CHAPTER 23

Biologically Plausible Models of Place Recognition and Goal Location

D. ZIPSER

In this chapter I describe three parallel network models that deal with some elementary aspects of spatial cognition—namely, recognizing places and locating goals. The performance of these models was examined using computer simulations in which the networks guide the movements of a simulated observer. These models are constrained by both the requirements of accomplishing the tasks and by a desire to achieve some degree of biological plausibility. The requirement of biological plausibility has the advantage of generating models that can have immediate relevance to psychology and neurobiology but it has the disadvantage of limiting the range of problems that can be approached. This limitation is not one of principle, but grows out of our currently limited understanding of how computation is organized in the brain. More general problems can often be attacked when the constraint of biological plausibility is not invoked and unrestricted symbol manipulation is allowed. Before tackling biologically plausible models, it is useful to look at what has been done in this unrestricted computing context because it will give some insight into the major issues in spatial cognition.

From the realm of "pure" artificial intelligence research, for example, we have the *fuzzy map* theory (Davis, 1981; McDermott, 1980). The

---

The first third of this chapter is a modification of the article "A Computational Model of Hippocampal Place Fields" by D. Zipser, which appeared in *Behavioral Neuroscience*, 1985, 99, 1006-1018. Copyright 1985 by the American Psychological Association, Inc. Adapted with permission.

goal here was to develop a system that could deal with the spatial relations between objects so that queries about them could be answered in the context of a language understanding system. Among the questions the fuzzy map system is designed to answer are: "Am I headed towards home? How do I get from here to Carnegie Hall? What buildings are between Woolsie and Sterling? What is the nearest Chinese restaurant to here? What street is the best route from Yale to the hospital?" These questions are answered in a linguistic framework and new information is assimilated through language so no attempt was made to guide actual navigation behavior or to learn from the environment.

Clearly, the range of possible facts about the spatial relationships between objects necessary to answer questions of this kind is far too great to be explicitly stored in memory. What is needed is a representation that stores basic information so that inference can be used to determine the required facts when the system is questioned. In the fuzzy map system, each landmark or object has its own reference frame fixed rigidly to it. The combination of object and associated reference frame is called a *frob*. Frobs are connected to each other to form maps by storing *mquants*, each of which contains information about the translation, rotation, or scale factors relating a pair of frobs. Queries about spatial relations are answered by searching a data structure consisting of mquants. An important feature of the fuzzy map formulation is that it separates the components of the transformation relating frobs so that rotation, translation, and scale factors are in separate mquants. This has a number of advantages, one of the most important of which is that partial information about the spatial relations between frobs can be used. For example, two frobs connected only by a scale factor can be compared with respect to size even if their other spatial relations are unknown. Another important feature of the fuzzy map concept is its "fuzziness." The components of the transformation between frobs are not specified exactly, but are specified as ranges. This allows inexact, semiqualitative information to be incorporated into the map data structure.

Another approach to spatial processing, in large scale environments, is found in the work of Kuipers (1978, 1983) who is concerned with developing psychological models. A particularly interesting feature of Kuipers' approach is the concept of *view*, which he defines as "the sensory image received by the observer at a particular point" (Kuipers, 1983, p. 217). The critical importance of views is that they are used to recognize places. Places are important because their representation can serve as a node to which related information can be attached. The critical operation on views is comparison to determine equality. For example, by comparing the current view with a data base of stored views, it is possible to gain access to all the information associated with the current place. The concept of view is very relevant biologically because there are neurons in the brain that fire only within the context of a particular view. One of the important things that can be associated with a place is information about getting from that place to another place. Kuipers represents the paths between places as a sequence of actions and views. The idea is that if you are at a particular place, and take a certain action, it will generate a new view. This view, in turn, can be used to index a new action, which will produce a new view, and so forth, tracing out a path to the goal. This strategy almost certainly corresponds to one of those used by humans and animals.

In addition to path information, it is also possible to associate actual knowledge about the direction and distance to a goal with a place. The existence of absolute knowledge of goal location, relative to external landmarks, has been clearly demonstrated in the case of rats that are forced to swim in a tank of milky water (Morris, Garrud, Rawlins, & O'Keefe, 1982). A small platform is placed below the surface of the cloudy water so that it is invisible to the rat. The rat is then put into the tank and swims around randomly until it locates the platform. The rat climbs up on the platform, rises up, and observes the landmarks in its environment. On subsequent immersions, the rat no longer searches randomly but swims directly to the remembered position of the platform. Since the rat will swim directly to the platform from any initial position in the tank, it must be computing the location of the platform, using remembered information about the location of external landmarks.

The models described in this paper were designed to emulate the kind of goal seeking behavior seen with the rat in cloudy water. Such models require that a place, or more generally, an environmental context, can be recognized. In addition, they need a mechanism to bind goal location to place representations and a way to use this information to guide behavior. Finally, the models need a mechanism to assimilate new information so that goals can be located in unfamiliar environments. The first model discussed is concerned with place recognition and it can be quite biologically plausible because a lot is known about the neurobiology of place recognition. The subsequent models of goal location must be more speculative since much less is known about the mechanism of this activity.

Place Recognition

One of the most remarkable correlations between single neuron activity and events in the world is found in the place-field cells of the hippocampus (Becker, Walker, & Olton, 1980; O'Keefe, 1976; O'Keefe & Dostrovsky, 1971; Olton, Branch, & Best, 1978). In the rat, for example, there are single units that fire at their maximum rate only when the animal is at a particular location relative to a set of distal landmarks. These locations are called place fields. Removing too many landmarks or radically altering their spatial relations abolishes the place-field response of these units (O'Keefe, 1979; O'Keefe & Nadel, 1979). Recently, it has become possible to make accurate measurements of the location, size, and shape of place fields, and to determine how they change when the environmental landmark cues are manipulated (Muller, Kubie, & Ranck, 1983). Quantitative experiments of this kind show that the response of place-field units is determined in a systematic, rule-governed manner by the configuration of the surrounding scene. Muller, Kubie, and Ranck have demonstrated this in a particularly elegant fashion. They showed that when every feature of the environment is scaled to a different size, the relative location of place fields and their areas scale with the environment. This, as we will see later, places a nontrivial constraint on the computational mechanisms that can be used to model location in rats.

To account for these experimental observations, I have developed a computational model that relates the configuration of distal landmarks to the location, size, and shape of place fields. The input to this model is the configuration of landmarks in the environment together with the location of the observer. The output of the model represents the activity of a place-field unit in the observer's brain. The model can be used to simulate experiments on a computer, the results of which can be compared with real experiments to test the correctness of the model. For example, suppose it is known from experiment that an animal can use a set of simple objects—such as rectangular cards—as place cues, and it is also known that the size of the retinal images of these cards is used as an indicator of location. Then the model can use this information to provide quantitative predictions of how the shape and location of place fields will change when the size, tilt, or location of these cards is changed. Conversely, suppose that the way in which place fields change when the environmental cues are manipulated has been determined experimentally. Then the model can help to determine what features of the environmental cues are being measured to locate the animal.

It might seem strange to try to construct a model to do this since so much information processing takes place between the input of sensory information and the excitation of place-field units. Obviously, no theory currently feasible can account for all this processing in detail, and I make no attempt to do so. Rather, my model employs a biologically plausible architecture to compute the activity of place-field units from the configuration of distal landmarks using only the geometrical constraints inherent in the problem itself and as few specific hypotheses about brain structure and function as possible. This approach implies that even if the model proves to be a valid predictor, it will not directly provide new physiological knowledge. What it will do is give a precise description of the computations being carried out by the place-field system. Once these computations are understood, the task becomes to explain how neural networks in the brain actually implement them. I feel that trying to explain how neurons might accomplish a clearly defined computation is a more tractable task than trying to relate an animal's behavior directly to the functioning of its neurons.

The Model

The basic question that the model must address is: How do the place-field neurons know that the observer, in whose brain they reside, is at the location where they are supposed to fire? I consider this to be a pattern-recognition problem in which a stored representation of a scene is compared to a representation of the current scene. Roughly speaking, the better the match between these two representations, the closer the viewer is to the viewpoint of the stored scene. The activity of place-field units can be thought of as reflecting the closeness of this match. A matching task of this kind could be very complicated if details from the entire scene had to be considered. However, the experiments show that only a few discrete objects need to be used. This greatly simplifies the modeling problem and also makes sense since location information about only a small set of objects is sufficient to locate a viewer in the environment. An example of how several measurements combine to specify location is shown in Figure 1 where the point *P* is uniquely located by its distance from the three landmarks *a*, *b*, and *c*.

In order to see how a place-field unit might make use of this kind of information, imagine that at some time when an observer was at location *P*, in Figure 1, representations of the objects *a*, *b*, and *c* were recorded in some way in memory. Further assume, that the descriptions of each of these objects contained the value of the distance to *P*. Suppose that at some later time the observer returns to the vicinity of *P*. The observer's sensory system can now generate a set of current descriptions representing the objects, together with their current distances. If these representations, other than their distance components, are not affected too much by the viewing position, then the current representation of each object will differ from its stored representation

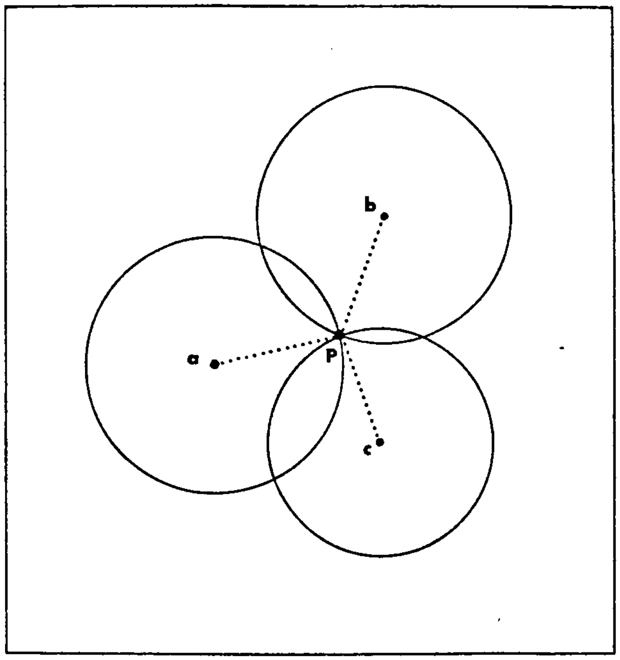


FIGURE 1. The location of point *P* is uniquely determined by its distance from the three landmarks located at *a*, *b*, and *c*. The circles are parametric curves along which the distance of an observer from a landmark remains the same. Other measures of location such as the size of the retinal image of a landmark also give parametric curves, but these are not generally circles centered at the landmark. (From "A Computational Model of Hippocampal Place Fields" by D. Zipser, 1985, *Behavioral Neuroscience*, *99*, p. 1007. Copyright 1985 by the American Psychological Association. Reprinted by permission.)

only to the degree that the object's current distance from the observer differs from its distance to *P*. A neuron whose output response is a summated measure of the similarity between these current and stored representations for each landmark will have the properties of a place-field unit.

