Selective Attention as Sequential Behavior: Modeling Eye Movements with an Augmented Hidden Markov Model

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

Abstract

Visual attention is an important problem in computer vision that has received little direct attention in computer vision research. We are pursuing a long-range program of research into the visual attention problem. This paper describes our first thrust in this program, the sequential aspect of attentive vision systems. We assume a spatially variant sensor with a fovea and periphery, and study foveal sequencing, or the problem of where next to concentrate high-resolution vision. We present a new explicit representation of attentional sequencing, called an augmented hidden Markov model (AHMM). An AHMM can model a sequence of *where* to place the fovea, or a sequence of *what* objects to look at. A dual-AHMM, combining a *what* and a *where* model with symmetric feedback, is also presented. Our model allows behavior to be learned and to be responsive to individual scene variations.

1 Overview

A system using real-world visual input for decision making must ignore the irrelevant, attend to the salient, and place reliable priorities on tasks and resources. Complexity analysis of the problem of matching visual images to models reveals that *pure parallelism is not enough* to overcome the visual computational burden, but that structure, in the form of a hierarchy of spatial resolutions and of abstractions, can render this visual task tractable [Tsotsos, 1987].

This material is based on work supported by the NSF under grant CDA-8822724, and by DARPA/AFOSR research contract no. AFOSR-89-0222. The government has certain rights in this material.

One way human systems overcome the visual computational burden is by the mechanism of *attention*, a topic extensively studied in the areas of psychology and neuroscience. We are pursuing a long-range program of research into the selective attention problem in computer vision.

We are currently exploring the thesis that attention provides mechanisms that control the *allocation* of visual processing preferentially within a scene, leaving open just which resources are being managed. Here we investigate the allocation of a spatially variant sensor such as one with a *peripheral visual field* of low resolution but wide angle, and a *high-resolution fovea* centered in the visual field. We present a novel approach to learning, representing, and generating explicit *attentional sequences* that direct such a sensor to view specific areas of a scene. The capability we describe is like a visual-motor skill — it emphasizes the efficient acquisition and use of relevant behavior for a repetitively-occurring situation, with the capability of adapting both to individual variations between problem instances and to slow variations in the expected situation. In a usual data-driven foveation (or region of interest) system, a sequence of eye movements *emerges* from a program reacting to the image data (often using low-level saliency, or "interest", operators). Our *explicit* model for such sequences can, for example, be trained on emergent sequences generated by other algorithms and be made to generate these sequences explicitly, whether they are continuous attentional paths or discontinuous, saccadic fixations. In other words the sequential attentional behavior can become automatic, or "compiled" into a lower-level, pre-attentive visual skill.

Our model is based on the hidden Markov model (HMM), which is roughly a generalization of a teachable, probabilistic finite state automaton. The HMM camera control model operates in a mode oblivious to the visual data. We introduce a modified model, called

an augmented HMM (AHMM), for the more typical case in which the movements should be responsive to (*i.e.* be modified by) visual cues.

Three models are described below. The first uses external feedback to affect the AHMM outputs (only). The second uses internal feedback to modify the internal parameters (probabilities) of the AHMM, thus affecting the generation likelihoods directly. These models can be applied for the purpose of generating a sequence of *where* to point the camera. The model can also be applied to generate a sequence of *what* to look at. The final model combines both a *what* and a *where* part, and a symmetrical scheme in which the two parts dynamically feed back to and support each other.

This paper is organized as follows. Section 2 discusses why attention, spatially-variant sensors, and attentional sequencing are important topics in computer vision. Section 3 describes in detail the AHMM models we have developed, and presents experiments applying the models. Conclusions and plans for future work are given in Section 4.

2 Motivation

Recently the advent of sophisticated controllable visual hardware has emphasized sensor control, or the "active vision" paradigm (*e.g.* [Bajcsy, 1988, Ballard, 1989, Brown, 1988, Burt, 1988]). Besides advocating the idea that vision and action modules should be designed to cooperate and support each other, active vision promotes a general re-examination of certain aspects of the human visual system [Ballard, 1989] for ideas on how to build computer vision systems. This section discusses two topics in this vein, anthropomorphic visual sensors and visual attention. Then the ideas of attentional sequencing and visual skills are presented.

2.1 Foveal and peripheral sensors

The fovea-periphery distinction is quite dramatic in the human visual system, but usually humans are not consciously aware of it. All of our high-resolution vision is performed by a fovea whose field is only 0.5 degrees of visual arc (about the extent of a quarter coin held at arms' length). The remaining large peripheral field only provides low-resolution vision.

A fovea, accompanied with attentional algorithms and control machinery for directing it, becomes a viable engineering solution when imaging, transmission, and computing bandwidth are limited. Several visual computations may also become easier in the context of a foveal sensory system. For example, various uses of stereo disparity become easier when coupled to camera vergence [Coombs, 1989], as do kinetic depth computations when coupled with fixation [Ballard and Ozcandarli, 1988].

A spatially variant sensory device can be created in several ways. Anthropomorphic VLSI sensors are being constructed (*e.g.* [Tistarelli and Sandini, 1990]). Software and hardware resolution pyramids are a classic technique (*e.g.* [Burt, 1988]). Another choice is simply to use two cameras with two different focal lengths. Finally, a simple electronic window could be used.

