Animate vision*

Dana H. Ballard

Computer Science Department, University of Rochester, Rochester, NY 14627, USA

Received March 1990
Revised June 1990

Abstract


Animate vision systems have gaze control mechanisms that can actively position the camera coordinate system in response to physical stimuli. Compared to passive systems, animate systems show that visual computation can be vastly less expensive when considered in the larger context of behavior. The most important visual behavior is the ability to control the direction of gaze. This allows the use of very low resolution imaging that has a high virtual resolution. Using such a system in a controlled way provides additional constraints that dramatically simplify the computations of early vision. Another important behavior is the way the environment "behaves". Animate systems under real-time constraints can further reduce their computational burden by using environmental cues that are perspicuous in the local context. A third source of economy is introduced when behaviors are learned. Because errors are rarely fatal, systems using learning algorithms can amortize computational cost over extended periods. Further economies can be achieved when the learning system uses indexical reference, which is a form of dynamic variable binding. Animate vision is a natural way of implementing this dynamic binding.

1. What is vision for?

We are accustomed to thinking of the task of vision as being the construction of a detailed representation of the physical world. Furthermore, this constructive process is regarded as being independent of larger tasks. From the Encyclopedia of Artificial Intelligence: “the goal of an image understanding system is to transform two dimensional data into a description of the three dimensional spatiotemporal world” and such a system “must infer 3-D surfaces, volumes, boundaries, shadows, occlusion, depth, color, motion” [58, p. 389]. However, a paradigm that we term animate vision1 argues that vision is

* Revised version of the paper that won the Artificial Intelligence Journal Best Paper Award at IJCAI-89, Detroit, MI.
This research was supported by NSF Grant No. DCR-8602958 and NIH Grant No. RO1 NS22407-01.
1 Why pick the term animate vision when there already is the notion of active vision? One problem with active vision is that it is readily confused with active sensing, which has been used for laser rangefinders, etc. Also it has been associated with multi-modal fusion [2] regardless of goals.
more readily understood in the context of the visual behaviors that the system is engaged in, and that these behaviors may not require elaborate categorical representations of the 3-D world. Animate visual systems have anthropomorphic features such as binocularity, focus, and most importantly high speed gaze control. While it is possible to build many different kinds of visual systems, such as those that have more than two cameras or use active sensing, what we are calling animate vision is directed towards specific computational advantages of having anthropomorphic features. The main purpose of this paper is to summarize these computational advantages.

Throughout the paper we stress that whatever models are produced must function in real time. As a research strategy, we shun general-purpose algorithms if they must appeal to vast increases in computing power in order to be practical. Instead our method is to look at ways to increase computational speed that exploit additional constraints introduced when the animate system is allowed to interact with its environment.

The goal of animate vision is the use of vision in behaviors associated with intelligence, and as such it has its roots in theories of robot behaviors. Brooks has argued for behaviors that do not require internal representations in a larger context [12, 13], and others have demonstrated the importance of active vision systems that integrate vision with behavior (Moravec [38], Bajcsy and Allen [1], Chen and Kak [17]) as well as demonstrating the advantages of knowing camera motions (Alimimous et al. [2]). Ullman has emphasized the use of task-directed programs that operate on the optic array [62]. Animate vision also has its roots in the study of vision of the lower animals. From studies of the frog, Aribb [3] has long been stressing the integral role of vision in behavior as a perception-action cycle. Many of the technical features of insect vision can be used by animate vision systems and some of these have recently been realized by Nelson [40]. However, our primary purpose is to develop the advantages of animate vision that are geared towards hand-eye coordination behaviors. (Although this paper heavily emphasizes the role of the visual system and treats the hand only to the extent needed to explore some interactions.)

To start to see how animate vision might be qualitatively different from passive vision, let us examine the structure and function of eye movements in the human visual system. The human eye is distinguished from current electronic cameras by virtue of having much better resolution in a small region near the optical axis. This region is termed the fovea, and has a diameter approximately one to two degrees of visual angle. Over this region the resolution is better by an order of magnitude than that in the periphery. One feature of this design is the simultaneous representation of a large field of view and local high acuity. Figure 1 from a study by Sandini and Tagliasco [57], shows graphically the kind of gains that can be achieved.

