data is (or previously has been) available. Following a camera movement the fovea is centered in the field of view, but afterwards the system can move the fovea within the field of view. The target location for a fovea movement is always within the field of view so it can be determined either from peripheral image data or by relations with other portions of the scene.

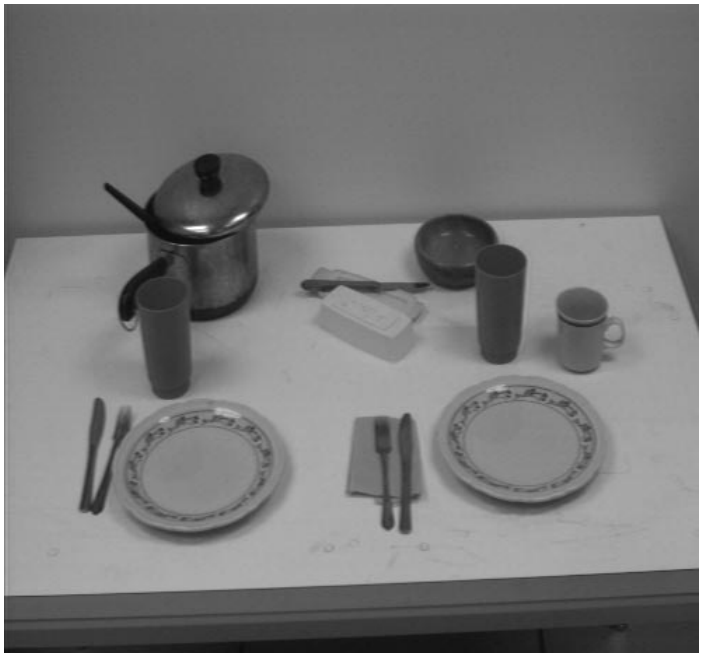

Figure 1: An example scene in the application domain.

Our goal is to support many different visual tasks efficiently. Each task can be specified by asking a question about the scene: Where is the butter? Is this breakfast, lunch, dinner, or dessert? Is this an informal or fancy meal? How far has the eating progressed? Is this table messy? We are particularly interested in more qualitative tasks.

3 Framework for Solving Simple Tasks

3.1 Bayes Nets

A Bayes net is a way of representing the joint distribution of a set of variables in a way that is especially useful for knowledge representation (see [9] for details). For example, Figure 2 shows a highly simplified Bayes net that describes a place setting for a meal. Nodes in the net represent variables. Here, nodes drawn with solid lines denote parts of a place setting. The variable *setting* has four possible

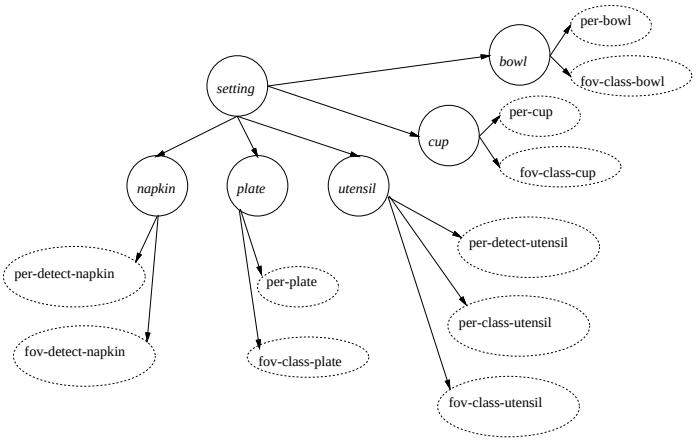

Figure 2: An example of a Bayes net that describes a place setting. Action nodes are drawn using dotted lines.

values ($breakfast, lunch, dinner, dessert$) that denote the respective types of meals. A *plate* can be either *paper* or *ceramic*. Links in the net represent conditional probabilities, for example the link from *setting* to *napkin* represents $P(napkin \mid setting)$, which says whether a napkin is expected at each of the possible meals.

