SHAPE RECOGNITION AND ILLUSORY CONJUNCTIONS

Geoffrey E. Hinton and Kevin J. Lang

Computer Science Department
Carnegie-Mellon University
Pittsburgh PA 15213

Abstract

One way to achieve viewpoint-invariant shape recognition is to impose a canonical, object-based frame of reference on a shape and to describe the positions, sizes and orientations of the shape's features relative to the imposed frame. This computation can be implemented in a parallel network of neuron-like processors, but the network has a tendency to make errors of a peculiar kind: When presented with several shapes it sometimes perceives one shape in the position of another. The parameters can be carefully tuned to avoid these "illusory conjunctions" in normal circumstances, but they reappear if the visual input is replaced by a random mask before the network has settled down. Treisman and Schmidt (1982) have shown that people make similar errors.

1. Introduction

People can recognize objects from a wide range of viewpoints, even if they have never seen them from these precise viewpoints before. There is psychological evidence (Rock, 1973; Marr and Nishihara, 1978) that people do this by imposing an appropriate frame of reference on the object to be recognized. They then redescribe the retinotopic features of the object relative to the imposed frame and thus they get an object-based description that does not depend on viewpoint. For example, given a square tilted at 45°, a person can impose an object-based frame tilted at 45°. Relative to this frame, the object has horizontal edges at top and bottom and vertical sides, so it is a square. Alternatively, a vertical frame can be imposed and all the edges will then be seen as diagonal and the object will be recognized as a diamond. The fact that we can see the same shape in either way is good evidence that we impose object-based frames in order to separate out the effects of viewpoint from the intrinsic shape of the object.

In general, it is difficult to choose the appropriate frame to impose on a collection of retinotopic features even if we have already solved the problem of segmenting out a set of features that all belong to a single object. We would like to use the frame that leads to a familiar object description, but the frame is required in order to access object descriptions so it is hard to see how object descriptions can give top-down guidance in picking the frame. When we see an upside-down capital R, for example, we recognize it by imposing an upside-down frame of reference, but how do we know to use this frame *before* we have recognized it as an R? Hinton (1981a) suggests that this chicken-and-egg problem can be solved by using a cooperative computation in a parallel network. The computation performs a parallel, iterative search which settles on both the reference frame and the shape description at the same time.

This paper describes a computer simulation which supports that suggestion and it shows that the network makes a peculiar kind of error when it is presented with several shapes which are then removed before the network has had time to settle on a percept of any one of them. The error consists of recognizing one shape in the position of another. The network was *not* designed to produce such errors. Rather, they appear to be an inevitable consequence of this method of shape recognition and it actually requires considerable fine-tuning to stop them from occurring in normal circumstances (i.e. when the input is not removed prematurely).

1.1 Illusory Conjunctions

Treisman and Schmidt presented subjects with cards containing a row of three colored letters surrounded by two black digits. The cards were presented for about 100ms and were then replaced by a mask of random features. The task was to first report the two black digits by saying a two digit number and then to report the positions and colors of as many letters as possible. Subjects often made errors which consisted of the shape of one letter in the position of another.[2] They were sometimes very confident in these errors, claiming to actually see the illusory conjunction rather than simply guessing. Also there was no distance effect. If a letter changed position it was as likely to move two places away as one. These results are counter-intuitive. Why should people get a clear percept of a combination of shape and location that isn't in the stimulus, and why is there no distance effect?

1.2 An Overview of the Network

The network used in the simulation contains four different kinds of units. At the highest level there are single units that stand for specific letters. These are connected to units that stand for object-based features — strokes or junctions whose position and orientation are described relative to the appropriate object-based frame. Each object-based feature is connected to all the retinotopic features that could depict it. Each of these connections is gated by a "mapping unit" that represents a particular choice of object-based frame (see figure 1.1).

When a single mapping unit is active it opens up connections that pair each retinotopic feature with exactly one object-based feature. So if

[1] This research was supported by a grant from the System Development Foundation

[2] Letters could also take on the color of other letters in the display. We shall not mention color again, but it fits the model we present provided there is a central location for representing the color of the object currently being perceived

the frame of reference is known in advance the retinotopic features can be made to activate the appropriate object-based features by simply turning on the right mapping unit and inhibiting all the others.

Each pairing of an activated object-based feature and an activated retinotopic feature sends activation to the corresponding mapping unit, so if the shape is known in advance it is easy to find the reference frame by simply activating the object-based features of the shape. If the shape contains f features there will be f correct pairings of an activated object-based feature with an activated retinotopic feature. These f pairings will all send activation to the *same* mapping unit and it will therefore be able to win a competition among the mapping units even though some of its rivals receive input from other spurious pairings of activated object-based and retinotopic features.

When neither the reference frame nor the shape are known in advance, the network will still settle to a consistent state when presented with a single familiar letter. Initially many different mapping units will be active and the activity in the object-based units will represent the superposition of many different ways of mapping the retinotopic features. However, combinations of object-based features that correspond to familiar letters will receive top-down support from the letter units and so they will be enhanced. Once this happens, the mappings that led to these features will be enhanced because the input to a mapping unit depends on the product of activity levels in the object-based and retinotopic units. This mutual enhancement is a non-linear cooperative process that eventually leads to one set of object-based features and one mapping unit becoming dominant.