Figure 1 visually understates the situation for the human system, where the fovea is less than 0.01% of the visual field area! With the small fovea at a premium in a large visual field, it is not surprising that the human visual system has special behaviors (saccades) for quickly moving the fovea to different spatial targets [42]. The first systematic study of saccadic eye movements in the context of behavior was done by Yarbus [68]. A selection of his data are shown in Fig. 2. Subjects were given specific tasks pertaining to a familiar picture. The figure shows the traces for three minutes of viewing as a subject attempts to solve different tasks: (a) give the ages of the people; (b) surmise what the family had been doing before the arrival of the "unexpected visitor"; and (c) remember the position of the people and the objects in the room. This data shows what has been confirmed by several other studies: Subjects use scanning patterns that are highly sensitive to the particular task at hand [43-45]. Of the traces in Fig. 2, the last is most remarkable, since it is so similar to the task of so many computer vision programs: we conjecture that since the eye movement traces show a specialized signature for this task as well, it is not done routinely. Instead, the overall impression of these traces is that the visual system is used to subservc problem-solving behaviors and such behaviors often do not require an accurate model of the world in the traditional sense of remembering positions of people and objects in a room.

The above data on the fovea and saccades hint also at how dynamic a process visual behavior must be. Saccades at the rate of three per second are routine in visual problem solving. Furthermore most of the brain structures that represent visual information are retinally indexed. This means that their state is changed with each eye movement. This raises a technical puzzle for human visual perception: How can the world appear to be stable when the data collecting process is so dynamic? We believe that this is a profound question with a surprising answer: The visual system provides the illusion of three-
dimensional stability by virtue of being able to execute fast behaviors. This point may be very difficult as it is so counter-intuitive, but it has been arrived at in different forms by many different researchers. For example, Rosenschein has stressed the importance of implicit knowledge representation by a behaving "situated automation" [54, 55]. This may have been the point of Gibson's "affordances" [24]. O'Regan and Lévy-Schoen emphasize the use of the world as a "memory buffer" that can be accessed by visual behaviors [48]. Dickmann's self-driven car makes extensive use of a dynamic model of the roadway [29]. At any rate, having a particular embodiment forces one to deal with performance issues: One has to act in a timely manner under resource constraints. One way to do this would be to have an elaborate internal representation as a form of "table look-up." But in a dynamic world, the cost of maintaining the correspondence between the representation and the world becomes prohibitive. For this reason animate vision systems may have to travel light and depend on highly adaptive behaviors that can quickly discover how to use current context.

2. The animate vision paradigm

The central asset of animate vision is gaze control. Gaze control is the collection of different mechanisms for keeping the fovea over a given spatial target. The single most distinguishing feature of the human visual system is its high-speed gaze control mechanisms. As animals, we move in relatively fixed environments, but we also have to deal with other moving objects, animate and inanimate. Although we must function in the presence of different kinds of motion, our visual system works best when the imaged part of the world does not move. However, for a variety of behaviors, such as running after moving objects and hand-eye coordination, the complete visual field cannot be stabilized. Instead, stabilization can be achieved for a region near a point in the world near the optical axes that commands the viewer's gaze. That point is termed the point of fixation and is defined by the intersection of the two optical axes.

Gaze control mechanisms fundamentally change computational models of vision. Without them the visual system must work in isolation, with the burden of solving difficult problems with many degrees of freedom. With them a new paradigm emerges in which the visual calculations are embedded in a sensory-motor behavioral repertoire. Rather than thinking of visual processing as separate from cognitive or motor processing, they are interlinked in terms of integral behaviors. These behaviors need not always be successful but they must be timely: Some competence may be sacrificed for timely performance. This viewpoint has many different kinds of advantages.

1 Here we are neglecting the very small motions of the eye [42, p. 95] as unimportant in a behavioral context.
1. **Animate vision systems can use physical search.** The system can move the cameras in order to get closer to objects, change focus, or change the point of view [29, 49, 65]. Often this visual search is more effective and less costly than algorithmic search on a single image, which may not even have the desired object in its field of view [41].

2. **Animate vision can make (approximately) known camera movements.** Since these movements are self-generated, they provide additional constraints on the imaging process [2]. This facilitates the computational process dramatically: properties that are difficult to compute with a fixed camera system are much more easily computed with a moving camera system. One of the first demonstrations of this advantage was Bandyopadhyay’s computation of rigid body motion parameters [8].

3. **Animate vision can use exocentric coordinate frames.** The ability to control the camera’s gaze, particularly the ability to fixate targets in the world while in motion, allows a robot to choose external coordinate frames that are attached to points in the world (see Fig. 3). Behaviors based on fixation point relative coordinates allow visual computations to be done with less precision.

![Diagram](image)

Fig. 3 Much previous work in computational vision has assumed that the vision system is passive and computations are performed in a viewer-centered frame (A). Instead, biological and psychophysical data argue for a world-centered frame (B). This frame is selected by the observer to suit information-gathering goals and is centered at the fixation point. The task of the observer is to relate information in the fixation-point frame to object-centered frames (C).

4. **Animate vision can use relative (or qualitative) algorithms.** The fixation point reference frame allows visuo-motor control strategies that servo relative to that frame. These are much simpler than strategies that use egocentric coordinates.

