Where to Look Next Using a Bayes Net: The TEA-1 System and Future Directions *

Raymond D. Rimey
The University of Rochester
Computer Science Department
Rochester, New York 14627
rimey@cs.rochester.edu

1 Introduction

A task-oriented system is one that performs the minimum effort necessary to solve a specified task. Depending on the task, the system decides which information to gather, which operators to use at which resolution, and where to apply them. We have been developing 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 where to point a camera (or fovea) and what vision modules to run is made using a value/cost utility measure, where value is based on average mutual information measured between nodes in the Bayes net that correspond to expected action results and to the goal of the task. This paper summarizes our latest implementation, called TEA-1, and outlines some directions for future work.

2 TEA-1: A Framework for Studying Task-Oriented Vision

This section summarizes the TEA-1 system, our second implementation of TEA, a general framework of a task-oriented computer vision system. A detailed description of TEA-1 and the *expected area* net are located in [17] and [18] respectively. Earlier work, mainly on TEA-0, appears in [15, 16].

A Pointable Spatially-Varying sensor. A key component in an active vision system is a spatially-varying sensor that can be pointed in space (using a pan-tilt platform) to selectively view a scene. We assume a sensor that provides a peripheral image that is a low-resolution image of the entire field of view from one camera angle, and a fovea that is a small high-resolution image that can be selectively moved within the field of view. 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. Spatially-varying sensors can be constructed in many ways: using special sensor array chips, *e.g.* [3, 19], two cameras with different focal lengths, hardware resolution pyramids [5], or programmed in software [4].

Domain and Example Tasks. Our example domain is table settings. Each task can be specified by asking a question about the scene. We are particularly interested in more qualitative tasks and those that require a variety of scene information to solve: Is this breakfast, lunch, dinner, or dessert? Is this an informal or fancy meal? How far has the eating progressed?

Main Control Loop. The TEA system gathers evidence visually and incorporates it into a Bayes net until the task, specified as a question, can be answered to a desired degree of confidence. TEA runs by iteratively selecting the evidence gathering action that maximizes an expected utility criterion involving the cost of the action and its benefits of increased certainties in the net: 1) List all the executable actions. 2) Select the action with highest expected utility. 3) Execute that action. 4) Add the resulting evidence to the Bayes net and propagate its influence. 5) Repeat, until the task is solved.

Bayes Nets. Nodes in a Bayes net represent random variables with (usually) a discrete set of values (*e.g.* a *utensil* node could have values *(knife, fork, spoon)*). Links in the net represent (via tables) conditional probabilities that a node has a particular value given that an adjacent node has a particular value. 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. Evidence, produced by running a visual action, directly supports the pos-

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

sible values of a particular node (*i.e.* variable) in the net. There exist a number of evidence propagation algorithms, which recompute belief values for all nodes given one new piece of evidence. Several references provide good introductions to the Bayes net model and associated algorithms, *e.g.* [6, 10, 11, 14, 20].

Composite Bayes Net. TEA-1 uses a composite net, a method for structuring knowledge into several separate Bayes nets [17]. A `PART-OF` net models subpart relationships between objects and whether an object is present in the scene or not. An *expected area* net models geometric relations between objects and the location of each object (see next paragraph). Associated with each object is an `IS-A` tree, a taxonomic hierarchy modeling one random variable that has many mutually exclusive values [7, 14]. Task specific knowledge is contained in a *task* net. There is one *task* net for each task, for example "Is this a fancy meal?", that TEA-1 can solve. Each of the separate nets in the composite net, except the *task* net, maintains its BEL values independently of the other nets. Evidence in the other nets affects the *task* net through a mechanism called packages, which updates values in evidence nodes in the *task* net using copies of belief values in the other nets.

Expected Area Net. An *expected area* net models geometric relations between objects and the location of each object [18]. Each node in the *expected area* net corresponds with a node in the `PART-OF` net and identifies the area in the scene in which that object is expected to be located. The location of an object in the scene is specified by the two camera angles, $\theta = (\phi_{pan}, \phi_{tilt})$, that would cause the object to be centered in the visual field. The height and width of an object's image is also specified using camera angles. Thus a node in the *expected area* net represents a 2-D discrete random variable, θ. $BEL(\theta)$ is a function on a discrete 2-D grid, with a high value corresponding to a scene location at which the object is expected with high probability. A link from node A to node B has an associated conditional probability, $P(\theta_B \mid \theta_A)$. Given a reasonable discretization, say as a 32x32 grid, each conditional probability table has just over a million entries. Such tables are unreasonable to specify and cause the calculation of new belief values to be very slow. We have developed a way to limit these problems [18]. The values of the conditional probabilities are computed using a special simplified distribution called a relation map. A relation map assumes that object A has unity dimensions and is located at the origin, and is scaled and shifted appropriately to obtain values of the conditional probability. When the set of expected locations for an object covers a relatively small area of the entire scene, the table of $P(\theta_B \mid \theta_A)$ values contains a large number of essentially zero values that can be used to speed up the belief propagation computation.

