Studying Control of Selective Perception Using T-World and TEA

Raymond D. Rimey*

The University of Rochester
Computer Science Department
Rochester, New York 14627-0226
rimey@cs.rochester.edu

Abstract

We hypothesize that selective perception allows more accurate solutions to visual tasks to be found in less wall-clock time than non-selective techniques. The best way to assess the practical truth of this hypothesis is by studying, designing and building complete vision systems — the issues are fundamentally systems issues. On the other hand, special-case systems are not convincing: we present the T-world problem *as an abstraction of an interesting class of real-world vision problems. T-world has enough structure to support basic study of fundamental tradeoffs inherent in selective computer perception. Our complete system is called TEA-1: it is a purposive and sufficing vision system that solves a version of the T-world problem. TEA-1 is a fully implemented system, and extensive experiments in the laboratory and simulation have explored the key factors that make the selective perception approach appealing, analyzing how each factor affects the overall performance of TEA-1 when solving a set of automatically generated (in simulation) T-world domains and tasks.*

1 Selective Perception

1.1 General Concepts

Purposive vision. A purposive vision system works to achieve a goal (*i.e.* solve a visual task) in minimal wall-clock time. Goal-directed operation can make the system fast by limiting the amount of data processed and by limiting the extent to which that data is processed.

Sufficing vision. Sufficing vision is the use of (usually simple, cheap, and general) vision modules whose output is ambiguous unless considered in a known context.

*This material is based on work supported by DARPA Contract MDA972-92-J-1012. The Government has certain rights in this material.

It is essentially impossible to design a vision module that performs well in all possible contexts. However, it is quite possible that through partial visual analysis or prior knowledge, the vision system has some information about the situation in the scene and the contexts in which objects appear or are expected to appear. The idea of sufficing vision is that vision modules are designed for and only executed in such contexts. Two things are required in order for sufficing vision to work in a system. First, a system is needed that establishes contexts and uses them to specify exactly what vision modules to run. Second, a large repertoire of sufficing vision modules is needed.

Historically, vision modules have been engineered to produce relatively high-level outputs, ones humans can reason about. Sufficing vision systems may be less transparent: A human may find it difficult to understand or specify exactly what a vision module in a sufficing vision system is really doing and why, since the context may not be known to the human and the significance of the extracted information within that context may not be obvious. While in general it may be difficult to design and integrate such vision modules, there are two causes for optimism. First, learning techniques may be able to tune and select modules in the context of the whole system, even though humans can not easily specify the modules explicitly and *a priori*. Second, one aspect of the sufficing vision idea is that existing (relatively simple) vision modules may be more useful than they may seem, when intelligently applied and interpreted within specific contexts.

For example, when looking for the carrots at a dinner table it may be sufficient to look for a big blob of orange and then check the orange things are roughly elongated. In another situation it may be sufficient simply to look for a big blob of orange. In some contexts a cup can be detected simply by finding a circle with a radius in a specified range. Figure 1 shows the performance of such a cup detection action as the con-

text in which it is executed is increasingly narrowed.

Selective perception. A selective perception system is purposive and sufficing.

Control of selective perception. The general problem that we are interested in is to control (select actions and make decisions in) a computer vision system that has a repertoire of actions (sufficing vision modules) such that the system operates in a purposive manner.

Hard things to do. Solving the control of selective perception problem involves several things that are hard to do: A visual task must be represented in the system, and the system must use that representation to operate in a task-oriented manner. Generally the system must decide what to do to solve the task, which includes the following things. It must decide where in the scene to look. (We assume a pointable camera and foveal window to make this problem more explicit.) It must decide what information to look for. It must decide how to look for it. (Only imperfect information can be obtained, and all information gathering actions have execution costs. We assume a large repertoire of such information gathering actions.) The system must decide what the current best solution to the task is, and whether that solution is good enough or more information from the scene is needed.

1.2 Claims

General framework. Bayes nets and decision theoretic techniques provide a *general, reusable framework* for constructing computer vision systems that are purposive and sufficing.

Task representation. Bayes nets provide a general, reusable framework to *represent tasks*. Task-oriented behavior emerges from the combination of the representation and general decision making procedures.

Decision making. General decision making procedures can be based on "goodness" functions constructed around core decision theoretic elements. The ability to use imperfect information as evidence (which is what any realistic vision modules outputs) and probabilistic models for the cost and performance of a vision module are advantages of a decision theoretic approach and are basic capabilities needed in any purposive and sufficing vision system.

Framework for sufficing vision. The points above are particularly important because they enable the construction of a system based on sufficing vision.