A different, but equivalent, view of the network is that the retinotopic units gate the connections between the mapping units and the object-based units. Each retinotopic feature is consistent with various possible conjunctions of a mapping with an object-based feature and so it allows all such pairs to support each other. The network settles into a state which allows as many of these pairs as possible to be active, subject to the constraint that only one mapping unit must be active in the final state. This view is helpful in understanding illusory conjunctions. It is the retinotopic features which determine the consistency between object-based features and mappings. If they are removed before the network has finished settling, there is nothing to prevent the object-based units and the mapping units from settling to inconsistent states.

2. The Simulation

To test the claim that a parallel model of shape recognition can explain the psychological evidence on illusory conjunctions, a simulation of such a model was performed on a Symbolics 3600 lisp machine. A network of continuous valued, neuron-like units was presented with the retinotopic features of several letters, and the activation levels of the units were repeatedly updated starting from balanced initial values and using a deterministic, synchronous relaxation algorithm until a letter had been located and identified. When allowed to run to completion, the network never made any errors. However, when a random mask replaced the image before the network was finished, illusory conjunctions occurred.

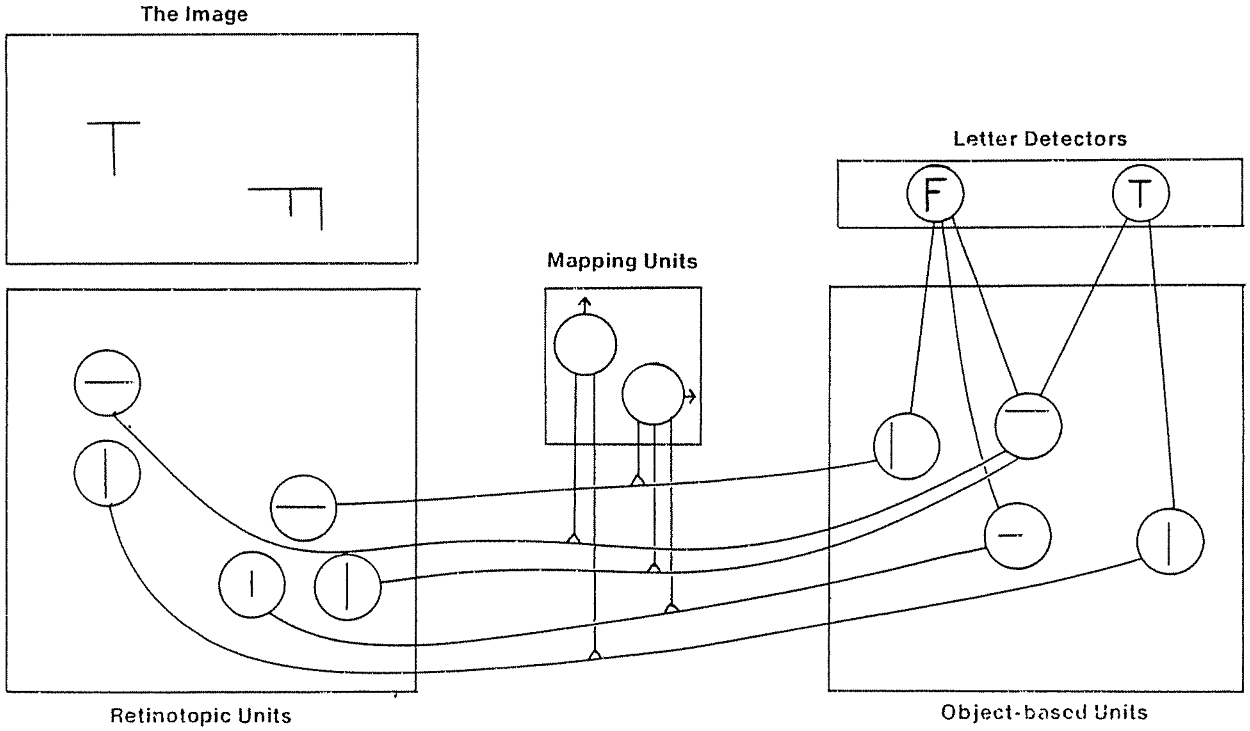

Figure 1.1: This shows four types of unit and how they are connected.
Only the units and connections relevant for one particular input are shown.

2.1 Structure of the Network

The units of the net are organized into four sets that are called planes to emphasize their spatial interpretation. The retinotopic plane contains units that represent retinotopic features that could be extracted by low-level visual processing. The mapping plane contains units that represent pairings of rotations and translations which are used to map the retinotopic image into a standard orientation on the object plane so that the image can be identified. The mappings used in this simulation do not perform scaling, so the letters presented on the retinotopic plane must all be of the same size. Units on the retinotopic, mapping, and object planes are connected by the special three-way links that are describe above. Each letter unit is connected to the features on the object plane which define the shape of the letter.