The model deals with the problem of implementing a place-field unit that can carry out the required matching by breaking the task into two stages, each of which is carried out by a "layer" of simple processors roughly analogous to neurons or small neural networks. This arrangement is shown in Figure 2. Each unit in the first layer is specific for one landmark and one place field, and each is able to compare the current value of a location parameter with a value of the same parameter stored in the unit. The output of these units remains 0 while their landmarks are not in the current scene. When a first-layer unit's

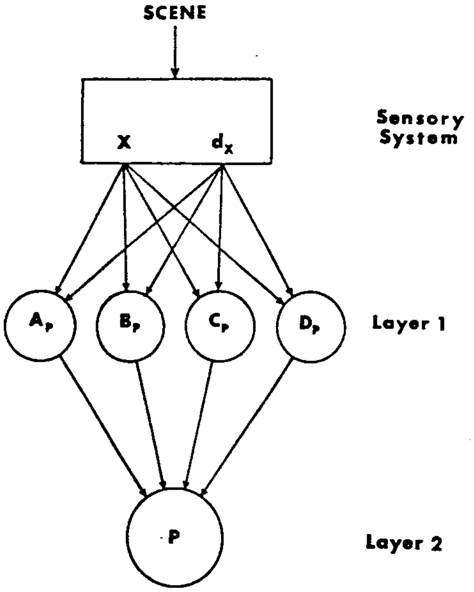


FIGURE 2. The model is shown here as implemented in a network of processors. The "sensory system" processes input from an environmental scene and produces two outputs. These are X, which is a description of a landmark in the current scene, coded so that it will be recognizable by units in Layer 1, and d_x, which is the value of a location parameter, such as distance, valid for the current position of the observer and landmark X. Each unit in Layer 1 is concerned with just one landmark and place field. When a description of this landmark appears at the output of the sensory system, the current value at the d_x output is compared with the value, d_a, stored in the unit. Thus, if the value at $X = A$, then the unit labeled A_p will generate an output by applying its response function to $d_x - d_a$. The procedure used to evaluate the model assumes that the whole scene is scanned while the observer remains still, and it also assumes that the outputs of Layer 1 units, once computed, remain fixed until the observer moves again. The Layer 2 unit integrates the outputs of the Layer 1 units to get an overall measure of how closely the current scene matches the scene as viewed from the place field centered at P. (From "A Computational Model of Hippocampal Place Fields" by D. Zipser, 1985, *Behavioral Neuroscience*, 99, p. 1008. Copyright 1985 by the American Psychological Association. Reprinted by permission.)

landmark is detected by the sensory system, its response increases as the current and stored location parameter values become closer. The response of these first-layer units reaches its maximum value when the current and stored location parameters are equal. The task of the sensory system is to detect the landmarks and generate descriptions of them that can be recognized by the processors in the first layer. The sensory system must also measure the current value of whatever parameters are used to determine location. The model, however, is not concerned with the details of how this is done. It is also not concerned with how this information is represented or transmitted to the processors in the first layer. The computations involved in the model do not require any detailed knowledge of these processes.

The output of the processors in the first layer serves as input to the unit in the second layer, which acts like a place-field neuron. Only one of these is shown in Figure 2 because the current model is only concerned with the output of a single place-field unit. The output of the second-layer unit must be a monotonically increasing function of the sum of its inputs from the first layer in order to represent the similarity between the current scene and the scene as recorded at the center of the place field. The output of this second-layer unit is used to define the location and shape of place fields in simulated experiments.

So far, the response functions for the units in each of the two layers have only been specified in general terms. In order to precisely define the model for computer simulation, some real functions must be chosen for these roles. The properties of the model are not very sensitive to the exact nature of these functions so long as they have the required general properties. This insensitivity to detail is illustrated in Figure 3 where place fields have been calculated using the two different first-layer tuning functions shown below:

$$R1 = \begin{cases} 0 & \text{if } |d_x - d_x^*| > \sigma \\ 1 & \text{otherwise} \end{cases}$$

$$R1 = \exp - \left[(d_x - d_x^*)^2 / \sigma^2 \right] \tag{1}$$

where $R1$ is the first-layer unit response function, d_x and $d_x{}^*$ are the stored and current values of a location parameter (distance from landmark X to P in the case of Figure 3), and σ is the matching criteria parameter. The second-layer unit response function is

$$R2 = \begin{cases} 0 & \text{if } \sum\limits_x R1(x) - \Theta \leqslant 0 \\ \sum\limits_x R1(x) - \Theta & \text{otherwise} \end{cases} \tag{2}$$

where $R2$ is the second-layer unit response function, x is the set of all landmarks used for the match, and Θ is the threshold parameter. The outlines of the place fields in Figure 3 indicate the curves along which the function $R2 = \Theta$. While the two place fields are not identical, the outlines and locations generated by the two different first-layer matching functions are quite similar. Since the step function and the Gaussian function used are about as different as this kind of monomodal

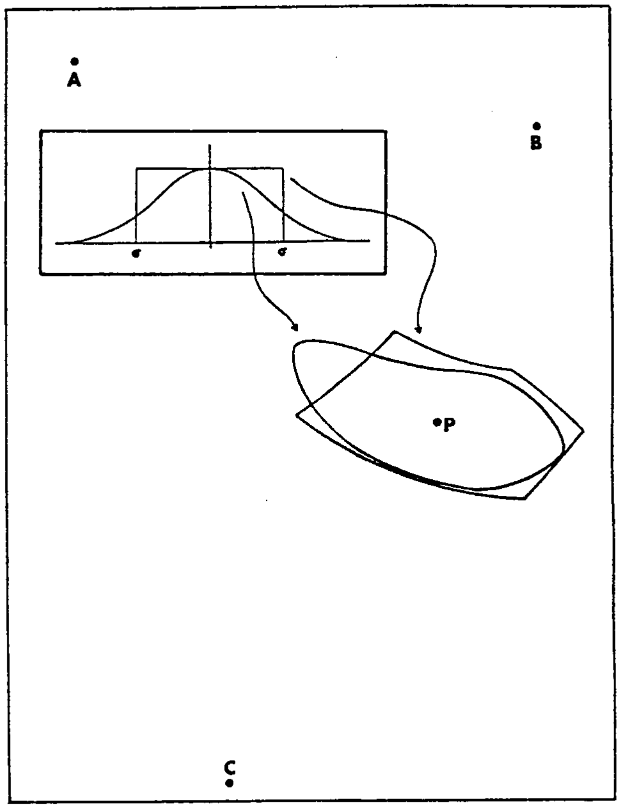


FIGURE 3. The boundaries of the place fields centered at *P* have been computed using two different Layer 1 response functions. The location parameters used were the distances to landmarks at *A*, *B*, and *C*. The two Layer 1 functions used are plotted in the insert and the arrows indicate which function generates which place-field boundary. (From "A Computational Model of Hippocampal Place Fields" by D. Zipser, 1985, *Behavioral Neuroscience*, *99*, p. 1009. Copyright 1985 by the American Psychological Association. Reprinted by permission.)

tuning function can get, the similarity in place-field shape indicates the general insensitivity of the model to the nature of these functions. While the outlines of the place fields generated are similar, the internal structure of the fields are different. The Gaussian matching function (Equation 1) gives the kind of smooth peak observed experimentally in place fields.

It is clear that what has been described is not a single model but a class of computational schemata related by the way in which they use location parameters in a two-layer architecture. The individual models in the class differ in just what features are measured to serve as location parameters and in which response functions are used in each layer.

Location Parameters

In Figure 1, distance was used as a measure of landmark location. Any other measurable feature whose value is a function of observer location can also be used. Somewhat more formally, a location parameter is a function depending on the relative location of landmark and observer that has a single value for each point in the environment. The inverse is not true. That is, a single value of a location parameter may map to many different locations. In the examples used so far, distance has been the location parameter, but in fact, distance is probably not being used by the rats in the Kubie, Muller, and Ranck (1983) experiments. This is demonstrated in Figure 4 which shows what happens when a triangle of landmarks a, b, c, is expanded by dilation to a larger triangle a', b', c'. The experimental results on environmental dilation predict that the place field should increase in size and still be centered at P. However, if the remembered distances are used, there is no longer a triple intersection at P. Actually, there is no triple intersection anywhere, which implies that there will be either no place field at all or, at best, one with a greatly *reduced* size, just the opposite of what is actually observed. This suggests that remembered values of the absolute distance to landmarks are not being used by the rat to activate place-field units in the dilation experiments.

If distance is not being used as a measure of landmark location, what is? Something is required that, like distance, defines curves in the environment whose intersections can be used to localize the viewer, but, unlike distance, scales properly with dilation. Among the measurable features that have these properties are the visual angle subtended by two points on a landmark and the retinal area of the landmark image. Angle and area differ from distance in several ways. Unlike distance, they cannot refer to single points but require landmarks with some finite extent. Also, unlike distance, angle and area can be determined by a single measurement. Distance requires either two measurements or knowledge of the absolute size of the landmark. Because the notion of "landmark" is somewhat arbitrary, measurements can often just as easily be made that reflect the relationship between landmarks. For example, the gap between two trees could make just as good a landmark as each of the trees themselves. Retinal area, it should be noted, is a very simple thing for a visual system to compute since after a figure-ground separation, all that is required is that the number of

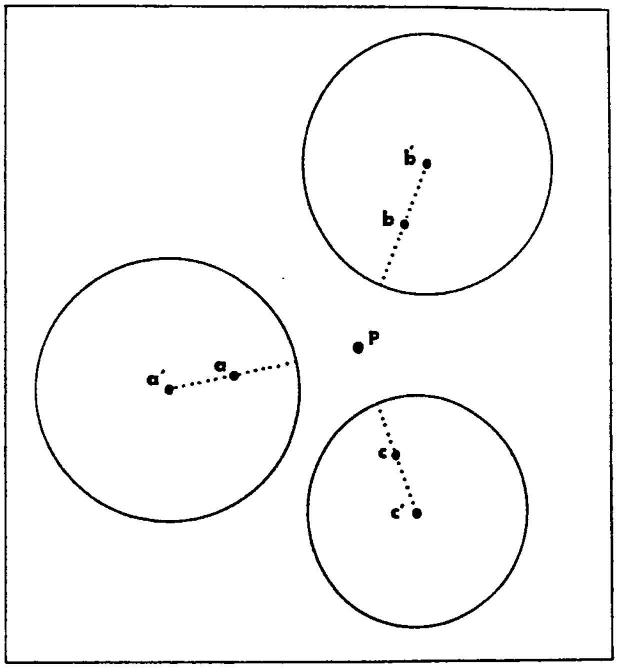


FIGURE 4. The scene from Figure 1 characterized by landmarks at *a*, *b*, and *c* has been dilated by a factor of 1.5 which has relocated the same landmarks at *a'*, *b'*, and *c'*. The circles indicate the curves along which the location parameters (the distances, indicated by the dotted lines in this case) have their undilated values. These circles no longer intersect at *P* making it unlikely that a place-field mechanism that dilates with the scene could depend on remembered distances. (From "A Computational Model of Hippocampal Place Fields" by D. Zipser, 1985, *Behavioral Neuroscience*, *99*, p. 1010. Copyright 1985 by the American Psychological Association. Reprinted by permission.)

units in the figure be totaled. Of course, other factors must be included to explain conscious perception that corrects for apparent retinal size.