The Bayes net formalism also includes a form of inference. Formally, belief in the values for node X is defined as $BEL(x) = P(x \mid \mathbf{e})$, where $\mathbf{e}$ is the combination of all evidence present in the net. Elegant solutions have been developed [4, 9] for incorporating a single piece of evidence into the net and for propagating its effect to all other nodes in the net.

3.2 Adding Actions to a Bayes Net

A node X in our Bayes net has *action nodes* (drawn with dotted lines) connected to it, each representing a variable that is a visual action's "evidence report", which contains a score for each possible value of the Bayes net node X. An action's evidence report affects the BEL values of the parent node X. Before the action is executed, the action node is a "chance" node like most of the nodes in the net, and the node contains the BEL of the evidence report. After an action is (successfully) executed the action node is changed to be an "instantiated" node (see [9]) and set to the value of the evidence report. For example, in Figure 2 the `per-class-utensil` action might generate the following evidence report $(4.9, 1.4, 3.2)$, which contains the scores for each of the objects, $(fork, knife, spoon)$. Action nodes are needed for computing the utility of actions given the current evidence present in the net.

All actions in the system are constructed from one or more low-level *vision modules*. Examples of some low-level vision modules are: color histogram matching and location-finding, (edge magnitude image) template matching, and a Hough transform for circles.

One or more vision modules may be used in a *visual action*. The Bayes net in Figure 2, used in one of our experiments with TEA-0, contains 11 visual actions. As an example, following is a more detailed summary of the actions related to plates:

- `per-plate`. Use Hough transform for plate-sized circles to detect plate in peripheral image. Detection succeeds or fails. The location is saved. Use color histogram to classify plate as paper (blue) or ceramic (green), using a window centered on the plate in the peripheral image.
- `fov-class-plate`. Plate location detected previously. Move fovea to plate. Use color histogram to classify plate as paper (blue) or ceramic (green), using fovea image data.

Complex actions like `per-plate` above are used in TEA-0, but are decomposed into simpler actions in TEA-1 (*e.g.* detect X, move peripheral window, classify X using peripheral window data, move fovea, classify X using foveal data, move camera). Some actions have *preconditions* that must be satisfied before they can be executed. TEA-0 uses only one kind of precondition: know the location of object X.

3.3 Adding Relations to a Bayes Net

Any node in a Bayes net that is bound to an object found in the scene will have the *location* of that object stored at that node. Otherwise, each node X has an *expected area* for the expected object associated with that node. An expected area is determined by applying geometric relations with each node Y_i connected to node X. A geometric relation between X and Y_i uses the location in node Y_i if it is available, otherwise the expected area at node Y_i is used. The expected area at node X is calculated strictly from relations; The location at node X is not used in the calculation of the expected area.

In TEA, an expected area is represented as a scalable bitmap denoting a subset of the planar scene area whose size depends on the object it is being related to. The resolution of the grid is currently the same as that of the peripheral images. TEA-0 allows relations between siblings, but in our latest work TEA-1 will have relations between parent and child nodes.

Each node Y_i produces an expected area for the object at node X. All these expected areas must be combined to obtain the final, single, expected area for the object at node X. In general it will be useful

to characterize the relation depicted by the maps as "must-be", "must-not-be" and "could-be". Combination of two "must-be" maps would then be by intersection, and in general map combination would proceed by the obvious set-theoretic operations corresponding to the inclusive or exclusive semantics of the relation. In TEA-0 the relations are "could-be", and the maps are unioned.

3.4 Calculating an Action's Utility

Let the action node α have the node A as its parent, then the utility $U(\alpha)$ of an action α is of the form

$$U(\alpha) = \frac{V(\alpha)}{C(\alpha)}.$$

$C(\alpha)$ is the cost of executing the action:

$$C(\alpha) = r_A C_0(\alpha).$$

$C_0(\alpha)$ is the execution time of action α on the entire peripheral or foveal image, and r_A is the percentage of the image covered by the expected area of the object associated with node A. For foveal images, $r_A = 1.0$ since the entire fovea is always processed. Before any actions have been executed, no objects have been located, and so all r_A values are 1.0. Over time, as other objects in the scene are located and as more and tighter relations are established, the value of r_A will approach zero.