2.1.1 The Retinotopic Plane

The features on the retinotopic plane are based on an imaginary 13 by 13 array. The two types of features that are represented by units on the plane are strokes and junctions. A stroke is defined to be a horizontal or vertical line segment whose length is 2, 3, 4, or 5 pixels. There are 1092 units of this type on the retinotopic plane. Junctions occur at the endpoints of segments. A corner occurs when two strokes meet at a 90 degree angle at their endpoints. A T-joint occurs when the endpoint of one stroke meets the interior of another stroke at a 90 degree angle. Any endpoint that is not a corner or a T-joint generates a free-end feature. Every junction unit also has an orientation of up, down, right, or left, so that the T-joint on the left side of an E is different from the T-joint on the top of an I. There are a total of 2028 junction units on the retinotopic plane. It is redundant to have both stroke units and junction units, since the information in either set can be derived from the other. This redundancy is crucial when the network needs to map several letters onto the object plane at the same time without confusing them. The junctions contain information about which stroke goes with which other stroke and similarly the strokes contain information about which junctions go together. Every unit in the network has a real-valued activation level between zero and one, but the units on the retinotopic plane are always clamped to zero or one in this simulation.

Figure 2-1 shows the display used to illustrate the state of the network. Retinotopic units are drawn on a 13 by 13 array of squares that suggests the image from which the features could have been extracted. Stroke units are drawn as stippled rectangles with the same length, orientation, and position as the strokes which they represent. Corner units are drawn as white triangles in the corners of the squares. T-joint units are shown by white triangles at the sides of the squares. Free-end units are drawn as small white squares on the ends of the strokes. The orientation of each junction unit is represented by the orientation of its symbol in the obvious way.

2.1.2 The Object Plane

The object plane is basically just a smaller version of the retinotopic plane. It uses 100 stroke units and 300 junction units to encode all possible features in its 5 by 5 array of object-based locations. In figure 2-1, the object plane is drawn the same size as the retinotopic plane, even though it contains fewer units. This provides the space required to clearly represent the activation level of each unit by the size of its symbol. The area of a junction symbol is exactly proportional to activation of its unit, but for stroke units, the correspondence is only approximate.

2.1.3 The Mapping Plane

The mapping plane contains units that represent rules for matching features on the retinotopic plane with features on the object plane. The 324 units on the mapping plane are the cross product of nine x-translations, nine y-translations, and four rotations. These mappings specify all the ways that the object plane can be associated with a 5 by 5 subset of the retinotopic plane. In figure 2-1, the mappings are drawn as small black triangles in a 9 by 9 array of squares which corresponds to the central portion of the retinotopic plane. The

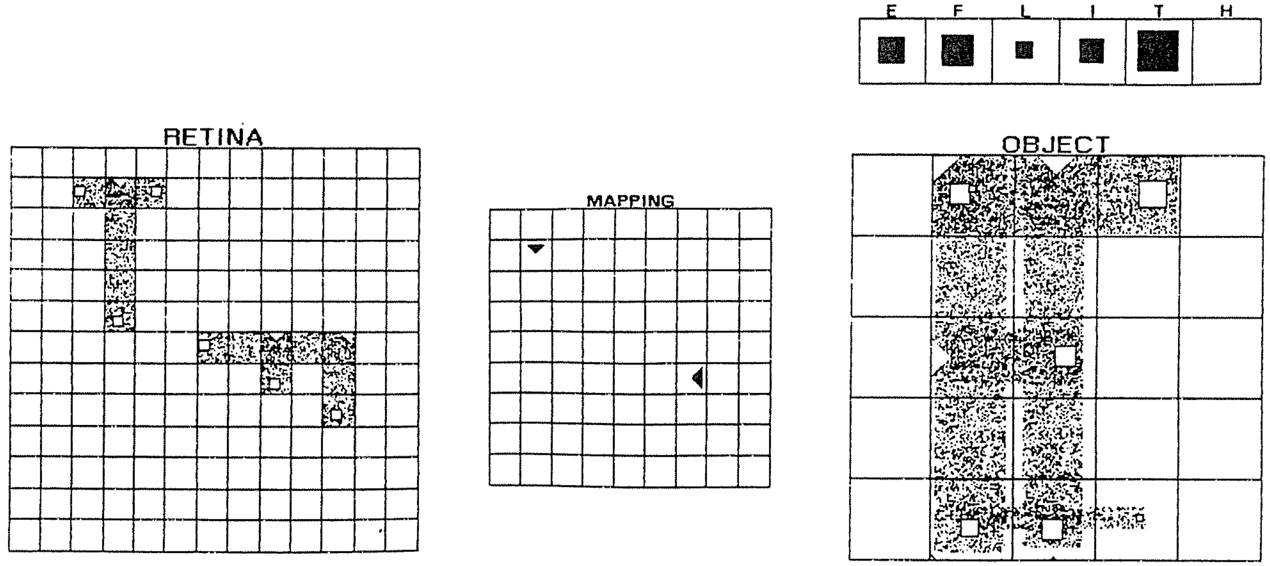

Figure 2-1: The Network

square in which a triangle is drawn shows the point which will be mapped to the center of the object plane. The rotation which a mapping performs is indicated by the orientation of its triangle. Try looking at Figure 2-1 to compare the positions of the vertical T and the sideways F with the two active mappings that associate them with their canonical images on the object plane. The activation levels of mapping units are indicated by the area of the triangles which represent them.