T-world. The T-world problem (Section 2) captures the key problem characteristics that can be exploited by a selective perception system. T-world is an adequate problem for easily studying some of the

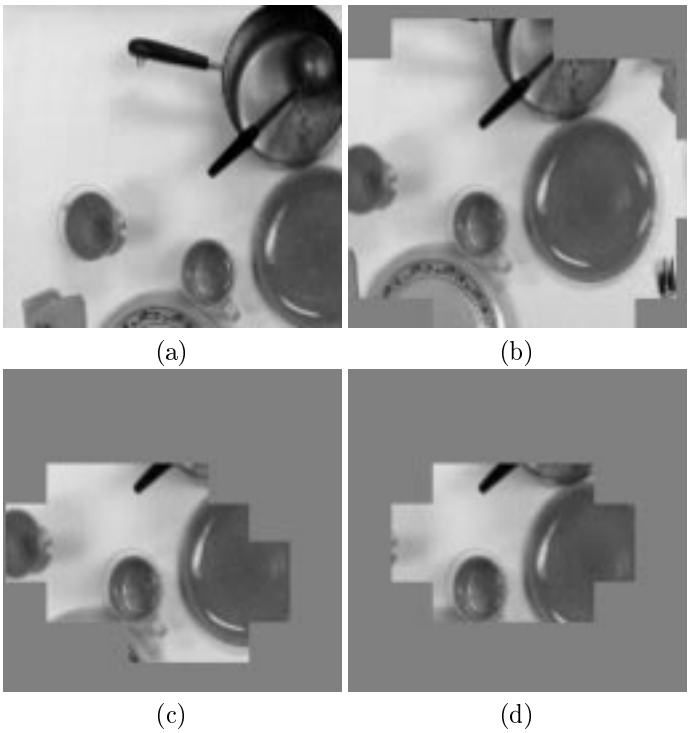


Figure 1: The area in an image where a cup is expected to be located changes over time as more objects are located via actions. The performance of a cup detection action (a Hough transform for a circle with a radius in a specified range) is affected by the changing area. (a) Before any objects have been located, the area is actually larger than the field of view, and the creamer container is mistakenly detected as the cup. (b) After the tabletop has been located, the area is only slightly larger than the field of view, and the cup is correctly detected but this is lucky since so many other objects are still in the unmasked area. (c) After the tabletop and plate have been located, the area is fairly small and the cup is correctly detected. (d) After the tabletop, plate and napkin have been located, the cup is again correctly detected.

basic issues in selective computer perception.

Selective vs. non-selective perception. A (high-level) computer vision system (with limited resources and working in complex environments) that is purposive and sufficing is better than one that is not, meaning that it solves tasks in less wall-clock time.

We believe that the best way to support many of our claims is by studying, designing and building complete vision systems — the issues are fundamentally systems issues.

2 The T-world Problem

This section defines the T-world problem, a formalization of some key problem characteristics that can be exploited by a selective perception system.

A scene. A scene consists of many objects within a large two-dimensional rectangular area. Each object has a location (for its centroid), rectangular dimensions, a type, and a set of properties. Each property has a set of possible values. There may be any number of objects in a scene. Objects may overlap each other, but this does not affect the performance of a visual action (see below). The objects in a scene may be organized into a set of mutually exclusive groups, and the groups may have subgroups, subsubgroups, *etc.* A group is a spatially local collection of objects. The size and number of groups and the subgroup structure are determined by the domain rules (see below).

The sensor. The sensor, called a camera, is a fixed-size rectangular window that is in the plane of the scene and is much smaller than the extent of the scene. The window may be moved (by a camera movement action, see below) to any location in the scene, specified by the coordinates for the center of the window in a two-dimensional world coordinate system. The window defines the camera's field of view. A fixed-size rectangular window, called the "fovea", is much smaller than the field of view's size, and may be moved around inside the field of view.

A low spatial-resolution image that covers the entire field of view is available and is called the "peripheral image". A high spatial-resolution image that covers the fovea is available and is called the "foveal image". (Alternatively, a resolution pyramid of images may be used.)

A domain. A "domain" consists of a set of scene types and a set of probabilistic rules for each scene type that specifies the number, type, location (and grouping structure) and properties of objects in a scene.

A task. A task is defined as determining the value of a task variable, which is a (probabilistic) function of a subset of the number, type, location and properties of objects in the scene.

Camera movement actions. Given a specified location in a scene (in a two-dimensional world coordinate system) a "camera movement action" moves the camera so it is centered on the specified location. This action always moves the camera exactly to that location. The time to execute a camera movement action is a function of the distance moved.

Visual actions. Visual actions try to obtain information about a portion of a scene visible inside the field of view (*i.e.* from image data). There is a large collection of visual actions designed to obtain many different types of information. We currently use two types of visual action: one tries to detect a specific type of object in an image, and the other tries to obtain the value of a specified property of a specified object in an image.

The behavior of an action depends on whether the target object is truly in the field of view or not, the true type, location and properties of the target object and of all the other objects in the field of view. An action may have a precondition that must be satisfied before the action can be executed.

The performance of an action is a function of several parameters, which must be specified for each action: the image resolution (currently either foveal or peripheral resolution, and generally a level in a resolution pyramid), the image area to process, and the length of time to process a unit of image data. The time to execute a visual action is a function of the same parameters. Note that several actions may have the same purpose, but different performance and cost characteristics.

The problem. Given a scene from an identified T-world domain and a specified T-world task, the problem is to sequentially collect evidence from the scene to support a decision about an answer to the task, with a desired level of confidence, so that the total wall-clock time for executing the actions is minimized. Solving the problem involves the following general steps: decide what action to execute next, execute that action, incorporate the results from that action, decide on the answer to the task, and decide whether to gather more evidence or to stop.