Actions. TEA-1 uses the following description of an action:

- *Precondition.* The precondition must be satisfied before the action can be executed. There are four types of precondition: that a particular node in the *expected area* net be instantiated, that it not be instantiated, that it be instantiated and within the field of view for the current camera position, and the empty precondition.

- *Function.* A function is called to execute the action. All actions are constructed from one or more low-level vision modules, process either foveal image or peripheral image data, and may first move the camera or fovea.

- *Adding evidence.* An action may add evidence to several nets and may do so in several ways (see [14]): 1) A chance node can be changed to a dummy node, representing virtual or judgemental evidence bearing on its parent node. 2) A chance node can be instantiated to a specific value. Object locations get instantiated in the *expected area* net. 3) Evidence weight can be added to an `IS-A` type of net.

Actions in TEA-1 that must move the camera to the expected location of a specific (expected) object, say X, will move the camera to the center of mass of the expected area for object X. Every action related to object X creates a mask that corresponds to the portion of the current image data (after a camera movement) that is covered by the expected area of object X (when thresholded to a given confidence level), and then processes only the image data covered by that mask. Each kind of object usually has several actions associated with it. TEA-1 currently has 20 actions related to 7 objects. For example, the actions related to plates are: The `per-detect-template-plate` action moves the camera to a specified position and uses a model grayscale template to detect the presence and location of a plate in the peripheral image. `Per-detect-hough-plate` uses a Hough transform for plate-sized circles for the same purpose. `Per-classify-plate` moves the camera to a specified position, centers a window in the peripheral image there, and uses a color histogram to classify that area as paper or ceramic. `Fov-classify-plate` moves the

fovea (but not the camera) to a specified location and uses a color histogram to classify the area as paper or ceramic.

Calculating an Action's Utility. TEA-1's utility function for an action has the following features: 1) An action's value is determined relative to the needs of the current task. 2) An action's cost is proportional to the amount of image data processed. 3) The utility accounts for the future value of establishing the location of an object, for example by a peripheral object-detection action. 4) It also accounts for the impact of making expected areas smaller so that future actions will have lower costs.

The utility $U(\alpha)$ of an action α is fundamentally modeled as $U(\alpha) = V(\alpha)/C(\alpha)$, a ratio of value $V(\alpha)$ and cost $C(\alpha)$. The value of an action, how useful it is for the task, is based on Shannon's measure of average mutual information, $V(\alpha) = I(T, e_\alpha)$, where T is the variable representing the goal of the task and e_α is the combination of all the evidence added to the composite net by action α. An action α related to a specific object X has a cost proportional to the amount of image data that it processes, estimated using the current expected area for object X and the average execution time per pixel for the action. See [17] for details and the specific utility function used in TEA-1

An important feature of the TEA-1 design is that a different *task* net is plugged into the composite net for each task the system is able to solve. The calculation of an action's value depends on the *task* net. Thus the action utilities directly reflect the information needs of the specific task, and produce a pattern of camera and fovea movements and visual operations that is unique to the task.

Experimental Results. The task of deciding whether a dinner table is set for a fancy meal or for an informal meal was encoded in a (very simple) *task* net, and TEA-1 was presented the scene shown in Figure 1, which shows a "fancy" meal. The sequence of actions executed by TEA-1 is summarized by the table in Figure 1. The *a priori* belief of the table setting being fancy is 0.590, compared with 0.410 that it is informal. As the system executed actions to gather specific information about the scene, the belief that the setting is a fancy one approaches 0.974. The graphics in the bottom of the figure illustrate the sequence of camera movements executed by the system. Figure 2 illustrates the execution of a few actions in the sequence, showing each action's results after any camera (or fovea) movement has been made and the expected area mask has been applied.

time	$U(\alpha)$	α, an action	$BEL(i)$
0		*a priori*	0.410
1	10.0	`table`	0.400
2	10.5	`per-detect-hough-cup`	0.263
3	42.8	`per-classify-cup`	0.343
4	11.3	`per-detect-hough-plate`	0.340
5	11.9	`per-classify-plate`	0.041
6	20.9	`per-detect-utensil`	0.041
7	58.8	`per-classify-utensil`	0.033
8	4.3	`per-detect-napkin`	0.026
9	3.3	`fov-classify-cup`	0.026
10	2.4	`fov-classify-plate`	0.026
11	1.7	`per-detect-hough-bowl`	0.026
12	0.6	`per-detect-butter`	0.026
13	0.4	`fov-verify-butter`	0.026