2.1.4 The Letter Plane

The six units on this plane represent the letters E, F, L, I, T, and H. Each unit's activation level is shown by the area of the black square in its box.

2.1.5 Links

The first type of link in the network is a simple two-way link between a letter unit and a feature unit on the object plane. The existence of such a link means that the feature is part of the definition of the letter. There are 47 of these links in the current model. The second type of link is a three-way link between a retinotopic feature, a mapping, and an object feature. There are 129,600 of these links, which means that some sort of optimization technique is needed to reduce the space and time requirements of the simulation. Although it is conceptually attractive to view the links as channels between the retinotopic plane and the object plane which are gated by mappings, one may think of them as channels between mappings and object-based features which are are gated by retinotopic features. In fact, the clamping on the retinotopic plane means that these channels are either completely open or completely closed, so it is possible *in a simulation* to scan the retinotopic plane for features that are on, and then wire up only the corresponding links between the mapping plane and the object plane. This reduces the network to about 3,000 two-way links that can be implemented in the same way as the links between letters and object features.

2.2 Relaxation

The deterministic relaxation algorithm employed in this simulation proceeds in synchronous cycles which have three phases. At the beginning of each cycle, the units exchange activation via their links. Next, a competition phase cuts the units' activation down by factors which depend on their size. Finally, all activations are scaled up by constant factors which are computed separately for each plane so that the total activation in the plane is normalized to a predetermined quantity. In more intuitive terms, each unit is a competitor of the other units on its plane but is an ally of the units to which it is connected.

In the following sections, units are designated by lower-case Greek letters near the beginning of the alphabet. Their activation levels are represented by the corresponding Latin letters. These activations are functions of time, which is measured in iterations and indicated by subscripts on the activations.

2.2.1 Propagation of Activation

Let a_i denote the activation level of unit α at time i. Then the activation of α at time $i+1$ is given by

$$a_{i+1} = \tau a_i + (1-\tau) \sum_{\beta \in \mathcal{L}(\alpha)} w_{\alpha\beta} \cdot b_i \qquad (2.1)$$

where $\mathcal{L}(\alpha)$ is the set of all units which are linked to α, $w_{\alpha\beta}$ is the weight of the link between α and β, and τ is a constant that is associated with α's plane. The purpose of τ is to adjust the rate at which activation levels change. Setting τ close to 1.0 causes the units to evolve slowly and feel little influence from their links. It turns out that τ also affects the competitive behavior of a plane. A winner-take-all selection cannot occur unless τ is set high enough to preclude an equilibrium state where the fresh activation that each unit gets from its links balances out the effect of competition on that unit.

With these facts in mind, it is easy to make rough estimates of appropriate τ values for the model. The letter and mapping planes should evolve slowly so they will have time to communicate their hypotheses, and should settle down to a state where only one unit has been selected on each plane. The high values of τ which were originally chosen worked moderately well, but some experimentation was required to locate the values of 0.995 for the letter plane and 0.98 for the mapping plane that always worked correctly. The object plane, on the other hand, should have fast reaction time but weak competition so that the other planes can exchange information quickly and keep many options open. In fact, a number of features should remain alive even in the final state. The first guess of $\tau = 0.5$ has worked for every version of the model ever tested, so the network is clearly not very sensitive to the exact value of this parameter.

2.2.2 Competition within Planes

Without competition, the network rapidly settles into an uninteresting equilibrium state where a large number of units are on and the activation boosts that are being transmitted over the links balance each other out. In order to force progress towards a solution, it is necessary to exert pressure on the units which will cause weakly supported ones to die out. The method of competition used in this simulation consisted of taking the activation level of each unit and raising it to some power greater than one. Since the activations are between zero and one, this forces them to shrink by an amount that is a function of their size, with small activations getting beaten down more than large ones. Furthermore, the discrimination against weak units increases as the exponent increases, which provides the simulator with the ability to control the intensity of competition. The precision of the competition control mechanism and the smoothness of activation decay under the influence of the competition algorithm were the keys to achieving interesting behavior from the network while using monotonic competition schedules. The exact schedules which proved to be acceptable are discussed in section 2.4.

2.2.3 Normalization

The final step in an iteration is to normalize the activation levels in each plane so that they add up to a target sum. This total corresponds to the number of units that ought to be on in a solution state. For the mapping and letter planes, the target sum is clearly 1. For the object plane, it is harder to say what the total should be, since the number of units that ought to be on at the end depends on the letter that is being recognized. To solve this problem, a new target sum is computed before every iteration by having the letter units vote for how much activation they would like to see on the object plane.

Once a target sum has been chosen for a plane, the normalization is performed by adding up the activations of all the units on the plane, and then multiplying each activation by the ratio of the target sum to the actual sum.