3 TEA-1: A Decision-Theoretic Solution for Control of Selective Perception in the T-world Problem

TEA-1 is an implemented, compact, flexible, selective computer vision system, which solves a version of the T-world problem and has a solid foundation

of well-established formalisms — Bayesian statistics and decision theory. TEA-1 uses Bayes nets for representation and a cost and benefit analysis extending over action sequences to decide which visual or non-visual action to perform next. We believe TEA-1's current design provides a general software tool sufficient to study a variety of basic issues in high-level and low-level selective analysis and behavior in computer perception.

A probabilistic knowledge representation is appropriate for a selective system, and Bayes net and Dempster-Shafer approaches are two obvious alternatives. We choose the Bayes net approach because it is flexible and easy to use, and works well for the variety of tasks and domains we have in mind. We developed a version of Bayes nets, called a composite Bayes net, which consists of domain-specific knowledge and a specification of the desired task. The composite net includes a new application of Bayes nets to represent relative object locations and geometric relations. A task is specified by a net that makes explicit the relation of evidence needed to accomplish a specific perceptual task to the components of the domain-dependent knowledge representation.

TEA-1's design assumes all the details in the T-world definition in Section 2. This includes a pointable camera, peripheral and foveal images, and a repertoire of visual actions. The visual actions are based on generic, easily-tuned, sufficing vision utilities (histograms, Hough transforms) from our software library. These sufficing algorithms are in general simple and fragile; in a known context they are simple and robust. TEA-1 can use any visual operator that can be characterized by TEA-1's action schema, which includes specification of preconditions and cost and performance models.

Decisions about what to do next are based on a goodness function constructed around the following core elements: the expected value of sample information for the result produced by actions, the expected cost of actions, and the probability that an object is in a particular subset of the image data. Detailed descriptions of decision making in the original TEA-1 design are available in [9, 10, 11, 12]. Considerable modifications were made for the final TEA-1 design described in [8].

TEA-1 programs can transparently run either with a T-world simulator providing input and accepting output or in the laboratory (for a dinner table domain). There are actually two simulator programs: One program simulates an instance of T-world (scene, camera, actions, *etc.*) as specified by a database of rules and models. The other program automatically generates the database files that specify new instances of T-world domains, and scenes and tasks for each domain. The same program automatically generates the knowledge representation structures used by the TEA-1 system.

3.1 An Example Run

Here we illustrate how the TEA-1 system basically works by presenting an example run of the system in the laboratory using real image data and a collection of simple visual actions. The domain is dinner table settings. The task is to decide whether a table is set for a fancy meal or for an informal (also called notfancy) meal, a decision that involves acquiring and assessing probabilistically the values of six scene characteristics (such as the type of cup, presence of napkin, and type of plate). The domain and task nets used are straightforward, see [9] for details. TEA-1 was presented the scene shown in Figure 2, which shows a fancy meal.

The sequence of actions executed by TEA-1 is summarized in Figure 3(a), and the values of the belief in the goal over time are plotted in Figure 3(b). So if the scene is actually fancy then the value of $BEL(notfancy)$ should progress from its *a priori* value to a lower value. Here, better performance means that $BEL(notfancy)$ gets lower faster. The *a priori* belief of the table setting being informal is 0.794, compared with 0.206 that it is fancy. As the system executes actions to gather specific information about the scene, the belief that the setting is an informal one approaches 0.108, correctly saying that the scene is a fancy one. Figure 4 illustrates the execution of a few actions in the sequence, showing each action's results after any camera (or fovea) movement has been made.

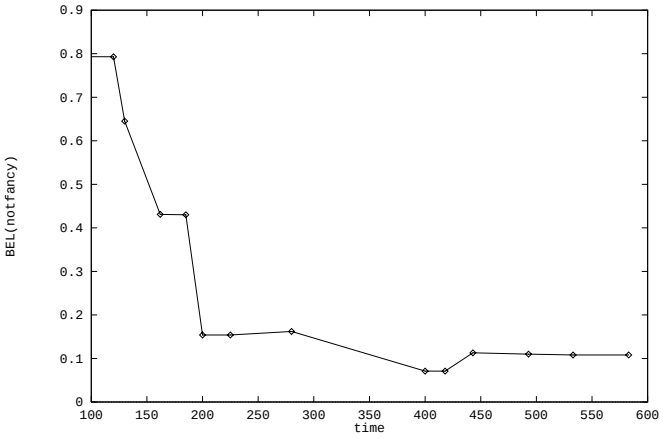
4 Factors Affecting TEA-1's Performance When Solving T-world Problems

We are currently analyzing the relationship of several key factors to the overall performance of a selective perception system, using T-world and TEA-1 and the following approach: (1) automatically generate a large number of simulated T-world domains, scenes, and tasks; (2) run TEA-1 on the generated scenes and tasks; and (3) compute the average solution time over all scenes for each task. This approach lets us show how each factor affects the average solution time. Factors falling in three categories are being analyzed: control method, scene structure, and system parameters [8].

