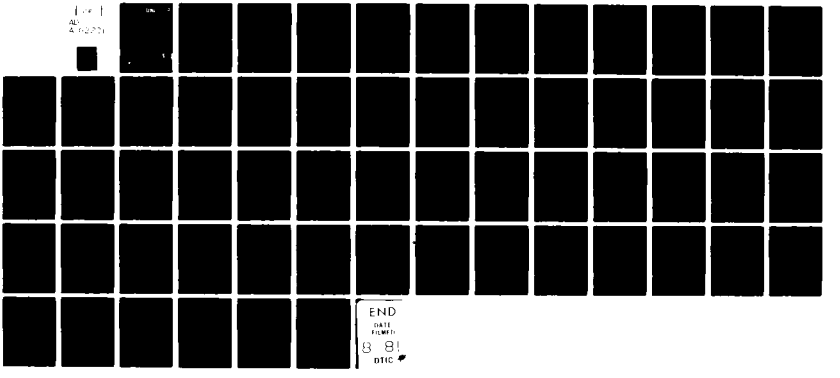
AD-A102 221 ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE F/G 9/2
COMPUTING WITH CONNECTIONS,(U)
APR 81 J A FELDMAN, D H BALLARD N00014-78-C-0164
UNCLASSIFIED TR-72 NL



END
DATE FILMED
8 81
DTIC

LEVEL II

12

AD A102221

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC FILE COPY

rochester

DTIC
ELECTE
S JUL 30 1981

Department of Computer Science
University of Rochester
Rochester, New York 14627

81 6 15 169

# Computing with Connections

J.A. Feldman and D.H. Ballard
Computer Science Department
University of Rochester
Rochester, NY 14627

TR72
April 1981

## Abstract

There is a rapidly growing interest in problem-scale parallelism, both as a model of animal brains and as a paradigm for VLSI. Work at Rochester has concentrated on connectionist models and their application to vision. This paper lays out a framework for dealing with such problems. The framework is built around computational modules, the simplest of which are termed p-units. We develop their properties and show how they can be applied to a variety of problems.

To show how the framework can be applied to computational problems in vision, two specific examples are developed in some detail. In the first, we describe how spatially distributed data can be associated with a complex concept. In the second, we discuss the shape from shading problem and show how a global parameter, such as light source position, interacts with the calculation of a spatially distributed parameter such as surface orientation.

The preparation of this paper was supported in part by the Defense Advanced Research Projects Agency, monitored by the ONR, under Contracts N00014-78-C-0164 and N00014-80-C-0197

## Table of Contents

## 1. Introduction

Animal brains do not compute like a serial computer. Comparatively slow (millisecond) neural computing elements with complex, parallel connections form a structure which is dramatically different from a high-speed, predominantly serial machine. Much of current research in the neurosciences is concerned with tracing out these connections and with discovering how they *transfer information*. One purpose of this paper is to suggest how connectionist theories of the brain can be used to produce testable, detailed models of interesting behaviors.

Artificial intelligence and articulating cognitive sciences have made great progress by employing models based on conventional digital computers as theories of intelligent behavior. But a number of crucial phenomena such as associative memory, priming, perceptual rivalry, and the remarkable recovery ability of animals have not yielded to this treatment. The other major goal of this paper is to lay a foundation for the systematic use of massively parallel connectionist models in the cognitive sciences, even where these are not yet reducible to physiology.

The connectionist view of brain and behavior is that all encodings of importance in the brain are in terms of the relative strengths of synaptic connections. The fundamental premise of connectionism is that individual neurons *do not transmit large amounts of symbolic information.* Instead they compute by being *appropriately connected* to large numbers of similar units. This is in sharp contrast to the conventional computer model of intelligence prevalent in computer science and cognitive psychology. While the connectionist view has a much stronger physiological foundation, explicit models of behavior have been almost exclusively cast in the framework of computer-like information processing models. Connectionism has been associated with a pre-computational view that knowing the connection structure of a system is all that is required for its understanding. Recent advances in digital hardware, vision research, and the theory of computation have caused renewed interest in highly parallel computational models more in keeping with the connectionist paradigm. It now appears to be feasible to construct models which are simultaneously structurally and functionally sound.

The fundamental distinction between the conventional and connectionist computing models can be grasped in the following example. When we see an apple and say the phrase "wormy apple," some information must be transferred, however indirectly, from the visual system to the speech system. Either a sequence of special *symbols* that denote a wormy apple is transmitted to the speech system, or there are special *connections* to the speech command area for the words. Figure 1 is a graphic presentation of the two alternatives. The path on the right described by double-lined arrows depicts the situation (as in a computer) where the information that a wormy apple has been seen is encoded by the visual system and sent as an abstract message (perhaps frequency-coded) to a general receiver in the speech system which decodes the message and initiates the appropriate speech act. We have not encountered anyone who will defend this model as biologically plausible.

Figure 1: Connectionism vs. Symbolic Encoding.

The only alternative that we have been able to uncover is described by the path with single-width arrows. This suggests that there are (indirect) links from the units (cells, columns, centers, or what-have-you) that recognize an apple to some units responsible for speaking the word. The connectionist model requires only very simple messages (e.g. stimulus strength) to cross a channel but puts strong demands on the availability of the right connections.

Over the past few years, we have been exploring the efficacy of formulating detailed models of intelligent behavior directly in connectionist terms. This kind of effort is in the tradition of McCullogh-Pitts machines and Perceptrons and has long been viewed as a good way of attacking problems in low-level vision. Until recently, work in this mode has been mainly just suggestive: examining properties of networks, attempting to match wave-forms, etc. There was little of the detailed specification of non-trivial behavioral models which characterizes AI and cognitive psychology. Currently, a great deal of successful vision work in this laboratory and elsewhere has its basis in highly parallel models [Hanson and Riseman, 1978]. One particularly fruitful insight for us has been the correspondence between the so-called Hough techniques [Ballard, 1981a] and

connectionist models. We are continuing to work on detailed parallel models of visual functions [Ballard, 1981b; Sabbah, 1981; Ballard and Sabbah, 1981] and some examples will be used as illustrations in this paper.

But the connectionist dogma suggests that all mental functions, not just low-level vision, can be well described in terms of richly connected networks transmitting very simple signals. We have done some preliminary work [Feldman, 1980; 1981] on laying out the advantages and difficulties in such an approach. The purpose of this paper is to prepare a solid foundation for the construction of detailed connectionist models. This involves defining a set of primitive units, considering some of their properties, and using these to solve some problems that seem to be prerequisite to any widespread use of connectionist models.

The body of this paper has four sections. Section Two contains the basic definitions for a tractable and biologically plausible neuron-level computing unit. Although there is a rich tradition of neural modeling research, much of which will be useful to us, our definitions depart from standard ones. A primitive unit can have both symbolic and numerical state, can treat its inputs non-uniformly, and need not compute a linear function. A particularly important construct is the use of groups or "conjunctions" of input connections. Some important special cases and some simple examples, based on lateral inhibition, are presented. Encapsulation techniques are suggested as a basis for simplifying larger problems.

Section Three is concerned with the general computing abilities of networks of our units. The crucial point is achieving a single coherent action in a diffuse set of units. Winner-take-all (WTA) networks are introduced as our solution to this problem for single layers. More generally, we define and study the idea of a stable coalition of units whose mutual reinforcement has the effect of a single action, perception, etc.

Section Four concentrates on some specific computations and how they can be effectively performed within the model. We begin with computing simple functions like multiplication and show how general parameters can be treated. Modifiers and mappings are used to show how connections can effectively be treated as dynamic. An extension of this idea allows us to treat time-varying data like speech.

In Section Five we tackle some additional classic problems for connectionism and apply our ideas to some more problems in visual perception. A representation for conjunctive concepts such as "big blue cube" is laid out and applied to the description of complex objects. Finally, as another indication of the way we intend to proceed, a fairly detailed connectionist model of shape-from-shading computations is presented.

## 2. Neuron-Like Computing Units

### Definitions

As part of our effort to develop a generally useful framework for connectionist theories, we have developed a standard model of the individual unit. It will turn out that a "unit" may be used to model anything from a small part of a neuron to the external functionality of a major subsystem. But the basic notion of unit is meant to loosely correspond to an information processing model of our current understanding of neurons. The particular definitions here were chosen to make it easy to specify detailed examples of relatively complex behaviors. There is no attempt to be minimal or mathematically elegant. The various numerical values appearing in the definitions are arbitrary, but fixed finite bounds play a crucial role in the development. The presentation of the definitions will be in stages, accompanied by examples. A compact technical specification for reference purposes is included as Appendix A.

Each unit is a computational entity comprising

{q} -- a set of *discrete states*, < 10

p -- a continuous value in [-1,1], called *potential* (accuracy of 10 digits)

v -- an *output value*, integers $0 \leq v \leq 9$

i -- a vector of *inputs* $i_1,...,i_n$

and functions from old to new values of these

p <- f(i,p,q)

q <- g(i,p,q)

v <- h(i,p,q)

*which we assume, for now, to compute continuously.* The form of the f, g, and h functions will vary, but will generally be restricted to conditionals and functions found on hand calculators. There are both biological and computational reasons for allowing units to respond (for example) logarithmically to their inputs. The "<-" notation is borrowed from the assignment statement of programming languages. This notation covers both continuous and discrete time formulations and allows us to talk about some issues without any explicit mention of time. Of course, certain other questions will inherently involve time and computer simulation of any network of units will raise delicate questions of discretizing time.

### P-Units

For some applications, we will be able to use a particularly simple kind of unit whose output v is proportional to its potential p (rounded) and which has only one state. In other words

$p <- p + \beta \Sigma i_k$ $\quad [-1 \leq p \leq 1]$

$v = \alpha p - \theta$ $\quad [v = 0...9]$

where $\beta$, $\alpha$, $\theta$ are constants

The p-unit is somewhat like classical linear threshold elements (Perceptrons [Minsky and Papert, 1972]), but there are several differences. The potential, p, is a crude form of memory and is an abstraction of the instantaneous membrane potential that characterizes neurons.

The restriction that output take on small integer values is central to our enterprise. The firing frequencies of neurons range from a few to a few hundred impulses per second. In the 1/10 second needed for basic mental events, there can only be a limited amount of information encoded in frequencies. The ten output values are an attempt to capture this idea. A more accurate rendering of neural events would be to allow 100 discrete values with noise on transmission (cf. [Sejnowski, 1977]). If it turns out that local "graded" potentials cannot be effectively quantized, the definitions will have to be extended to allow local exchange of continuous information. Transmission time is assumed to be negligible; delay units can be added when transit time needs to be taken into

account.

**Example 1**

One problem with the definition above of a p-unit is that its potential does not decay in the absence of input. This decay is both a physical property of neurons and an important computational feature for our highly parallel models. One computational trick to solve this is to have an inhibitory connection from the unit back to itself.

Figure 2: Self-Inhibition and Decay.

We will follow the usual notation that a connection with a circular tip is inhibitory. More complex networks will sometimes be specified by a connection table instead of a diagram. Informally, we identify the negative self feedback with an exponential decay in potential which is mathematically equivalent. We will specify this more carefully below and add the notion of *weights* on inputs.

The first step is to elaborate the input vector i in terms of received values, weights, and modifiers:

$$\forall j,\ i_j = r_j \cdot w_j \cdot m_j \qquad j = 1,\ldots,n$$

where $r_j$ is the *value* received from a predecessor [r = 0...n]; $w_j$ is a changeable *weight*, unsigned [0 $\leq w_j \leq$ 1] (accuracy of 10 digits); and $m_j$ is a synapto-synapuc *modifier* which is either 0 or 1.