Since location parameters, by definition, have a single value at each point in space, the properties of a particular parameter can be visualized by plotting the locus of a single value of the parameter. Examples of the curves generated in this way for distance, angle, and area are given in Figure 5. These are the curves that arise assuming the landmark is a square card that looks the same from both sides. Actually, most natural landmarks look different from the back, so that generally only the curves on one side of the landmark will play a role in place-field location. The shape of these location parameter curves will depend on the shape of the landmark.

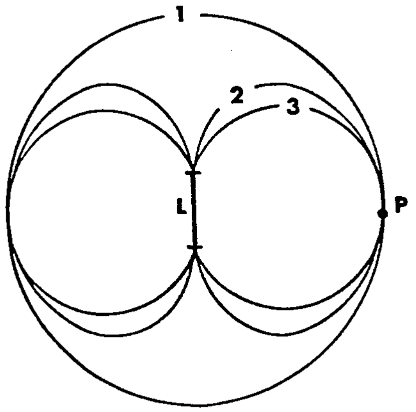


FIGURE 5. L is a landmark and P is a viewpoint. Curve 1 is a locus of all viewpoints a constant distance from the center of L. Curve 2 is a locus of all viewpoints from which a square standing vertically on L would have a constant area on a spherical retina. Curve 3 is a locus of all viewpoints where a line of length L would subtend a constant visual angle. The visual angle, A, subtended by a line of length a is given by:

$$A = 2 \arctan \sqrt{(s-b)(s-c)/s(s-a)}$$

where b and c are the distances from the endpoints of the line to the focal point and $s = \frac{1}{2}(a+b+c)$. Note that A is invariant to dilation by a factor d since replacing a, b, and c by da, db, and dc does not affect the value of the above equation. The area of any object can be approximated by the sum of the areas of a set of triangles which cover its visible surface. The retinal area of a triangle is:

$$Area = 4f^2 \arctan \sqrt{\tan(S/2)\tan((S-A)/2)\tan((S-B)/2)\tan((S-C)/2)}$$

where f is the focal length; A, B, and C, are the visual angles of the three sides of the triangle; and $S = \frac{1}{2}(A+B+C)$. Since A, B, and C are dilation invariant, so is the retinal area of the object. (From "A Computational Model of Hippocampal Place Fields" by D. Zipser, 1985, *Behavioral Neuroscience*, *99*, p. 1011. Copyright 1985 by the American Psychological Association. Reprinted by permission.)

Simulated Experiments

In order to see how well the model works, experiments of the kind used to study place fields were simulated on a computer. The simulated experimental environment, as detailed above, is shown in the computer generated display in Figure 6 (details of the simulations are given in the legend to this figure). The outer rectangle, which represents the limit of the simulated environment, contains the landmarks (shown as straight lines). The simulation assumes that the animal has a place field centered at point P. The location of the simulated observer is systematically moved over a uniform lattice of points

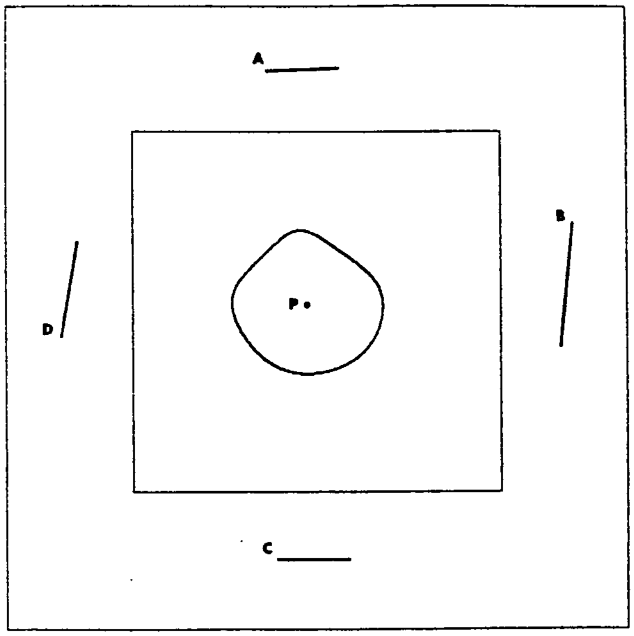


FIGURE 6. This figure shows the output from a typical simulated experiment on the screen of a Symbolics 3600 LISP machine. The experiment is initialized by first placing landmarks, (simulated square cards in this case) in the outer square. The size, position, and number of landmarks can be input explicitly as a set of numbers or determined interactively by the use of a "mouse." Once the scene is set, an initial viewpoint *P*, which will be the center of the place field, is chosen. Then values for σ and Θ are picked and the simulation is started. First, the simulation program calculates the areas of the landmarks from the viewpoint *P*. These areas are used as the stored values of the location parameter. Then the viewpoint is systematically scanned through a raster of 145,000 points in the inner rectangle. At each of these points the retinal areas of landmarks are computed and used as the current value of the location parameters in evaluating the response of the Layer 2 unit that represents the place-field unit. A mark is placed in the inner rectangle wherever the value of the Layer 2 unit just crosses threshold. The curve generated by these marks is the border of the place field. The landmarks in this figure are quite symmetrically placed around the scene that leads to a fairly symmetrically shaped place-field border. Asymmetrically placed landmarks would produce a less symmetrical field (see Figures 7 and 8). The size and, to some degree, the shape of the place field are determined by the choice of σ and Θ. See the text and Figure 7 for more about σ and Θ. (From "A Computational Model of Hippocampal Place Fields" by D. Zipser, 1985, *Behavioral Neuroscience*, 99, p. 1012. Copyright 1985 by the American Psychological Association. Reprinted by permission.)

covering the inner rectangle. The observer is limited to this inner rectangle so as to keep the landmarks distal because the model cannot deal with either proximal landmarks or the complexities of having the observer wander around in between landmarks. At each viewpoint, the activity of the place-field unit is computed using the model. In this, and all following simulations, the Gaussian matching function (Equation 1) and a threshold function (Equation 2) are used for the response functions of the first- and second-layer units respectively, and retinal area is the location parameter. The shape and location of place fields are visualized by plotting their boundaries. At each viewpoint scanned in the inner rectangle, information from all landmarks is used. That is, no account is taken of viewer orientation. The significance of this simplification will be discussed later.

More About Landmarks

In the experiments that originally demonstrated the dependence of place-field unit activity on distal landmarks, a variety of objects such as a fan, a white card, a buzzer, and a light, all arranged at 90° intervals around the experimental area, were used (O'Keefe & Nadel, 1979). The only manipulation carried out on these cues was either to remove them or to interchange their locations. While this served well to demonstrate the basic phenomenon in a qualitative fashion, a more precisely defined set of landmarks and closer control over their location is required to determine the quantitative relationship between landmark configuration and place-field unit activity. At the present time not enough information is available to determine what would constitute a good, simple set of landmarks. In the computer simulations described here, planar, square landmarks of different sizes, that stand vertically at any position in the outer rectangle, are used. As viewed from above, in the computer-generated display, these square landmarks appear as straight lines. Squares were chosen because the computation of their projected retinal areas is reasonably tractable. The computation of the projected retinal area of an arbitrarily shaped object from a large number of different positions is quite complex. In all the rest of the simulations described here, the retinal area of these squares will be the location parameter used. It seems reasonable that a set of such square cards of different sizes, textures, and reflectances might well serve as adequate cues in real place-field experiments. Indeed in the Kubie, Muller, and Ranck (1983) experiments, a rectangular card played an important role as an environmental cue.

Place-Field Shape and Size

The two parameters σ and Θ in Equations 1 and 2 affect the size and, to some degree, the shape of the simulated place fields. The way

σ and Θ affect place fields is shown in Figure 7, in which σ and Θ are systematically varied while the landmarks and place-field center are kept fixed. The results shown in Figure 7 indicate that although these two parameters do not have identical effects on the shape and size of place fields, their effects are quite similar, particularly in the region of moderately high Θ and moderately low σ, which give realistic looking place fields. Real place-field units are probably subject to a good deal of inhibitory interaction with other hippocampal units and many other modulating influences. For this reason, the model described here is unlikely to give a very accurate picture of the shape variation of place fields with landmark manipulation.

Place-Field Location

Extensive simulations indicate that both the location, shape, and size of place fields is affected by the number, size, and location of the landmarks. The effect of landmark manipulation on the *shape* of place fields is relatively small and might be difficult to observe experimentally. However, the much larger effects of changing landmark location, size, or orientation on the *location* and *size* of place fields can easily be detected experimentally and will serve as the main test of the model.