Figure 2: The scene of a fancy meal used in the example run.

k	t	camera decision	$G(\alpha)$	α
0	0	-	-	*a priori*
1	120	move to (15,15)	9.00000	`table`
2	130	don't move	0.00169	`per-detect-butter`
3	162	move to (21,8)	0.00235	`per-detect-napkin`
4	185	don't move	0.00171	`per-detect-plate`
5	200	don't move	0.00522	`per-classify-plate`
6	225	don't move	0.00028	`per-detect-utensil`
7	280	don't move	0.00050	`per-classify-utensil`
8	400	don't move	0.00105	`fov-classify-utensil`
9	418	don't move	0.00003	`per-detect-cup`
10	443	don't move	0.00007	`per-classify-cup`
11	493	don't move	0.00000	`fov-classify-cup`
12	533	don't move	0.00000	`fov-verify-napkin`
13	583	don't move	0.00000	`fov-classify-plate`

(a)



(b)

Figure 3: Summary of an example run using the goodness function approach to control on the scene of a fancy meal. (a) Each line of the table corresponds with one cycle of the decision algorithm, which means decide on and execute the best camera movement action, decide on and execute the best visual action α, and incorporate the results. The best visual action α has the goodness $G(\alpha)$ shown. Actions that process peripheral and foveal image data have the prefixes `per-` and `fov-`. (b) The variation in belief $BEL(notfancy)$ in the goal over time t for the example run. The plot begins at $t = 100$, right after the table-locating action, which always runs first, has been executed. The BEL value after that action is insignificantly different than the *a priori* belief at $t = 0$.

4.1 Control Method

We have explored a variety of goodness functions for deciding what camera movements to make and what visual actions to execute. For example, Figure 5 shows the performance of TEA-1 solving a set of T-world tasks with six different goodness functions for visual actions, while leaving all other factors unchanged. We have also explored several different goodness functions for making camera movement decisions, and have compared the goodness function approach with a state-space search of all possible action sequences [8, 10, 12].

Many fundamental questions remain, however, as to the most effective control and cost/benefit evaluation mechanisms. For example, the original TEA-1 design used a $G_{rate} = value/cost$ measure, though some (camera movement calculations) used $G_{imm} = value - cost$. Computing the goodness of an action as G_{imm} emphasizes finding the best single action to perform now, but maximizing G_{rate} ensures the fastest improvement over time, which is also an important consideration. The current TEA-1 design maximizes G_{imm}, and will be modified to maximize a combination of G_{rate} and G_{imm}. Calibration between value and cost is necessary in either method. The original design used a slightly ad hoc value measure, based on average mutual information, while the new design uses the expected value of sample information, which can be calibrated more accurately with a cost measure

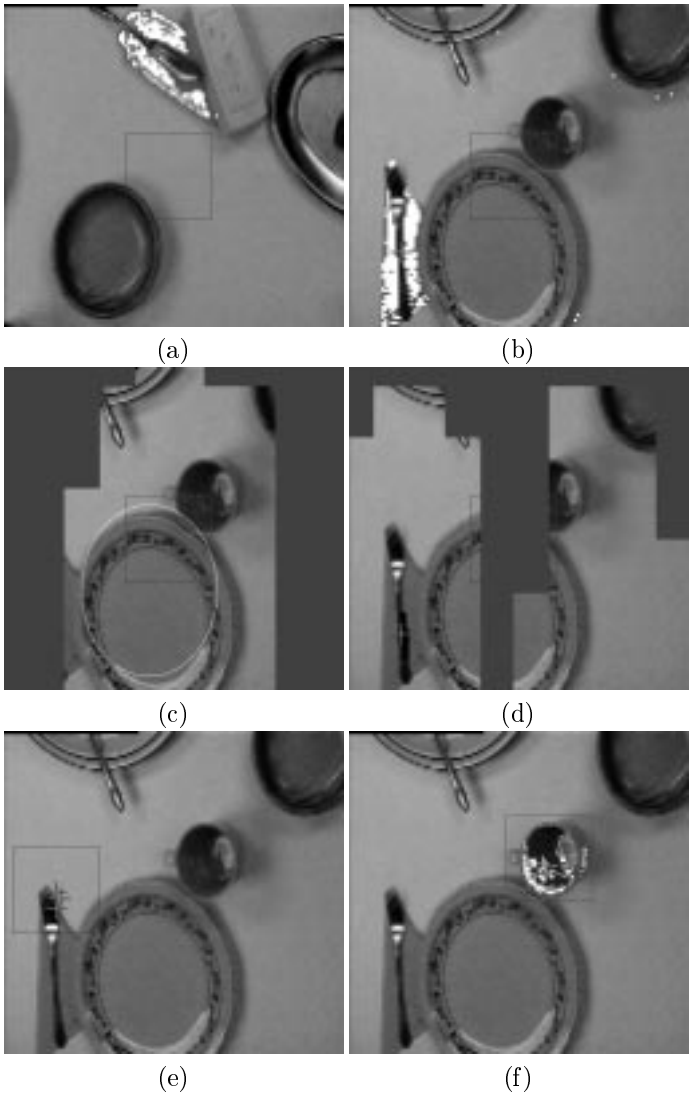


Figure 4: Processing performed by several individual actions during the example run on a fancy meal. (a) Results at step $k = 2$ from the `per-detect-butter` action. (b) Results at step $k = 3$ from the `per-detect-napkin` action, after the camera has moved. (c) Results at step $k = 4$ from the `per-detect-plate` action. Note that previous actions have narrowed down the possible locations for the plate, as reflected by the expected area mask (the blocked out area). (d) Results at step $k = 6$ from the `per-detect-utensil` action. (e) Results at step $k = 8$ from the `fov-classify-utensil` action. Note that this action moves the fovea window and uses foveal image data. (f) Results at step $k = 11$ from the `fov-classify-cup` action, another foveal action.