2.2 Attention

The topic of visual attention has been identified and extensively studied by researchers in psychology and the neurosciences ([Poggio and Hurlbert, 1985, Posner and Presti, 1987, Humphreys and Bruce, 1989]). Attention is usually identified with the limited availability of resources. In the spotlight model of attention a fixed size "spotlight" can be shifted around to enhance visual processing in the area it covers. Covert attention is shifted within the visual field, but is not associated with eye movements, whereas shifts in overt attention are directly linked with eye movements. "Popout" phenomena, studied by Treisman (*e.g.* [Treisman and Gelade, 1980, Treisman, 1985]) and others, examine the relation between preattentive immediate vision and serial attentive vision.

Ullman's "Visual Routines" paper [Ullman, 1984] explores how ideas from the psychological and neuroscience research on visual attention can be applied to computer vision, and it enunciates several interesting ideas that have inspired our work. Ullman proposes five primitive visual routines: attention shift (*i.e.* controlling and moving the location of attention), indexing (*i.e.* selecting specific locations for further processing), bounded activation, boundary tracing, and marking. Attention shift and indexing are the primitive routines that mainly concern us.

2.3 Attention as sequential behavior: Foveal or attentional sequencing

The psychological literature contains much work on eye movements, the most direct evidence for visual attention (though usually for two-dimensional stimulii). Yarbus' book [Yarbus, 1967] documents graphic traces of eye movements as humans examine scenes for relatively long times (three minutes). Some intriguing observations are: Subjects always foveate *only* select areas in the scene, those containing "relevant objects". For a single task, a given subject repeatedly uses a foveation sequence with only minor variations. For different tasks, the general sequence that all subjects use is highly dependent on the task.

Yarbus's work led to the idea that an object is represented for visual recognition as a *scanpath* – a time-ordered sequence of features perceived at each fixation, along with motor commands that link the fixations [Noton and Stark, 1971]. Subsequent work [Stark and Ellis, 1981] presented a simple probabilistic model for fixation sequences. Didday and Arbib [Didday and Arbib, 1975] describe how similar behavior can emerge from parallel operations on foveal and peripheral image data. A modern piece of work with a fovea-periphery distinction, controllers to determine the location of the next fixation, *and* experiments is that of [Bolle *et al.*, 1990]. Three controllers are reported: A simple scanning process to move the fovea in a task- and data-independent way; one whose candidate is the largest so-far unexplained region; and one that tries to resolve conflicts between the interpretation so far and the model database. Similar objectives were accomplished with resolution pyramid hardware [Burt, 1988], where foveation was implemented as coarse to fine search through the pyramid.

Control of a fovea and blending its output over fixations is an increasingly common topic [Abbott and Ahuja, 1989, Browse and Rodrigues, 1988, Yeshurun and Schwartz, 1989]. The control algorithms are typically just to examine next the area of highest interest (using some bottom-up measure), using memory or a salience-reduction method to avoid re-examining an area, and perhaps an exploratory urge. Clark and Ferrier [Clark and Ferrier, 1988] report experiments using a similar scheme for controlling a binocular robot.

2.4 Visual skills

Two aspects of the previous work stand out. First, previous systems have generally produced *sequences of fovea movements*. Given limited resources, sequential allocation is obviously a reasonable strategy. Secondly, almost all previous work has studied *emergent sequences*. Sequence steps are produced as the output of processes operating on the image data. They are not represented, remembered, or available in advance.

The alternative is *explicit sequences*: Sequence representations are maintained at some level, and can be modified, retrieved, and generally appear in computations. Explicit representations of foveation sequences are rarely proposed, and seldom implemented. The main exception is the experiments of [Stark and Ellis, 1981], and perhaps the idea of motor programs (*e.g.* [Wright, 1990]). Indeed, foveation sequences are not a convincing way to represent objects for recognition.

However, explicit sequences may well be a good way to represent a strategy for certain types of skilled observation of a structured environment. Emergent sequences, computed a step at a time, are analogous to motor behavior, mediated primarily by general perception and by more or less cognitive involvement. Emergent foveation sequences must be rederived each time, but should be the same for the same situations. Remembering such a sequence efficiently captures the effects of the cognition applied to the task domain. A remembered sequence can be regenerated while also being fine-tuned using current visual feedback data. Thus, a remembered (explicit) sequence is analogous to skilled motor behavior.

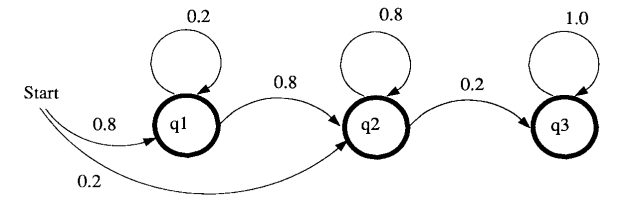
For example, when entering into your new office for the very first time, you look around in an undirected way. You tend to look at "interesting" objects, but also at areas and objects that in retrospect are not of any significance or never change. Over time, you develop a pattern of where you look upon entering your office. Perhaps you first look towards the terminal in the near corner, expecting to see an office mate there, and then across to your desk in another corner, scanning another office mate's desk along the way. (The emergent behavior developed a pattern that has now been learned as an explicit behavior, or habit.) While executing the habitual eye movement behavior you often modify it according to visual cues along the way. For example, you can tell that no one is at the terminal using your peripheral vision, and if not, your eyes begin moving towards your desk sooner. If there is some interesting new object on the office mate's desk or your own, your eyes are attracted to it before continuing along their typical path. (The habitual behavior is modified by cues in visual feedback.)

Other examples abound, for example, in industrial visual inspection or driving skills.

In summary, the skill or habit analogy is appealing. A method to model a visual skill, so it could be learned and executed with visual feedback, would be useful. Since knowledge is "compiled" into a visual skill, simpler vision processing would be sufficient, thus reducing computational demands. The model could also specialize itself and adapt to its environment, compensating for single-instance and slowly trending variations in the world.