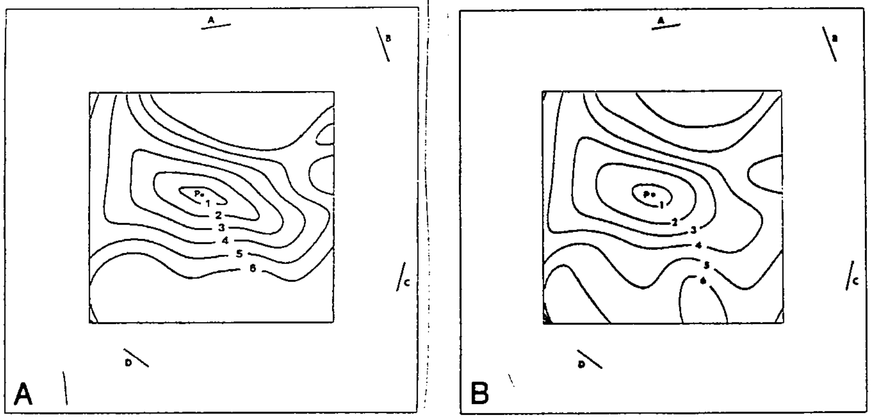


FIGURE 7. This figure was generated in the same way as Figure 6 except that rather than displaying the place-field boundary for just one set of values of σ and Θ, a range of values was used to create a contour map of the place fields. *A*: σ has been scanned while Θ is held at 2.4. The σ values for the different curves are: $\sigma_1 = 0.1$, $\sigma_2 = 0.25$, $\sigma_3 = 0.4$, $\sigma_4 = 0.55$, $\sigma_5 = 0.7$, $\sigma_6 = 0.85$. *B*: σ is fixed at 0.6 and Θ is scanned: $\Theta_1 = 3.9$, $\Theta_2 = 3.4$, $\Theta_3 = 2.9$, $\Theta_4 = 2.4$, $\Theta_5 = 1.9$, $\Theta_6 = 1.4$.

For example, the one hard quantitative experimental result that we have in this area comes from the dilation experiments (Kubie et al., 1983). Experiments similar to this were simulated using the model described here. The results of this simulation are shown in Figure 8. First, a place field centered at P was generated using landmarks A, B, and C. All the coordinates specifying the location of A, B, and C were then multiplied by 2 to give a dilation. The dilated scene was then translated back to a convenient location in the outer rectangle to

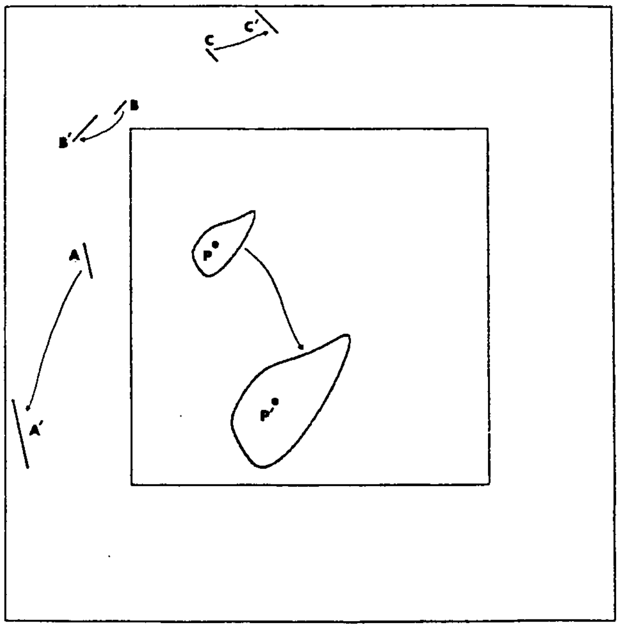


FIGURE 8. This figure shows the result of a simulated dilation experiment. First, the landmarks A, B, and C of the indicated size were set up and the viewpoint P chosen as the place-field center. The field centered at P was generated by the simulation. Then the dilation, which increased the size of each landmark and the distance between them by a factor of two, was computed using the known coordinates of the landmarks. This enlarged scene was then shifted so that it could be put in the outer rectangle. This resulted in landmarks A', B', and C'. A place-field boundary was then computed by the simulation using the same values of the stored location parameters originally determined at P. This led to the generation of a new larger place field at a different location. When the point P was dilated and translated by the same amount as the landmark A', B', and C', it mapped to a location indicated by P'. The fact that P' occupies the same position in the new place field as P did in the original, together with the fact that the new place field is a two-fold magnification of the original, are consequences of the dilation properties of retinal area used as a location parameter. (From "A Computational Model of Hippocampal Place Fields" by D. Zipser, 1985, *Behavioral Neuroscience*, 99, p. 1015. Copyright 1985 by the American Psychological Association. Reprinted by permission.)

give landmarks A', B', and C'. The same dilation and translation were applied to the point P itself to give the point P'. The landmarks A', B', and C' were used to generate a new place field using the same recorded values of the location parameters employed to generate the original place field centered at P. The new place field is centered at P' and has the same shape but is twice the size of the original. This result demonstrates that when retinal area is used as a location parameter, the model gives the expected result for a dilation experiment. While the actual shape and size of a place field depends on model parameters, the way it dilates is the result of the inherent mathematical properties of the location parameter used—area in this case. (See the caption of Figure 5 for proof that the area has the expected dilation properties.) Other simulations in which the size or location of landmarks was changed in ways other than uniform dilation, produced large shifts in the location and size of the place fields. The finding of similar place-field position shifts accompanying corresponding landmark manipulations in real experiments serves as strong confirmation that the model described here embodies the correct rules relating landmark configuration to place-field properties.

Scope of the Place-Field Model

The model place-field described here is both oversimplified and incomplete, yet within its domain of applicability, it is testable by experiment. One way in which the model is incomplete is that it ignores the problem of left-right asymmetry. Early experiments that indicated that distal landmarks determined place fields also showed that interchanging the landmarks on the right and left side extinguished the location-specific activity of place-field units. This indicates that the animal has the ability to distinguish right from left, whereas the model as described is symmetrical. The ability to distinguish right from left is a relatively straightforward feature to incorporate into the model and one way of doing this is described later in this chapter in connection with goal-finding models.

Another oversimplification of the place-field model is that it ignores the problem of viewer orientation. It is known from experiment that many place-field units show sensitivity to the orientation of the observer (McNaughton, Barnes, & O'Keefe, 1983). For example, a rat running through a maze arm in one direction may excite a place-field unit at a particular location that is not excited when the rat runs back through the same location in the opposite direction. Orientation is important for goal location and a mechanism for incorporating it into a place-field system is described in the next section.

Goal Location

Knowing where you are is only worthwhile if this information can be used for some useful cognitive purpose. For example, to answer the question "How do I get home from here?", information about the location of a goal (home in this case) must be available. For the rest of this chapter, I will deal with the problem of developing mechanisms that can answer this question. The problem is trivial, of course, if the goal is perceivable from the current location. The problem becomes interesting only when the goal is too far away or too inconspicuous to detect. In all of what follows, it will be assumed that the observer must find the way home using a set of locally visible landmarks and that the observer has no access to absolute direction information such as "Which way is north?" and so forth. This does not significantly reduce the generality of the problem, since the determination of an absolute direction such as north would, in general, depend on knowledge about the configuration of the local landmarks anyway.

The unoriented place fields described above are insufficient for goal location. The reason for this is that in the absence of absolute direction information, the location of a goal must be given relative to both the location *and* orientation of the observer. One way to approach this issue is to incorporate orientation into the place field itself. I call an oriented place field a *view field.* Many of the place fields observed in the hippocampus are, in fact, oriented as pointed out above. That is, some hippocampus neurons fire at a particular place only when the rat is pointed in a specific direction. Orientation can be incorporated into place fields by adding some weak, topological constraint, such as classifying landmarks as being either on the right, in the center, or on the left. Distinguishing right, left, and center is a fairly straightforward task for the sensory system of any bilaterally symmetrical animal.

In the following sections I describe two models of goal-finding. One, the *distributed view-field model,* uses a large number of oriented place-fields, each with associated goal location information, scattered around the environment. The other model is based on a more extensive computational analysis and requires fewer units but more elaborate calculations.

The Distributed View-Field Model

The distributed view-field model for goal location is based on the network shown in Figure 9 which is used as the "cognitive map" to guide an observer to a goal. The first layer of units, called object units,

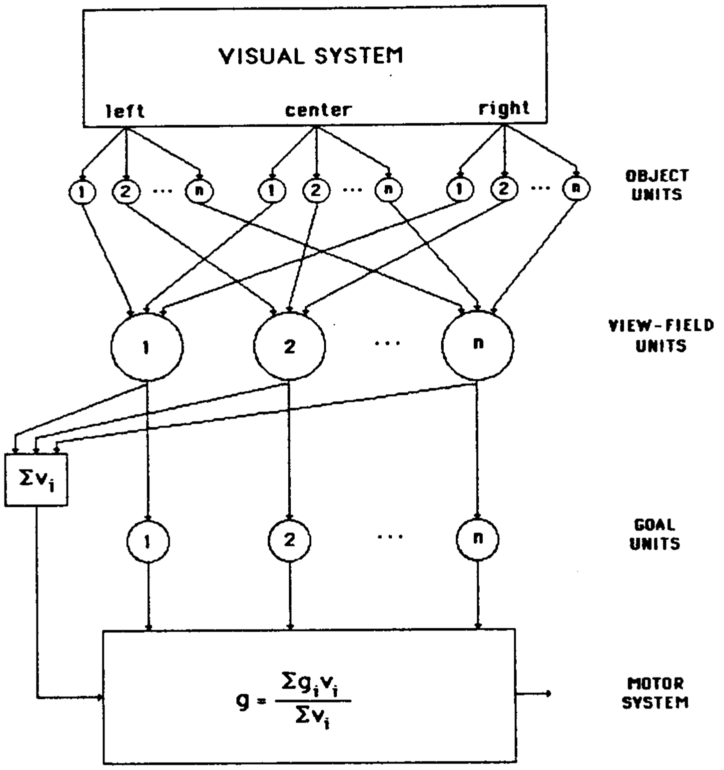


FIGURE 9. Network used in the distributed view-field model. The visual system output consists of three sets (left, center, and right) of three numbers representing an object's type, distance, and orientation relative to the observer. The output of an object unit is 0 unless its recorded type number matches its current input from the visual system. In this case, the output is the product of two Gaussian tuning functions (see Equation 1), one for distance and one for orientation. The output of the view-field units is the same kind of sum given in Equation 2. The output of the goal units is two components of a vector pointing to the goal each multiplied by the input from a view unit.

is analogous in function to the Layer 1 units of the preceding place-field model shown in Figure 2. The primary difference is that the object units in Figure 9 are divided into left, center, and right groups. Object units are gated on only when the landmarks they recognize appear in the appropriate region of the visual field, i.e., left, center, or right. The view-field units in Figure 9 are analogous to the Layer 2 units in Figure 2. Together, the object units and the view-field units recognize an oriented place field, i.e., a view field, in the same general way as the corresponding units in Figure 2. However, now the output of the view-field unit also depends on orientation as well as location. Once a view-field unit for a particular place is created, information about the location of a goal can be associated with it. In the network shown in Figure 9, goal location is encoded in an additional unit whose output is interpreted by the motor system as the direction of the goal. To actually head to a goal from any particular location, an observer with this kind of network would first examine the landmarks in front of it, which would lead to the activation of a set of object units and then to the activation of an associated view unit. This in turn would activate a goal unit, the output of which could direct the observer to the goal.