$V(\alpha)$ is meant to be the value of the action, how useful it is for achieving the task's goal. All actions in a computer vision system are *information gathering* actions. Therefore, the value of an action is strictly a measure of the information the action provides:

$$V(\alpha) = I(target, \alpha),$$

where the task's goal is represented by the node *target*. I is Shannon's measure of average mutual information (see, *e.g.* [9]):

$$I(X,Y) = \sum_x \sum_y BEL(x,y) log \frac{BEL(x,y)}{BEL(x)BEL(y)}$$

where

$$BEL(x,y) = BEL(x \mid y)BEL(y).$$

The values of $BEL(x)$ and $BEL(y)$ are respectively available at nodes X and Y in the Bayes net. The values of $BEL(x \mid y)$ can be calculated by temporarily instantiating node Y to each of its values, propagating beliefs, and taking the resulting $BEL(x)$ as $BEL(x \mid y)$ [9].

It is important to "look ahead" at the future impact of executing an action. Therefore we use the following "lookahead" utility function. Recall that action node α has the node A as its parent.

$$U(\alpha) = \frac{V(\alpha) + V(\beta)}{C(\alpha) + C(\beta)} + \sum_{X \in Rel(A)} \Delta U(X) \qquad (1)$$

where

$$\beta = argmax_{\gamma \in LocPre(A)} \frac{V(\gamma)}{C(\gamma)}$$

The first term in equation (1) accounts for the future value of establishing the location of an object. Action α might detect and locate an object, but not provide any information ($I = 0$) toward the task node in the Bayes net, however it does locate the object and thereby satisfy the preconditions of other actions that in turn will provide information useful for accomplishing the task. The interpretation of the first term in equation (1) is: Let β be the "best" action with a precondition that is satisfied by executing action α. The new utility of action α is an average over both α and β, more specifically an average of the value and cost of the two actions α and β. $LocPre(A)$ is the set of actions with the precondition of knowing the location of the object associated with node A.

The second term in equation (1) anticipates the impact of the expected areas that action α will generate (by establishing relations with other objects). $Rel(A)$ is the set of all nodes that are directly helped by location information about the object associated with node A. In other words, nodes A and X are siblings or a parent-child pair and they have a relation map defined between them. Each node in $Rel(A)$ contributes a term $\Delta U(X)$ to the utility:

$$\Delta U(X) = \max_{\gamma \in Actions(X)} (U(\gamma) * (1/r - 1)).$$

r is the percent reduction in the expected area for node X's object assuming that the location of node A's object is known, and is computed by applying the relation map, but it is applied using the expected rather than the known size of node A's object.

3.5 Experimental Example

The TEA-0 system is an initial implementation that follows the technical framework outlined above. TEA-0 works in a simplified domain (a single place setting) and solves the following task: decide which meal the place setting is for *breakfast*, *lunch*, *dinner* or *dessert*. The task is further simplified by assuming that the scene could contain only a napkin, plate, cup, bowl, and a single utensil. The entire scene can

be viewed in one image so the system does not use camera movements. A relationship between the possible objects and the type of meal was contrived and encoded as the Bayes net model shown in Figure 2. The goal is to obtain high values for $BEL(setting)$. The scene was an overhead view of a single place setting, like one of those in Figure 1. The values of $BEL(setting)$ before any actions have been executed are:

$BEL(setting)$	
0.100	$bfast$
0.300	$lunch$
0.500	$dinner$
0.100	$dessert$

. The initial list of executable actions is:

$U(\alpha)$	
0.376946	`per-plate`
0.156381	`per-detect-napkin`
0.020900	`per-detect-utensil`
0.011891	`per-cup`
0.000509	`per-bowl`

. The system ends up executing the following sequence of actions: `per-plate`, `per-detect-napkin`, `per-detect-utensil`, `per-class-utensil`, `per-cup`, `fov-class-utensil`, `per-bowl`, after which the task belief values correctly indicate that the place setting is most likely set for a dinner meal:

$BEL(setting)$	
0.004	$bfast$
0.066	$lunch$
0.867	$dinner$
0.063	$dessert$

. The remaining actions have little effect on the task belief.