The weights are the only thing in the system which can change with experience. They are unsigned because we do not want a connection to change sign. The modifier or gate greatly simplifies many of our detailed models in Section 4. One could, of course, use extra units instead, but the biological evidence for blocking inhibition is solid.

**Lateral Inhibition, Several Cases**

Mutual lateral inhibition is widespread in nature and has been one of the basic computational schemes used in modeling. We will present two examples of how it works to help aid in intuition as well as to illustrate the notation. The basic situation is symmetric configurations of p-units which mutually inhibit one another. Time is broken into discrete intervals for these examples. The examples are too simple to be realistic, but do contain ideas which we will employ repeatedly.

**Example 2: Two P-Units Symmetrically Connected**

Suppose v $\doteq$ 10p, $w_1$ = .1, $w_2$ = .05(-). It is easier to use P = 10p internally and round output:

$$P(t+1) = P(t) + r_1 - (.5)r_2 \qquad r_j = \text{received}$$
$$v = \text{round } (P) \quad [0...9]$$

Referring to Figure 3, suppose the initial input to the unit A.1 is 6, then 2 per time step, and the initial input to B.1 is 5, then 2 per time step.

Figure 3: Two P-Units Symmetrically Connected, with Table.

This system will stabilize to the side of the larger of two instantaneous inputs.

It is interesting to also look at a continuous version of this example. The continuous approximation to the defining equations for Example 2 can be written:

$$P'_1 = 2 - .5P_2$$
$$P'_2 = 2 - .5P_1$$

where $P_1$ is ten times the potential of A and $P_2$ of B and where $P'_1$ is the derivative of $P_1$ with respect to time. This system of linear differential equations can be solved by standard techniques for the initial conditions $P_1 = 6$, $P_2 = 5$. The solutions are

$$P_1 = 4 + 1/2e^{1/2t} - 3/2e^{-1/2t}$$
$$P2 = 4 - 1/2e^{1/2t} - 3/2e^{-1/2t}$$

First note that the last term in each equation is a negative exponential and can be neglected. The resulting relation indicates clearly the rapid decay of $P_2$ and rise of $P_1$. Linear systems theory is only an approximation to our models which in general are nonlinear. For example, the above equations do not take into account the fact that the potentials saturate. Nonetheless, the theory can be an important aid in understanding some properties of our networks.

**Example 3: Two Symmetric Coalitions of 2-Units**

$v \doteq 10p$
$w_1 = .1$
$w_2 = .05$
$w_3 = .05(-)$
$P(t+1) = P(t) + r_1 + .5r_2 - .5r_3$
$v = \text{round}(P)$
A,C start at 6; B,D at 5;
A,B,C,D have no external input for t>1

Figure 4: Two Symmetric Coalitions of 2-Units, with Table.

This system converges faster than the previous example. The idea here is that units A and C form a "coalition" with mutually reinforcing connections. The competing units are *A vs. B and C* vs. D. Example 3 is the smallest network depicting what we believe to be the basic mode of operation in connectionist systems. One can imagine, e.g., that C and D are competing phonemes and that A and B are words which incorporate C and D, respectively.

We have already described the graphical notation which will often be used in examples. The alternative method is to describe, for each unit, the outgoing connections to other units in tabular form. Each outgoing $v_j$ (only one for basic units) will have a set of entries of the form

(<receiving unit>.<index>,<±>,<type>)

where any of the last three constructs can be omitted and given its default value. The <±> field specifies whether the link is excitatory (+) or inhibitory (-) and defaults to +. The <index> is the input index j in $r_j$ at the receiving end. This index can be used for specifying different weights as in the examples above. Indexed inputs also allow for functionally different use of various inputs and many of our examples will exploit this feature. The <type> is either normal, modifier (m), or learning (x), the default being normal.

For example, the diagram of Example 2 could be replaced by the table:

A: B.2, -
Z1
B: A.2, -
Z2
Y1: A.1
Y2: B.2

where units labeled Y, Z, etc., designate unnamed sources and sinks.

Competing coalitions of units will be the organizing principle behind most of our models. Consider the two alternative readings of the Necker cube shown in Figure 5. At each level of visual processing, there are mutually contradictory units representing alternative possibilities. The dashed lines denote the boundaries of coalitions which embody the alternative interpretations of the image. A number of interesting phenomena (e.g. priming, perceptual rivalry, subjective contour) find natural expression in this formalism. We are engaged in an ongoing effort [Sabbah, 1981; Ballard, 1981b] to model as much of visual processing as possible within the connectionist framework. This paper is largely an exercise in developing standard mechanisms for this and other specific modeling projects.

Figure 5: Necker Cube.

### Q-Units and Compound Units

Another useful special case arises when one suppresses the numerical potential, p, and relies upon the finite-state set {q} for modeling. If we also identify each input of i with a separate named input signal, we can get classical finite automata. A simple example would be a unit that could be started or stopped from firing.

One could describe the behavior of this unit by a table, with rows corresponding to states in {q} and columns to possible inputs, e.g.,

| | $i_1$ (start) | $i_2$ (stop) |
|---|---|---|
| Firing | Firing | Null |
| Null | Firing | Null |

The table above is a tabular presentation of our simplified generic function, g = g(i,q) which describes state changes. In a similar manner, the computation v <- h(i,p,q) could be simplified to v <- h(q), e.g.,

v <- if q = Firing then 6 else 0.

This could also be added to the table above.

We have already employed a variety of graphical and textual descriptions of units and collections of them. The paper will continue to use different representations, but these are all instances of the general definition. One of the most powerful techniques employed will be encapsulation and abstraction of a subnetwork by an individual unit. For example, assume that some system had separate motor abilities for turning left and turning right (e.g. fins). We could use two start-stop units to model a turn-unit.

Figure 6: A Turn Unit.

There are two important points. The compound unit here has two distinct outputs, where basic units have only one (which can branch, of course). In general, compound units will differ from basic ones only in that they can have several distinct outputs.

The main point of this example is that the turn-unit can be described abstractly, independent of the details of how it is built. For example, using the tabular conventions described above,

| | Left | Right | Values | Output |
|---|---|---|---|---|
| a gauche | a gauche | a droit | $V_1=7$, | $V_2=0$ |
| a droit | a gauche | a droit | $V_1=0$, | $V_2=8$ |

where the right-going output being larger than the left could mean that we have a right-finned robot. There is a great deal more that must be said about the use of states and symbolic input names, about multiple simultaneous inputs, etc., but the idea of describing the external behavior of a system only in enough detail for the task at hand is at the core of our enterprise. This is one of the few ways known of coping with the complexity of the magnitude needed for serious modeling of biological functions. It is not strictly necessary that the same formalism be used at each level of functional abstraction and, in the long run, we may need to employ a wide range of models. For example, for certain purposes one might like to expand our units in terms of compartmental models of neurons like those of [Perkel, 1979]. The advantage of keeping within the same formalism is that we preserve intuition, mathematics, and the ability to use existing simulation programs.

The idea of encapsulation used in compound units is vital, but one should not think that only a small number of units are involved in output; rather, only a *small fraction* of the units in the subsystem are output units. Some simple biological systems (such as in leech [Stent et al., 1978] or lobster [Warshaw and Hartline, 1976]) might be able to be completely modeled on the above scale. But we are more concerned here with complex systems like human vision, etc. For this purpose we will need yet more abstraction techniques (see below). In human vision even loose coupling will involve a large number of connections between subsystems, e.g. vestibular and vision.

**Units Employing p and q**

It will already have occurred to the reader that a numerical value, like our p, would be useful for modeling the amount of turning to the left or right in the last example. It appears to be generally true that a single numerical value and a small set of discrete states combine to provide a powerful yet tractable modeling unit. This is one reason that the current definitions were chosen. Another reason is that the mixed unit seems to be a particularly convenient way of modeling the information processing behavior of neurons, as generally described. The discrete states enable one to model the effects in neurons of abnormal chemical environments, fatigue, etc. One example of a unit employing both p and q non-trivially is the following crude neuron model. This model is concerned with saturation and assumes that the output strength, v, is something like average firing frequency. It is not a model of individual action potentials and refractory periods.

We suppose the distinct states of the unit $q \in$ {normal, recover}. In *normal* state the unit behaves like a p-unit, but while it is *recovering* it ignores inputs. The following table captures almost all of this behavior.

| | | $-.1<p\leq.9$ | $p>.9$ | Output Value |
|---|---|---|---|---|
| (incomplete) | normal | p<- p+S1 | p<- -p/ recover | v <- a p - q |
| | recover | normal | <impossible> | v <- 0 |

Here we have the change from one state to the other depending on the value of the potential, p, rather than on specific inputs. The recovering state is also characterized by the potential being set negative. The unspecified issue is what determines the duration of the recovering state--there are several possibilities. One is an explicit dishabituation signal like those in Kandel's experiments [Kandel, 1976]. Another would be to have the unit sum inputs in the recovering state as well. The reader might want to consider how to add this to the table.

A third possibility, which we will use frequently, is to assume that the potential, p, decays toward zero (from both directions) unless explicitly changed. Example 1 showed how this implicit decay $p \leftarrow p_0 e^{-kt}$ can be modeled by self inhibition. In this case, the decay constant, k, would

determine the length of the recovery period.

### Disjunctive Firing Conditions

It is both computationally efficient and biologically realistic to allow a unit to respond to one of a number of alternative conditions. One way to view this is to imagine the unit having "dendrites" each of which depicts an alternative enabling condition.

Figure 7. A Unit with Disjunctive Inputs.

In terms of our formalism, this could be described in a variety of ways. One of the simplest is to define the potential in terms of the maximum of the three separate computations, e.g.,

$$p \leftarrow p + \mathrm{Max}(i_1+i_2,\ i_3+i_4,\ i_5+i_6-i_7)$$

It does not seem unreasonable (given current data) to model the firing rate of a unit as the maximum of the rates at its active sites. Units whose potential is changed according to the maximum of a set of algebraic sums will occur frequently in our specific models.

One could replace this unit with three simple p-units plus a maximum unit and get a similar effect. (Note that the potentials of the p-units wouldn't be equal.) But it appears to be easier to understand and analyze systems of units that describe intuitively coherent computations. Another reason for employing disjunctive units is that they appear to be wide-spread in nature. The firing of a neuron depends, in many cases, on local spatio-temporal summation involving only a small part of the neuron's surface. So-called dendritic spikes transmit the activation to the rest of the cell. It also turns out that inhibitory inputs sometimes block such internal signals that are upstream of the point of inhibition, rather than just sum with them. It is possible to model a dendritic tree with inhibitory blocking inputs all within our formalism for a single unit, or as a simple network. One can model each section of the dendritic tree as a unit which sends output to the unit body unless it is blocked by a modifier ($m_j$) input, corresponding to blocking inhibition in neurons. One advantage of keeping the processing power of our abstract unit close to that of a neuron is that it helps inform our counting arguments. When we attempt to model a particular function (e.g. stereopsis), we expect to require that the number of units and connections as well as the execution time required by the model are plausible.

### Conjunctive Connections

The max-of-sum unit is the continuous analog of a logical OR-of-AND (disjunctive normal form) unit and we will sometimes use the latter as an approximate version of the former. The OR-of-AND unit corresponding to Figure 7 is:

$$p \leftarrow p + \alpha\ \mathrm{OR}\ (i_1 \& i_2,\ i_3 \& i_4,\ i_5 \& i_6 \& (\mathrm{not}\ i_7))$$

This formulation stresses the importance that nearby spatial connections *all* be firing before the potential is affected. Hence, in the above example, $i_3$ and $i_4$ make a *conjunctive connection* with the unit.