3 An augmented hidden Markov model for explicit sequences with visual feedback

This section presents an explicit representation — an augmented version of a hidden Markov model — for attentional sequences modified by visual feedback.



v1 (left)	0.5	0.7	0.0
v2 (up)	0.5	0.3	1.0

Figure 1: Example of a simple HMM.

3.1 Hidden Markov models

We need the ability to learn, represent, generate and even classify sequences of (2-D or 3-D) spatial locations for foveations or attention. The hidden Markov model (HMM) has been widely used to classify signals in speech recognition systems, but in fact it has all the abilities we desire. Although it is a very general model for sequences it has not been used much in other fields, such as computer vision. The key points about HMMs are summarized here. For details see the excellent tutorial by Rabiner [Rabiner, 1989, Rabiner and Juang, 1986]. Methods to incorporate feedback into an HMM, the final ability we need, are presented in following sections.

An HMM is somewhat like a probabilistic finite state machine. It is formally defined as $\lambda = (A, \pi, B)$ with states $Q = \{q_1, ..., q_N\}$ and symbols $V = \{v_1, ..., v_M\}$. The probability of transitioning at time step t from state q_i to state q_j is given by $A = \{a_{ij}\}$ where $a_{ij} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)$. The initial state is determined by $\pi = \{\pi_i\}$ where $\pi_i = P(q_i \text{ at } t = 1)$. If the HMM is in state q_j it produces symbol v_k according to the probability $B = \{b_j(k)\}$ where $b_j(k) = P(v_k \text{ at } t \mid q_j \text{ at } t)$. Note that the symbol sequence is observable, but that the state sequence is "hidden" (not observable).

Figure 1 shows an example HMM. The state sequence it models will tend to contain a short subsequence of q1 states, a larger subsequence of q2 states, and q3 states until the end of the sequence. Assuming that symbols v1 and v2 were associated with "left" and "up"-ward (incremental) eye movements, and that the HMM remained in state q1 for several steps, then it would tend to move the eyes to the upper left during that time.

The graph structure of an HMM may be arbitrary. Graphs with a left-to-right flow, such as in Figures 1, 5 and 6, are often appropriate for non-cyclic sequences. In practice the particular graph structure is crucial to performance and its selection requires experimentation.

HMMs have three associated capabilities, at least one of which is used in any application. They can classify sequences, they can be learned from examples, and they can generate sequences. Each of these capabilities is well known: detailed algorithms for implementing them may be found in [Rabiner, 1989, Rabiner and Juang, 1986] and will not be repeated here.

A single HMM λ_i is associated with each possible class ω_i. An observed sequence O is classified as the most likely class, according to $P(O \mid \lambda_i)$. The parameters λ_i for each HMM are estimated separately using a "training set" of examples and an algorithm to maximize $P(O \mid \lambda_i)$ over the set.

Sequences are generated using random numbers chosen from the appropriate probability distributions in λ_i. The length of a sequence is determined by a bound on $P(O \mid \lambda_i)$ or by using a fixed length. The random method, when *visual feedback* is added, gives robustness to local and long-term variations in the world.

3.2 Using HMMs for oblivious eye movements

To apply HMMs to attentional sequences, the symbols must be related to spatial locations or movements. Following are a few choices.

- *Incremental-position movement.* The symbols are $v_i \in [0, 7]$, the eight chain code ("compass point") directions. The eye rotates a fixed increment so that the image shifts by an increment in the given direction. Here, the attentional sequence is a relatively smooth path, of the sort arising in contour following, doing vision through a reduction tube, or in some other situations [Ullman, 1984].

- *Large-position movement.* The symbols are $v_i = (\theta, R)$ where θ and R are quantized direction and distance. Each movement is relative to the current eye location. Alternatively, the symbols can be $v_i = (x, y)$ where each movement is in a fixed coordinate system. Here, an eye movement sequence is similar to a series of saccades.

- *Object sequencing.* The symbols are feature vectors describing objects. If image analysis produces such feature vectors throughout the image, the sequence defines a series of locations (possibly ambiguous, see Section 3.5) in the image.

In all cases a training set of example sequences can be created from the emergent movement decisions output by some other algorithm, or a human using a

pointer on several example images. Alternatively, if technical considerations permit, recordings of human eye fixations generated during the task can be used.

3.3 Augmented hidden Markov models: external feedback modification of oblivious eye movements

This section presents a trivially augmented AHMM, which will serve as an example to explain feedback modified eye movements and saliency-based feedback cues. The following sections present more sophisticated AHMMs.

The external feedback AHMM. Let the oblivious sequence of symbols normally generated by an HMM be $O = O_1, ..., O_T$, and let the feedback sequence be $S = S_1, ..., S_T$. S_i is a symbol representing the feedback cue at index i in the sequence. We assume that feedback symbols and output symbols are both of the same type. The AHMM simply outputs the feedback modified path, $M = M_1, ..., M_T$, according to the following equation.

$$M_i = \begin{cases} S_i & \text{with probability } \alpha \\ O_i & \text{otherwise} \end{cases} \quad (1)$$

α is a parameter from 0.0 to 1.0 that regulates the amount of influence from the feedback sequence. With $\alpha = 0.0$ the feedback data is completely ignored. With $\alpha = 1.0$ the HMM is ignored and the feedback sequence is tracked exactly.

Computation of visual cues. In our experiments we have used feedback cues based on local maxima of a saliency image. S_i is simply the direction (actually a chain code symbol) towards the local maximum of the saliency data. Generally, of course, any other kind of feedback cue can be used. A saliency image is computed as a weighted sum of several feature images, which are themselves computed from the original intensity image.