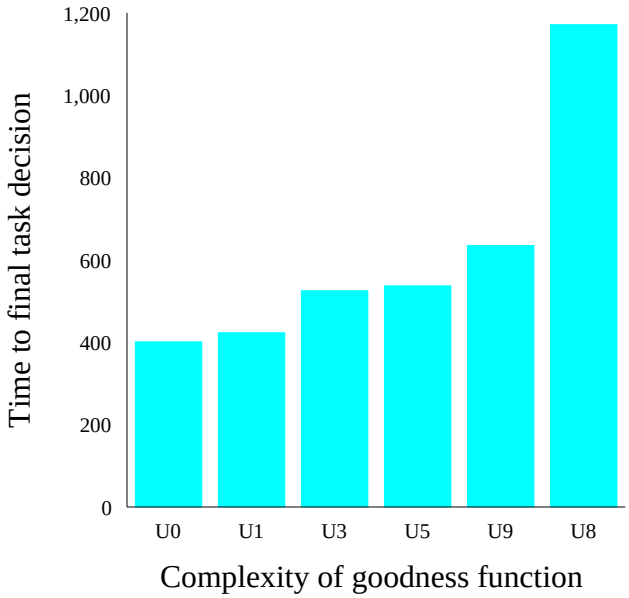


Figure 5: Performance when visual action goodness functions of varying complexity are used. Each bar depicts the total amount of time taken while executing actions until a final decision was made about the solution to a task, averaged over 5 simulated T-world scenes. U8 at the far right is a constant goodness, or random choice of operation, and sophistication increases from right to left. All other factors in the TEA-1 system and in the T-world scenes where left at their default typical values. These results should be considered as a rough comparison only, because the camera movement goodness function should ideally be calibrated to the range of goodness function values, which obviously differs for each goodness function.

and also enables the decisions about the solution to the task and whether to continue gathering evidence to be formalized.

4.2 Scene Structure

Several aspects of scene structure can have a significant impact on performance, because geometric relations define contexts in a scene, which help locate things in the scene and thus help obtain more accurate information more cheaply, and because fewer camera movements may be needed.

We are currently studying several specific scene-structure factors via simulated T-world domains, such as: number of groups and number of subgroup levels that objects are organized into, average number of geometric relations between objects, and shape (and type) of geometric relation distributions between objects. For example, Figure 6 shows several different ways that a scene of nine objects could be structured. Figures 7 and 8 show how the performance of TEA-1 varies for simulated T-world domains containing 16 objects, when the objects are organized into groups containing different numbers of objects and when the spatial extent of the groups is varied.

Another factor that we classify under scene structure is whether it is inherently easier to detect and obtain properties of some objects than others. For example, certain large objects (like runways) may be easier to find than small objects (like service vehicles), while highly constraining the location (or properties) of smaller objects. Wixson demonstrated that the efficiency of actively searching for a specified target object in a room can be improved by a factor to 2 to 8 when a related intermediate object is located first [15].

The T-world problem contains significantly more opportunity for and kinds of scene structure than Wixson's object search problem, which contains only the simple "look near" relation. The solution to a T-world problem can involve several types of information about several objects, rather than simply detecting one object. It will be interesting to compare, for example, the performance gains obtained by using relations with more than one object to those obtained by using relations with only one object.

Another factor in T-world is whether all properties have the same impact on the task, meaning how much effect knowing the value of a specific property has on the value of the task variable. Figure 9 shows how the performance of TEA-1 varies when objects have different kinds of impact on the task.

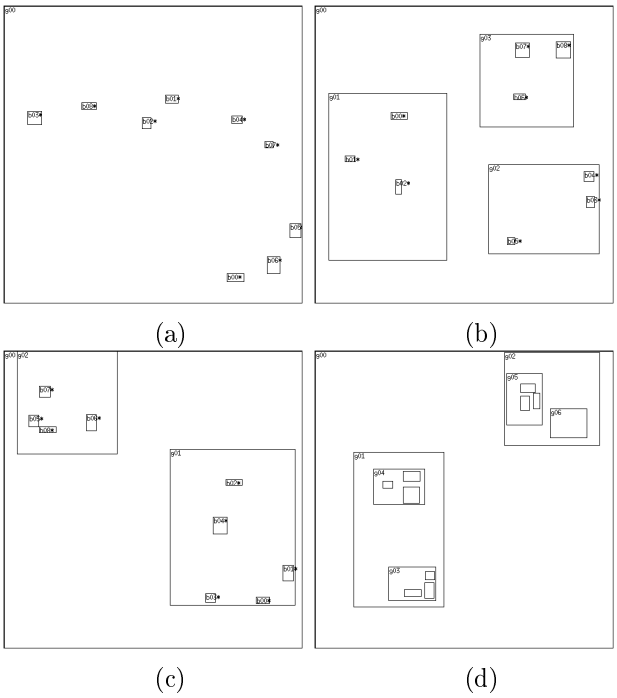


Figure 6: Some examples of how a scene of nine objects could be structured. The smallest squares are objects. The other squares depict groups or subgroups. (a) No grouping. (b) Three groups of three objects. (c) Two denser groups, with five and four objects in each. (d) A subgroup structure.