There are several problems to be solved before this general strategy can actually be applied to controlling the movements of an observer. First, it is impossible to have a unique view-field unit for every place and orientation in the environment. At best, one could hope to have a rich supply of view-field units whose place fields spanned the environment. Another problem is that even with a large number of view-field units, it is unlikely that the observer is ever at the exact center of a place field. This means that some form of interpolation is required to compute the location of the goal from the current position of the observer. Another problem is to specify how information about the environment gets assimilated by the network in Figure 9 in the first place. Aside from the technical issues of recruiting the appropriate units and connecting them together with the desired weights, there is the conceptual problem that arises from the need to know the location of a goal at the time that its association with a particular view field is made. There is no trivial solution to this problem. One approach is to remember and constantly update the location of a particular goal as the observer moves through the environment. This active memory of goal location can then be used to establish the place-field goal relationships when new units are recruited. Once enough units have been instantiated, the territory can be thought of as familiar and it will no longer be necessary to have an active memory of the location of a goal in order to find it. This is the strategy used in the distributed view-map model described here. This approach has the drawback that it demands both processing resources and memory accuracy. These requirements could possibly be met for modest periods of time and over short distances with no distractions, but they could cause problems in more extended situations.

In the distributed view-map model, the viewer has a large set of view fields scattered over the environment. Associated with each of these view fields is a vector pointed towards the goal. If the observer is in the center of one of these view fields and oriented in the optimal direction, the associated goal unit indicates the direction of the goal. However, the observer is seldom at the center of a view field so information from a set of nearby view fields must be interpreted. The first step in this process is recognition of the currently visible set of left, center, and right landmarks that are used to gate the appropriate ensemble of view fields. Associated with each of these view fields is a vector pointing home. This is illustrated in Figure 10, which shows a viewer, *V*, oriented towards the central landmark, *C*, and nine view fields, each labeled *m*. Of course, the oriented view fields are not actually written on the ground in the environment but are encoded in the brain of the observer. Although the whole ensemble of view fields that have the same set of three landmarks will be gated on, the activity of each individual unit will depend on how closely its parameters match the current situation. The closer the viewer is to the center and orientation of any particular view field, the more active the unit representing it becomes. In the distributed view-field model, the observer interpolates the location of the goal by using a weighted average of the direction vectors of all of the active goal units. The way this weighted average is formed is that each goal unit multiplies its output vector, indicating goal location by its input from a view-field unit. The motor system then sums all of these weighted vectors and normalizes them to the sum of the activity

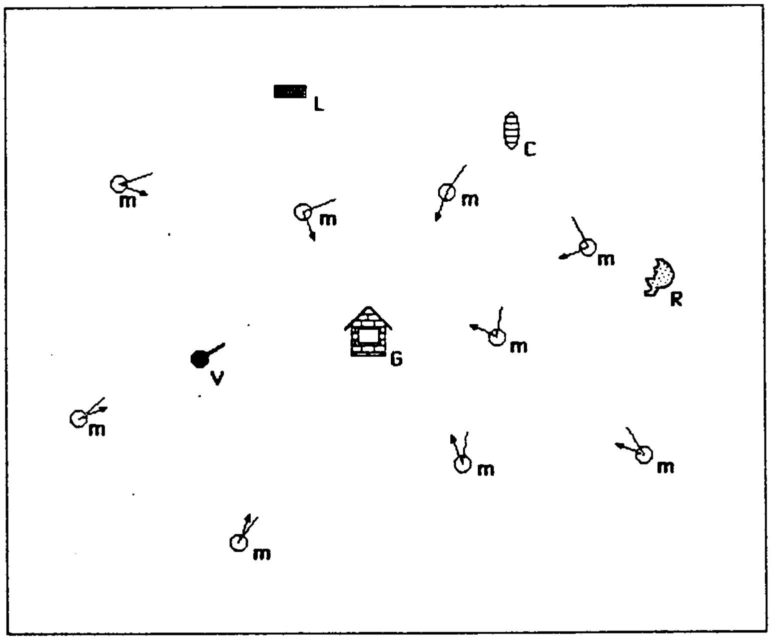


FIGURE 10. A diagrammatic representation of the view-field model. *G* is the goal, *V* is the observer, *L*, *C*, and *R* are landmarks. The positions at which view fields have been recorded by the observer are indicated by *m*s. Each view field consists of a line representing the orientation of the observer when the view field was recorded and an arrow representing the direction of the goal relative to this orientation.

of all the view-field units (see Figure 9). Thus, the goal vectors associated with close-matching view fields play a large part in the sum, since their associated units are highly active, whereas the vectors from less closely fitting view fields play a smaller role. This strategy requires the view fields to be reasonably well distributed around the environment.

In order to test the model, the network of Figure 9 was used to actually control the behavior of a simulated observer moving around in a model environment. Before describing this simulation in detail, it's necessary to deal with the problem of getting environmental information into view fields. Rather than building in spatial information, a procedure for assimilating it while the viewer moves around was incorporated into the model. The naive viewer was supplied with a network similar to that shown in Figure 9, but in which all object units and goal units had zero values for all their relevant parameters. At fixed time intervals, while the observer explored the environment at random, a signal is sent to recruit object and goal units. Upon receipt of this signal, the object units being recruited record the values necessary to recognize and locate the currently visible set of landmarks, while the goal unit records the location of the goal relative to the current position and orientation. The object units receive their information from the current output of the visual system, whereas the goal unit to be trained obtains its values from a working memory. Once all the available units are recruited, no more view fields can be learned and all subsequent goal-directed travel is dependent on the existing set of view fields.

To observe the properties of the distributed view-field model, a simulated observer was developed with two motivational states, "curiosity" and "guilt." In the curiosity state, the observer explores the environment by a random pattern of movement. In the guilt state, the observer attempts to return home using the information recorded in its network of view-field units. The assimilation of new information takes place at all times, independent of the motivational state of the observer. Thus, new information can be assimilated either during random exploration or during return to home.

The actual simulation program to carry out all this is quite complex, because in addition to simulating the cognitive mapping network of Figure 10, the environment, together with some information about the observer's location and visual system, had to be modeled. This project was expedited by the availability of a general purpose parallel simulating system called P3 (Parallel Process Programmer). The P3 system is described in more detail in a separate chapter of this book, but briefly, P3 is used to simulate networks consisting of an arbitrary number of unit processors connected together by signal lines. The complete network of interconnected units is called a P3 plan. Each unit in the plan must have a computer program, called a method, associated with it that updates its outputs according to some function of its current inputs and state variables. The input to units described in a P3 plan can come only from other units in the plan. That is, there is no outside world in the P3 description of a network. This means that the environment, as well as the nervous system network within the model, must be described by individual units. The simulated model can interact with the outside world in two ways. It can be initialized and observed as it functions, using facilities provided by the P3 simulation system. In addition, the method programs can communicate with the investigator through the keyboard and display using the available features of the method language. The methods can also use the file system. However, all communication between units must be via connections described in the P3 plan.

A simplified view of the P3 plan used to simulate the model described here is shown in Figure 11. There is one very complex unit called the *environ unit* which carries out a multitude of tasks. It keeps a record of and displays for observation the current position of the viewer, the location of the goal (home in this case), and the position of all relevant landmarks. The output of the environ unit provides the viewer-centered information necessary to activate the object units in the model. In this way, the environ unit serves as the sensory system of the viewer. Another output of the environ unit is working memory, which has the current goal location used by goal units when they are assimilating new information. The input to the environ unit comes from the *mover unit* which serves as the motor system for the viewer. The environ unit uses this information to update the location and the orientation of the observer. Thus, we see that the environ unit serves as the entire world outside the particular part of the brain being simulated by the mapping network under test, that is, it constitutes the musculature, the visual system, and the short-term memory of the observer together with the physical world outside the viewer. Obviously the method program for the environ unit is quite complex. The shaded units in Figure 11 represent the major parts of the cognitive map networks under test. These are the elements that are changed significantly when different models of the viewer's space representation system are studied. The rest of the P3 plan can remain essentially the same, constituting a major saving in effort.

The model and the associated methods were coded in the P3 system. An example of what the screen of the Symbolics 3600 LISP machine looks like during the progress of a simulation of the model is shown in Figure 12. Information about the state of various units, at one time cycle, is displayed in the large panel on the left. The "world display" window of the environ unit appears in the panel on the right. The behavior of the model is most easily analyzed by examination of the

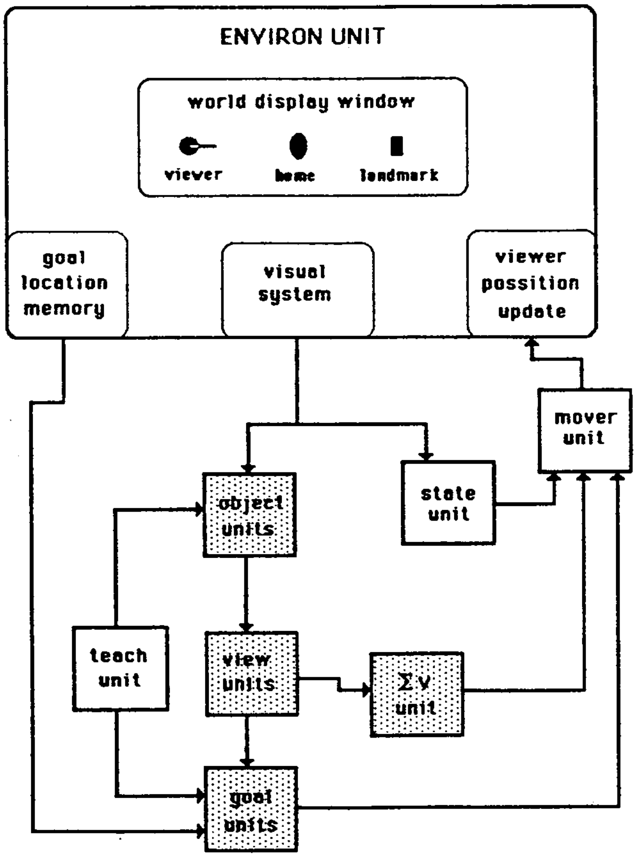
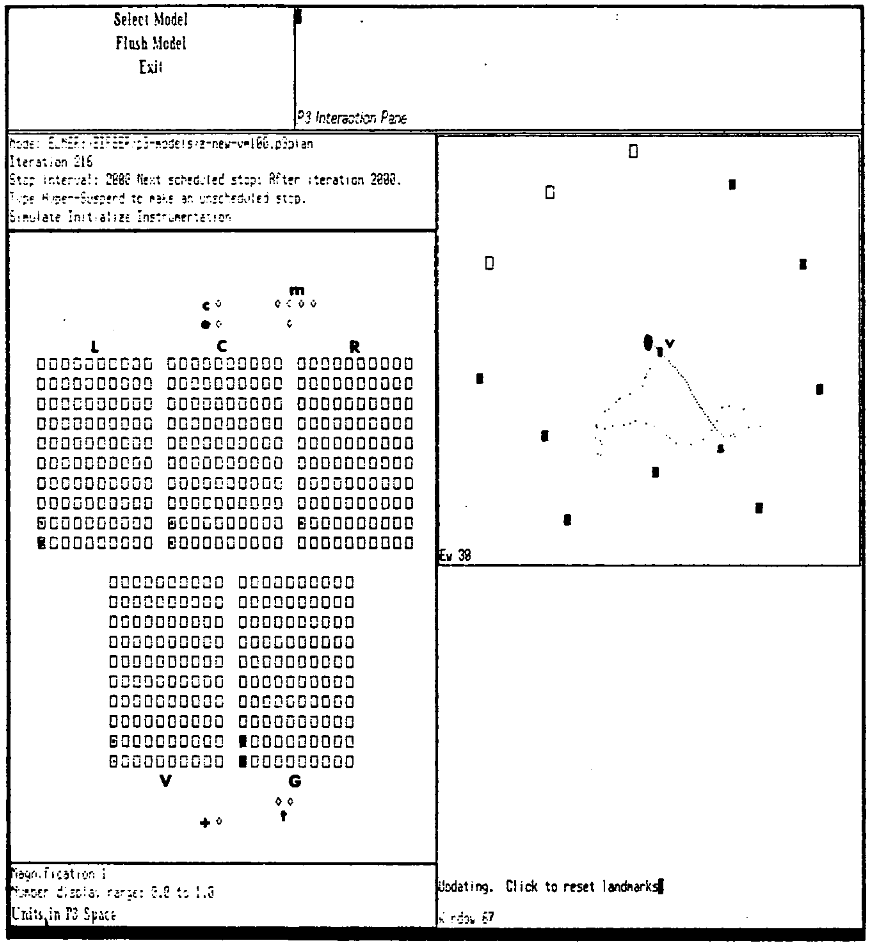