A Gaussian neighborhood function can be used to compute the local maximum. However, this would cause problems when generating incremental-position ("smooth path") eye movements, since there would be a tendency for the path to turn around towards a maximum behind it and not depart from a maximum it had reached. We handle the problem by using a neighborhood function that emphasizes saliency points in the direction of the path, and saliency points away from the current location.

The above method is applied separately to the foveal and peripheral images to compute a peripheral feedback cue $S_i^{(p)}$ and a foveal cue $S_i^{(f)}$.

Modifying incremental- position movement. Recall that in an incremental-position movement application each O_i is a chain code symbol. Before the eye moves, this sequence defines a path through the peripheral image. First, a periphery modified path, with elements $M_i^{(p)}$, is generated according to

$$M_i^{(p)} = \begin{cases} S_i^{(p)} & \text{with probability } \alpha \\ O_i & \text{otherwise.} \end{cases} \quad (2)$$

Then, foveal modification is performed according to

$$M_i^{(f)} = \begin{cases} S_i^{(f)} & \text{with probability } \beta \\ M_i^{(p)} & \text{otherwise} \end{cases} \quad (3)$$

resulting in the final sequence $M^{(f)} = M_1^{(f)}, ..., M_T^{(f)}$. Note that the eyes must actually be moved to make the appropriate foveal data available to compute $S_i^{(f)}$.

Modifying large- position movement. Large-position movement might, for example, use symbols of the form $O_i = (x, y)$, which denote absolute positions ("targets") in the periphery. Feedback is determined from the maximum of the peripheral saliency image in the neighborhood of the target. The maximum is computed using a Gaussian neighborhood function centered on the target. Unfortunately equation (1) does not provide a satisfactory form of modification. The AHMM presented in the next section addresses this problem. (Modifications using foveal data before a large-position movement is performed are not appropriate since the target is usually outside the fovea. However, once the target is reached, the foveal data can be used to perform a small adjustment movement.)

3.4 An augmented hidden Markov model with internal feedback

The internal feedback AHMM has parameters that vary as a function of time, denoted as $\lambda^{t+1} = (A^{t+1}, \pi, B^{t+1})$. The AHMM operates as follows. Assume that the AHMM is at time step t, that it has already output a sequence of symbols $O_1, ..., O_t$, and that the current state is q_i. The feedback symbol S_t is available, where the value of S_t is v_k. S_t is used to modify λ^t into λ^{t+1}, then the AHMM uses λ^{t+1} to generate the next symbol, O_{t+1}. Obviously the choice of the next generated symbol depends partially on what the few most recent feedback symbols have been. The λ^t parameters also slowly decay over time to their original values. So as long as consistent feedback symbols are available, their effect on the AHMM parameters will endure, but eventually the parameters return to their original values. The initial state probabilities π do not vary in time since a feedback symbol is not available at time $t = 0$. The equations for computing λ^{t+1} are summarized below. See [Rimey and

Brown, 1990] for their derivation and a more complete explanation.

A weighting factor. The equations for modifying the AHMM parameters use three key values: i, k, and w_j^t. The value i is determined from the state q_i of the AHMM at the current time step, t. The value k is determined from the value of the current feedback symbol, $S_t = v_k$. The values for i and k are known and will be assumed in all the equations below.

The final key value is w_j^t, a probabilistic weight computed from i and k: w_j^t is the probability of being in state q_j at time $t+1$, given the information that the AHMM is in state q_i at time t and will output symbol v_k at time $t+1$. The equation for computing w_j^t is

$$w_j^t = \frac{a_{ij}^t b_j^t(k)}{\sum_{l=1}^{N} a_{il}^t b_l^t(k)} \qquad 1 \le j \le N. \tag{4}$$

The current feedback symbol (S_t) is assumed to be a prediction of the next output symbol (O_{t+1}), so w_j^t provides an indication of how consistent the immediately possible state transitions are with the current feedback, and it can be used to bias the state transition probabilities.

Modification of a_{ij}^t. New values for the transition probabilities, denoted by $\tilde{a}_{ij}^{t+1}$ for the time being, that are most consistent with the feedback are $\tilde{a}_{ij}^{t+1} = w_j^t$. Generally it seems desirable to modify a_{ij}^t slowly rather than completely replace it. Therefore only a fraction, $r_f w_j^t$, is mixed with the current value. The new equation is

$$\tilde{a}_{ij}^{t+1} = r_f w_j^t + (1 - r_f) a_{ij}^t \qquad 1 \le j \le N \tag{5}$$

where r_f ($0 \le r_f \le 1$) is a modification gain. Larger gain values emphasize the feedback modified transition probability over the original probability.

Modification of $b_j^t(l)$. The emission probabilities in the AHMM must be updated for each state q_j that can be reached in one time step from the current state q_i. The equations for the updated emission probabilities, denoted $\tilde{b}_j^{t+1}(l)$, and already incorporating a mixing gain, are as follows.

$$\tilde{b}_j^{t+1}(l) = \frac{e_j^t(l)}{\sum_{m=1}^{M} e_j^t(m)} \quad 1 \le j \le N, \quad 1 \le l \le M \tag{6}$$

$$e_j^t(l) = s_f w_j^t d_j^t(l) + (1 - s_f) b_j^t(l) \tag{7}$$

$$d_j^t(l) = \begin{cases} 1.0 & \text{if } l = k \\ 0.0 & \text{otherwise.} \end{cases} \tag{8}$$

The denominator in equation (6) ensures that $\tilde{b}_j^{t+1}(l)$ is a valid probability. The modification gain is s_f ($0 \le s_f \le 1$), where small gain values emphasize the original emission probability over the feedback modified ones. The key term in these equations, the one contributed by the feedback, is $w_j^t d_j^t(l)$.