### Change

For our purposes, it is useful to have all the adaptability of networks be confined to changes in weights. While there is known to be some growth of new connections in adults, it does not appear to be fast or extensive enough to play a major role in learning. For technical reasons, we consider very local growth or decay of connections to be changes in existing connection patterns. Obviously, models concerned with developing systems would need a richer notion of change in connectionist networks (cf. [von der Malsburg and Willshaw, 1977]). Learning and change will not be treated technically in this paper, but the definitions are included for completeness. We provide each unit with a change function c:

$$\mu \leftarrow c(i,p,q,x,\mu)$$

where $\mu$ is the intermediate-term memory vector, i, p, and q are as always, and x is an additional single integer input ($0 \leq x \leq 9$) which captures the notion of the importance and value of the current behavior. Instantaneous establishment of long-term memory (which does not seem plausible) would be equivalent to having $\mu = w$. We are assuming that the consolidation of long-term changes is a separate process.

We assume that important, favorable or unfavorable, behaviors can give rise to faster learning. The rationale for this is given in [Feldman, 1980; 1981], which also lays out informally our views on how short- and long-term learning could occur in connectionist networks. We are working on a more technical presentation of our model of change along the lines of this paper. Obviously enough, a plausible model of learning and memory is a prerequisite for any serious scientific use of connectionism. But we have found that an examination of networks for carrying out the basic building blocks is already enough for one report.

## 3. Networks

Our general idea of temporal behavior in networks is that of *relaxation*. The independent inputs together with the various inter-unit connections are sufficient to cause the networks to behave in an appropriate manner; each unit should converge to a potential value between -1 and 1. Much work has been done on relaxation, from classical Gauss-Sidel iterations to more modern applications in vision (e.g. [Rosenfeld et al., 1976; Marr and Poggio 1976; Prager 1980; and Hinton, 1980]). With a few exceptions, previous work has assumed linear behavior (or linear with a threshold). As one of the exceptions, Prager used a non-linear model and noted that it enabled him to use more complicated updating conditions than those in a linear system. Our model also breaks with the linear traditions in its use of conjunctive connections and state tables. While we still use linear approximations to analyze the stability of the system, the non-linear units are closer to actual neurons in behavior and allow vast simplifications in network design.

### Winner-Take-All Networks

A very general problem that arises in any distributed computing situation is how to get the entire system to make a decision (or perform a coherent action, etc.). This is a particularly important issue for the current model because of its restrictions on information flow and because of the almost linear nature of the p-units used in many of our specific examples. One way to deal with the issue of coherent decisions in a connectionist framework is to introduce winner-take-all (WTA) networks, which have the property that only the unit with the highest potential (among a set of contenders) will have output above zero after some settling time. Biologically necessary examples of this behavior abound; ranging from turning left or right, through fight-or-flight responses, to interpretations of ambiguous words and images.

There are a number of ways to construct WTA networks from the units described above. We will discuss several of these, both because of the importance of WTA capabilities and because it is the first non-trivial problem treated here. The question of identical values (ties) is an important one, but will be deferred for a few paragraphs. Our first example of a WTA network will operate in one time step for a set of contenders each of whom can read the potential of all of the others. (The fan in/out of neurons is about 1,000-10,000.) Each unit in the network computes its new potential according to the rule:

$$p \leftarrow \text{if } p > \max(i_j, .1) \text{ then } p \text{ else } 0.$$

That is, each unit sets itself to zero if it knows of a higher input. This is fast and simple, but probably a little too complex to be plausible as the behavior of a single neuron. There is a standard trick (apparently widely used by nature) to convert this into a more plausible scheme. We replace each unit above with two units; one computes the maximum of the competitor's inputs and inhibits the other. This is shown in Figure 8.

Figure 8: Paired Units for Max WTA.

There are a number of remarks in order. It is not biologically unreasonable to view the firing rate of a neuron to be the maximum of the rates of its separate sites of spatio-temporal summation. The circuit above can be strengthened by adding a reverse inhibitory link, or one could use a modifier on the output, etc. Obviously one could have a WTA layer that got inputs from some set of competitors and settled to a winner when triggered to do so by some downstream network. This is an exact analogy of strobing an output buffer in a conventional computer. Another set of standard ideas (here from theoretical computer science) enables us to build WTA networks among sets of contenders larger than the allowable fan-in of units. We just arrange the competitors in a tournament tree [Aho, Hopcroft, and Ullman, 1974] and have the winners at each level play off. The time required is the height of the tree which is the logarithm (to the base fan in) of the size of the set of contenders and is small for all realistic situations.

The question of ties remains to be considered. Since we are assuming only a limited range of output values, quite a few contenders might appear to be equal. Depending on how the WTA network is being employed, one might want several different ways of treating this situation. A

common idea in computer science is to order the units in some way and have the first in order win in the case of ties. This is easy to implement by having units turn off if a predecessor is higher or equal to itself. In some situations, a random choice might be appropriate. There is reason to believe that essentially random effects break ties in real neural networks. Randomness can be achieved in our scheme, e.g., by adding a randomly changing hierarchy, but we will not be using random selection in this paper.

There are two more basic ways of treating ties that deserve mention. One could try to resolve ties by looking ever more closely at the values of potential among contenders. This would amount to having "rounds" of competition. First, all units whose high-order digit was sub-maximal would drop out. Then there would be a play-off based on the second digit, etc. This could be combined with the tournament tree, but, in the end, one still might have to contend with ties. There do seem to be situations where some fine-tuning is called for, but the most common situation appears to be quite different.

Recall that the purpose of WTA networks was to identify a clear winner out of a set of contending actions, perceptions, etc. Both in nature and in our models, this rarely occurs in the form of pure competition among a single layer of contenders. For example, the choice of which word should be assumed to have been heard is influenced by phonemic, semantic, contextual and general considerations. We believe that WTA type structures exist, but that they are normally part of coalitions spanning many layers. Ties in a single WTA layer do not require specific resolution because the coalition interaction normally will produce a unique overall winner. The idea of coalitions among members of different competing layers was discussed briefly in Example 3 and will receive a great deal of attention below.

Multi-layer coalition networks could employ MAX-based WTA circuits, but it often seems more appropriate to algebraically combine the outputs of units. For this reason, and to tie in with some important related work, we will now consider WTA circuits based primarily on p-units (which algebraically sum their inputs).

First we present an abstract solution of the WTA problem which ignores quantization and bounds. Suppose we have a symmetric network of n+1 p-units, each of which equally inhibits all the others, i.e.,

$$p \leftarrow p - 1/10n \sum_{j \neq k} v_j + ? \; ; \; (v = 10p)$$

If we add one extra unit, AVE, which computes the average of all active (non-zero) outputs and feeds it (with + polarity) to all the units, we get the desired subnetwork.

$$p \leftarrow p - 1/10n \sum_{j \neq k} v_j + 1/10b \sum_{b} v_j$$

where b is the number of non-zero inputs to AVE.

This network has the required behavior because each unit has its potential increased by the difference between the average of all outputs and the average of all but its own output. Units whose output is above average will increase while the others decrease. As units go to zero and drop out, more units go below average. One instance of this would be when a subnetwork with all $p_j$ initially equal got outside signals which favored one unit. Notice that AVE is not a p-unit, since it counts non-zero inputs.

A possible problem arises if one takes saturation into account. If two units were both near saturation, they might easily both reach saturation before the WTA network settled down. For any network there will be a difference so small that the intent is that the two values are considered identical. For differences larger than this, one can design the WTA network to converge slowly enough to prevent multiple unequal units from reaching saturation. This is accomplished by giving less weight to the positive input from the AVE unit, still assuming that the output can have

continuous values.

Quantization of output values (here 0...9) adds interesting additional issues. For a sufficiently large network, ten distinct values will not be enough to resolve the difference between the two averages. There are a variety of computational tricks to exploit the limited dynamic range available. Some of these, like tournament trees and successive digit comparisons, were mentioned in the discussion of ties. But the restriction to simple signals is at the heart of our approach and should not be evaded. We should not build models in which WTA networks involve a large number of alternatives nor should we expect very delicate decisions to be made by a single competitive network.

**The Question of Delicacy**

One problem with previous neural modeling attempts is that the circuits proposed were unnaturally delicate (unstable). Small changes in parameter values would cause the networks to oscillate or converge to incorrect answers. We will have to be careful not to fall into this trap, but would like to avoid detailed analysis of each particular model for delicacy. What appears to be required are some building blocks and combination rules that preserve the desired properties. For example, the WTA subnetworks of the last example will not oscillate in the absence of oscillating inputs. This is also true of any symmetric mutually inhibitory subnetwork. This is intuitively clear and could be proven rigorously under a variety of assumptions (cf. [Grossberg, 1980]).

One useful principle is the employment of lower-bound and upper-bound cells to keep the total activity of a network within bounds. The idea is an extension of the AVE cell used in the WTA example. Suppose that we add two extra units, LB and UB, to a network which has coordinated output. The LB cell compares the total (sum) activity of the units of the network with a lower bound and sends positive activation uniformly to all members if the sum is too low. The UB cell inhibits all units equally if the sum of activity is too high. Notice that LB and UB can be parameters set from outside the network. Under a wide range of conditions (but not all), the LB-UB augmented network can be designed to preserve order relationships among the outputs $v_j$ of the original network while keeping the sum between LB and UB.

We will often assume that LB-UB pairs are used to keep the sum of outputs from a network within a given range. This same mechanism also goes far towards eliminating the twin perils of uniform saturation and uniform silence which can easily arise in mutual inhibition networks. Thus we will often be able to reason about the computation of a network assuming that it stays active and bounded. We also require that individual units be viewed as part of different subnetworks, which may be simultaneously active. The general issue of interacting subnetworks entails nothing less than the whole enterprise, but we can tackle the question of bounds. If we view each output value $v_j$ in a set of networks comprising n-units as the axis of an n-dimensional space, the UB and LB cells correspond to bounding hyper-planes in this space. The simultaneous imposition of these conditions defines a convex hull, in which the solution must lie. (Geoff Hinton pointed this out.) This could turn out to have singularities if some simultaneous solutions are impossible, but this condition can be checked for in advance.

One problem with the AVE and UB-LB solutions is that they assume that these units can compute *all* of the activity of a network. As we have mentioned, the saturation of potential and limited data transfer rate mean that only an approximation is possible for networks of significant size. Other results in the literature (e.g. [Grossberg, 1980]) have similar limitations. We will have to place less reliance on precise calculations by large networks and more on cooperative computation.

**Stable Coalitions**

For a massively parallel system like we are envisioning to actually make a decision (or do something), there will have to be states in which some activity strongly dominates. We have shown some simple instances of this, in Examples 2 and 3 and the WTA network. But the general idea is that a very large complex subsystem must stabilize, e.g. to a fixed interpretation of visual input.

The way we believe this to happen is through mutually reinforcing coalitions which instantaneously dominate all rival activity. The simplest case of this is Example 3, where the two units A and B form a coalition which suppresses C and D. Phenomenologically, the two renderings of the Necker Cube in Figure 5 can be viewed as alternative stable coalitions. Formally, **a coalition will be called stable when the output of all of its members is *non-decreasing*.**

What can we say about the conditions under which coalitions will become and remain stable? We will begin informally with an almost trivial condition. Consider a set of units {a,b,...} which we wish to examine as a possible coalition, $\pi$. For now, we assume that the units in $\pi$ are all p-units and are in the non-saturated range and have no decay. Thus for each u in $\pi$,

$$p(u) \leftarrow p(u) + Exc - Inh,$$

where Exc is the weighted sum of excitatory inputs and Inh is the weighted sum of inhibitory inputs. Now suppose that $Exc|\pi$, the excitation from the coalition $\pi$ only, were greater than INH, the largest possible inhibition receivable by u, for each unit u in $\pi$, i.e.,

$$(SC) \qquad \forall\ u \in \pi\ ;\ Exc|\pi > INH$$

Then it follows that

$$\forall\ u \in \pi\ ;\ p(u) \leftarrow p(u) + \delta \qquad \text{where } \delta > 0.$$

That is, the potential of every unit in the coalition will increase. This is not only true instantaneously, but remains true as long as nothing external changes (we are ignoring state change, saturation, and decay). This is because $Exc|\pi$ continues to increase (recursively) as the potential of the members of $\pi$ increases. Taking saturation into account adds no new problems; if all of the units in $\pi$ are saturated, the change, $\delta$, will be zero, but the coalition will remain stable.