FIGURE 11. A simplified plan of the P3 simulation of the distributed view-field model. The shaded boxes represent the part of the model that contains the specific map information. The mover is actually comprised of several units that compensate for the delay between observation and resulting movement. This compensation is required because other motions may occur during this period. The symbols in the world display window of the environ unit are those used in the computer display that it drives (see Figure 12).

world display window. The observer, marked *V* in the figure, is just about to reach the goal at the center, having started towards the goal at a point marked *S*. The path of the observer during the "guilt" state, when it is trying to go home, is characterized by evenly spaced dots. The path of the observer during the "curiosity" state is seen as a set of much more arbitrarily spaced dots. One dot is added to these paths at each cycle of the simulation. At the outer extremes of the field, the landmarks appear as rectangles. In this particular simulation, eleven landmarks were used. Although they all look the same on the screen,



```
Model: [illegible]p3-models>z-new-v=100.p3plan
Iteration 216
Stop interval: 2000 Next scheduled stop: After iteration 2000.
Type Hyper-Suspend to make an unscheduled stop.
```

```
Magnification 1
Number display range: 0.0 to 1.0
Units in P3 Space
```

```
Updating.  Click to reset landmarks
```

FIGURE 12. Symbolics 3600 LISP machine display of the distributed view-field model. See the text for explanation.

they are in fact distinguished by the viewer. The three landmarks that constitute the current view are highlighted by making them hollow rectangles slightly larger than the unviewed landmarks.

Figure 12 represents the state of the system after 216 iterations of the simulation. Only 10 out of the pool of 100 view units have been recruited, so this is relatively early in the exploration phase. Examining the path from *S* to *V* in the world display window, shows that the viewer is able to go directly home along a moderately straight path. In general, this is the case early in a simulation, before many view fields have been recruited. The panel on the left shows the output activity of the object units, the view-field units, and the goal units. No information is being displayed about the teach unit, the environ unit, the state unit, or the mover unit. Actually, the mover unit consists of a group of several units that work together to provide the appropriate motion information based on the output of the array of goal units. Some caution needs to be used in looking at a state display of this type since there is a delay of one cycle between each layer of the network. Thus, what's seen at the output of the goal units is a consequence of events occurring one cycle earlier in the view units, and two cycles earlier in the *L*, *C*, and *R* object units. The important point to notice here is that only two goal units show any output activity at all, and both of these have activity of the same sign. This means that at this point the movement of the viewer is being controlled by only a small set of view fields. This is consistent with the fact that the simulation is early in the assimilation phase.

After the assimilation phase is over and all units have been recruited, the pattern of view fields is as shown in Figure 13. This figure was generated from the world display window of the environ unit, which has the option of showing the actual positions corresponding to the observer's store of view fields. Figure 14 shows the path of the observer attempting to get home after simulation cycle 3081 when the assimilation phase is long over. The path starts at *S*, and the observer, after much thrashing about, has reached point *V*, and not yet gotten close enough to home for the home detector to have triggered. Note that the path starts at *S* and continues in a relatively straight fashion, but not directly towards home. Before passing too far beyond the goal, the observer has made a sharp turn to the left and moved towards the goal on a very uncertain and constantly changing path. Then, after approaching quite close to the goal, the observer again turned, this time in a path that took it somewhat away from the goal. As we look at Figure 14 at the current cycle, the observer has been thrashing around in the general vicinity of the goal but has not been able to actually reach it. Eventually the observer will get home, but it will take a great many cycles. This behavior is quite typical of what happens when an attempt is made to go home after all 100 view-field units have been recruited. To the right of the world display window, the current states of the array of the view-field units and goal units are shown. These are not both from the same cycle, rather the view-field units are shown from one cycle before the goal units so that the active view-field units are the ones that determined the activity of the goal units in the lower array. Notice that a large number of view-field units are active and that a correspondingly large number of goal field units are active. However,

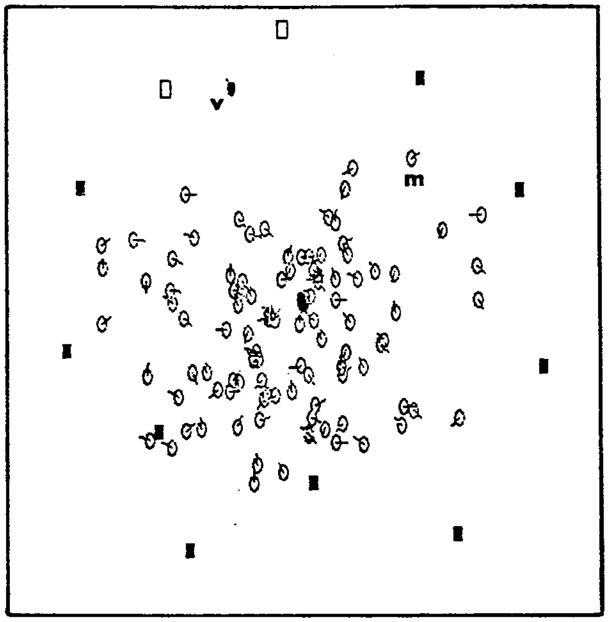


FIGURE 13. The distribution of view fields after all 100 available units have been recruited. A typical view field, labeled *m*, has a line indicating the orientation of the observer at the time it was recorded. The goal vector is left out for clarity but it would point to the dark oval at the center. The current position of the observer is indicated by *v*.

some goal units have positive output values as indicated by solid rectangles, while others have negative values as indicated by open rectangles. The reason for this is that some of the view fields are on opposite sides of the goal, which provides the observer with conflicting information about the sign of the direction of the goal. This conflicting information is the cause of the viewer's inability to get to the goal in a smooth fashion. This conflict situation only causes a serious problem when the observer is quite close to the goal because when it is further away, information from view fields on the opposite side of the goal contribute very little to the weighted sum of goal vectors. Although the problem is inherent in the model, by adjusting various parameters that control the density of view-field placement and the strength of their interaction, it is possible to reduce the size of the region of ambiguity to the point where it is small compared to the area in which the goal is directly perceivable.

The distributed view-field model simulated above has the virtue of simplicity, since the viewer is not required to carry out any complex

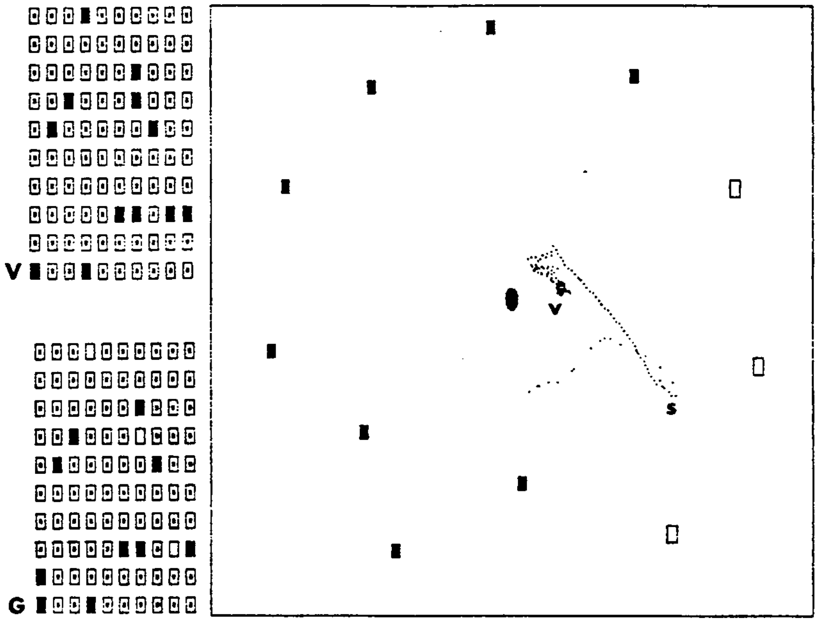


FIGURE 14. Behavior of observer, *V*, in the distributed view-field model with all units recruited. The state of the view-field units, *V*, on the 3080th cycle and of the goal units on the next cycle are shown on the left. The reason for the complex path followed by the observer is given in the text.

computations to navigate. Its algorithm for assimilating new information is extremely unsophisticated, operating only on the basis of time. One price paid for this simplicity is the need for a large number of units to store assimilated information. One way in which this algorithm might be improved is to assimilate new information only when the present information is inadequate. This inadequacy is easy to detect since the unit that sums the activity of all the view-field units can also be used to determine when there is insufficient information available. The way this is done is to take advantage of the fact that when the sum of the activities of the view-field units is high, the observer is in familiar territory; when this sum is low, the viewer is in unfamiliar territory where it would be wise to assimilate some new information. This would do away with redundant view fields that are generated by the clock method. On balance, the distributed view-field model meets the criterion of biological plausibility since it is based on the concept of place-field units that are known to exist. These units may well play a role in goal navigation since the hippocampus, where they are located, is required for navigation to unperceived goals. The magnitude of the ambiguity problem that this model has in the vicinity of the goal can probably be reduced to a point where it is not any more troublesome than the uncertainties about goal location that would arise from other sources of error in any model.