5. **Gaze control can segment areas of interest in the image precategorically.** That is, one can isolate candidate visual features without first associating them with models using the degrees of freedom of the gaze control mechanisms. For example, one can use the blurring introduced by self-motion while fixating to isolate the region around the point of fixation [16]. Similarly, one can use regions of near zero disparity produced by a binocular vergence system.

6. **Animate systems can exploit environmental context.** Gaze control leads naturally to the use of object-centered coordinate systems as the basis for spatial memory. Object-centered coordinates have a great advantage over egocentric coordinates in that they are invariant with respect to observer motion. Keeping track of relations between object-centered frames allows for simplified object location strategies.

7. **Animate vision is tailored for learning algorithms that use indexical reference.** Gaze control with a high resolution fovea to isolate visual features is tailor-made for systems that use indexical reference [1, 64]. Such systems provide a controlled access to the environment, making it much easier to access stored plans. Furthermore, such systems are tailor-made for reinforcement learning algorithms. The vast reduction in the state space provided by indexical reference makes the use of such brute force learning algorithms possible. In turn, learning algorithms allow visual behaviors to learn just those features that are useful for solving the problem in very specific contexts. This leads to further computational economies.

3. **The fixation frame**

One of the most central aspects of animate vision is the use of an exocentric coordinate frame termed the frame of fixation. This frame provides direct access to information from a small region near the fixated point. Of particular importance is the information associated with early vision [33]. Early vision builds retinotopically indexed maps of important environmental features such as depth, color, and velocity. Despite extant work in this area over the past decade, the construction of such maps with computational models has proven to be very difficult. A primary reason for this may have been that the assumption of a passive vision system. In an animate vision system, the degrees of freedom of the cameras are under the control of the animal. Akimnon et al. [2] show in a general way how such assumptions can stabilize the computation of those
3.1 Using the fixation frame

To illustrate the advantages of using the fixation frame, we developed a computational model of motion parallax. Motion parallax, or kinetic depth, is the sensation of depth obtained by moving the head while fixating an environmental point in a static scene. If the observer has little forward motion, objects in front of the fixation point appear to move in the opposite direction to the motion while objects behind the fixation point move in the same direction. (For a more general analysis that includes forward motion, see [52].) The apparent velocity is proportional to the distance from the fixation point [19]. Under these conditions it is easy to compute scaled depth (depth/fixation depth), which is a monotonous function of spatial and temporal derivatives of the image intensity function and has a zero value at the fixation point. By implementing this strategy on our robot we verified that a depth estimate can be obtained in real time over a 400 × 400 pixel image without iteration [7]. This result shows that the early vision computations of animate vision, at least in the case of kinetic depth, are considerably simpler than fixed camera vision, as first noted by Aloimonos et al. [2]. Table 1 compares the two paradigms.

3.2 Gaze control

The small size of the fovea, together with the rapid movements humans can make, places a premium on gaze stabilization mechanisms. Perhaps for this reason a number of separate mechanisms for human gaze control have evolved. As Table 2 shows, the eye movement system has a number of different systems that function to control gaze under different circumstances. In addition there is the accommodation system that acts to focus the lens. We argue that the ability to control gaze can greatly simplify the computations of early vision, but what of the complexity of gaze control itself? If that should turn out to be prohibitively difficult it would negate the value of this paradigm. Fortunately, all our experimental work to date argues that this will not be the case [6], as does work by Clark and Ferrier [18]. Figure 4 shows our animate vision system. Currently we use a “dominant eye” control protocol whereby the dominant camera controls the system pitch and its own yaw coordinate using a simple correlation tracking scheme [16]. The non-dominant camera uses a novel vergence correction algorithm [47] based on the cepstral filter [69] to correct its own yaw error. Brown [14, 15, 53] has recently shown how these and other components can work together synergistically. These components run in real time. At the moment there are many differences with a reasonable human model, but the performance is sufficiently good to allow us to explore vision while fixating in real time. Details may be found in [16].
relative result is easier to obtain, it motivates the question as to whether other visual behaviors might in fact use relative vision. In fact many examples can be found that suggest that relative quantities are used and that their computation is simpler. For example, many psychophysical tests suggest that the way the image is interpreted depends on occlusion cues such as shown in Fig. 5 [39]. It is not easy to make such judgements from an arbitrary viewing position, as would be required by a viewer-centered hypothesis. The kinetic depth result suggests that the notion of a fixation point may be implicit behind the analysis even though we might not be aware of it. Our perceptual system is structured to make accurate judgements relative to an object-centered frame at the fixation depth. Simplistically, imagine that one keeps two maps: one for structures that are judged to be in front of or at the fixation depth, and one for structures that are behind the fixation depth. The different interpolation rules can be fixed for each map. This structure is much simpler than that which would be required for viewer-centered maps. Such maps would have to be able to make corrections based on comparisons of all possible pairs of depth values.