4 Framework for Solving More Complex Tasks

More complex questions will require a more complex Bayes net structure than used by TEA-0. The organization we are pursuing for the TEA-1 system is shown in Figure 3. It consists of three separate tree structures: a `PART-OF` tree, `IS-A` trees, and a *task* tree.

The `PART-OF` tree, an example of one is shown in Figure 4, models the physical structure of the scene. We assume the scene and all the objects in the scene can be modeled as a hierarchy of parts. All nodes in this net have the same set of possible values: *present* and *notPresent*. The conditional probability on each network link indicates the likelihood that a subpart exists. Each object's location and expected area are stored within the object's node in the `PART-OF` network.

An `IS-A` tree (for example, Figure 5) models an abstraction hierarchy for each instance of an object in the scene. The `IS-A` net is a special kind of network

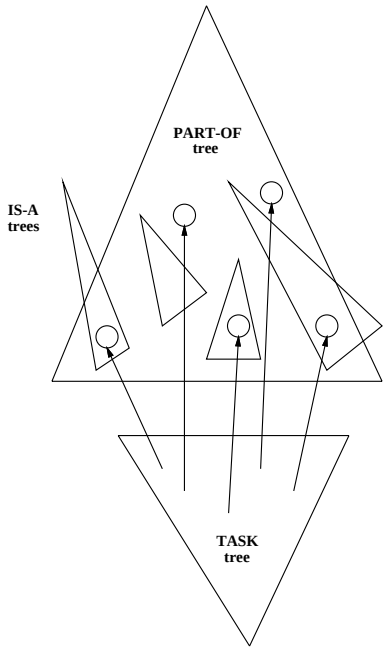

Figure 3: The organization of a large Bayes net used by TEA.

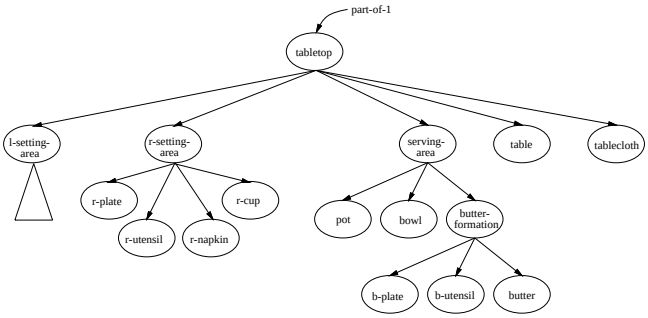

Figure 4: A PART-OF Bayes net.

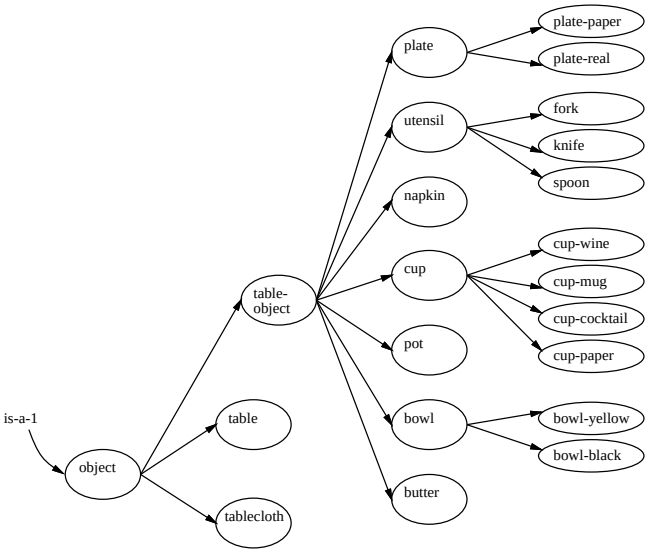

Figure 5: An IS-A Bayes net.