The β-Coefficient Model

In this section, a navigation scheme called the β-coefficient model (β model for short) is described that overcomes some of the drawbacks of the distributed view-field model described previously. Rather than requiring many separate view fields, the β model needs to record information about a set of landmarks and a goal only once. This recorded information can then be used to locate the goal anywhere the landmarks are visible. The β model also indicates goal location unambiguously, so the problem of sign that caused confusion near the goal in the view-field model is eliminated. Since only one recording is required, the actual number of units needed for the β model can be much less than for the view-field model. The β model can do away with the need for an updated memory of goal location whenever the set of landmarks to be used are visible from the goal itself since in these cases the recording can be made at the goal. If additional recordings of new sets of landmarks are made at locations where both a new and an old set are visible, the need for an updatable memory of goal location can be eliminated because the old set of landmarks can be used to establish the goal location for the new set. The price paid for these good features is an increase in computational complexity together with a loss of an immediately obvious link to current neurobiological data. Both of these drawbacks could be more apparent than real. More about this after the β model has been examined in detail.

To understand the β model, it's necessary to delve a little more deeply into the computational aspects of representing location. The problem we wish to solve is illustrated in Figure 15. An observer located at position V records the position of the three landmarks L, C, and R together with the location of the goal G. At some later time, the same observer is imagined to be at point V' at which the locations of the landmarks L, C, and R can be determined. The observer's problem is to locate the goal G, which cannot be seen from the position V', using the information recorded at point V. It can be shown that there exist three constants, β_L, β_C, and β_R, which depend only on the

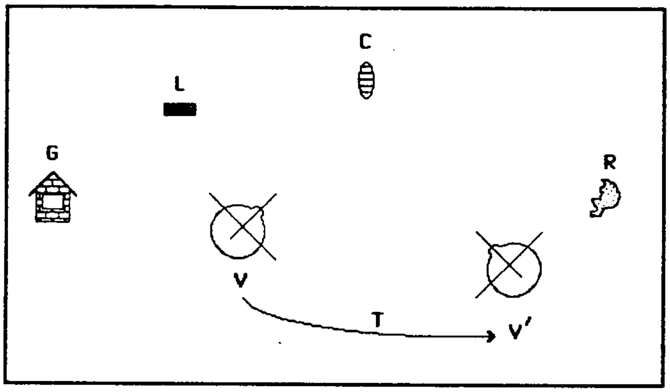


FIGURE 15. The observers at V and at V' can both locate the landmarks L, C, and R in their viewer-centered reference frames but only at V can the goal G be perceived. The problem is to solve for T so that the observer at V' can locate G.

relative locations of L, C, R and G, and have the following nice properties:

$$x'_G = \beta_L x'_L + \beta_C x'_C + \beta_R x'_R$$
$$y'_G = \beta_L y'_L + \beta_C y'_C + \beta_R y'_R$$
$$1 = \beta_R + \beta_C + \beta_C$$

where $x'_G, y'_G, x'_L, y'_L, x'_C, y'_C$, and x'_R, y'_R are the coordinates of G, L, C, and R in the reference frame of the observer at V'. (For a derivation of the βs see Appendix A.) Since the βs depend only on the relative locations of the landmarks and the goal and not on the location of the observer, they can be computed at any place where the location of the landmarks and the goal are known, that is, at V in our example. Because the βs are constant, it is only necessary to have a single record of their values to locate the goal from anywhere the landmarks can be viewed. The significance of this set of β equations relating the location of G to the location of the landmarks is that the coordinates of the goal can be obtained as a linear combination of the current values of the landmark coordinates and the remembered β. This means that neural-like elements can perform the computations if their synaptic weights correspond to the values of β.

A network using the β-coefficient concept and consisting of three units is shown in Figure 16. The job of the context gate unit is to determine that the current left, center, and right landmarks are the same as were used to originally program the β units. When the context unit recognizes the appropriate set of landmarks, it gates on the β units.

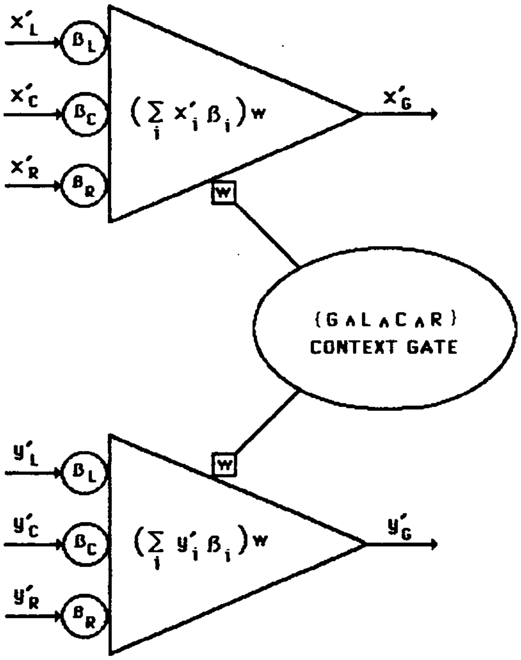


FIGURE 16. A pair of units that compute the x and y components of goal location using the method of β-coefficients. In the units, the βs are implemented as weights or lines coming from a sensory system that can measure the current values of the coordinates of three landmarks, L, C, and R. Each of the triangular β units is gated by a context gate unit (oval shape) which has an output of 0 unless the desired goal is G and the current landmarks are L, C, and R in the appropriate configuration, in which case the output of the gate unit goes to 1. If the value of $w = 1$ also, the correct value of the goal coordinates appears at the output of the β units.

These units perform the appropriate linear summation operations and their output represents the coordinates of the goal. One of the sets of units in Figure 16 is required for every combination of three landmarks with a goal. The output of the appropriately gated β units is an exact indication of the location of the goal from any point in the environment. This means that there will be no problem approaching the goal from any angle and there will be no confusion from information coming from the opposite side of the goal as there was in the previous example. Note that only one set of β units and a context unit is required for each goal-landmark combination as opposed to many units for each combination in the distributed view-field model described above. If information about the configuration of the environment is recorded at the goal itself, as may be the case for the rat on the platform in cloudy water, no working memory of goal location is required.

An important issue in the β model is how the βs are initially computed and recorded. Since we can get no hints from neurobiology, a wide range of possibilities are available. For example, we might imagine that the βs are computed locally by each β unit when it is recruited. Such a computation could take the form of a learning rule that adjusted the value of the βs in accordance with an error signal generated by comparing the current output of a β unit with the correct output, i.e., the coordinates of the goal that, of course, must be known when the βs are programmed. Another approach is to have a single, central β computer that calculates the appropriate values of β and transmits them to the newly recruited β units for recording. In the absence of hard data, it is a matter of taste which of these one considers to be more biologically plausible.

There is a completely different way to go about programming the β units that does not involve setting the values of β at all. In this approach it is assumed that there is a pre-existing pool of β units each with a set of fixed β values. It is then the values of the strengths of connections between a context unit and the β units in the pool that are set to produce the desired result. The idea is that the output of each β unit is weighted by the value of the weight, w, which connects it to a context unit. If these ws are chosen correctly, the sum of the weighted outputs of all the β units in the pool gives the appropriate goal coordinates. In this system a set of ws rather than a set of β units have to be learned. This could have some advantages, for example, the number of β units in the pool is fixed and can be much smaller than the number of context units. This means that fewer units are required to instantiate a w programming version of the model than a β programming version. Another advantage of the w version is that setting the w values is synonymous with recruiting the β units so that recruitment and programming are accomplished as a single operation rather than as two separate ones. As with the computation of β, the determination of the ws can be local to the β units or carried out by a central w computer.

In the simulation used here to test the β model, a version of w programming that had a pool consisting of only four β units with a central w computer was used. The way this system worked to program the β units with the correct connection strengths to the context units is explained in Appendix B.

A simulation analogous to that used to test the view-field model was employed to evaluate the β-coefficient scheme. Only those parts of the P3 plan that determine the representation of the spatial map were changed. The plan of this model is shown in Figure 17. The context units are very simple recognizers. Their output is 0 unless the particular set of left, center, and right landmarks to which they are sensitive is currently being viewed, in which case their output goes to 1. The teach

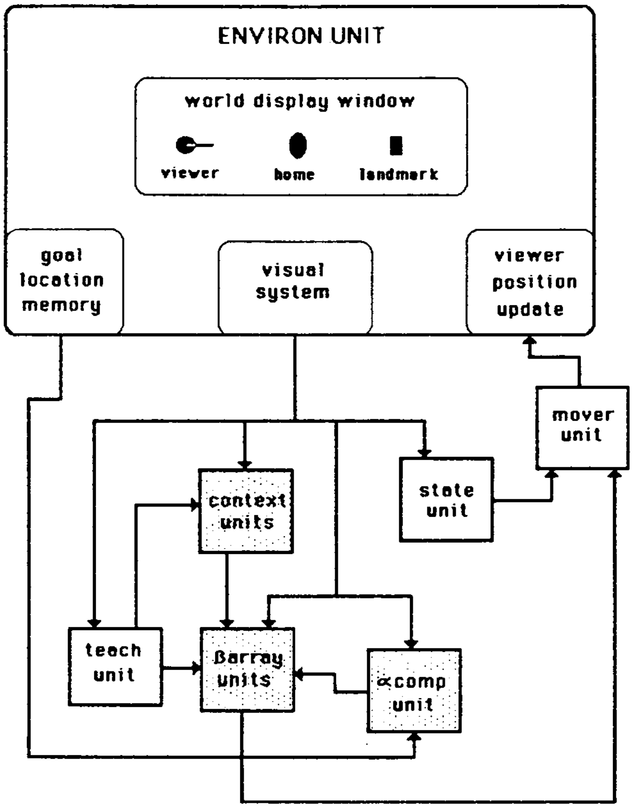


FIGURE 17. A simplified diagram of the P3 plan used to simulate an observer with a β array map of its environment. The shaded units constitute the map and they are the only significant difference between this plan and the one shown in Figure 11.

unit supplies a signal to the context units and the β units to recruit them when new contexts are assimilated. The teach unit requires modification since it no longer makes sense to teach at uniform time intervals. The teach unit produces a pulse only when no context units are active, that is, when the current view is unfamiliar.