The condition that the excitation from other coalition members alone, $Exc|\pi$, be greater than any possible inhibition INH for each unit may appear to be too strong to be useful. Observe first that INH is directly computable from the description of the unit; it is the largest negative weighted sum possible. If inhibition in our networks is mutual, the upper-bound possible after a fixed time $\tau$, $INH\tau$, will depend on the current value of potential in each unit u. The simplest case of this is when two units are "deadly rivals"--each gets all its inhibition from the other. In such cases, it may well be feasible to show that after some time $\tau$, the stable coalition condition will hold (in the absence of decay, fatigue, and changes external to the network).

There are a number of interesting properties of the stable coalition principle. First notice that it does not prohibit multiple stable coalitions nor single coalitions which contain units which mutually inhibit one another (although excessive mutual inhibition is precluded). If the units in the coalition had non-zero decay, the coalition excitation $Exc|\pi$ would have to exceed both INH and decay for the coalition to be stable. We suppose that a stable coalition yields control when its input elements change (fatigue and explicit resets are also feasible). To model coalitions with changeable inputs, we could add boundary elements, whose condition was

$$Exc|\pi + Input > INH$$

and which could disrupt the coalition if its Input went too low.

**An Artificial Example**

The coalitions of units needed to model biologically interesting functions will be large and heterogeneous. We do not yet have mathematical results that enable us to characterize the behavior of general coalitions. What we can do now is develop an artificial example of a coalition and establish the critical aspects of its behavior. This has proven to be useful to us both in aiding intuition and in constraining the choice of weights for real models. (Paul Shields of U. Toledo and Stanford provided the basic analysis.)

The artificial coalition consists of M+1 rows, each of which has N+1 units which compete by mutual lateral inhibition (Figure 9). We are assuming here that each unit can have potential and output values of unlimited range and accuracy and the output is exactly the potential. This makes it

possible to express the competition in a row as a strictly linear rule:

$$X_{mn} \leftarrow X_{mn} - \beta \sum_{j \neq n} X_{mj} + \textit{Coalition Support}$$

Figure 9: Artificially Symmetric Coalition Structure

If we further assume that each coalition is exactly a column and provides positive support proportional to the sum of its members, the rule becomes

$$(*) \qquad X_{mn} \leftarrow X_{mn} + \alpha \sum_{i \neq m} X_{in} - \beta \sum_{j \neq n} X_{mj}$$

Under all these assumptions (*) defines a linear transformation, T, on the collection of values $X_{mn}$ viewed as a "vector" in the sense of linear algebra. This transformation is sufficiently regular that we can characterize all of its eigenvalues and eigen "vectors." Recall that an eigenvalue, $\lambda$, and the associated "vector" X have the property that $TX = \lambda X$. Any such coalition structure, X, will be stable because repeated applications of the relaxation rule (*) will just multiply every element repeatedly by the related $\lambda$. What is interesting here is that the configurations of $X_{mn}$ which have this property are easy to discuss in terms of our model.

Suppose that $X_{mn}$ were such that each column had every one of its elements equal. This might be a good resting state for the structure because any row would provide the same answer as to the relative strengths of the various possibilities. The rule (*) becomes:

$$X_{mn} \leftarrow X_{mn} + \alpha \cdot M \cdot X_{mn} - \beta \sum_{j \neq n} X_{mj}$$

because all M other elements of its column are equal to $X_{mn}$. If we further assume that each row has the sum of all its elements equal to zero, the remaining summation above must be equal to $-X_{mn}$ and we get:

$$X_{mn} \leftarrow X_{mn} + \alpha \cdot M \cdot X_{mn} + \beta X_{mn}$$

or

$$X_{mn} \leftarrow (1 + \alpha M + \beta) X_{mn}$$

which says that $(1 + \alpha M + \beta) = \lambda_1$ is an eigenvalue for T, working on "vectors" with constant columns and zero row-sums. The condition of a zero sum for a row captures the idea of competition quite nicely; the fact that this requires negative values to be transmitted is not a serious problem. It is the assumptions of unbounded scale and accuracy that limit the application of these results even in the case of purely row-column coalition structures.

The fact that constant-column, zero-row-sum configurations are stable for this structure is important, but there are several other points to be made. Notice that several columns could have the same constant value; the problem of ties cannot be resolved by such a uniform system. There are also other eigenvalues and "vectors" which do not correspond to desirable states of the system. These are:

| Eigenvalue | "vector" X |
|---|---|
| $1 + \alpha M - \beta N$ | matrix of all 1 |
| $1 - \alpha - \beta N$ | rows equal, column-sum zero |
| $1 - \alpha + \beta$ | row-sum and column-sums all zero |

By computing the multiplicity of the four eigenvalues, one can show that the total multiplicity is $(N+1)(M+1)$, so that there are no other eigenvalues. The critical point is that powers of a linear system like T will converge to the direction specified by its largest eigenvalue. If we make sure to choose $\alpha$ and $\beta$ so that $\lambda_+ = 1 + \alpha M + \beta$ is the largest eigenvalue, then repetitions of (*) will converge to the desired constant-column zero-row-sum state. This requires (for $\alpha$, $\beta$ positive) that

$$1 + \alpha M + \beta > \alpha + \beta N - 1$$

or

$$(**)\quad 2 > \beta N - \alpha M + (\alpha - \beta).$$

We can ignore $\alpha - \beta$ which is a small fraction. Recall that $\beta$ is the weight given to the competitors and $\alpha$ the weight given to collaborators. Condition (**) states that if the coalitions are given adequate weight, the system will settle into a state with uniform columns (coalitions). The obvious choice of $\beta = 1/N$ and $\alpha = 1/M$ comfortably meets condition (**). The problem that occurs if $\beta$ is too small is that mutual inhibition will have no effect and the system will converge to the state where all columns have their initial average value. The relative importance of competition and collaboration will be a crucial part of the detailed specification of any model. There appears to be no reason that discrete values, bounded ranges and overlapping coalitions should change the basic character of this result, but the detailed analysis of a realistic coalition structure for its convergence properties appears to be very difficult. More generally, there will need to be ways of assessing the impact of finite bounds and discrete ranges on systems whose continuous approximation is understood, a classic problem in numerical analysis.

## 4. Distributed (Massively Connected) Computing

The main restriction imposed by the connectionist paradigm is that no symbolic information is passed from unit to unit. This restriction makes it difficult to employ standard computational devices like parameterized functions. In this section, we present connectionist solutions to a variety of computational problems.

### Using a Unit to Represent a Value

A cornerstone of our approach is the dedication of a separate unit to each value of each parameter of interest, which we term the unit/value principle. We will show how to compute using unit/value networks and present arguments that the number of units required is not unreasonable. In this representation the output may be thought of as a confidence measure. If a unit representing depth = 2 saturates then the network is expressing confidence that the distance of *some object* from the retina is two depth units. There is much neurophysiological evidence to suggest unit/value organizations in less abstract cortical organizations. Examples are edge sensitive units [Hubel and Wiesel, 1979] and perceptual color units [Zeki, 1980], which are relatively insensitive to illumination spectra. Experiments with cortical motor control in the monkey and cat [Wurtz and Albano, 1980] indirectly hint at a unit/value organization. Our hypothesis is that the unit/value organization is widespread, and is a fundamental design principle.

Although many physical neurons do seem to follow the unit/value rule and respond according to the reliability of a particular configuration, there are also other neurons whose output represents the range of some parameter, and apparently some units whose firing frequency reflects both range and strength information. Both of the latter types can be accommodated within our definition of a unit, but we will employ only unit/value cells in the remainder of this paper.

In the unit/value representation, much computation is done by table look-up. Previous ideas such as WTA networks, scaling networks, and deadly rivals still apply; they describe the dynamic behavior of the table. Here we discuss the implications of the tables themselves, which are at the core of what we mean by computing with connections.

As a simple example, let us consider the multiplication of two variables, i.e., $z = xy$. In the unit/value formalism there will be units for *every* value of x and y that is important. Appropriate pairs of these will make a connection with another unit cell representing a specific value for the product. Figure 10 shows this for a small set of units representing values for x and y. Notice that the confidence (expressed as output value) that a particular product is an answer is a linear function of the maximum of the sums of the confidences of its two inputs. Note that the number of xy units need not be as large as the product of the number of x and y inputs for the table to be useful. Furthermore, the x and y inputs make *conjunctive connections* with their z-unit.

Figure 10: Computing with Table Look-Up Units.

### Modifiers and Mappings

The idea of function tables can be extended through the use of *variable mappings*. In our definition of the computational unit, we included a binary modifier, m, as an option on every connection. As the definition specifies, if the modifier associated with a connection is zero, the value v sent along that connection is ignored. There is considerable evidence in nature for synapses on synapses and the modifiers add greatly to the computational simplicity of our networks. Let us start with an initial informal example of the use of modifiers and mappings. Suppose you wanted to ignore the telephone in your office, but answer it at home. One intuitive way to do this is shown by Figure 11.

Figure 11: Modifier ($m_j$) on a Connection.

The circular connection between links denotes a binary modifier. You probably don't want to inhibit your own initiation of phone calls from the office, just the link between the ring and your action. Of course, there are ways of encoding this without using modifiers, but it is easy to see how modifiers permit whole behavior patterns to depend on a state change. By convention, we will assume that a modifier *blocks* the connection when its source unit is active. Technically, m <- if v = 0 then 1 else 0, where v is the output value of the unit which is the source of m.

A slightly more complex use of mappings is for disjunctions. Suppose that one has a model of grass as green except in California where it is brown (golden).

Figure 12. Grass is Green Connection Modified by California.

Here we can see that grass and green are potential members of a coalition (can reinforce one another) except when the link is blocked. This use is similar to the cancellation link of [Fahlman, 1979] and gives a crude idea of how context can effect perception in our models. Note that in Figures 11 and 12 we are using a shorthand notation. A modifier touching a double-ended arrow actually blocks two connections. (Sometimes we also omit the arrowheads when connection is double-ended.)

Mappings can also be used to select among a number of possible values. Consider the example of the relation between depth, physical size, and retinal size of a circle. (For now, assume that the circle is centered on and orthogonal to the line of sight, that the focus is fixed, etc.) Then there is a fixed relation between the size of retinal image and the size of the physical circle for any given depth. That is, each depth specifies a *mapping* from retinal to physical size, i.e.,

Figure 13: Depth Network.

Here we suppose the scales for depth and the two sizes are chosen so that unit depth means the same numerical size. If we knew the depth of the object (by touch, context, or magic) we would know its physical size. The network above allows retinal size 2 to reinforce physical size 2 when depth = 1 but inhibits this connection for all other depths. Similarly, at depth 3, we should interpret retinal size 2 as physical size 8, and inhibit other interpretations. Several remarks are in order. First, notice that this network implements a function phys = f(ret,dep) that maps from retinal size and depth to physical size, providing an example of how to replace functions with parameters by mappings. For the simple case of looking at one object perpendicular to the line of sight, there will be one consistent coalition of units which will be stable. The network does something more, and this is crucial to our enterprise; the network can represent the consistency relation R among the three quantities: depth, retinal size, and physical size. It embodies not only the function f, but its two inverse functions as well (dep = $f_1$(ret,phys), and ret = $f_2$(phys,dep)). (The network as shown does not include the links for $f_1$ and $f_2$, but these are similar to those for f.) Most of Section 5 is devoted to laying out networks that embody theories of particular visual consistency relations.