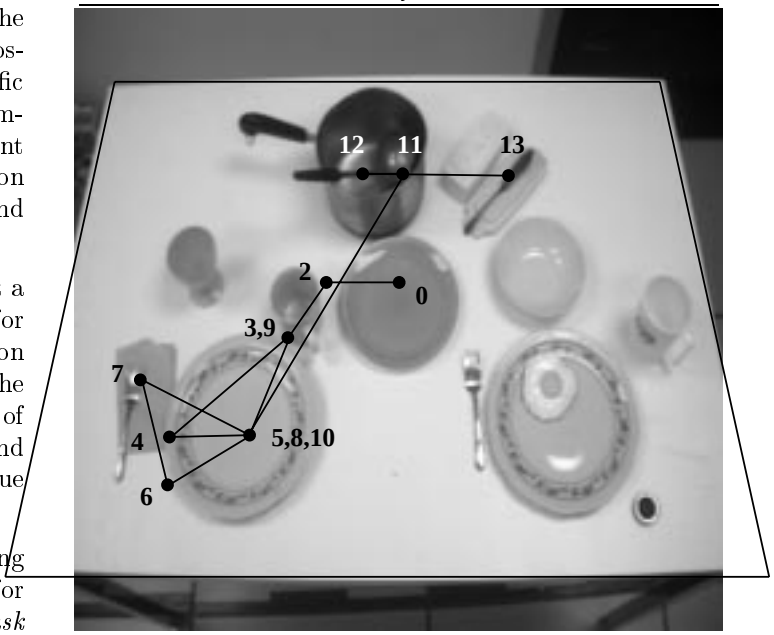

Figure 1: The sequence of actions selected and executed by TEA-1 is shown in the table at the top. Each line corresponds to one cycle in the main control loop. The belief values listed are those after incorporating the results from each action. The $BEL(i)$ column shows $BEL(informal)$, and $BEL(formal) = 1 - BEL(i)$. The path drawn on the wide-angle picture of the table scene at the bottom illustrates the camera movements made in the action sequence.

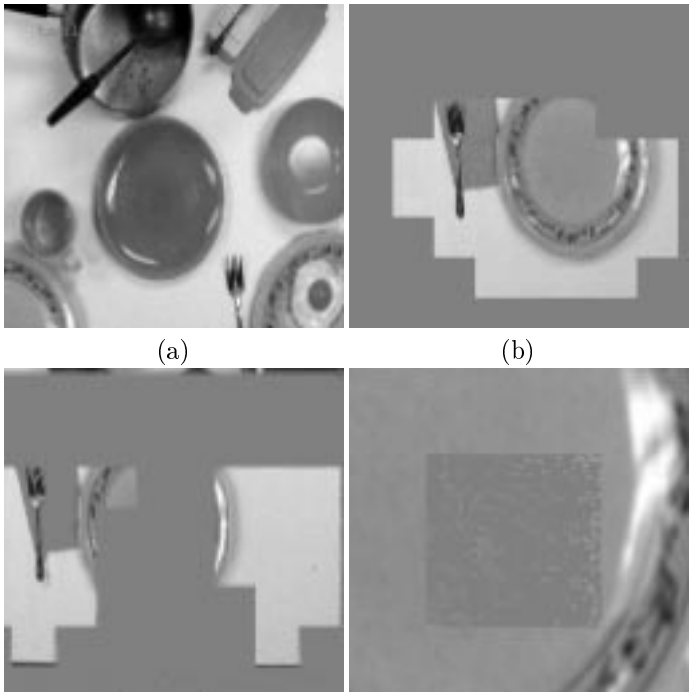

Figure 2: Processing performed by individual actions. Image pixels outside the expected area mask are shown as gray values. (a) The camera's initial viewpoint. (b) Results from the per-detect-hough-plate action executed at time step 4. (c) Results from the per-detect-napkin action executed at time step 8. The mask prevents the red napkin from being confused with the pink creamer container just above the plate. (d) Results from the fov-classify-plate action executed at time step 10. A zoomed display of the fovea centered on the plate is shown.

3 Future Directions

Several people are investigating the use of Bayes nets and influence diagrams in sensing problems. Levitt's group was the first to apply Bayes nets to computer vision [2, 12, 13]. Dean's group is studying applications in sensor-based mobile robot control, using a special kind of influence diagram called a temporal belief network (TBN) [8, 10]. More recently, they have used sensor data to maintain an occupancy grid, which in turn affects link probabilities in the TBN [9]. A sensor and control problem involving a real milling machine is solved using influence diagram techniques in [1].