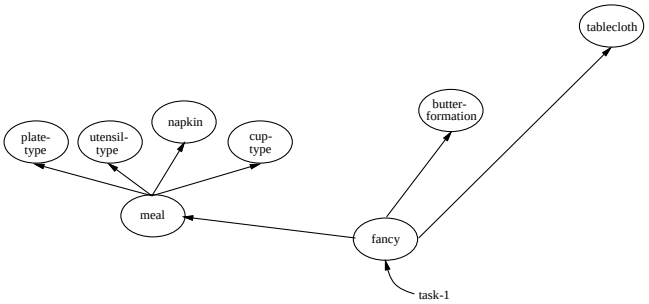

Figure 6: A *task* Bayes net.

because the leaf nodes are mutually exclusive (*i.e.* the object can only be one thing). Belief nets with this special property have been developed [4, 9].

One of our scientific goals is to make a tight formal and practical coupling between "task specific knowledge" and visual actions. Task specific knowledge is contained in the *task* net (for example, Figure 6), and is thus distinguished from other types of knowledge. One feature of task knowledge is that subtask nodes could be shared by several tasks. Questions such as "Is this a fancy meal?" may be answered using a range of image clues. Some simple tasks, such as "Where is the butter?", do not require a task tree since they only involve one particular node in a tree.

We want to use the task tree to add task-specificity to the utility function: basically the idea is to relativize the utility calculation to the task by "projecting" the knowledge in the PART-OF and IS-A trees onto the task tree and computing utilities there. We are in the early stages of developing the notation, semantics, and implementation of the interacting trees. We wish to develop general formulae for these utilities, and to study the benefits gained by more complex utility functions, as opposed to simpler utility functions and more complete planning.

5 Conclusions and Future Work

We are pursuing two main streams of work. One stream develops the TEA systems, a progression of systems that support increasingly sophisticated task-oriented vision by providing solutions to the "where to look next" problem. The second stream of work uses and extends the TEA framework to explore broader and more advanced issues in task-oriented vision: foveal - peripheral vision algorithms, qualitative visual tasks, limited-context vision algorithms that gain in robustness or accuracy by being applied in well-understood circumstances, incremental visual actions whose results monotonically improve as more time is spent on them, representations of 3-D and dynamic spatial relations, head-shifting and viewpoint planning, and processor scheduling.

5.1 "Where to Look Next"

TEA-0: Using relations. An initial version of TEA-0 has been implemented and it should give the basic idea of our approach. TEA-0 will be completed by implementing the full version of relations, and by enabling camera movements.

TEA-1: Projecting utilities through a task tree. The main feature of TEA-1 is the addition of multiple interacting trees, which permit the system to solve more complex tasks. A preliminary design for projecting utility calculations through the task tree is complete and we have begun implementing it. A deeper issue is that relational information should be used to modify probabilities and beliefs, not just costs. Relational evidence must then be formulated in a probabilistic framework, and expected areas used like beliefs.

TEA-2: Planning. TEA-0 and TEA-1 are "myopic", making decisions by only looking one step ahead. The anticipatory utility function is an improvement, trying to pack look-ahead into the utility of a single action. Ultimately our problem involves full-scale planning, in which sequences of actions are evaluated as to their expected utility. We intend to develop a simple planning system (for computer vision) using Bayes nets. We do not propose "planning research" *per se*, but rather shall likely use some STRIPS-like planning algorithm. The idea is to substitute a search in action space rather than to try to pack all the intelligence into a (quasi-static) utility function.