The idea of modifiers is, in a sense, complementary to that of conjunctive connections. For example, the network of Figure 13 could be transformed into the following network (Figure 14).

Figure 14: An Alternate Depth Network.

In this network the variables for physical size, depth, and retinal size are all given equal weight. For example, physical size = 4 and depth = 1 make a *conjunctive connection* with retinal size = 4. Each of the variables may also form a separate WTA network; hence rivalry for different depth values can be settled via inhibitory connections in the depth network.

To see how the conjunctive connection strategy works in general, suppose a constraint relation to be satisfied involves a variable x, e.g., f(x,y,z,w) = 0. For a particular value of x, there will be triples of values of y, z, and w that satisfy the relation f. Each of these triples should make a conjunctive connection with the unit representing the x-value. There could also be 3-input

conjunctions at each value of y,z,w. Each of these four different kinds of conjunctive connections corresponds to an interpretation of the *relation* f(x,y,z,w) = 0 as a *function*, i.e., $x = f_1(y,z,w)$, $y = f_2(x,z,w)$, $z = f_3(x,y,w)$, or $w = f_4(x,y,z)$. Of course, these functions need not be single-valued. This network connection pattern could be extended to more than four variables, but high numbers of variables would tend to increase its sensitivity to noisy inputs. Hinton has suggested a special notation for the situation where a network exactly captures a consistency relation. The mutually consistent values are all shown to be centrally linked (see Figure 15).

Figure 15: Hinton's Notation.

When should a relation be implemented with modifiers and when should it be implemented with conjunctive connections? A simple, non-rigorous answer to this question can be obtained by examining the size of two sets of units: (1) the number of units that would have to be inhibited by modifiers; and (2) the number of units that would have to be reinforced with conjunctive connections. If (1) is larger than (2), then one should choose modifiers; otherwise choose conjunctive connections. Sometimes the choice is obvious: to implement the brown Californian grass example of Figure 12 with conjunctive connections, one would have to reinforce all units representing places that had green grass! Clearly in this case it is easier to handle the exception with modifiers. On the other hand, the depth relation R(phy,dep,ret) is more cheaply implemented with conjunctive connections.

In physical neurons, there is a feature that makes modifiers more powerful than our examples suggest. Inhibitory connections can block inputs from entire dendritic subtrees, and this could simplify certain networks.

### Time and Sequence

Connectionist models do not initially appear to be well-suited to representing changes with time. The network for computing some function can be made quite fast, but it will be fixed in *functionality*. There are two quite different aspects of time variability of connectionist structures to discuss: long-term modification of the networks (through changing weights) and short-term changes in the behavior of a fixed network with time. There are a number of biologically suggested *mechanisms for changing the weight* ($w_{ij}$) of synaptic connections, but none of them are nearly rapid enough to account for our ability to hear, read, or speak. The ability to perceive a time-varying signal like speech or to integrate the images from successive fixations must be achieved (according to our dogma) by some dynamic (electrical) activity in the networks.

As usual, we will present computational solutions to these problems that appear to be consistent with known structural and performance constraints. These are, again, too crude to be taken literally but do suggest that connectionist models can describe the phenomena. As a first example, consider the problem of controlling a simple physical motion, such as throwing a ball. It is not hard to imagine that for a skilled motor performance we have a fixed sequence of unit-groups that fire each other in succession, leading to the motor sequence. The computational problem is that there is a unique set of effector units (say at the spinal level) that must receive input from each group at the right time.

Figure 16: A Simple Sequencer Using Modifiers.

Figure 16 depicts a situation where two effectors, $e_1$ and $e_2$, get activity from four sequential groups of three units each. At odd intervals, the middle layer masks the upper connections, and at even intervals, the lower. We assume that each column gets activated synchronously and in order. The main point is that a succession of outputs to a single effector set can be modelled as a sequence of time-exclusive groups representing instantaneous coordinated signals. Moving from one time step to the next could be controlled by pure timing, or (more realistically in many cases) by a proprioceptive feedback signal. There is, of course, an enormous amount more than this to motor control, and realistic models would have to model force control, ballistic movements, gravity compensation, etc.

The sequencer model for skilled movements was greatly simplified by the assumption that the sequence of activities was pre-wired. How could we (still crudely, of course) model a situation like speech perception where there is a largely unpredictable time-varying computation to be carried out. The idea here is to combine the sequencer model of Figure 16 with a simple vision-like scheme. We assume that speech is recognized by being sequenced into a buffer of about the length of a phrase and then is relaxed against context in the way described above for vision. For simplicity, we will assume that there are two identical buffers, each having a pervasive modifier ($m_j$) innervation so that either one can be switched into or out of its connections. We are particularly concerned with the process of going from a sequence of potential phonemes into an interpreted phrase. Figure 17 gives an idea of how this might happen.

Figure 17: A Pheneme Sequence Buffer.

We assume that there is a separate unit for each potential phoneme for each time step up to the length of the buffer. The network which analyzes sound is connected identically to each column, but conjunction allows only the connections to the active column to transmit values. Under ideal circumstances, at each time step exactly one phoneme unit would be active. A phrase would then be layed out on the buffer like an image on the "mind's eye," and the analogous kind of relaxation cones involving morphemes, words, etc., could be brought to bear. The more realistic case where sounds are locally ambiguous presents no additional problems. We assume that, at each time step, the various competing phonemes get varying activation. Diphone constraints could be captured by (+ or -) links to the next column as suggested by Figure 17. We are now left with a multiple possibility relaxation problem--again exactly like that in visual perception. The fact that each potential phoneme could be assigned a row of units is essential to this solution; we do not know how to make an analogous model for a sequence of sounds which cannot be clearly categorized and combined. Recall that the purpose of this example is to indicate how time-varying input could be treated in connectionist models. The problem of actually laying out detailed models for language skills is enormous and our example may or may not be useful in its current form. Some of the considerations that arise in distributed modeling of language skills are presented in [Arbib and Caplan, 1979].

### Conserving Connections

It is currently estimated that there are about $10^{11}$ neurons and $10^{15}$ connections in the human brain and that each neuron receives input from about $10^3$-$10^4$ other neurons. These numbers are quite large, but not so large as to present no problems for connectionist theories. It is also important to remember that neurons are not switching devices; the same signal is propagated along all of the outgoing branches. For example, suppose some model called for a separate, dedicated path between all possible pairs of units in two layers of size N. It is easy to show that this requires $N^2$ intermediate sites. This means, for example, that there are not enough neurons in the brain to provide such a cross-bar switch for layers of a million elements each. Similarly, there are not enough neurons to provide one to represent each complex concept at every position, orientation, and scale of visual space. Although the development of connectionist models is in its perinatal period, we have been able to accumulate a number of ideas on how some of the required computations can be carried out without excessive resource requirements. Two of the most important of these are described below. A third important idea is that of sequencing, but that will be deferred to Section 5 in order to develop it in the context of a detailed example from vision.

### Fixed Resolution Computation

In the multiplication example of Figure 10 it might seem that $N_x N_y$ units are required to implement this simple function and that in general the number of units would grow exponentially with the number of arguments. However, there are several refinements which can drastically reduce the number of required units. The principal way to do this is to fix the number of units at the *resolution* required for the computation. Figure 18 shows the network of Figure 10 modified when less computational accuracy is required.

Figure 18: Modified Table Using Less Units.

When the number of variables in the function becomes large, it might seem that the fan-in or number of input connections might become unrealistically large. For example, with the function z = f(u,v,w,x,y,z) implemented with 100 values of z, when each of its arguments can have 100 distinct values, would require an average number of inputs per unit of $10^{12}/10^2$, or $10^{10}$! However, there are simple ways of trading units for connections. One is to replicate the number of units with each value. This is a good solution when the inputs can be partitioned in some natural way as in the vision examples in the next section. Another is to use intermediate units when the computation can be decomposed in some way. For example, if f(u,v,w,x,y,z) = g(u,v)o h(w,x,y,z), where o is some composition, then separate tables for f(g,h), g(u,v), and h(w,x,y,z) can be used. The outputs from the g and h tables can be combined in conjunctive connections according to the composition operator o via a third table to produce f. In perception the transition from u,v,w,x,y,z to g,h to f corresponds to changes in level of abstraction.

**Low Resolution Grain**

Suppose we have a set of units to represent a vector parameter v composed of components (r,s). Suppose that the number of units required to represent the subspace r is $N_r$ and that required to represent s is $N_s$. Then the number of units required to represent v is $N_rN_s$. It is easy to construct examples in vision where the product $N_rN_s$ is too close to the upper bound of $10^{11}$ units to be realistic. Consider the case of trihedral vertices, an important visual cue. Three angles and two position coordinates are necessary to uniquely define every possible trihedral vertex. If we use 5 degree angle sensitivity and $10^5$ spatial sample points, the number of units is given by $N_r \approx 5\times10^3$ and $N_s = 10^5$ or $5\times10^8$! How can we achieve the required representation accuracy with less units?

In many instances, we can take advantage of the fact that the *actual occurrence* of parameters is *low density*. What we mean by this in terms of trihedral vertices is that in an image, such vertices will rarely occur in tight spatial clusters. (If they do, one cannot resolve them as individuals simultaneously.) However, even though simultaneous proximal values of parameters are unlikely, they still can be represented accurately for other computations.

The solution is to decompose the space (r,s) into two subspaces, each with unilaterally reduced resolution.

> Instead of $N_rN_s$ units, we represent v with two spaces, one with $N_r'N_s$ units where $N_r' \ll N_r$ and another with $N_rN_s'$ units where $N_s' \ll N_s$.

To illustrate this technique with the example of trihedral vertices we choose $N_s' = 0.01N_s$ and $N_r' = 0.01N_r$. Thus the dimensions of the two sets of units are:

$$N_s'N_r = 5\times10^6$$

and

$$N_sN_r' = 5\times10^6.$$

The choices mean that we have one type of unit which accurately represents the angle measurements and fires for any trihedral vertex in a given visual region, and another set of units which fire only if a vertex is present at the precise position. Figure 19 shows the two cases.

Figure 19: Fuzzy Resolution Trick.

If the vertex enters into another relation, say R(v,α), where both its angle and position are required accurately, one simply conjunctively connects pairs of appropriate units from each of the reduced resolution spaces to appropriate α-units. The conjunctive connection represents the *intersection* of each of its components' *fields*.

*This resolution device is a variant of a general result due to [Hinton, 1980]; namely,* that connections from overlapping sets of units can produce fine resolution with less units. An important limitation of this technique, however, is that the input must be sparse. If inputs are too closely spaced, "ghost" firings will occur (Figure 20). Another point is that the resolution device is essenually a units/connections tradeoff, but as the brain has many more synapses than neurons, the tradeoff is attractive.

Figure 20: "Ghost" Firings.

## 5. Some Implications for Early Visual Processing

Although early visual processing appears to be particularly well-suited for connectionist treatment, there are a number of serious problems. Some of these arise from the immense size of the cross product of the spatial dimensions with those of other interesting features such as color, velocity, and texture. Thus to explain how image-like input such as color and optical flow are related to abstract objects such as "a blue, fast-moving thing," it becomes necessary to use all the techniques of the previous sections. We will work through an example in detail and show a solution which uses realistic numbers of units, connections, and connections per unit. The main trap to avoid is a solution that requires $X!^k$ units where X is the spatial dimension, f is the number of measurement values per modality (color, distance, velocity, etc.), and k is the number of modalities.

The above example omits the details of the transformation involved in relating image-like features, like primary color measurements, to a percept, like "blue." To remedy this deficiency, we will work through a second example, the calculation of shape from shading, which emphasizes this kind of transformation.