Time decay. Equations (5) and (6) are the basis for modification at any instant in time. Since these modifications should only be maintained as long as they are justified by feedback information, the modifications are made to decay in time as follows

$$a_{ij}^{t+1} = r_d \tilde{a}_{ij}^{t+1} + (1 - r_d)\hat{a}_{ij} \qquad 1 \le j \le N \tag{9}$$

$$b_j^{t+1}(l) = s_d \tilde{b}_j^{t+1}(l) + (1 - s_d)\hat{b}_j(l) \qquad 1 \le j \le N, \quad 1 \le l \le M \tag{10}$$

where $\hat{a}_{ij}$ and $\hat{b}_j(l)$ are the original values, which do not change over time. The decay gains are r_d and s_d ($0 \le r_d, s_d \le 1$). Small decay gains cause the probabilities to decay quickly to their original values. These equations give the final values for a_{ij}^{t+1} and $b_j^{t+1}(l)$.

Multiple feedback signals. The above AHMM can easily be extended to the case that several different feedback symbols are available from different feedback sources. The feedback symbols at time t are denoted by a *set* $\mathbf{S_t}$. For example, this set could contain either simultaneous peripheral and foveal feedback symbols, or multiple peripheral feedback symbols. The updating equations are similar to those above, and can be found in [Rimey and Brown, 1990].

Modifying incremental- position movement. The above AHMM can be used in a straight-forward manner to generate incremental-position movements. Multiple feedback symbols, for simultaneous peripheral and foveal feedback, are used.

Modifying large- position movement. The above AHMM can be more easily applied to large-position movements than could the first AHMM we presented. The AHMM operates in two stages during each time step. First it generates a preliminary output symbol (target location). Then a Gaussian neighborhood function is centered on the target location. Local maxima are detected and used as multiple feedback to modify the AHMM parameters. Lastly, the AHMM generates the final version of the output symbol for the time step. The internal feedback modifications in the AHMM result in a form of (application independent) averaging of the feedback symbols and the "oblivious" output symbol, essentially because the internal probabilities are averaged over time. Multiple feedback paths can also be used, for example, by using several of the largest local peripheral-saliency maxima (rather than just one). Large-position movements using foveal feedback are still not possible because the fovea does not view the target area.

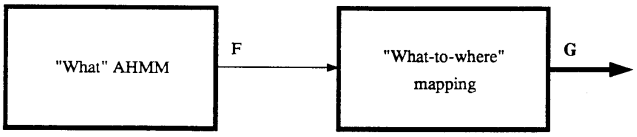


Figure 2: The "what" version of the AHMM. A Darker signal path denotes a set rather than a single signal value.

Adding new states. The above AHMM does not permit an existing state to add a new non-zero emission probability, or to add a new link, or for the graph to add a new state. The modification equations above were derived assuming that the feedback sequence reflects variations that are generally still consistent with the underlying modeled sequence. If the underlying sequence (in the real world) has changed in a fundamental but local way it may be necessary to insert a new state into the AHMM graph. This situation may be detected by a small value of

$$\max_{1 \le j \le N} \hat{a}_{ij}^t \hat{b}_j^t(k) \tag{11}$$

in which case a slow semi-permanent modification may be initiated in which a new state is added.

3.5 A mutually supporting what-where feedback model constructed from two internal feedback AHMMs

This section describes a hybrid AHMM that elegantly incorporates both "what" and "where" sequence models and uses them to play off and mutually support each other.

A what-AHMM. The "what" version of the AHMM, called a what-AHMM and shown in Figure 2, contains two sections. The first section is an internal feedback AHMM whose output symbols F_i, called *what-symbols*, are feature vectors intended to describe an object or characteristics of objects. Such feature vectors are assumed to have been computed for each pixel in the peripheral image. The second section of the what-AHMM performs a "what-to-where" mapping, meaning that it maps a feature vector into the set of image coordinates $\mathbf{G_t}$ in the current image where instances of those feature vectors (or similar ones) exist. Actually it maps to the eye movement commands, called *where-symbols*, that would cause those locations in the image to be centered on the fovea.

If each $\mathbf{G_t}$ contains exactly one element, the output sequence will fixate the desired objects in the scene. Each $\mathbf{G_t}$ does not generaly contain exactly one element, so some method must be developed to select among the choices. One option is to use a where-AHMM to help pick among the choices. In fact, the what-AHMM can be made to help the where-AHMM with its *own* choices.

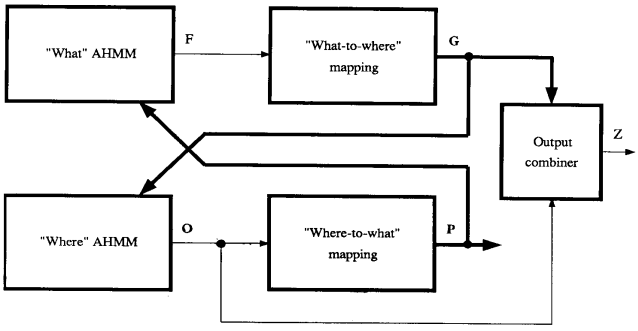


Figure 3: The "what-where" version of the AHMM. A Darker signal path denotes a set rather than a single signal value.

A what-where-AHMM. The what-where-AHMM shown in Figure 3 contains three distinct parts: a what-part and a where-part which both feed back to each other, and an output combiner. The what-part is like the what-AHMM described above. It uses feedback, but the feedback is a sequence of *sets* of what-symbols $\mathbf{P_t}$, the output of the where-part at time step t.