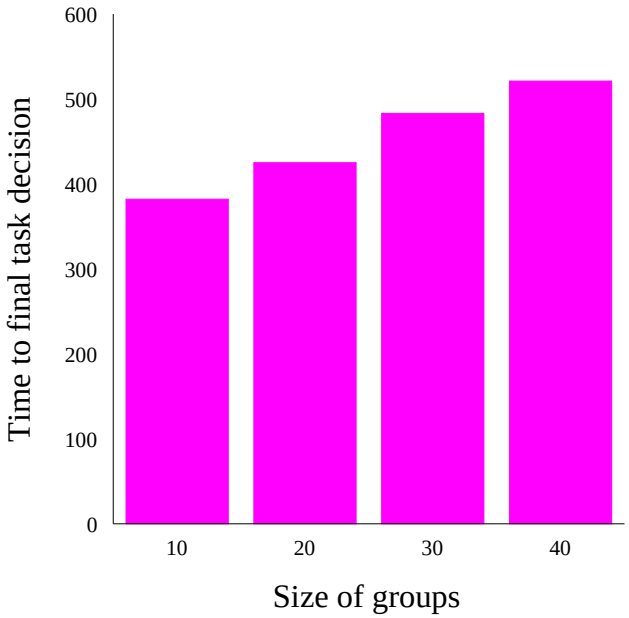


Figure 7: Performance when the spatial size of groups is varied. Here 16 objects are organized into 4 groups of 4 objects each. Each bar depicts the total amount of time taken while executing actions until a final decision was made about the solution to a task, averaged over 5 simulated T-world scenes. The size of a group is specified as a percentage of the scene width, and increases from left to right in the chart. The solution time increases for larger groups because more camera movements are needed to locate objects and because relations between objects in any one group are less constraining.

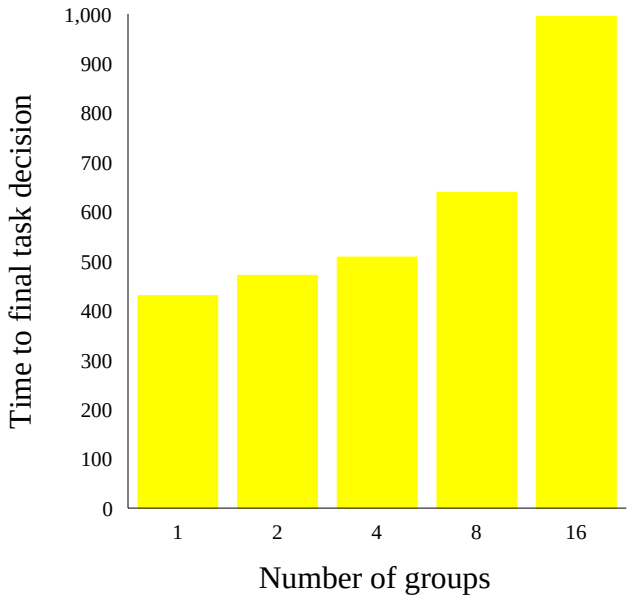


Figure 8: Performance when the number of groups is varied. Here 16 objects are organized into N groups containing 16/N objects each. All groups have the same spatial size. Each bar depicts the total amount of time taken while executing actions until a final decision was made about the solution to a task, averaged over 10 simulated T-world scenes. Each bar is for a different value of N, which increase from left to right. The solution time increases when there are more groups because more camera movements are needed to locate objects and because there are fewer constraining relations between objects in any one group.

4.3 System Parameters

The system parameters category includes: performance model of a visual action (a table of probabilities), relative costs of visual and non-visual actions, size of the camera's field of view, size of the fovea, relative speed of computation (in the multiprocessor version of TEA-1). For example, in [12] we showed the effect of varying the cost of a camera movement, which is generally to stretch out the solution times. (Wixson discusses the effect of some similar system parameters on an object search task in [15].)

Varying some system parameters will change the system's overall pattern of behavior, meaning the best sequence of actions to execute, which can produce interesting effects and raises interesting issues. For example, making camera movements more expensive means that more time is spent analyzing more of the things visible at each camera fixation.

4.3.1 Cycles

The combination of cheap camera movements and "anytime" visual actions [3] could encourage cyclic fixation and analysis patterns to emerge. T-world circumvents knowledge engineering and other practical difficulties in experiments with complex scenes, so we can study these issues.

With cheap vision, humans may not use their innate powers of representation and memory and may prefer just to update short-term memory. This strategy seems to be found in humans [1] in repetitive sequential hand-eye tasks. On the other hand, human eye fixations during even simple tasks clearly show evidence of rational sequential control ([17], and see Figure 10). Further, vision is expensive when peripheral vision is reduced, when there are distractors, noise, low contrast, *etc.* Humans *do* manage their visual resources, even for static scenes, and their management strategies are open to investigation through several avenues. We are hopeful that we can relate decision-theoretic control to human performance by using modern eye-, head-, and hand-tracking technology to observe humans performing T-world tasks.