### Objects

The visual field contains objects that are disjoint. This separateness is manifest in groups of spatially registered features such as texture and color which distinguish the objects. Thus we regard the problem of detecting an object as a matter of determining which of several possible features of color, texture, motion, and shape it has. In fact we can view these features as having ranges of associated parameter values. For example, an object could be described as having the properties "fast" and "blue." The quotes signify that the property is not a single value but incorporates an appropriate range of values. The property "blue" might be any of a set of primary red, green, and blue values, each of which satisfies some relationship B(r,g,b) which defines the percept "blue." In a one-dimensional retina, we might imagine arrays of spatially registered color sensors, each appropriately connected to a property unit, as shown in Figure 21.

Figure 21: Feature Measurement:
A Unit which Responds to a Range of Blue.

Figure 21 shows two kinds of units: a property unit and spatially-registered sensor units. The sensor units represent five different values of each of three parts of the color spectrum. In our primitive design, a property unit has a high potential value (=1) if any of the spatial sensor units for that measurement value have a high potential, i.e.,

Blue <- ANY(X) Blue(X)

The non-spatially registered units represent ranges of feature values, e.g. the property "blue." Objects can be described in terms of combinations of these properties. The property units, in turn, receive inputs from groups of spatially registered primary color unit measurements. Here we take advantage of the disjunctive nature of different groups of inputs to differentiate between different parts of space. The number of connections to a property unit can always be reduced by replicating the property unit and connecting the replicated unit's outputs to a single property unit.

### Tuning

In Figure 21 the "blue" unit will respond to any values of primary inputs in the appropriate ranges of "blue." This can make it susceptible to noise; for example, consider the case where the object has some specific value of "blue" in the appropriate range and there are also random similar values of "blue" at other image points. One way of ruling out these extraneous values is to *tune* the "blue" unit to respond to only the appropriate set of primary color measurements. This is done by using a fine-grained set of perceptual color units. Within this set there are many units corresponding to colors in the range defined by "blue" (although only one is drawn in Figure 22). Tuning the "blue" unit is accomplished by conjunctively connecting the appropriate color units to their corresponding values of primary measurements at the "blue" unit inputs. This is shown in

Figure 22. The fine-tuned blue unit receives input from all parts of space and sums its input. By firing only the appropriate fine-grain color unit, the "blue" unit is made to respond to only the corresponding set of its activated inputs. Note that this is an instance of the general tuning method (discussed in Section 4).

Figure 22: Tuning the "Blue" Unit with a Precise Value Unit.

A problem arises in the simple circuit of Figure 21 when the visual input contains more than one object, that is, more than one group of spatially registered features. The simple network of Figure 22 cannot detect the spatial distinctness of the two groups. To make this problem more concrete, let us consider two spatially distinct items, one blue (B) and fast (F) and the other red (R) and textured (T). In the simple network we must expect all feature units have a high potential, i.e., B, T, R, and F, and there is no grouping of the two appropriate pairs, BF and RT. This is an instance of the general problem of multi-attribute concepts which has been viewed as a major obstacle to connectionist schemes.

One solution to this is to elaborate all BF(x) units, but this poses two problems. First, there are a large number of units, i.e., $(N_m)^2 N_x$ where $N_m$ is the number of feature groups and $N_x$ is the spatial quantification. For the retina (even the fovea) this number becomes unrealistically large. A solution is to allow pairs of coarse-grained property units which still do not use the spatial registration explicitly. In Figure 23 we show the circuit for a BF cell which assumes a high potential only if its inputs from sensor units are spatially registered. This is done by making spatially registered units have conjunctive connections. That is, appropriate values of color at x = 7 and velocity at x = 7 would make a conjunctive connection, but appropriate values that were not spatially registered would not. Of course, the BF unit might have to normalize its input in the manner of Section 2, if there were many visual features present. Note that the velocity measurement portion of the network is not drawn but that it is nearly identical. Velocity sensors are connected to fine-grained velocity units in the same way as color sensors are connected to fine-grained color units. Models for the various parts of velocity-sensitive networks have been explored by [Horn and Schunck, 1980; Barnard and Thompson, 1979; and Ballard, 1981c].

Figure 23: A BF Unit Detects the Existence
of Spatially Registered "Blue" and "Fast" Percepts.

Despite all these ways of economizing, it is still combinatorically implausible to have complex cells such as BTRM(...). However, there is a way around this problem using multiple connections from object units. A blue-textured-fast (BTF) object unit can be synthesized from BT and BF units, i.e., BT & BF $\simeq>$ BTF. What the $\simeq>$ symbol means is that the implication is not guaranteed, but very *likely*, given that the BF and BT units are tuned. The BF and BT units detect spatial registration directly via connections like those in Figure 23, but the BTF unit does not. We are saying essentially that, in general, simple (here, pairwise) conjunctions can be kept spatially registered by conjunctive connections, and that more complex property combinations can be synthesized from them. Complex combinations that are important to an individual are presumed to have new units *recruited* [Feldman, 1980; 1981] to represent them explicitly.

**Sequencing**

One might imagine that the network in Figure 23 is adequate and that given visual input, it will converge to appropriate potential values. However, there is an easy way to see that, in general, this will not happen for all percepts. First consider the fine-grained set of feature values in Figure 22. Since units in this network receive inputs from all parts of space, diffusely valued spatial points can easily obscure a set of contiguous spatial points with a single value. For example, motion from other parts of space can obscure the motion of a particular object (in feature space). Another way to understand this is to suppose a set a features formed a single, multi-dimensional space. In this space objects are clear, but because of the enormous size of the space, it must be represented with different projections. The projections are the familiar subspaces of color, velocity, etc.

There is a way to use the subspaces to cause the appropriate percepts to fire by building conjunctions of features in a sequential manner. This circumvents the basic problem described above. To develop the sequential solution, we introduce *spatial* units, such as those shown in Figure 24. There is a spatial unit for each spatial position and each intrinsic parameter (e.g. color).

Figure 24: A Blue-Fast Unit showing
Spatial Context Units.

Spatial units receive input from percept units, feature units, and sensors. If all three of these are present, the unit will adjust the potentials of all of the sensor units upwards. We assume that upper and lower bound units will adjust the potentials of the entire sensor network. The net result will be that spatial sensor units not receiving appropriate input will have no effect on the property units.

With respect to Figure 24, we suggest the following scenario for a "blue," "fast-moving," "horizontal" surface, where blue stands out in the color feature space, but fast-moving and horizontal do not in their respective feature spaces. First, the blue percept unit causes input from blue valued spatial positions to be favored. Under this restriction, one of the other features is now distinct, further raising the potential of active sensor cells at those positions. The effect is as if a blue filter were placed in front of the sensory input. Now the third feature is detected. At this point the composite network indicates that there is a blue, fast-moving, horizontal surface in the visual field at the position specified by high-confidence spatial units.

Our solution to the problem of detecting spatially-registered features is not unique but does require much less than $Xf^k$ units. Table 2 summarizes the connection and unit requirements in terms of spatial complexity estimates.

Table 2

k = number of modalities
X = number of distinct spatial values
F = number of distinct "fuzzy" features/modality
f = number of distinct fine-resolution features/modality

| Unit Type | No. of Units | Connections/ Unit | Connects to Units of Type |
|---|---|---|---|
| Sensors (S) | kXf | F | ff, C |
| Fine-features (ff) | kf | X | FF, C |
| Fuzzy-features (FF) | $F^2$ | $X(f/F)^2$ | C |
| Spatial (C) | kXF | $F^2$ | S, ff |

**Motor Control of the Eye**

To see how this notion of distributed objects might work in motor control, we offer a simplistic model of vergence eye movements. (The same idea may be valid for fixations, but control probably takes place at higher levels of abstraction.) In this model retinotopic (spatial) units are connected directly to muscle control units. Each retinotopic unit can if saturated cause the appropriate contraction so that the new eye position is centered on that unit. When several retinotopic units

saturate, each enables a muscle control unit independently and the muscle itself contracts an average amount.

Figure 25 shows the idea for a one-dimensional retina. For example, with units at positions 2, 4, 5, and 6 saturated, the net result is that the muscle is centered at 17/4 or 4.25. This idea can be extended if we assume the retinotopic units have overlapping fields such as those used by [Hinton, 1980]. This kind of organization is consistent with studies of the organization of the superior colliculus in the monkey [Wurtz and Albano, 1980].

Figure 25: Distributed Control of Eye Fixations.

Notice that each retinotopic unit is capable of enabling different muscle control units. The appropriate one is determined by the enabled x-origin unit which inhibits commands to the inappropriate control units via modifiers.

One problem with this simple network arises when disparate groups of retinotopic units are saturated. The present configuration can send the eye to an average position if the features are truly identical. Also, the network can be modified with additional connections so that only a single connected component of saturated units is enabled by using additional object primitives. A version of this motor control idea has already been used in a computer model of the frog tectum [Didday, 1976].

There are still many details to be worked out before this could be considered a realistic model of vergence control, but it does illustrate the basic idea: local spatially separate sensors have *distinct, active* connections which could be averaged at the muscle for fine motor control.

**Shape from Shading**

In a previous sub-section we showed how *spatially distributed information* could be connected to a global object unit. There the issue was primarily one of feasibility. With a simple model, there were large numbers of global features, yet it was possible to detect them all. Relationships between image-like inputs and features were assumed but not stressed. In this section we present an example of such relations by showing a network which computes surface orientation from an intensity array.

The specific example we will use is that of shape-from-shading. It is well-known that given the orientation of a surface with respect to a viewer, its reflectance properties and the location of a single light source, that the brightness at a point of the viewer's retina can be determined. That is, the reflectance function $R(\Theta,\Phi,\Theta_S,\Phi_S)$, where $\Theta,\Phi$ and $\Theta_S,\Phi_S$ are orientations of the surface and source respectively, allows us to determine $I(x,y)$, the normalized intensity in terms of retinal coordinates. However, the perceptual problem is the reverse: given $I(x,y)$ and $R(...)$, determine $\Theta(x,y),\Phi(x,y)$ and $\Theta_S,\Phi_S$.

In general, the problem of deriving $\Theta(x,y),\Phi(x,y)$ and $\Theta_S,\Phi_S$ is underdetermined. However, Ikeuchi [1980] showed that the surface could be determined locally once $\Theta_S,\Phi_S$ was specified. This method has been extended [Ballard, 1981b] to the case where $\Theta_S,\Phi_S$ is initially unknown.

The algorithm is outlined as follows. For a single light source, the intensity at a point on a retina can be described in terms of the orientation of the normal of the corresponding surface point and the surface orientation. That is, in spherical coordinates,

$$I(x,y) = R(\Theta,\Phi,\Theta_S,\Phi_S) \qquad \text{(Eq. 5.1)}$$

where the angles $\Theta$ and $\Phi$ are functions of x and y. The viewer is assumed to be looking down the z axis (towards its origin) under orthonormal viewing conditions. $\Theta$, $\Phi$, $\Theta_S$, and $\Phi_S$ are spherical angles measured with respect to this frame. Now by minimizing $(I-R)^2$ and appending a smoothness constraint on $\Theta$ and $\Phi$ we have [Ikeuchi, 1980] an expression for the local error (if the estimate for $\Theta$ and $\Phi$ is unreliable) as follows:

$$E(x,y) = (I-R)^2 + \lambda((\nabla^2 O)^2+(\nabla^2\Phi)^2)$$

where $\lambda$ is a Lagrange multiplier. For a minimum, $E_O$ and $E_\Phi = 0$. Skipping some steps, this leads to

$$O(x,y) = O_{ave}(x,y)+T(x,y)R_O \qquad \text{(Eq. 5.2a)}$$
$$\Phi(x,y) = \Phi_{ave}(x,y)+T(x,y)R_\Phi \qquad \text{(Eq. 5.2b)}$$

where

$\Phi_{ave}(x,y)$ is a local average

and

$$T(x,y) = (1/16\lambda)(I-R)$$

In solving these equations, $O_S$ and $\Phi_S$ are assumed to be known. Ikeuchi used a parallel-iterative method where the $\Phi_{ave}$ and $O_{ave}$ are calculated from a previous iteration.