2.2.4 Initial State

Each relaxation of the network begins in a balanced initial state where the units on a given plane have equal activations that add up to the target sum defined in section 2.2.3.

2.2.5 Orientation Bias

The network described so far is unrealistic because it gives no preference to letters in their upright orientation. This can be corrected by modifying equation (2.1) to read

$$a_{i+1} = \tau \upsilon a_i + (1 - \tau) \sum_{\beta \in \mathcal{L}(\alpha)} w_{\alpha\beta} \cdot b_i \tag{2.2}$$

where υ is a factor that equals 1.0 for a unit α on the letter or object plane. For α on the mapping plane, υ equals 0.2 if the unit being updated is rotated from the vertical and the mapping with the same translation but no rotation has nonzero activation. Otherwise, υ equals 1.0 for the mapping plane also. The purpose of υ is to deflate rotated mappings that are directly competing with nonrotated mappings. However, this rule is not strong enough to kill off mappings by itself, because the deflating effect of υ quickly comes into equilibrium with the positive boost coming over the links. The only effect is to de-emphasize rotated mappings in the presence of upright mappings so that the latter can win through ordinary competition.

2.3 Assignment of Weights

Each link possesses a weight which can be thought of as an amplification factor for signals sent over that link. Weights are needed to balance the network so that the simple relaxation technique is able to find an appropriate final state for every input. The main problem is that the unit with the most links often wins the competition even when other units are receiving more support per link. For example, the letter unit for E will always beat the unit for L just because E has more features and therefore more links. The solution to this problem is to compute a link's weight using a function that takes into account the connectivity of both of the link's endpoints.

2.3.1 The Weighting Fun<ction

The weight $w_{\alpha\beta}$ of the link between units α and β is given by

$$w_{\alpha\beta} = u_{\alpha\beta} \cdot u_{\beta\alpha} \tag{2.3}$$

where $u_{\alpha\beta}$ is a weighting factor that compensates for the number of connections α has with units on β's plane. Specifically,

$$u_{\alpha\beta} = \frac{\left(\sum_{\gamma \in \pi(\alpha)} \|\mathcal{L}(\gamma) \cap \pi(\beta)\|\right) / \|\pi(\alpha)\|}{\|\mathcal{L}(\alpha) \cap \pi(\beta)\|} \tag{2.4}$$

where $\mathcal{L}(\alpha)$ is the set of units which are linked to α and $\pi(\alpha)$ is the plane on which α is located. For a concrete example, let α be the letter unit which represents E and β be a feature unit which is linked to α. Since E has links to 4 stroke units and 6 junction units, the denominator of equation (2.4) is 10. The numerator is the average number of links to the object plane from units on the letter plane, namely 47 / 6 = 7.8, so $u_{\alpha\beta}$ = 7.8 / 10 = .78 is the weighting factor for E. The unit for L only has links with 2 stroke units and 3 junction units, so its weighting factor is $u_{\alpha\beta}$ = 7.8 / 5 = 1.54. Thus the weighting factors for units on the letter plane cancel out inbalances caused by the different number of features found in the various letters.

The mirror image factors $u_{\alpha\beta}$ for α on the object plane and β on the letter plane take care of a more subtle problem. Letter units can support each other by sending activation indirectly through units representing shared features. Unfortunately, the six letters fall into the cliques {EFL}, {IT}, and {H} based on shared feature counts. Without these weighting factors, the survival of a letter unit depends on the size of its clique. Finally, when α and β are on the object and mapping planes, $u_{\alpha\beta}$ = 1 since every object unit is linked to every mapping unit and *vice versa.*

2.3.2 Tuning the Weights

Equation (2.3) provides a good approximation to the weights required to balance the network. However, some hand tuning needs to be performed to ensure proper handling of superimposed images on the object plane. This extra tuning can be implemented by amending the definition of $w_{\alpha\beta}$ to read

$$w_{\alpha\beta} = f_{\alpha\beta}(u_{\alpha\beta}) \cdot f_{\beta\alpha}(u_{\beta\alpha}) \tag{2.5}$$

where $f_{\alpha\beta}$ is an adjustment function that is associated with the ordered pair $(\pi(\alpha), \pi(\beta))$.

The main challenge in balancing the letters is keeping subset relationships untangled. For example, the features of F are basically a subset of the features of E, so it is difficult to find a set of weights for which E is not identified as F and F is not identified as E. The problem becomes much worse when more than one letter is being mapped onto the object plane at the same time, because F + L looks like E, F + I looks like E + T, and so on. Since the features being mapped onto the object plane contain redundant information and equation (2.3) balances things out about right, a one-dimensional adjustment based on discriminating between large letters and small letters is sufficient to solve these difficulties. A function which performs this discrimination is $f_{\alpha\beta}(x) = x + k$ for unit α on the letter plane and unit β on the object plane. Since k is added to the weighting factor for each link to a letter unit, it provides a boost which is a function of the letter's size. A k value of 0.1 prevents confusion with up to three letters on the retina.