The current TEA-1 system design, incorporating *expected area* nets, provides a framework that enables a computer vision system to make decisions about moving a camera and fovea around and about selectively gathering information.

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. 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, which we call the "how to look" problem: foveal - peripheral vision algorithms, qualitative visual tasks, limited-context vision algorithms that gain in robustness or accuracy by being applied in well-understood circumstances, and incremental visual actions whose results monotonically improve as more time is spent on them.

3.1 Where to Look Next

Deciding Between Fovea and Camera Movements. Deciding where to move a camera (or fovea) is an interesting problem. TEA-1 does the simplest thing possible by moving to the center of the expected area of one object. If several objects of interest should fall in the field of view, then it may for example be better to move the camera to the center of that set of objects. In our experiments to date, TEA-1 has relied mainly on camera movements to get the first piece of information about an object, while fovea movements are mostly used for verification. This behavior is determined by the costs and other parameters associated with actions. Another interesting problem is to consider the tradeoffs between a camera and a fovea movement. A camera movement is expensive and an action following one processes a completely new area

of the scene, which means there is risk of not finding anything, but if something is found it will likely have large impact for the task. Alternatively, a fovea movement is cheap but produces image data near an area already analyzed, so there is a good chance of finding some new information, but it will tend to have a small impact on the task.

TEA-2: Planning. TEA-0 and TEA-1 are "myopic", making decisions by only looking one step ahead. They try to pack a look-ahead capability into the utility function 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 TEA-2, a simple planning system for computer vision that uses Bayes nets. 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.

3.2 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 applied in a limited context. For example, TEA-1 can use simple color histograms for object identification only because camera movements and expected area masks limit the processing to a small area. We want to explore limited context effects that arise naturally in task-oriented vision when the vision problem is known to be simplified (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, and anytime algorithms [10]). Such actions are generalizations of TEA'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.

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. By experimenting with TEA and analyzing how it gathers evidence for a variety of different tasks we hope to learn something about the information requirements of tasks. Also we expect to encounter interesting new problems for the visual actions and knowledge representation needed in answering qualitative questions such as "Is this table messy?".

Multiple Domains. We believe that a task-oriented vision system should be verified using more than one domain. We have begun work involving a new domain, model trains, which will probably focus on dealing with real-time constraints. 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.

Acknowledgments

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 implemented many of the actions and low-level vision modules, and is currently working with the model train domain, as an undergraduate project.

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. Bajcsy. An active observer. In *Proceedings: DARPA Image Understanding Workshop*, pages 137–147, 1992.

[4] R. M. Bolle, A. Califano, and R. Kjeldsen. Data and model driven foveation. In *Proceedings: IEEE International Conference on Pattern Recognition*, pages 1–7, 1990.

[5] P. J. Burt. Smart sensing within a pyramid vision machine. *IEEE Proceedings*, 76(8):1006–1015, 1988.

[6] E. Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, 1991.

[7] 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.

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

[9] T. Dean and J. Kirman. Representation issues in Bayesian decision theory for planning and active perception. In *Proceedings: DARPA Image Understanding Workshop*, pages 763–768, 1992.

[10] T. L. Dean and M. P. Wellman. *Planning and Control*. Morgan Kaufmann, 1991.

[11] M. Henrion, J. S. Breese, and E. J. Horvitz. Decision analysis and expert systems. *AI Magazine*, 12(4):64–91, 1991.

[12] 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.

[13] W. B. Mann and T. O. Binford. An example of 3D interpretation of images using Bayesian networks. In *Proceedings: DARPA Image Understanding Workshop*, pages 793–801, 1992.

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

[15] R. D. Rimey. Where to look next using a Bayes net: An overview. In *Proceedings: DARPA Image Understanding Workshop*, pages 927–932, 1992.

[16] 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.

[17] R. D. Rimey and C. M. Brown. Task-oriented vision with multiple Bayes nets. In A. Blake and A. Yuille, editors, *Active Vision*, pages 217–236. MIT Press, 1992.

[18] R. D. Rimey and C. M. Brown. Where to look next using a Bayes net: Incorporating geometric relations. In *Proceedings: European Conference on Computer Vision*, pages 542–550, 1992.

[19] A. S. Rojer and E. L. Schwartz. Design considerations for a space-variant visual sensor with complex-logarithmic geometry. In *Proceedings: IEEE International Conference on Pattern Recognition*, 1990.

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