The where-part contains two sections, similar to those in the what-part. First it has an internal feedback AHMM, which outputs a sequence of where-symbols O_t. In this model the where-symbols are intended to be large-position movement symbols. Secondly it has a "where-to-what" mapping which determines for each where-symbol O_t the location in the current image it corresponds to, and outputs the set of feature vectors $\mathbf{P_t}$ in that local area of the image. The AHMM in the where-part uses as feedback a sequence of sets of where-symbols $\mathbf{G_t}$, which is the output of the what-part.

Finally, the output combiner determines the overall output of the what-where-AHMM. The overall output at time step t is Z_t, a where-symbol (*i.e.* an eye movement command), selected as the element of the set $\mathbf{G_t}$ that has the smallest distance to the symbol O_t.

Operation of the what-where-AHMM is as follows. At each time step, each of the two parts produces a set of feedback symbols that reflects its own preference for action. Each then updates its own preferences taking the other's into account, and then generates its own final preference for action at that time step. The set of final preferences is reduced to a single output symbol

by the output combiner.

Incorporating high-resolution (foveal) feature vectors. So far, the what-where-AHMM has used only peripheral image data so its feature vectors (what-symbols) should be considered to be low-resolution feature vectors. After each eye movement, new fovea data is available to compute a high-resolution feature vector, essentially a verification of what the low-resolution feature vector suggested might be at that location. A negative verification might be used to modify further the AHMM, for example, to move to a new location containing one of the other instances of the same feature vector.

3.6 Experiments

The AHMM eye movement models have been implemented using the Rochester Head and its associated image processing hardware [Brown, 1988]. Computer control of the two cameras on the Head permits individual camera pans, a shared tilt, and either smooth path or saccadic movements. See [Rimey and Brown, 1990] for a complete description of our experiments.

Figure 4 shows a Lab scene typical of those used in the experiments — a table top with a variety of objects. All figures here contain both peripheral and foveal components: The majority of a figure is a low resolution peripheral image (128x128, zoomed 4x), while the center contains the high resolution foveal image (also 128x128). For visual cues in these experiments we used a simple saliency image that was easy to implement, the equally weighted sum of five features derived from the Sobel edge operator and the grayscale variance. The graphics superimposed on all figures illustrate the points which would be fixated if the cameras were to execute a movement sequence. Generation of an oblivious or peripheral modified sequence does not require camera movement, whereas a foveal modified sequence does require it.

External feedback AHMM, incremental- position movement. Figure 5 shows the graph structure of the AHMM used. A mouse was used to create a training set containing 30 sequences. An AHMM was trained on that set and used to generate the oblivious path shown in white in Figure 4. Such a path might correspond to knowing the desired object is normally kept on the left side of the desk. Peripheral image data modified the oblivious path, resulting in the path shown in gray. Here the peripheral data keeps the path from overshooting the stuffed animals. However, later it does not effectively pull the path closer to either the soda cans or the pile of small parts, as would be preferred. Finally, the AHMM was run also using foveal data modification, obtaining the path shown in black.

Figure 4: Incremental-position movement sequence. Parameters: $\alpha = 0.3$ and $\beta = 0.4$. Oblivious path (white), peripheral (gray) and foveal (black) modified paths.

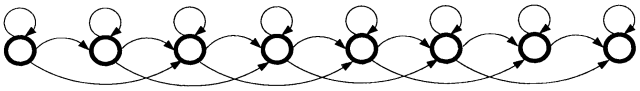


Figure 5: AHMM graph for incremental-position movement experiment.

The foveal saliency data attracts the latter half of the path to the small pile of parts. The two modified paths in these results were produced with the peripheral gain $\alpha = 0.3$ and the foveal gain $\beta = 0.4$ (equations (2)-(3)). Experiments with larger and smaller values for α and β have verified that the model can provide more and less aggressive path modification.

External feedback AHMM, large- position movement. The large-position movement experiments use AHMM symbols that are (coarsely quantized) absolute retina positions, (x, y). Figure 6 shows the AHMM graph structure used. An algorithm similar to that in [Clark and Ferrier, 1988] was used to generate 20 training examples, thus showing how the AHMM can learn the emergent behavior generated by some other algorithm. In this case, the other algorithm iteratively fixated the point in the image with a maximum saliency value, zeroed out the local saliency

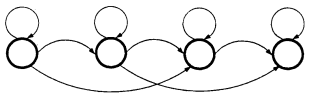


Figure 6: AHMM graph for large-position movement experiment.

Figure 7: Large-position movement sequence generated using AHMM which was trained on examples produced by "other" algorithm. Oblivious path (white) and periphery modified path (gray).

around that maximum, and then went to the next largest maximum, until 5 fixations were made. Figure 7 shows an oblivious sequence (white) generated by the trained AHMM, and the sequence modified using peripheral salience (gray). Note how the modified sequence has been drawn to the locally more interesting areas of the image.

Internal feedback AHMM, incremental- position movement. The internal feedback AHMM has been investigated with experiments parallel to the incremental-position experiments for the external feedback AHMM.[1] The same AHMM graph structure (Figure 5) was used. The same trained AHMM parameters were also used, except that any zero valued probabilities were changed to have very small non-zero values. These AHMM parameters served as the initial (fixed in time) version of the model. The saliency image used in these experiments was simply the Sobel edge magnitude image.

[1] These experiments are still in progress.

Figure 8: Incremental-position movement sequence, small foveal feedback gain. Gains are: $r_{f,(p)} = s_{f,(p)} = 0$, $r_{f,(f)} = s_{f,(f)} = 0.2$, $r_d = s_d = 0.9$.