The next adjustment function is necessary because the clique effect described in section 2.3.1 is much less pronounced than the other imbalances between letters. The weighting factor which primarily addresses this effect should therefore be de-emphasized by setting $f_{\alpha\beta}(x) = \sqrt{x}$ for units α on the object plane and β on the letter plane.

2.4 Competition Schedules

The behavior of the network while it is settling down into a solution state is primarily determined by the rules for varying the amount of competition through time. By changing these rules, a variety of behaviors can be induced, most of which are pathological when there are multiple letters in the input. The network tends to get wedged into either an uninteresting equilibrium state where too many units are active, or an inconsistent final state where the winners on the various planes do not correspond with each other or the input. The next two sections describe a strategy for managing the settling process and the effect of replacing the input with a backward mask before the network has finished settling.

2.4.1 A Successful Search Strategy

The ideal method of running the simulation would be to gradually increase the competition on all of the planes, so that a variety of options would remain open as long as possible, with a consensus eventually emerging as to what was seen where. The final selection of a letter unit and a mapping unit would occur simultaneously. Figure 2-1 shows why this approach doesn't quite work. The simulation has progressed to the point where the correct pair of mappings and the correct pair of letters are winning. However, due to slight inbalances in the amount of support the various units are giving to each other, the letter unit for T is winning its competition, while on the mapping plane the unit transmitting F is ahead. If the simulation were to proceed with the same gradual competition on both planes, the units which are currently ahead would win, resulting in a spontaneous illusory conjunction. Although the weights could be adjusted to balance the activations in this example, the new weights would make the problem worse for some other pair of letters.

A better solution to the problem is to allow one plane to decide first and then transmit its choice to the other plane so that it can choose the corresponding option. During a relaxation based on this strategy, the competition increases gradually at first, as in the ideal simulation described above. A small number of likely mappings emerge as likely candidates, and the images which they specify are superimposed on the object plane. The letter units also evolve slowly, relying on the redundant features which encode the letters to sort out the combined image they see on the object plane. After the network has run for a while in this consensus-building phase, the competition is turned up on the letter plane to force a choice. Figure 2-2 shows the state of the network shortly after this letter selection process has begun. The unit for L is leading because all of its features are strongly on, whereas some of the features for E are less activated. (Without the weight tuning described in §2.3.2, E would be winning at this point because it is linked to units containing more total activation.) After the unit for L has won, the image on the object plane will look much more like an L than an E, and the correct mapping will win after about 40 more iterations.

Although the scenario just described sounds quite reasonable, it is very difficult to adjust the competition schedules so that the network never makes mistakes. A workable solution depends on getting the rate of mapping competition exactly right, and is bounded by the following problems. In figure 2-2 it is apparent that the mapping for E is far ahead of the mapping for L, even though the letter unit for L is going to win. If the mapping competition does not occur slowly enough, the mapping unit for L will be so far behind at this point that it will have no hope of catching up. If, however, the mapping competition occurs too slowly, the spurious mappings[3] visible in figure 2-2 will sufficiently distort the relative activations of the various features on the object plane to cause either outright misidentification of the letters, or the kind of letter blending mentioned in section 2.3.2.

A competition schedule that works is shown in figure 2-3, which is a plot of the competition control parameters for the three active planes as functions of time. These parameters are the exponents used in the competition scheme described in §2.2.2. The behavior of the network is sensitive to the exact shape of the parameter curves, and simple piecewise-linear functions were not sufficient to eliminate all errors. It is important to note that the primary type of error which occurs is illusory conjunction, and that all of this tuning effort is necessary to eliminate them, not to introduce them. The final result of the adjustment process was a network that would correctly select and identify a letter from any set of one, two, or three upright letters encoded on the retinotopic plane.

[3] These mappings are actually partial symmetries of the letters, which explains why the figure resembles an X-ray diffraction pattern