5 Conclusions

Extensive references can be found in our other papers [8, 9, 10, 11, 12]. Our work on task-based vision is most directly comparable to [5], which put a carefully designed version of model-based hypothesis verification vision into a Dempster-Shafer setting, and the related [4, 16]. Our work is also closely related to the computer vision and mobile robot applications of decision theory in [6] and [2] respectively.

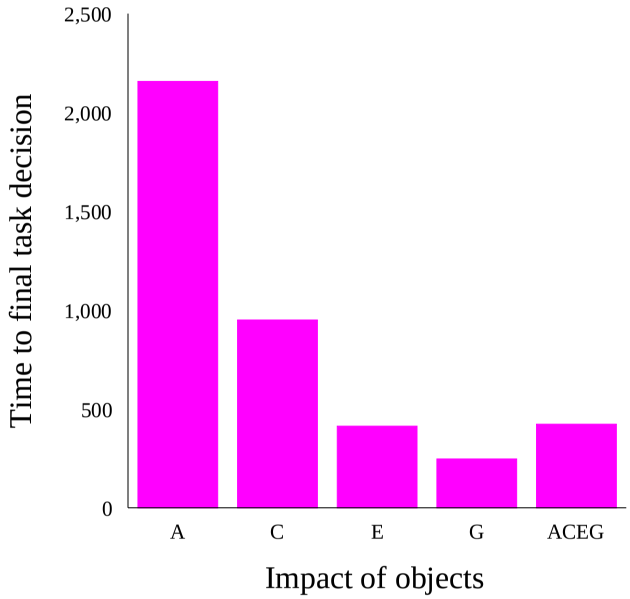


Figure 9: Performance when objects have different kinds of impact on the task. Each bar depicts the total amount of time taken while executing actions until a final decision was made about the solution to the task, averaged over 5 simulated T-world scenes. The first four bars depict the results when all objects have approximately the same average impact, where the average impact increases from left to right. Obviously, stronger impacts yield shorter solution times. Objects in realistic scenes have a wide variety of impacts. The rightmost bar shows the result when the impacts of the objects in a scene are an equal mix of the impacts used for the previous four bars. The solution time here is quite fast because the system performs an optimization that takes into account the expected impacts of (and costs of obtaining) object properties. (The leftmost bar is an estimate, since TEA-1 could not reach a final decision at the required confidence level before running out of objects in the scene.)

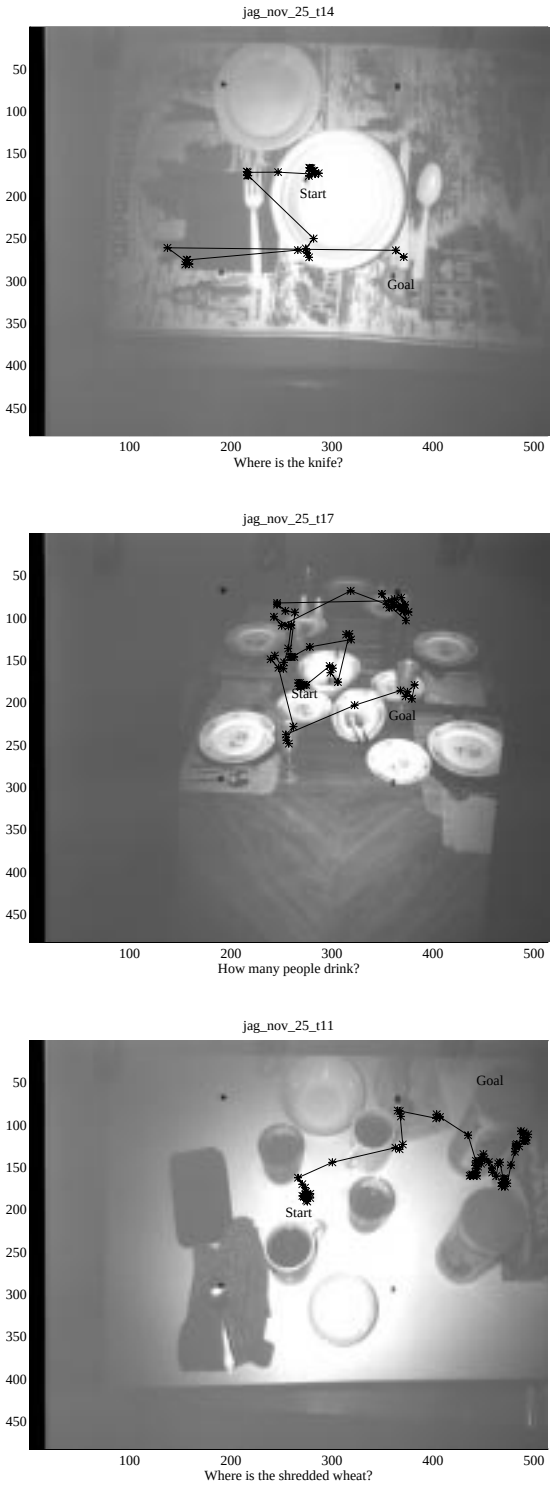


Figure 10: Eye-fixations in visual tasks. (top) Where is the knife? S habitually places knife to left of plate. (middle) How many people drink? S uses plates somehow. (bottom) Where is the Shredded Wheat? S examines objects sequentially, checks can label, moves on.