The notion that the computational results of early vision are intrinsically relative can be challenged by obvious counter-examples. We can reach our

Fig. 4. The University of Rochester's animate vision system. The "robot-head" has three motors and two CCD high-resolution television cameras providing input to a DataCube MaxVideo image processing system. One motor controls pitch of the two eye platform, and separate motors control each camera's yaw. The motors have a resolution of 2,500 positions per revolution and a maximum speed of 800 second. The robot arm, a Unimation 302, has a workspace consisting of most of the volume of a sphere with a two-meter radius, and a top speed of about one meter per second. The first such system, built at the University of Pennsylvania by Bishop [4], demonstrated the potential for vision with controlled cameras. It had vergence and accommodation and zoom control. The main drawbacks were its slow speed and limited workspace.

The importance of vergence in gaze control is dramatically demonstrated by Olson and Potter [17]. Without vergence, very large disparities on the order of half the image dimension can be obtained. These pose difficulties for algorithms that use stereo to build depth maps. With vergence, the disparities for the objects of interest can be kept small. In fact, most models of human stereopsis postulate a fusional system that brings the disparities within the range of a detailed correspondence process [21, 33, 69].

3.3. Relative vision

The kinetic depth computation naturally produces a relative result; for absolute depth the calculations must include direction of gaze. Since the

Fig. 5. (a) Ken Nakayama's [39] illusion of subjective contours using stereo (not to scale). When fixated, if the relative disparities are such that the triangle is in front of the circles, subjective contours are seen; if behind, then they are not. (b) The letter "A" is easier to see if its components result from real occluding boundaries. This can be explained if the occluders can invoke a fixation depth that is in front of the plane of the A's components.
D. H. Ballard

1. Behavioral movement

2. Behavioral motion

3. Behavioral cues

4. Visual behaviors

A feature of the kinetic depth result is that it is an integral part of a visual behavior. When looking at a moving target, this interpretation is no longer valid. It is the case that the kinetic depth result is an isolated case where behavior makes a large difference in the comprehension of the world. However, in the case of vision, some very minor changes in the environment show that there are many examples where vision simplifies the sense of motion (the behavior of the animal system). Table 2 summarizes some of these results.
initialized, no communication between them except indirectly through their effects on the robot and the environment. Our preliminary experiments have revealed a remarkable level of coherent behavior using this strategy [67].

4.1. Quickly computable features

The real-time stress of animate vision requires that the kinds of visual cues used are easily computable. In a human system, the short fixation times are about 0.25 seconds and cortical neurons typically fire at rates of 10 spikes per second, leaving 2.5 spikes per fixation. In a sequential computer system using a 500 × 500 pixel image at video frame rates the demand is also great as the system must compute at roughly 10^7 pixels/second. Algorithms that require 10^8 instructions per pixel are common, leading to a demand of 10^10 instructions/second.

One feature that is easily computed is color. Color has been neglected recently as a useful cue, although it has been used in earlier work (Feldman and Yakimowsky [22], Garvey [23], Beveridge et al. [9]). One reason for this neglect may have been the lack of good algorithms for color constancy. However, recently there has been great progress in correcting for both the chromaticity of the illuminant [32, 56] and for geometric effects such as specularities [27]. Another reason that color may not have been so successful is that it has been associated with a Mondrian-like view: one color per object. But many objects are multi-colored and this fact can prove very useful, as will be shown in the next section. A third reason for the neglect of color may be that it is not intrinsically related to the object's identity in the way that other cues, e.g., form, are. This view is well represented by Biederman [10]:

> Surface characteristics such as color and texture will typically have only secondary roles in primal access... we may know that a chair has a particular color and texture simultaneously with its volumetric description, but it is only the volumetric description that provides efficient access to the representation of CHAIR

but it is easily challenged. There are many examples from nature where color is used by animals and plants to send clear messages of enticement or warning. The manufacturing sector uses color extensively in packaging to market goods (e.g., Kodak). Animate vision systems can also use representations that are heavily personalized to achieve efficient behaviors, and color is an important feature for such representations. For example, it may not be helpful to model coffee cups as being red and white, but mine is, and that color combination is very useful in locating it. Another obvious example is commercial food packaging. We can readily describe the color of food packages for the kind of eggs and milk we buy even though these colors do not generalize: they will not work for another supermarket chain.

* Parallel architectures help but not with infeasible algorithms [61].
In summary, there have been various reasons for not using color, but most of these are now less compelling, particularly in the light of recent technical advances in color constancy and in reconsideration of the behavioral context in which color can be used. More importantly, color has two very important properties that make it a useful feature. Given that reasonable color constancy can be achieved, color has enormous value in vision as a cue because it is a property of individual photoreceptors. This means that if it is a very useful cue under conditions of low spatial resolution, precisely the conditions that exist in the periphery of the retina. The second useful property of color is invariance. The colors of an object typically are invariant to wide ranges in field of view and to several different kinds of occlusion.