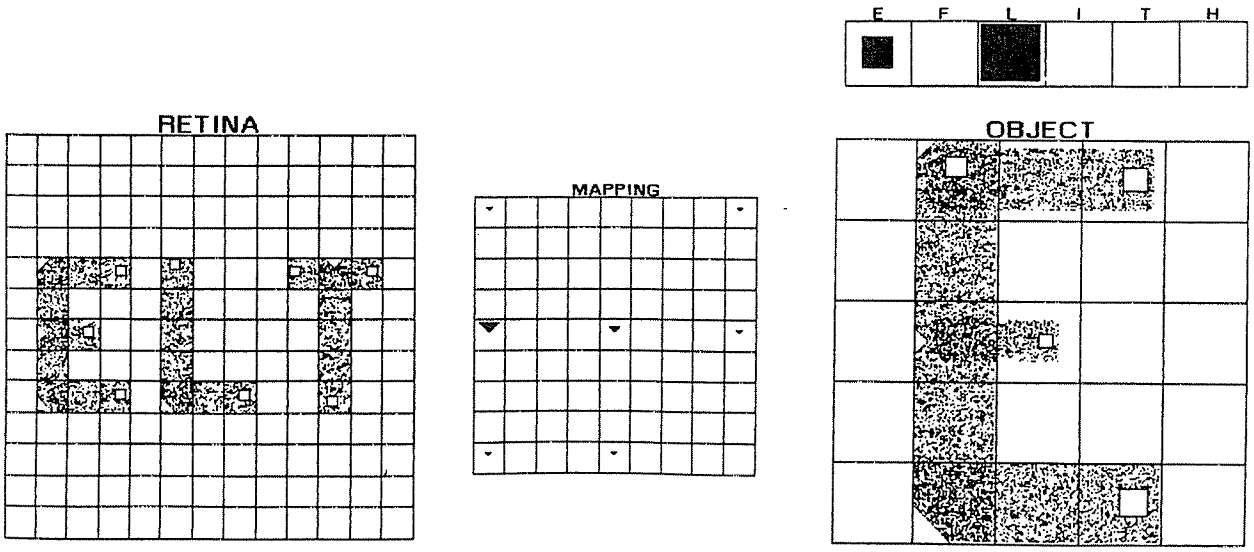

Figure 2-2: A Normal Simulation

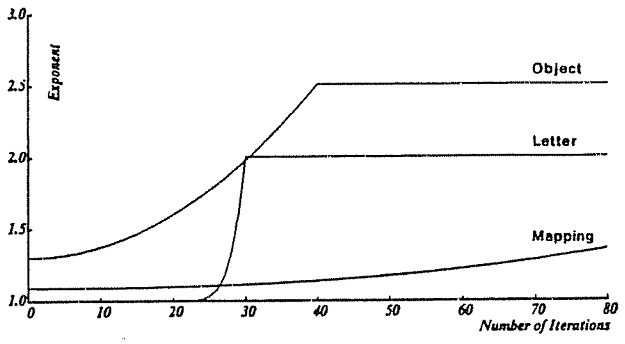

Figure 2-3: Competition as a Function of Time

2.4.2 Backward Masking

In the absence of interference, the network always works correctly. On the other hand, if the input is replaced by noise at some point in the relaxation process, a variety of errors can occur, depending on when the disruption occurs. If a backward mask replaces the input very early, the network will end up in a random state completely unrelated to its original input. If the mask is introduced in the middle of the simulation when the selection process is already underway, the relaxation will either finish correctly or produce an illusory conjunction, depending on how the noise interacts with the unstable activations in the network. For an example of a simulation that produced an illusory conjunction, look at figure 2-4, which belongs to the same sequence as figure 2-2. The letter unit for L has gone on to win, and there is a fairly clear image of an L on the object plane, but the mapping units no longer have any meaningful information on the retinotopic plane with which they could correlate that image. Consequently, the leftmost mapping unit proceeds to win in this case, which would produce the perception of an L on the left to any sort of binding mechanism which looks at the letter and mapping planes to determine what was perceived. It is interesting to note that in this case it is the absence of meaningful information on the retinotopic plane, and not the noise *per se* that causes the error. Human subjects need a random mask to produce this type of error because information persists in their lower vision centers unless it is actively overwritten.

3. Discussion

The model we have described is incomplete in many ways. It ignores the problem of integrating perception across many fixations, though the scheme can be extended to allow this (Hinton, 1981b). It also uses a two-dimensional domain instead of a three-dimensional one, though the extension to 3-D would be feasible if the scheme could be made more efficient. At present, the network requires too much hardware. Even if we restrict ourselves to rigid transformations, our scheme requires N^2 gated connections to map N retinotopic features through N possible mappings into N object-based features.

One way to save hardware is to take advantage of the *structure* of the set of possible mappings. The scheme we presented would work for *any* set of one-to-one mappings. But the allowable spatial transformations are much more restricted than this. For example, small changes in the mapping cause small changes in the parameters of the object-based feature to which a given retinotopic feature maps. This means that it is possible to use "coarse coding" (Hinton, 1981b) to economize on units. Coarse coding uses activity in a unit to represent a whole collection of similar alternatives and it represents each specific alternative by the joint activity of many such units. It