General framework. We have presented the TEA-1 system, an example of one reasonable way to design a general, reusable framework for a purposive and sufficing vision system using Bayes nets and decision theoretic techniques. The operation of the TEA-1 system has been demonstrated on a table setting domain in the lab, simulated T-world domains, and a simulated (dynamic) model trains domain (see [14]), and we believe it can be made to work with a variety of real-world applications (see [8]) that can be mapped to T-world problems.

Other computer vision and robot systems have been built that use decision theoretic techniques, Bayes nets, Dempster-Shafer or similar modeling techniques, usually in the lower-level capacity of supporting sensor fusion for single object recognition or building environmental descriptions. TEA-1 addresses high-level computer vision, and is one of the first to consider how to control an actively pointable sensor and to emphasize purposive and sufficing control. Since TEA-1 is a fully implemented system, we have been able to perform extensive experiments in simulation and in the lab using complete runs on a large number of scenes.

Task representation. TEA-1 represents tasks using Bayes nets, and the TEA-1 approach can solve a variety of visual tasks (generally ones involving a collection of objects in a scene, as in the T-world problem for example).

Decision making. There are many approaches to control in a selective perception system ranging from brute-force and heuristic search through hand-crafted goodness functions to a formal planning system. Our work explores this spectrum of choices, studying and experimenting with some of the choices and exploring the issues in control and how each choice deals with those issues. In particular we have presented and evaluated several goodness functions for T-world problems, and have compared these to brute force search solutions.

T-world. We have defined the T-world problem, a simple class of vision problems that still contains many of the key factors motivating the selective perception approach. We believe T-world is an adequate problem for easily studying some of the basic issues in selective computer perception. We can explore the key factors that make the selective perception approach appealing by analyzing how each factor affects TEA-1's overall performance when solving a set of automatically generated (in simulation) T-world domains and tasks. One group of factors that we have explored is the amount of "structure" there is in a scene, meaning the various ways that objects may be grouped in

a scene and the spatial relationships between objects and groups.

Selective vs. non-selective perception. The analyses provided in [7, 13, 15] provide support for the selective perception approach. Control of selective perception is a kind of optimization problem, and we have presented experimental evaluation of some factors involved in that optimization process. Some of these results suggest how non-selective approaches perform: The relative size of the ACEG bar in Figure 9 demonstrates the importance of reasoning about the impact (for a specified task) and the cost of properties in the scene. The U8 bar (random choice of operation) in Figure 5, which is roughly analogous to a full reconstruction approach, depicts performance 2-3 times worse than the selective approach (the other bars). Removing the decision to stop or (randomly) gather more evidence from the U8 trials would produce even worse performance.

Acknowledgements

Chris Brown has been an invaluable advisor for my thesis work. Peter von Kaenel helped build parts of the T-world simulator and several vision modules and visual actions. Tim Becker improved modularity of the T-world/TEA-1 system, and parallelized several parts of it. Martin Jagersand obtained the eye movement recordings in Figure 10, using software being developed by Jeff Pelz.

References

[1] D. Ballard, M. Hayhoe, F. Li, and S. Whitehead. Hand-eye coordination during sequential tasks. *Phil. Trans. R. Soc. Lond. B*, 335, 1993.

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

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

[4] G. D. Hager. *Task-Directed Sensor Fusion and Planning: A Computational Approach.* Kluwer Academic, 1990.

[5] S. A. Hutchinson and A. C. Kak. Planning sensing strategies in a robot work cell with multi-sensor capabilities. *IEEE Journal of Robotics and Automation*, 5(6):765–783, 1989.

[6] 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.

[7] D. A. Reece and S. A. Shafer. Planning for perception in robot driving. In *Proceedings: DARPA Image Understanding Workshop*, pages 953–960, 1992.

[8] R. D. Rimey. *Control of Selective Perception Using Bayes Nets and Decision Theory.* PhD thesis, Department of Computer Science, University of Rochester, 1993. (Also available as Technical Report 468, Department of Computer Science, University of Rochester, December 1993).

[9] 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.

[10] R. D. Rimey and C. M. Brown. Task-specific utilities in a general Bayes net vision system. In *Proceedings: IEEE Conference on Computer Vision and Pattern Recognition*, pages 142–147, 1992.

[11] 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.

[12] R. D. Rimey and C. M. Brown. Control of selective perception using Bayes nets and decision theory. *International Journal of Computer Vision*, 12(2-3), 1994. To appear (Special Issue on Active Vision, Volume 2).

[13] J. Tsotsos. The complexity of perceptual search tasks. In *Proceedings: International Joint Conference on Artificial Intelligence*, pages 1571–1577, 1989.

[14] P. A. von Kaenel, C. M. Brown, and R. D. Rimey. Goal-oriented dynamic vision. Technical Report 466, Department of Computer Science, University of Rochester, August 1993.

[15] L. Wixson. Exploiting world structure to search for objects efficiently. Technical Report 434, Department of Computer Science, University of Rochester, July 1992.

[16] H. L. Wu and A. Cameron. A Bayesian decision theoretic approach for adaptive goal-directed sensing. In *Proceedings: International Conference on Computer Vision*, pages 563–567, 1990.

[17] A. Yarbus. *Eye Movements and Vision.* Plenum Press, 1967.