The performance of this model is superb. The viewer always goes exactly home at the appropriate time. The results of a typical simulation episode are shown in Figure 18. Notice the straight paths heading directly towards the center of the goal. The number of context units required in this model is roughly equal to the number of landmarks, plus a few extra to take care of cases where a landmark can be the central object with different sets of flanking objects. The loops in the paths, found near the environment window borders in Figure 18, are

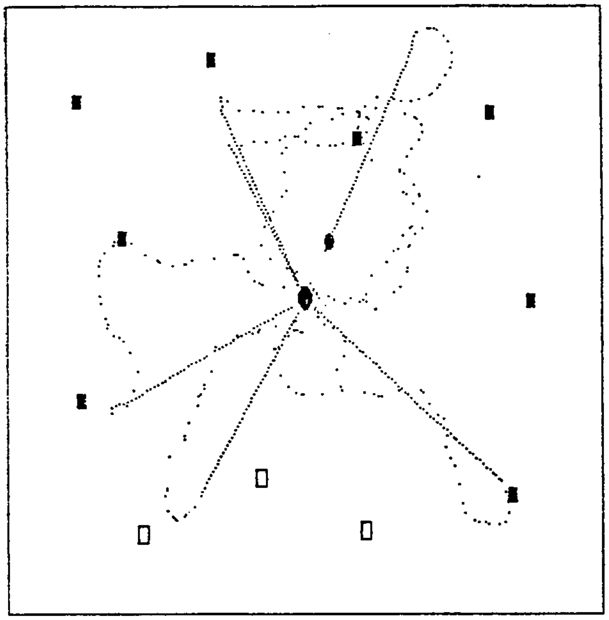


FIGURE 18. The performance of the β array model simulated with P3 and shown on the display of a Symbolics 3600 LISP machine. See text for details.

due to that fact that the state unit can detect when the observer leaves the area with landmarks. When this happens, the context unit instructs the move unit to turn the observer left until a set of three landmarks become visible. This feature is only included to keep the observer on the screen and has no significance for the network being tested. Even when the number of context units is insufficient to assimilate all possible views of the landmarks, the observer manages to find its way home. It simply searches until it finds an appropriate view for which it has assimilated information and then proceeds to use it to head home. Behavior degrades quite gracefully in this case and the number of landmarks has to exceed the number of context units by a great deal before serious difficulties are encountered.

Aside from being able to accomplish the task in an efficient and accurate fashion, the β-coefficient network has some other attractive features. It is, for example, able to explain the behavior of rats in the cloudy water experiment, since while on the platform, the rat can assimilate the appropriate β-related information for groups of room cues. The distributed view-field model described earlier has a much more difficult time with the cloudy water paradigm. It would require that the rat jump off the platform and swim around the tank to pick up a large set of view-field information. The problems that arose in the view-field scheme near the goal would be particularly annoying to the rat in this case since there are no visual cues to serve in goal detection, which become apparent as the rat approaches the submerged platform.

Conclusion

The three models that have been presented here differ from each other in the degree to which they encompass known experimental observations. The first model of place fields is based so strongly on neurophysiological data that its predictions are immediately amenable to testing. The second model, which adds a goal vector to place fields, is more speculative in the sense that it hypothesizes an explicit output role for the place-field neurons. It seems very likely that these neurons do play a role in navigation since removal of the hippocampus, which contains them, cripples navigational ability (Morris et al., 1982). Both of these models clearly meet the criterion of biological plausibility since they attempt to explain observable data in terms of neural-like models.

The β model is of a different sort. It attempts to explain the observed data by proposing a novel representational and computational scheme. This scheme is biologically plausible in the sense that it is implementable with neural-like elements operating in a highly parallel fashion. However, the β model can draw neither support or opposition from current experimental data. Indeed, this is just what we would expect from a theory that proposes a mechanism that is not completely intuitive but derives from a formal analysis of the problem domain. This is to be expected since it is very unlikely that experimentalists would find data relevant to the β model, even if it is correct, until they are specifically looking for it. This, of course, is exactly what theory is supposed to do, ask questions which would not be asked in the normal course of experimentation.

APPENDIX A

To see where the βs come from, we solve for the transformation T that maps all points in the V system to the V' system and then divide the expression for T into a constant part that depends on landmark and goal configuration and a variable part that depends on observer location. T must be found using only information about the recorded and current location of the landmarks L, C, and R. Once T has been obtained, it can be applied to the coordinates of the goal G in the V system to map them into the coordinates of G in the V' system. If information about three landmarks is used in two dimensions, homogeneous coordinates can be employed and T can be a linear transformation (Foley & VanDam, 1983). Having T linear will greatly simplify the nature of the computations necessary to use the transformation and will make the networks that carry it out much more biologically plausible. The derivation of T is shown below:

> Let $x_i = (x_i, y_i, z_i)$ be the homogeneous coordinates of the ith object. (In the case of Figure 15, $i = L$, C, R, or G.) Then T is defined so that $x'_i = x_i T$. To solve for T let
>
> $$X = \begin{pmatrix} x_L \\ x_C \\ x_R \end{pmatrix} \qquad X' = \begin{pmatrix} x'_L \\ x'_C \\ x'_R \end{pmatrix}$$
>
> Then, $T = X^{-1}X'$ where X^{-1} is the inverse of X that exists as long as L, C, and R are distinct and not collinear.

Now we have T defined in terms of the coordinates of the three landmarks recorded at V and the coordinates of the same landmarks observed from V'. Since we are using homogeneous coordinates, the third coordinate of all objects can be taken as 1.

The observer at V' is now able to compute T. Applying T to the recorded coordinates of the goal G will give the coordinates of G in the observer's current reference frame. That is,

$$x'_G = x_G T$$

or substituting for T,

$$x'_G = x_G X^{-1} X'. \qquad (1)$$

Equation 1 points out an interesting fact. Namely, the first two terms on the right-hand side are independent of viewer location and can be computed once—at the initial siting of the landmarks. This computation, in fact, yields three constants. Once determined, these three constants are sufficient to locate the goal G in any reference frame in which the locations of the landmarks L, C, and R can be determined. I call these constants β-coefficients and their definition is given below:

Define:

$$\beta = (\beta_L, \beta_C, \beta_R) = x_G X^{-1},$$

then:

$$\begin{aligned} x'_G &= \beta_L x'_L + \beta_C x'_C + \beta_R x'_R \\ y'_G &= \beta_L y'_L + \beta_C y'_C + \beta_R y'_R \\ 1 &= \beta_L + \beta_C + \beta_R . \end{aligned}$$

APPENDIX B

Four β units with fixed, predetermined values of β were used in the simulation described here. They were arranged in a 2 × 2 array as shown in Figure 19. When instructed by the teach unit to establish a new weight for connection to a newly instantiated context unit, each β unit calculated the appropriate weight by multiplying together the values of its two inputs from the α-computer. The problem is what should the outputs of the α-computer be to assure that each β unit generates the correct weights? Assuming that the contribution of each

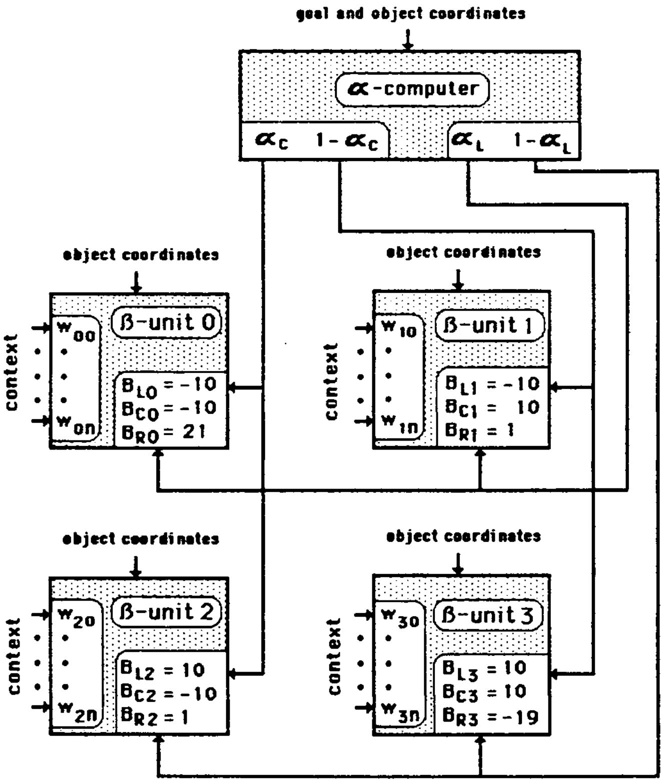


FIGURE 19. An array of four predetermined β units and the α-computer unit used to train them. Each β unit shows the actual values of the predetermined βs used in the simulation illustrated in Figure 18. The x and y components of the goal location vector output by each β unit are not shown in the figure for clarity. The calculations performed by the α-computer and β units when they are determining goal location and when they are being trained are given in the text.

of four predetermined β units to the x'_G component of the goal vector is given by

$$x'_{Gi} = w_{ij}(\beta_{Li}x'_L + \beta_{Ci}x'_C + \beta_{Ri}x'_R)$$

where $i = 0, 1, 2, 3$ are the four predetermined β units, w_{ij} is the weight for the ith β unit in the jth context, and β_{Li}, β_{Ci}, and β_{Ri} are the predetermined, constant β values for each β unit. The condition we want satisfied is $\sum_i x'_{Gi} = x'_G$. It was found that this is satisfied when[1]

$$\begin{aligned}
w_{0j} &= \alpha_{Lj}\alpha_{Cj}\\
w_{1j} &= \alpha_{Lj}(1-\alpha_{Cj})\\
w_{2j} &= \alpha_{Cj}(1-\alpha_{Lj})\\
w_{3j} &= (1-\alpha_{Lj})\ (1-\alpha_{Cj})
\end{aligned}$$

where

$$\begin{aligned}
\alpha_{Lj} &= (\beta_{Lj}-\beta_{L,0})/(\beta_{L,0}-\beta_{L,2})\\
\alpha_{Cj} &= (\beta_{Cj}-\beta_{C,0})/(\beta_{C,0}-\beta_{C,1})
\end{aligned}$$

and $\beta_{L,0} = \beta_{L,1}$, $\beta_{L,2} = \beta_{L,3}$, $\beta_{C,0} = \beta_{C,2}$, $\beta_{C,1} = \beta_{C,3}$.

---

[1] Note that the terms on the right of the equations are the outputs of the α-computer.