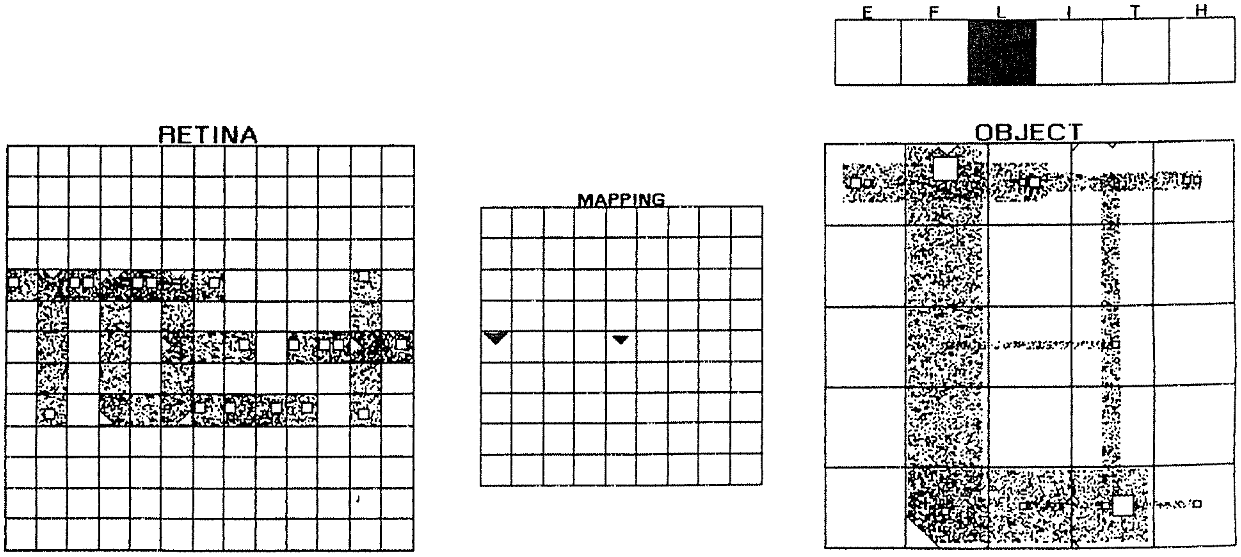

Figure 2-4: A Simulation Disrupted by a Random Mask

relies on local linearity: If B lies between A and C, then the representation of B must have the average of the effects of the representations of A and C. On a local scale, this holds for spatial features and their transformations.

An even stronger property of the set of rigid transformations is that each of them can be expressed as a matrix which operates on a vector containing the retinotopic parameters of one feature to produce a vector containing the object-based parameters of the transformed feature. Arbitrary one-to-one mappings cannot be expressed in this way. It should be possible to make great economies by representing features and mappings as collections of parameters rather than as single active units, though we know of no connectionist scheme which fully exploits this possibility.

Another way of economizing on gated connections is to introduce several sequential stages into the network (Ballard, 1985; Ballard and Sabbah, 1983). Each stage handles one aspect of the overall transformation. A typical decomposition is to let one stage handle translation and the next stage handle rotation and scaling. If there are six degrees of freedom in the total transformation and each degree of freedom has d discriminable values, this two stage method reduces the number of gated connections per retinotopic feature from d^6 to $2 \times d^3$. However, it also slows down the relaxation process and makes it much harder to ensure that it converges to a sensible solution.

A possible criticism of the simulation is that we have not *proved* that the network converges, and we have not given any *systematic* procedure for tuning the connection strengths. Hopfield (1982, 1984) and Hummel and Zucker (1983) have shown that in networks with symmetrical connections (like ours), there is an "energy" function which governs the behavior of the network. If each unit computes the derivative of the energy function and updates its state accordingly, the network is guaranteed to find an energy minimum. Our relaxation procedure has similarities to the model in Hopfield (1984). The use of a variable power law to suppress weak activations plays the same role as the variable gain in Hopfield's model. A further elaboration is to use a stochastic decision rule (Hinton & Sejnowski, 1983; Geman and Geman, 1984). This allows networks to escape from local energy minima and it also leads to a simple local algorithm for tuning the weights (Ackley, Hinton and Sejnowski, 1985). We have deliberately avoided using these more sophisticated relaxation techniques in this simulation because they introduce extra complexity which simply obscures the relationship between parallel cooperative models of shape perception and illusory conjunctions.

The use of coarse coding or multiple sequential stages or stochastic relaxation would not remove the tendency of these networks to produce illusory conjunctions, and so it would not affect our central result: Illusory conjunctions are a natural consequence of using relaxation to search for consistent states of a parallel network in which connections between retinotopic and object-based features are gated by mapping units that explicitly represent the viewpoint.

References

Ackley, D. H., Hinton, G. E., Sejnowski, T. J. A learning algorithm for Boltzmann machines. *Cognitive Science*, 1985, **9**, 147-169.

Ballard, D. H., Cortical connections and parallel processing: Structure and function. *The Behavioral and Brain Sciences* in press.

Ballard, D. H. & Sabbah, D. Viewer independent shape recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983, **PAMI-5**, 653-659.

Geman, S., & Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984, **PAMI-6**, 721-741.

Hinton, G. E. A parallel computation that assigns canonical object-based frames of reference. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vol 2. Vancouver BC, Canada. August 1981a.

Hinton, G. E. Shape representation in parallel systems. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vol 2. Vancouver BC, Canada. August 1981b.

Hinton, G. E. & Sejnowski, T. J. Optimal perceptual inference. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, Washington DC, June 1983.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA*, 1982, **79** 2554-2558.

Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA*, 1984, **81**, 3088-3092.

Hummel, R. A., & Zucker, S. W. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983, **PAMI-5**, 267-287.

Marr, D. & Nishihara, H. K. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society. London B*, 1978, **200**, 269-294.

Rock, I., Orientation and Form. New York: Academic Press, 1973.

Treisman, A. M. & Schmidt, H. Illusory conjunctions in the perception of objects *Cognitive Psychology*, 1982, **14**, 107-141.