To calculate $O_S$ and $\Phi_S$, we assume $O$ and $\Phi$ are known and use a Hough technique. First we form an array $A[O_S,\Phi_S]$ of possible values of $O_S$ and $\Phi_S$ initialized to zero. Now we can solve the Lamberian reflectance equation for $\Phi_S$. The Hough technique works as follows. For each surface element $O,\Phi$, and for each $O_S$ we calculate all $\Phi_S$ that sausfy Eq. 5.1 and increment $A[O_S,\Phi_S]$, i.e., $A[O_S\Phi_S] := A[O_S,\Phi_S]+1$. After all surface elements have been processed, the maximum value of A corresponds to the location of the point source. In [Ballard, 1981b] it is shown that calculation of the source location can proceed in parallel with that of $O(x,y)$ and $\Phi(x,y)$ and that the two calculations will converge.

**Implementation in Networks**

The above description of the shape-from-shading is geared to implementation on a *conventional* computer. We now describe how these computations can be realized with connections between networks of basic units. The general strategy is as follows. Variables in the above equations are represented by networks of units where each unit has a discrete value. The connections between value-units must be made in such a way that the networks converge to a set of value-units that have potential equal to unity. These units will represent a particular solution for the input intensity distribution.

The shape-from-shading calculations can be decomposed into two principal networks. One represents the surface orientations at retinal points and the other represents possible illumination angles. Thus there is a $\{O(x,y),\Phi(x,y)\}$-network and a $\{(O_S,\Phi_S)\}$ network. In addition, input values of $I(x,y)$ are assumed. The $O(x,y)$ sub-network is represented by units each representing a specific value of $O$ at a particular point x,y. This representation requires $N_O N_x N_y$ units. Assuming $N_x = N_y = 2^7$ and $N_O = 2^5$, the requirement is for $2^{19}$ or $< 10^6$ units. The illumination angle network uses units to represent *pairs* of values, one for $O_S$ and one for $\Phi_S$. The reason for this choice will be discussed momentarily.

With these provisos we describe the connections between networks that compute shape from shading in two parts. The first part describes connections from the $\{O,\Phi\}$ network to the network that detects illumination direction. The second part describes the connections in the other direction. For every position x,y and for each value of $O$, $\Phi$, and $I$ at that position, the appropriate values of $O_S$ and $\Phi_S$ which sausfy $R = I$ can be precalculated. Thus $O,\Phi,I$ triples, each representing a specific value, make conjunctive connections with $O_S,\Phi_S$ units. Figure 26 shows a representative connection.

Figure 26: A Portion of the Connections
Used to Detect Illumination Angle.

The $O_S,\Phi_S$ units are summation units. Each $O_S,\Phi_S$ unit sums the number of I,O,$\Phi$ input triples that are firing and their potentials are proportional to this sum. The proportionality constant may be known from physical considerations or may be adjusted by upper and lower bound units like those described in Section 2. The $O_S,\Phi_S$ unit with the highest potential is the one that is consistent with the maximum number of $\{O,\Phi\}$ and I units.

Two important considerations effect the design of the $\{(O_S\Phi_S)\}$ network. One is the need for good discrimination in the values of $O_S$ and $\Phi_S$. This led to the decision to use $(O_S,\Phi_S)$ pairs as units instead of $O_S$ and $\Phi_S$ units. In the latter case, solutions to the constraint relation that satisfied pairs of $(O_S,\Phi_S)$ values may be obscured in the individual $O_S$ and $\Phi_S$ networks owing to the reduced dimensionality. The second consideration is the average number of connections per unit. With $2^6$ values for $O_S$ and $\Phi_S$, there must be $2^{12}$ units in the $\{(O_S\Phi_S)\}$ network. An upper bound on the number of connections to this network from the $\{(O\Phi)\}$ network can be determined from straightforward counting arguments. At each point x,y there are $N_O N_\Phi N_I$ combinations, or $2^{15}$. With $N_x N_y = 2^{14}$, this leads to $2^{29}$ or $10^9$ total connections. Thus the average number of connections per $(O_S\Phi_S)$ unit is $2^{29}/2^{12}$ or $2^{17} = 3\times10^5$. If this is unreasonably large, a simple solution is to use auxilliary $O_S\Phi_S$ units, which sum subsets of the inputs to a $O_S\Phi_S$ unit. The $O_S\Phi_S$ unit then sums the outputs of the auxilliary units.

For the second part we consider connections from the $\{(O_S,\Phi_S)\}$ network to the $\{O,\Phi\}$ network needed to realize the constraint of Equations 5.2a and b. We will only consider the first equation (since the second is treated similarly). This represents a constraint $g(\Phi,O_{ave},I,O,O_S,\Phi_S) = 0$. Given values for these variables we can determine $R(O,\Phi,O_S,\Phi_S)$ and $R_O$ to see whether or not Equation 5.2a is satisfied. Thus a straightforward application of our technique would use conjunctive connections in groups of nine, as shown in Figure 27. For a particular O value we examine all the combinations of values for the other variables and connect the subset that satisfies Equation 5.2a to the O-unit.

Figure 27: Alternate Connection Schemes
for Computing Surface Orientation.

Here we have used four nearby values of O to compute $O_{ave}$. This implementation is unsatisfactory for two reasons: (1) the large number of inputs in the conjunctive connections would be noise-sensitive; and (2) there would be an unrealistically large number of connections on any one unit. To solve both of these problems we use $O_{ave}$ units for each O(x,y). While this only doubles the number of units in the $\{O,\Phi\}$ network, it drastically reduces the number of connections to an individual O-unit. Assuming our earlier figures, this number is

$$N_O N_{Oave} N_I N_\Phi N_{Os} N_{\Phi s} \text{ or } 2^{30} = 10^9.$$

This number is still very large, but can be further reduced by further unit/connection tradeoffs.

The introduction of the $O_{ave}$ unit to reduce connections represents a different kind of tradeoff from the simpler tradeoff used to handle high density connections in the illumination angle network. A specific value of $O_{ave}$ may be produced by several different combinations of nearby O's. Each of these groups of O's makes a conjunctive connection with the $O_{ave}$ unit. However, since we expect a unique value of O(x,y), the unit behaves differently than that in the illumination angle network. Rather than sum its inputs, each $O_{ave}$ unit adjusts its potential based on the maximum of its conjunctive groups.

## Other Networks Determine Boundary Conditions

The unit outline for shape from shading calculations does not include a discussion of boundary conditions. These can be calculated from other networks such as a disparity network. For example, the existence of a depth discontinuity d(x,y) in the disparity network could inhibit connections between the O and Φ either side of the discontinuity. In general, such networks will interact in many different ways to determine boundary conditions [Barrow and Tenenbaum, 1978]. Much additional work needs to be done to specify these interactions more precisely.

## Conclusions

We have now completed five years of intensive effort on the development of connectionist models and their application to the description of complex tasks. While we have only touched the surface, the results to date are very encouraging. Somewhat to our surprise, we have yet to encounter a challenge to the basic formulation. Our attempts to model in detail particular computations [Sabbah, 1981; Ballard and Sabbah, 1981] have led to a number of new insights (for us, at least) into these specific tasks. Attempts like this one to formulate and solve general computational problems in realistic connectionist terms have proven to be difficult, but less so than we would have guessed. There appear to be a number of interesting technical problems within the theory and a wide range of questions about brains and behavior which might benefit from an approach along the lines suggested in this report.

## Appendix: Summary of Definitions and Notation

A **unit** is a computational entity comprising:

{q} -- a set of *discrete states*, < 10

p -- a continuous value in [-1,1], called *potential* (accuracy of 10 digits)

v -- an *output value*, integers $0 \leq v \leq 9$

i -- a vector of *inputs* $i_1,...,i_n$

and functions from old to new values of these

p <- f(i,p,q)

q <- g(i,p,q)

v <- h(i,p,q)

which we assume, for now, to compute continuously. The form of the f, g, and h functions will vary, but will generally be restricted to conditionals and functions found on hand calculators.

### P-Unit

For some applications, we will use a particularly simple kind of unit whose output v is proportional to its potential p (rounded) and which has only one state. In other words

$p \leftarrow p + \beta \Sigma i_k$ $\quad |-1 \leq p \leq 1|$

$v = \alpha p - \theta$ $\quad |v = 0...9|$

where $\beta$, $\alpha$, $\theta$ are constants

### Connection Tables

In addition to graphical notation, the outgoing connections to other units can be described in tabular form. Each outgoing $v_j$ (only one for basic units) will have a set of entries of the form

(<receiving unit>,<index>,<±>,<type>)

where any of the last three constructs can be omitted and given its default value. The <±> field specifies whether the link is excitatory (+) or inhibitory (-) and defaults to +. The <index> is the input index j in $r_j$ at the receiving end. This index can be used for specifying different weights. Indexed inputs also allow for functionally different use of various inputs and many of our examples exploit this feature. The <type> is either normal, modifier (m), or learning (x), the default being normal.

### Conjunctive Connections

In terms of our formalism, this could be described in a variety of ways. One of the simplest is to define the potential in terms of the maximum, e.g.,

$p \leftarrow p + \text{Max}(i_1+i_2,\ i_3+i_4,\ i_5+i_6-i_7)$

The max-of-sum unit is the continuous analog of a logical OR-of-AND (disjunctive normal form) unit and we will sometimes use the latter as an approximate version of the former. The OR-of-AND unit corresponding to the above is:

$p \leftarrow p + \alpha\ \text{OR}\ (i_1\&i_2,\ i_3\&i_4,\ i_5\&i_6\&(\text{not } i_7))$

**Winner-take-all (WTA) networks** have the property that only the unit with the highest potential (among a set of contenders) will have output above zero after some settling time.

A **coalition** will be called stable when the output of all of its members is non-decreasing.

## References


Aho, A., J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

Anderson, J.R. and G.H. Bower. *Human Associative Memory.* Washington, DC: V.H. Winston and Sons, 1972.

Arbib, M.A. and D. Caplan, "Neurolinguistics must be computational," *The Brain and Behavioral Sciences 2*, 449-483, 1979.

Ballard, D.H., "Parameter networks," TR75, Computer Science Dept, U. Rochester, 1981a.

Ballard, D.H., "Shape and illumination angle from shading," Computer Science Dept, U. Rochester, 1981b.

Ballard, D.H., "3-d rigid body motion from optical flow," Computer Science Dept, U. Rochester, 1981c.

Ballard, D.H. and D. Sabbah, "On shapes," Computer Science Dept, U. Rochester, 1981.

Barnard, S.T. and W.B. Thompson, "Disparity analysis of images," TR 79-1, Computer Science Dept, U. Minnesota, January 1979.

Barrow, H.G. and J.M. Tenenbaum, "Recovering intrinsic scene characteristics from images," Technical Note 157, AI Center, SRI Int'l., April 1978.

"The Brain," *Scientific American*, September 1979.

Buser, P.A. and A. Roguel-Buser (Eds). *Cerebral Correlates of Conscious Experience.* Amsterdam: North Holland Publishing Co., 1978.

Collins, A.M. and E.F. Loftus, "A spreading-activation theory of semantic processing," *Psych Review 82*, 407-429, November 1975.

Didday, R.L., "A model of visuomotor mechanisms in the frog optic tectum," *Math'l Biosciences 30*, 169-180, 1976.