Figures 8-10 show the result of introducing varying amounts of foveal feedback into the model.[2] In these experiments the gain value of $r_{f,(f)}$ was set equal to $s_{f,(f)}$ and varied over the values 0.2, 0.5 and 0.8. The 0.2 value results in a path almost identical to the original, while the 0.5 value results in a path that begins to be drawn more towards the nearby objects in the scene (the soda cans) – a reasonable behavior, although perhaps not the best one. A value of 0.8 results in even stronger modifications, producing a path that is immediately drawn to the detergent boxes, however the effect of the AHMM's trained behavior eventually gains control and the path resumes its course to the lower left towards the stuffed animals.

4 Conclusions

Summary. We are pursuing a long-range research program with the goal of isolating, clearly defining, and studying selective attention as a problem area in its own right. In this paper we assume a spatially vari-

[2] These sequences are different than Figure 4 only because a different random number sequence and a slightly different scene was used when each set of these experiments was performed. The same random number sequence was used for each of the experiments shown in Figures 8-10.

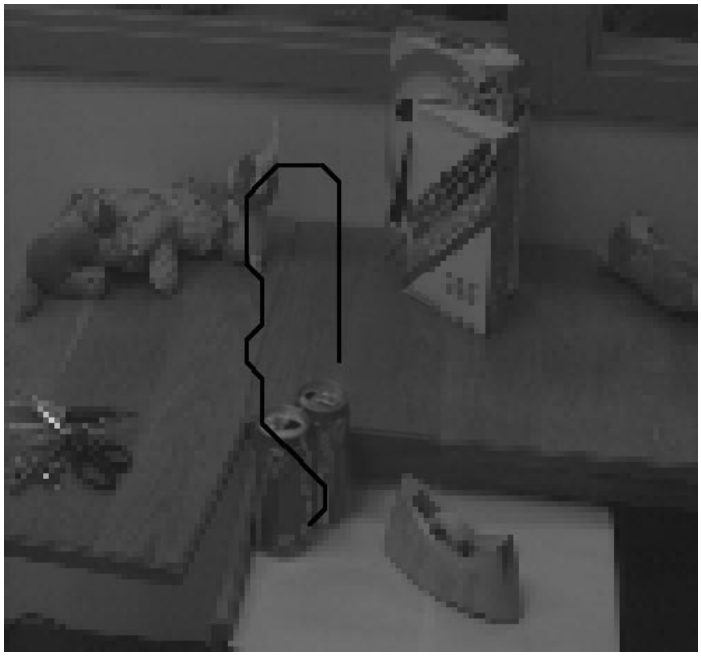


Figure 9: Incremental-position movement sequence, medium foveal feedback gain. Gains are: $r_{f,(p)} = s_{f,(p)} = 0$, $r_{f,(f)} = s_{f,(f)} = 0.5$, $r_d = s_d = 0.9$.

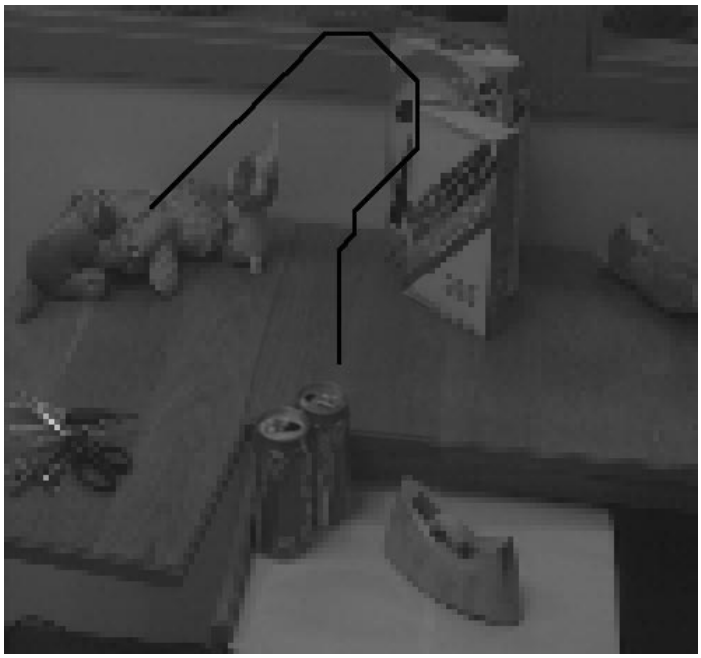


Figure 10: Incremental-position movement sequence, large foveal feedback gain. Gains are: $r_{f,(p)} = s_{f,(p)} = 0$, $r_{f,(f)} = s_{f,(f)} = 0.8$, $r_d = s_d = 0.9$.

ant sensor, such as a fovea and periphery, and study some aspects of attentional sequences. An augmented hidden Markov model (AHMM) is presented as a way to model explicit eye movement sequences while incorporating feedback from visual cues. AHMMs can deal with sequences of locations (where) or of object characteristics (what) or even both (dual what/where). A more detailed presentation of the theoretical and experimental results reported here can be found in [Rimey and Brown, 1990].

We conclude that AHMMs indeed are a promising way to acquire, represent, generate, and use probabilistic sequences for computer recognition. The AHMM is not a replacement for high- or low- level approaches to computing where or what to look at next. It provides a mechanism like a visual skill, for remembering how to allocate the visual sensor, but it is only one of several mechanisms (each with limited uses) that an attentive vision system might contain.