One way to take advantage of these properties of color is to use the color histogram. Given a discrete color space, the color histogram is obtained by integrating over the image array,

\[ b(c) = \int f(r, c) \, dr \]

The color vector \( c = (r, g, b) \) obtained from the tri-chromatic receptor array can be sensitive to gross lighting changes such as the 1/f fall-off from a point source. One way to compensate for this, observed in biological systems, is to use an opponent color space \( c' = (r + g, b, r - g) \). Figure 7 shows the three chromatic channels from a color camera together with the opponent color histogram.

4.2. What/where behaviors

Returning to the challenge of Fig. 2(c), it seems that for human vision, location and identities of objects are not routinely computed. Furthermore, the second cue to face is that when this is done, the resultant computation is a sequence of eye movements. We further suggest that very different algorithms are used depending on the task of the moment. A gross distinction that can be made is between identification algorithms that analyze the fixated area during fixation and location algorithms that direct the eyes to new targets. Support for this what/where distinction, made by Mishkin [36, 37], comes from studies of human and primate brains. A major feature of the gross organization of the primate visual brain is the specialization of the temporal and parietal lobes of visual cortex [34, 36, 37]. The parietal cortex seems to be subserving the management of locations in space whereas the temporal cortex seems to be subserving the identification of objects in the case where location is not the issue. In a striking experiment by Mishkin [36], monkeys with parietal lesions fail at a task that requires using a relational cue but have no trouble performing a very similar task that requires using a pattern cue. The reverse is true for temporal lesions.

Why should the primate brain be specialized into two separate areas that are crucial for different functions? If we think generally about the problem of relating internal models to objects in the world, then one way to interpret this dichotomy is as a suggestion that the general problem of associating many models to many parts of the image simultaneously is too difficult. In order to make it computationally tractable within a single fixation, it has to be simplified, either into one of location (one internal model) or identification (one world model). Table 4 makes this suggestion more concrete.

<table>
<thead>
<tr>
<th>Table 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>The biological organization of cortex into what/where modules may have a basis in computational complexity. Trying to match a large number of image segments to a large number of models at once may be too difficult.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>One</th>
<th>Many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image parts</td>
<td>Manipulation. Trying to do something with an object whose identity and location are known.</td>
<td>Identification. Trying to identify an object whose location can be located.</td>
</tr>
<tr>
<td>Location</td>
<td>Trying to find a known object that may not be in view.</td>
<td>Too difficult!</td>
</tr>
</tbody>
</table>

Fig. 7. Top: Red, green, and blue bands of "Arm & Hammer" image. The main body is yellow, the neck containing the hammer is red, the rope at the top is green, and the letters and numbers in blue and white. Bottom: Opponent-color histogram of "Arm & Hammer" image. Red-green axes run vertically, green at the top, red at the bottom. Blue-yellow axes run horizontally, yellow at the left, blue at the right. The yellow (red) peaks and black background (the center) peaks are the largest, and red and green peaks, as well as a small blue peak, are present. From Wilson and Ballard [6].
Both of these examples use the color histogram or spectogram as a central low-cost representation. This histogram can be used in two very different ways for different behaviors. If the location of a single known multicolored object is sought, the histogram of the current scene can be matched against that of the desired object. A robot can be trained to move toward the object by using this match function as a gradient. Let $M(h_m, h_x)$ be a function that scores the match between the object histogram and scene histogram at pose $x$. For example, one possible match function is simply $\|h_m - h_x\|$. Now the robot can move in a direction of maximum $dM/dx$. This works largely because different colors superpose in the color histogram, but if the spectral resolution is sufficient, they will not mix. The match function acts as a qualitative measure to direct the search. For an initial point one cannot depend on the object being within view, but if it is potentially viewable, an animate system can conduct a coarse saccadic scan of the view space and select good candidates by applying the match function to all of these discrete views. Figure 8 shows the results of doing this for two cases of looking for brightly colored objects.

Now let us turn to the complementary task: that of identifying an object whose location is known. The object can be isolated in various ways: one uses motion under fixation to blur nearby structure [46, 59]. If the image is assumed to be obtained from a single multicolored object, the histogram can be used as a multidimensional index into a database of multicolored objects. Given the notion of a match function $M(h_m, h_i)$, it is easy to find the model in terms of the best match, i.e.,

$$\left\{ m^* | M(h_m, h_i) = \max_m M(h_m, h_i) \right\}.$$

There are ways to perform this computation that are more efficient than a linear search through all the models, and details may be found in [60]. Figure 9 shows the results of matching nineteen objects one at a time into a database of nineteen objects, but taken from different poses. Calculations by Swain show that the three-dimensional histogram has a very large capacity, given that the multicolored objects are distributed in color space.