5.2 "How to Look"

Limited-Context Vision Algorithms. One claim of this work is that vision algorithms can be more robust and reliable if they are known to work in a limited context. For example, TEA-0 can use simple color histograms for object identification only because it has foveated a small area of the image previously. Similarly restricting input to a small volume of space means geometric hashing can work more reliably. We want to explore limited context effects that arise naturally in task-oriented vision when the vision problem is known to be simplified or better specified than normal (by camera actions, foveal processing, and generally by satisfaction of preconditions).

Incremental actions. We want to investigate vision modules that can run for different periods of time, improving their results the longer they run (*e.g.* some scale space algorithms, multi-feature classifiers). Such vision actions are generalizations of TEA-0's peripheral - foveal actions which produce a peripheral result at one cost and follow it up with a foveal action for a further cost. An evidence/time function can quantify the incremental benefit of such an action. New control strategies should then emerge, such as running a set of incremental actions cyclically to attain the maximum evidence per unit time from the set.

5.3 Task-Oriented Vision

Our idea of a true task-oriented vision system will be achieved by bringing together solutions to the "where to look next" and the "how to look" problems.

Multiple Tasks. We plan to solve multiple tasks in any given domain using the same set of visual actions. This exercise will test the generality of our knowledge representations and visual actions and probably encourage us to extend and modify both. Also we expect to encounter interesting new problems for visual actions used in answering qualitative questions such as "Is this desk messy?".

Multiple Domains. We believe that a task-oriented vision system should be verified using more than one domain. We shall seek out other domains. A possible domain is model trains to be monitored on a more or less complex system of tracks. Another is monitoring or searching the laboratory space in 3-D and performing head movements as well as camera movements, and ultimately dynamic scenes. Medical images are another possibility emphasizing reliability as opposed to active vision. Expanding the domains will doubtless mean that visual actions need to be re-engineered and improved to apply more generally. Difficulties in encoding or coping with new domains will motivate extensions and modifications to our formalisms. New domains may necessitate the use of more complex knowledge representations, in particular non-tree Bayes nets.

Acknowledgements

Chris Brown, my advisor, has been invaluable to me throughout this work by providing crystal judgement and ideas and asking key questions. Peter von Kaenel worked on the Hough transform for circles and is currently building the modules that match models using straight lines and circular arcs. He also built several of the visual actions.

References

[1] A. M. Agogino and K. Ramamurthi. Real time influence diagrams for monitoring and controlling mechanical systems. In R. M. Oliver and J. Q. Smith, editors, *Influence Diagrams, Belief Nets and Decision Analysis*, pages 199–228. John Wiley and Sons, 1990.

[2] J. M. Agosta. The structure of Bayes networks for visual recognition. In *Uncertainty in AI 4*, pages 397–405. North-Holland, 1990.

[3] R. C. Bolles. Verification vision for programmable assembly. In *Proceedings: International Joint Conference on Artificial Intelligence*, pages 569–575, 1977.

[4] P. B. Chou and C. M. Brown. The theory and practice of Bayesian image labeling. *International Journal of Computer Vision*, 4(3):185–210, 1990.

[5] T. Dean, T. Camus, and J. Kirman. Sequential decision making for active perception. In *Proceedings: DARPA Image Understanding Workshop*, pages 889–894, 1990.

[6] J. Feldman and R. Sproull. Decision theory and artificial intelligence II: The hungry monkey. *Cognitive Science*, 1:158–192, 1977.

[7] T. Garvey. Perceptual strategies for purposive vision. Technical Report 117, SRI AI Center, 1976.

[8] T. Levitt, T. Binford, G. Ettinger, and P. Gelband. Probability-based control for computer vision. In *Proceedings: DARPA Image Understanding Workshop*, pages 355–369, 1989.

[9] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman, 1988.

[10] R. D. Rimey and C. M. Brown. Task-oriented vision with multiple Bayes nets. Technical Report 398, Department of Computer Science, University of Rochester, November 1991.

[11] R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.