HMMs in computer vision. The HMM is a fairly general yet quite simple model and as such deserves consideration and investigation in areas other than speech understanding. To help illustrate the HMM's usefulness this section briefly mentions a few other problems to which it can be applied.

Classification of certain time-varying patterns is another application, *e.g.* recognizing non-rigid objects through their motion or temporal-texture characteristics. Another capability is object classification from view sequences. Here a view is characterized by a feature vector, the viewpoint can be rotated around the object, and the HMM can essentially (and compactly) learn the *topology* of features over a subset of the sphere of viewpoints (*i.e.* the object's feature-aspect graph). Finally, the HMM can be viewed as a trainable finite state machine, which may be a useful characterization for those AI researchers who are hand-crafting finite state machine variants.

Directions for future work. Our work on the AHMM has helped us to concentrate on the concept of attentional sequences. Our intention is to pursue similar efforts to learn more about other concepts important to an attentive computer vision system. Some related concepts we are considering are: visual masking, perceptual grouping, figure-ground separation, and the role of structure in vision. We are also interested in studying: Bayesian, decision theoretic frameworks, incremental and deictic representations, and real-time scheduling of tasks that can only compute partial results. Our current plan is (1) to continue developing and experimenting with the what-where-AHMM, and (2) to start a new thrust, studying attention as allo-

cation and scheduling.

Acknowledgements

We thank the entire Rochester vision group for comments on a early presentation of this work. Dana Ballard provided the key inspiration that led to the what-where-AHMM. He, as well as David Coombs, Steve Whitehead, and Mike Swain provided valuable comments on the written report.

References

[Abbott and Ahuja, 1989] A. L. Abbott and N. Ahuja. Surface reconstruction by dynamic integration of focus, camera vergence, and stereo. In *Proceedings: International Conference on Computer Vision*, pages 532–543, 1989.

[Bajcsy, 1988] R. Bajcsy. Active perception. *IEEE Proceedings*, 76(8):996–1005, 1988.

[Ballard and Ozcandarli, 1988] D. H. Ballard and A. Ozcandarli. Eye movements and visual cognition: Kinetic depth. In *Proceedings: International Conference on Computer Vision*, 1988.

[Ballard, 1989] D. H. Ballard. Reference frames for animate vision. In *Proceedings: International Joint Conference on Artificial Intelligence*, pages 1635–1641, 1989.

[Bolle *et al.*, 1990] 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.

[Brown, 1988] C. M. Brown. The Rochester robot. Technical Report 257, Department of Computer Science, University of Rochester, August 1988.

[Browse and Rodrigues, 1988] R. A. Browse and M. G. Rodrigues. Propagation of interpretations based on graded resolution input. In *Proceedings: International Conference on Computer Vision*, pages 405–410, 1988.

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

[Clark and Ferrier, 1988] J. J. Clark and N. J. Ferrier. Modal control of an attentive vision system. In *Proceedings: International Conference on Computer Vision*, pages 514–523, 1988.

[Coombs, 1989] D. J. Coombs. Tracking objects with eye movements. In *Proceedings: OSA Topical Meeting on Image Understanding and Machine Vision*, 1989.

[Didday and Arbib, 1975] R. L. Didday and M. A. Arbib. Eye movements and visual perception: A two visual system model. *International Journal Man-Machine Studies*, 7:547–569, 1975.

[Humphreys and Bruce, 1989] G. W. Humphreys and V. Bruce. *Visual Cognition: Computational, Experimental, and Neuropsychological Perspectives*. Lawrence Erlbaum, 1989.

[Noton and Stark, 1971] D. Noton and L. Stark. Eye movements and visual perception. *Scientific American*, 224(6):34–43, 1971.

[Poggio and Hurlbert, 1985] T. Poggio and A. Hurlbert. Spotlight on attention. *Trends in Neuroscience*, pages 309–311, 1985.

[Posner and Presti, 1987] M. I. Posner and D. E. Presti. Selective attention and cognitive control. *Trends in Neuroscience*, 10:13–17, 1987.

[Rabiner and Juang, 1986] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 1986.

[Rabiner, 1989] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE Proceedings*, 77(2):257–286, 1989.

[Rimey and Brown, 1990] R. D. Rimey and C. M. Brown. Selective attention as sequential behavior: Modeling eye movements with an augmented hidden Markov model. Technical Report 327 (revised), Department of Computer Science, University of Rochester, April 1990.

[Stark and Ellis, 1981] L. Stark and S. R. Ellis. Scanpaths revisited: Cognitive models direct active looking. In D. F. Fisher, R. A. Monty, and J. W. Senders, editors, *Eye Movements: Cognition and Visual Perception*. Lawrence Erlbaum, 1981.

[Tistarelli and Sandini, 1990] M. Tistarelli and G. Sandini. Robot navigation using an anthropomorphic sensor. In *Proceedings: IEEE International Conference on Robotics and Automation*, 1990.

[Treisman and Gelade, 1980] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[Treisman, 1985] A. Treisman. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31:156–177, 1985.

[Tsotsos, 1987] J. Tsotsos. A 'complexity level' analysis of vision. In *Proceedings: International Conference on Computer Vision*, 1987.

[Ullman, 1984] S. Ullman. Visual routines. *Cognition*, 18:97–157, 1984.

[Wright, 1990] C. E. Wright. Controlling sequential motor activity. In D. N. Osherson, S. M. Kosslyn, and J. M. Hollerbach, editors, *An Invitation to Cognitive Science, Volume 2, Visual Cognition and Action*, pages 285–316. MIT Press, 1990.

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

[Yeshurun and Schwartz, 1989]
Y. Yeshurun and E. L. Schwartz. Shape description with a space-variant sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1217–1222, 1989.