These two examples of location and indexing are very simple when treated as separate behaviors, but would be difficult to combine into a single behavior or algorithm using many models and many image fragments (see the "too difficult" entry in Table 4). For example, trying to locate many objects simultaneously forces the different objects to compete for the peripheral resources of the animate system. Also, if many different models are placed into the model histogram $h_m$ simultaneously, the effect of the cross-product of all the different colors is potentially devastating.

4.3. Spatial memory

The previous section explored one way of managing space, and that was homing. In the location task, a color signal was used to move the robot near a colored object. The homing behavior can be extended to a path using several landmarks, but each landmark must be in view at the appropriate time [41]. If the landmarks are not in view, the animate system has to resort to some kind of exhaustive search of visual space using its physical resources. Thus homing is
the errors in the measurement system itself, which are a function of the relative positions of the robot and target object. Another reason is that such maps are very expensive in terms of size, since only a small portion of the material is relevant to tasks that require it to be identified. A third reason is the expensive updating introduced by self-motion when the entire environment undergoes relative motion.

We have argued that animate vision allows the perception of properties of the world to be related to a coordinate frame that is attached to the world by using the abilities to fixate or pursue. However, this coordinate frame is only valid for the duration of the camera fixation; some additional structure is necessary for spatial memory. Thus for a variety of other reasons we need to introduce the notion of object-centered reference frames: (1) such frames allow the memory of objects’ locations with respect to each other; (2) objects may be in motion; and (3) objects may not be in view. An elegant way of relating this coordinate frame to object-centered frames (OCFs) posits an explicit representation of transformations between OCFs and the current view. If one assumes that the model and view have primitive parts, for example, line segments, matches between these parts determine particular values of the transformation that relates the stored model to the current view [5, 25].

Figure 2 can be used to summarize the proposal for spatial memory. The current view represents similar features but with respect to a frame that is centered on the current fixation point (as opposed to the camera frame used by passive systems). For example, if the fixation point is the object-centered frame origin, the transformation will only differ by a rotation, having a translation value of zero. Spatial memory stores relationships between object-centered frames. In a computational theory of active vision, eye movements have an integral role in the storing and retrieval of spatial information in the following ways:

1. The view transform $T_v$ contains the information necessary to foveate a visible object that has been recognized.
2. Stored relationships between objects, $T_{uo}$, can be used to transfer gaze from one object to another.

In contrast, egocentric or camera-centered systems attempt to maintain the transformations $T_v$ and $T_{uo}$, which is more computationally intensive.

As noted in the introduction, the fovea is an elegant solution to the problem of simultaneously having high spatial resolution and a wide field of view given a fixed amount of imaging hardware. The price paid is that the target must be foveated. Thus small objects in a cluttered periphery can be effectively invisible. This means that directed visual search strategies must be employed to find objects. Think of car keys: to be useful, at any one time they must be kept in a familiar relationship with a large object. We think this difficulty can be minimized by having a stored model database whereby small objects are linked...
to larger objects. To illustrate this proposal, we have built a two-dimensional eye movement simulator. Figure 10 shows the results from a test simulation. The problem is to locate a cup that is initially invisible in the periphery. Knowing that the cup is on the table, we first locate the table via a Hough transform technique [5] and then use the pose information to center the gaze. In this instance, once the gaze is centered on the table, the cup is within the high resolution fovea and can be found by using the same Hough transform technique, but now with the cup as the stored model. Here again, application of a system with a high precision fovea avoids the complexity of making fine-grained measurements over the full field of view.

Early work in vision attempted to use context in object recognition [23], but this work languished with the introduction of the Marr paradigm and its focus on low level vision. Since then, object recognition work has been very reluctant to use any kinds of context, with the result that object recognition is usually considered in a vacuum. The motivation for this is that general-purpose techniques that make few assumptions about the world would be more useful than special-purpose techniques. However, the disadvantage of this minimalist position is that methods with few assumptions typically fall back on search, which can lead to impractical computational demands. Instead of this minimalist notion of generality, animate vision advocates making maximal use of all the different kinds of constraints available. These are of two principal kinds.

(1) In a human or robot, one source of information is behavioral state. Humans have a vestibular system that measures linear and angular accelerations. This provides a short-term history of movements in the environment and also a measure of gravitational force. Another source is the human proprioceptive system, which provides the kinematic state as well as muscle torques.

![Diagram of foveal vision](image)

Fig. 10: A foveal vision system is an elegant solution to the problem of high spatial resolution and a wide field of view. The price paid is that small objects on the periphery are hard to see. However, known relationships with large objects can help. In (A), the cup cannot be easily seen, but in searching for the cup, one can first look for the table (B), which in this case brings the cup near the fovea, where it can be found.

(2) A second source of information is the local context in which objects appear. Objects are dependent on the surfaces of other objects for support. For example, chairs are supported by the ground plane, and pens are usually on tables. The design of objects in terms of support relationships constrains the way in which they interact with supporting surfaces. For example, chairs and cups usually have only three degrees of freedom, while in contact with their supporting surface: one rotation and two translation. If there were a way of exploiting these constraints it should make the recognition problem computationally simpler. The constraints supplied by behavioral state necessarily interact with those supplied by local context since, as animals, we have our own support needs. Thus we can use kinematics to directly measure the orientation of a ground plane or table surface with respect to visual coordinates.

Given the goal of recovering the view transform, how can the general ideas about context help? One of the simplest constraints that can be supplied by context is the knowledge of a supporting surface, the simplest of which is a plane. The viewing transformation has six parameters in general, but for most objects, the constraint of planar support reduces the degrees of freedom to three [7]. This is because most objects have very limited ways in which they can be supported by a plane. A normal kind of coffee mug (with a handle) will have four: right side up, upside down, and two ways of lying on its side. If we look at "mug ethology", the mug spends almost all of its time in the first position. This means that to find a coffee mug on a table, an overwhelmingly good bet is that it will be in one support relation with three degrees of freedom: two translation and one rotation. Since the degrees of freedom are the same for those of a two-dimensional planar problem, one might suspect that a pose computation is possible using only the two-dimensional image as advocated by Lowe [30, 31]. In fact this is possible and the mathematical form of these constraints is developed in [66]. This use of spatial information has emphasized the WHERE task of locating known objects. Just as important, but given short emphasis here, is the use of geometric cues in the WHAT task of object identification. Interestingly enough, much recent work in identification finds ways around computing pose directly, e.g. [26, 30, 31], by using features which are relatively view invariant.

5. Coordinated behaviors

The fixation frame with its small fovea allows the animate system to simplify its access to the environment. The idea is that, at any given instant only a relatively small number of features of the external world are registered but through perceptual actions the system can actively control the features that are
registered. A consequence of this ability is that with animate vision or more generally animate perception, systems can be built which learn to operate in a complex task domain without the associated explosion in the input feature vector required to represent all the elements of the domain.

Steven Whitehead has applied reinforcement learning ideas to the study of animate vision [63, 64]. Whitehead has been studying block stacking tasks. On each trial, the system is presented with a pile of colored blocks. A pile can consist of any number of blocks and they can be arranged in any configuration. Each block is uniformly colored and can be either red, green, or blue. The system can manipulate the pile by picking and placing objects. An object can be picked up only if its top is clear, and an object can be placed on another object only if the target object’s top is clear. When the system arranges the blocks into a successful configuration, it receives a positive reward and the trial ends. A successful configuration is some predefined set of states which represents a desired outcome. For example, one simple block stacking task is for the system to learn to pick up a green block. In this case, the successful configurations consist just of those states where the system is holding a green object. The objective of the system is to learn algorithms for arranging arbitrary configurations of blocks into successful configurations.

Most reinforcement learning systems have static sensory systems. That is, the semantics of the feature vectors that describe the external state are defined a priori. Further, the input vector is defined so that each state is “sufficiently discriminable.” Unfortunately as the complexity of the task domain increases, in particular as the number of “possibly relevant” objects in the task grows, the size of the static input vector (state representation) grows very quickly even though the number of relevant objects remains small. The problem is that with a static input vector if an object may be relevant to the task then it must be represented internally.

In contrast to static systems, systems using animate vision can avoid the combinatorial explosion of absolute representations by using “indexical representations”. The basic idea behind an indexical representation is that the system shouldn’t attempt to maintain an accurate representation of every item in the universe, but instead should only register objects and aspects (features) that are relevant to the task at hand [1]. For the block stacking problem, instead of assigning an absolute symbolic name to each item in the universe, such as “BLOCK 44”, the system only registers objects (and their features) according to the functional roles they play in solving the task, such as “THE BLOCK I AM FIXATING”. Whitehead’s system uses both a fixation frame and an attention frame as shown in Fig. 11. Details may be found in [64].

Fig. 11. The active perceptual system is divided into two parts: the fixation frame and an attention frame. The information registered in the focus (or fixation point) can be actively controlled by executing perceptual and gaze control “acts”. For example, fixation the block as shown causes its features to be registered in the state vector; attending to the triangle as shown causes its features to appear in the state vector. The two degrees of freedom in the state vector that can be independently controlled correspond to “markers”. One can think of placing a special marker on an object causing its properties to appear in the appropriate place in the state vector. The system used by Whitehead is slightly more complex but still only uses twenty bits total to represent the state of the world.

Conceptual points. We contend that searching huge state spaces such as those in blocks world domains may be impossible without the incorporation of these kinds of ideas in animate vision systems. First, learning by trial and error allows the agent to amortize building a policy function over its history. Once a good policy function is learned, applying it is cheap. Second, the reinforcement learning algorithm we use has a limited attention span, so that it gives up after expending a predetermined amount of resources. This is important because (a) a real-time system has to respond in a timely manner and (b) this strategy, in the context of repeated applications, causes the agent to gradually improve its competence [11]. The third advantage of this kind of learning derives from the use of indexical representation. This allows (a) the access of items by property instead of by category, and (b) run-time indexing. Access by property is efficient in the following way. Consider the problem of hanging a picture where a nail has to be driven into a wall. We do not really need a hammer, but something that could serve as a hammer. Plan access by category forces the identification of image items, followed by a check to determine the appropriate properties. Access by properties short circuits this process. Also, the fact that these properties are determined by what is in the environment at the moment filters out the consideration of strategies that would require unavailable items.

One problem such systems will have is the well-known credit assignment problem. If the reinforcements change the problem of how to change the
However, when one examines the mechanisms of human and animal visual perception in detail, or tries to build anthropomorphic robots, it quickly becomes apparent that the way the apparatus works at this level of abstraction, e.g., the fast sequential saccade searches, is incompatible with phenomenological notions of invariance and stability. Models of the visual system that work are compartmentalized with inconsistent representations and specialized behaviors that compete for the resources of the system. In this milieu, animate vision has a huge run-time component. Vision depends on the world being sufficiently stable so that behaviors can be executed on demand. Perhaps it is this ability to conduct behaviors that make assumptions about the world that provides the illusion of stable perception. Another way to say this is that: Animate systems that can rapidly change their coupling with the real world place a premium on maintaining elaborate representations of the world. However, it may be the case that memorizing such representations is unnecessary, since they can be rapidly and incrementally computed on demand.

The ability to have behaviors that learn to adapt to the local environment will have a profound effect on the design of animate vision algorithms. The discussion on color introduced the notion of a personalized representation: that is, associating features with an object that makes the behaviors concerning it especially easy to execute. One can think of many other cases that challenge traditional notions of invariance. For example, we do not think of our coasts as being rigid objects, yet they appear to be to our visual systems while they are hanging on coat racks, and this limited invariance can be exploited. The hope is that such algorithms may be able to discover which combinations of such features work in each problem instance. It could be the case that the general assumptions that define categories are almost never as useful as the special assumptions found by adaptive algorithms.

The study of animate vision is in its infancy, but we can already project that this paradigm will extend the capabilities of all kinds of computer vision systems, but particularly those of mobile vision platforms.

Acknowledgement

I would like to thank my colleagues Christopher Brown and Randal Nelson for critiquing these ideas. I am also especially grateful to the University of Rochester team of researchers, Tim Becker, David Coombs, Nat Martin, Tom Olson, Robert Potter, Ray Rimey, Michael Swain, Dave Tilley, Steve Whitehead, Lambert Wixson, and Brian Yamaguchi, all of whom have greatly helped refine the ideas herein. The color examples are derived from research projects headed by Michael Swain and Lambert Wixson. The learning example comes from a research project headed by Steve Whitehead. Peggy Meeker is responsible for the pleasing format of the manuscript.
Research Note

Connectionist hashed associative memory

Ronald L. Greene
Department of Physics, University of New Orleans, New Orleans, LA 70148, USA

Received April 1989
Revised January 1990

Abstract


This paper proposes the use of simple connectionist networks as hashing functions for sparse associative or content addressable memory. The robustness of such networks in the presence of noisy inputs, and the property that "similar inputs lead to similar outputs" permits (in a probabilistic sense) faster-than-linear retrieval of data which best fits the input. The input may be noisy or have partially specified feature vectors. Mathematical analysis is presented for the Boolean feature case using a network with randomly selected connection strengths.

1. Associative memories

One of the most interesting and important characteristics of human intelligence is the ability to recall detailed memories even when given only a small portion of the relevant information. For example, details of an event such as a particular dinner date can be recalled by the mention of a person's name, or the name of the restaurant, or by a particular dish, or by a variety of other cues. The ability to efficiently simulate an associative or content addressable memory with ordinary computer memory has immediate applications in artificial intelligence and in the storage/retrieval of information into/from large database systems [6].

Associative memory lies at the heart of memory-based reasoning, a relatively new approach to machine intelligence [11]. In a typical application, items stored in memory are made up of a large number of features which, for example, describe an object or set the context for an action. Inquiries might give a partially specified set of features and ask for a classification of the