Edelman, G. and B. Mountcastle. *The Mindful Brain.* Boston, MA: MIT Press, 1978.

Fahlman, S.E., "The Hashnet interconnection scheme," Computer Science Dept, Carnegie-Mellon U., June 1980.

Fahlman, S.E. *NETL, A System for Representing and Using Real Knowledge.* Boston, MA: MIT Press, 1979.

Feldman, J.A., "A connectionist model of visual memory," in G.E. Hinton and J.A. Anderson (Eds). *Parallel Models of Associative Memory.* Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, 1981.

Feldman, J.A., "A distributed information processing model of visual memory," TR52, Computer Science Dept, U. Rochester, 1980.

Friesen, W.O. and G.S. Stent, "Neural circuits for generating rhythmic movements," *Ann Rev Biophys Bioengg 7*, 37-61, 1978.

Grossberg, S., "Biological competition: Decision rules, pattern formation, and oscillations," *Proc Nat'l Acad Sci USA 77*, 4, 2238-2342, April 1980.

Hanson, A.R. and E.M. Riseman (Eds). *Computer Vision Systems.* NY: Academic Press, 1978.

Hinton, G.E., "Relaxation and its role in vision," Ph.D. thesis, Cognitive Studies Program, U. Edinburgh, December 1977.

Hinton, G.E., Draft of Technical Report, U. California at San Diego, 1980.

Hinton, G.E. and J.A. Anderson (Eds). *Parallel Models of Associative Memory.* Hillsdale, NJ:

Lawrence Erlbaum Associates, Publishers, 1981.

Horn, B.K.P. and B.G. Schunck, "Determining optical flow," AI Memo 572, AI Lab, MIT, April 1980.

Hubel, D.H. and T.N. Wiesel, "Brain mechanisms of vision," *Scientific American*, 150-162, September 1979.

Hummel, R.A. and S.W. Zucker, "On the foundations of relaxation labeling processes," TR-80-7, Computer Vision and Graphics Lab, McGill U., July 1980.

Ikeuchi, K., "Numerical shape from shading and occluding contours in a single view," AI Memo 566, AI Lab, MIT, revised February 1980.

Kandel, E.R. *The Cellular Basis of Behavior*. CA: Freeman, 1976.

Kilmer, W.L., W.S. McCulloch, and J. Blum, "Some mechanisms for a theory of the reticular formation," in M. Mesarovic (Ed). *Systems Theory and Biology*. NY: Springer-Verlag, 1968.

Kosslyn, S.M. and S.P. Schwartz, "Visual images as spatial representations in active memory," in A.R. Hanson and E.M. Riseman (Eds). *Computer Vision Systems*. NY: Academic Press, 1978.

Marr, D., "Representing visual information," in A.R. Hanson and E.M. Riseman (Eds). *Computer Vision Systems*. NY: Academic Press, 1978.

Marr, D.C. and T. Poggio, "Cooperative computation of stereo disparity," *Science 194*, 283-287, 1976.

Minsky, M., "K-Lines: A Theory of Memory," *Cognitive Science 4*, 2, 117-133, 1980.

Minsky, M. and S. Papert. *Perceptrons*. Cambridge, MA: The MIT Press, 1972.

Neisser, U. *Cognition and Reality: Principles and Implications of Cognitive Psychology*. San Francisco, CA: W.H. Freeman and Co., 1976.

Perkel, D.H. and B. Mulloney, "Calibrating compartmental models of neurons," *Am J Physiol 235*, 1, R93-R98, 1979.

Prager, J.M., "Extracting and labeling boundary segments in natural scenes," *IEEE Trans. PAMI 2*, 1, 16-27, January 1980.

Rosenfeld, A., R.A. Hummel, and S.W. Zucker, "Scene labelling by relaxation operations," *IEEE Trans. SMC 6*, 1976.

Russell, D.M., "Adaptive modules for low-level problem-solving," Computer Science Dept, U. Rochester; submitted to IJCAI, 1981.

Sabbah, D., "*Design of a highly parallel visual recognition system*," Computer Science Dept, U. Rochester; submitted to IJCAI, 1981.

Sejnowski, T.J., "Storing covariance with nonlinearly interacting neurons," *J Math Biology 4*, 4, 303-321, 1977.

Stent, G.S., W.B. Kristan, Jr., W.O. Friesen, C.A. Ort, M. Poon, and R.L. Calabrese, "Neuronal generation of the leech swimming movement," *Science 200*, June 1978.

Tomoka, T., "*Pattern separability in a random neural net with inhibitory connections*," *Biol. Cybernetics 34*, 53-62, 1979.

Ullman, S., "Relaxation and constrained optimization by local processes," *CGIP 10*, 115-125, 1979.

von der Malsburg, Ch. and D.J. Willshaw, "How to label nerve cells so that they can interconnect in an ordered fashion," *Proc Nat'l Acad Sci USA 74*, 11, 5176-5178, November 1977.

Warshaw, H.S. and D.K. Hartline, "Stimulation of network activity in stomatogastric ganglion of the spiny lobster, *Panulirus*," *Brain Research 110*, 259-272, 1976.

Wurtz, R.H. and J.E. Albano, "Visual-motor function of the primate superior colliculus," *Ann Rev Neurosci 3*, 189-226, 1980.

Wickelgren, W.A., "Chunking and consolidation: A theoretical synthesis of semantic networks, configuring in conditioning, S-R versus cognitive learning, normal forgetting, the amnesic syndrome, and the hippocampal arousal system," *Psych Review 86*, 1, 44-60, 1979.

Zeki, S., "The representation of colours in the cerebral cortex," *Nature 284*, April 1980.

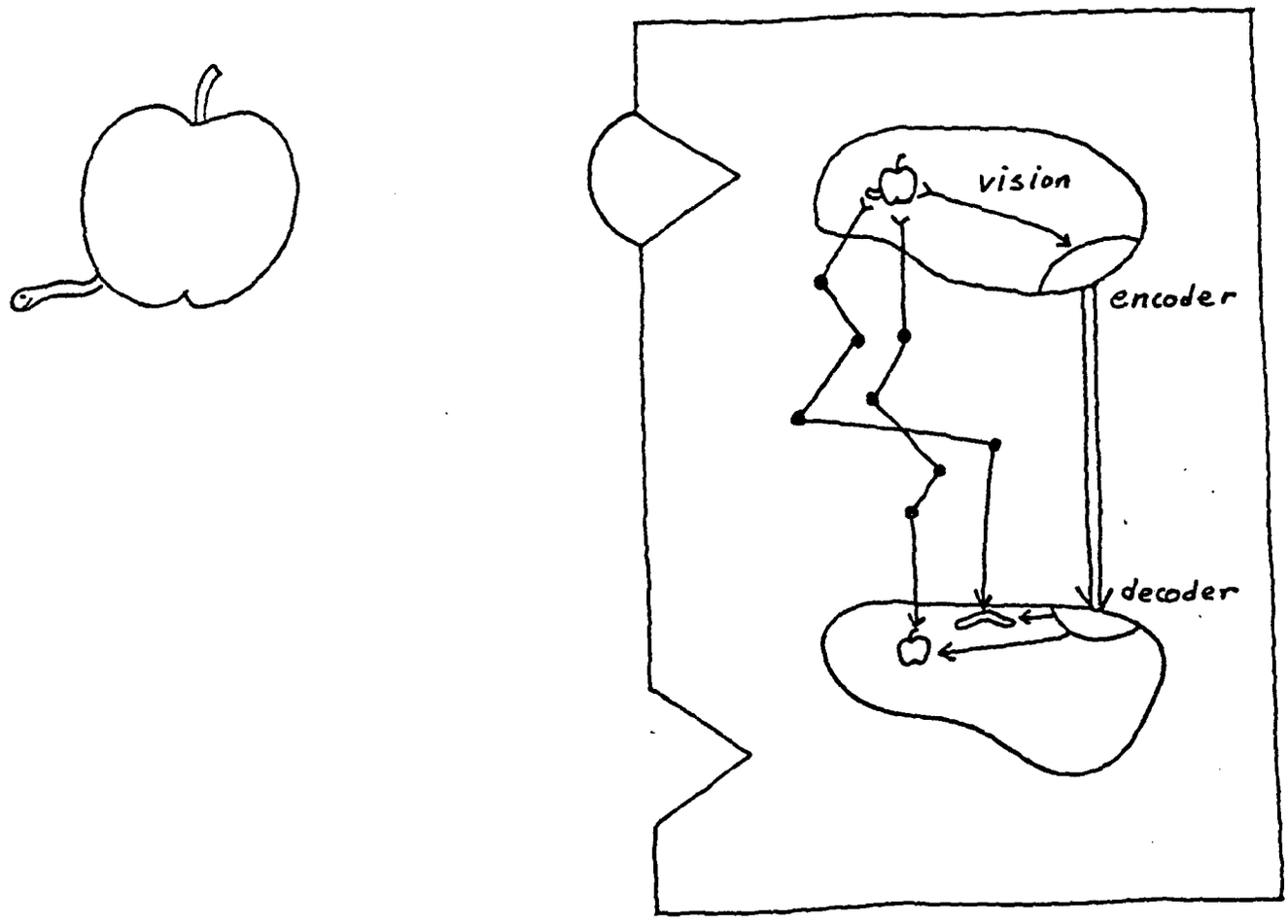
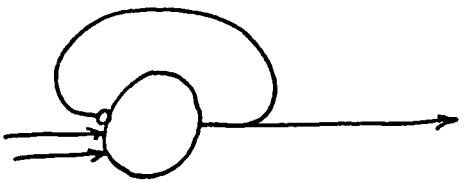



Figure 1

Connectionism vs. Symbollic Encoding

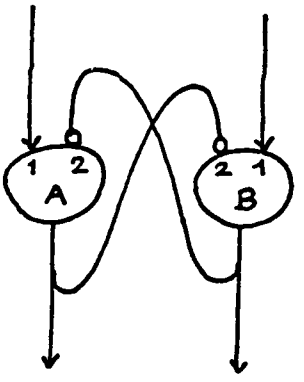
⟹ Assumes some general encoding

⟶ Assumes individual connections



if isolated: $\rho = \rho_0 e^{-k(t-t_0)}$

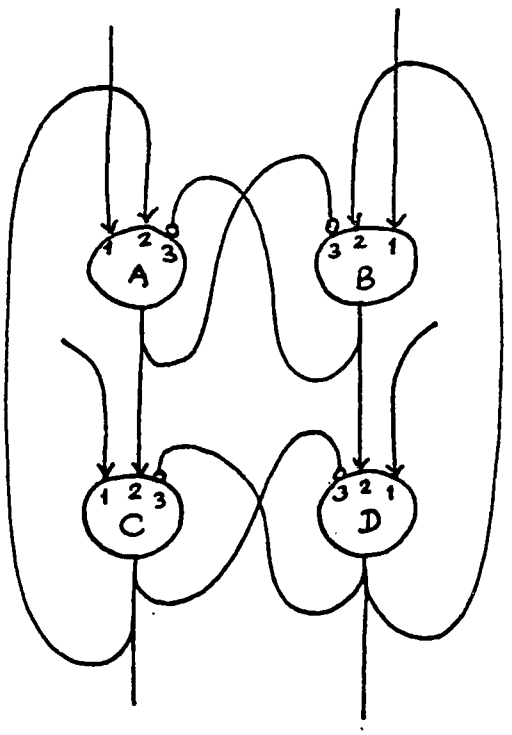
Figure 2. Self-inhibition and Decay




Suppose $A_1$ received an input of 6 units, then 2 per time step
$B_1$ " " " " 5 " " 2 " " "

| t | P(A) | P(B) |
|---|---|---|
| 1 | 6 | 5 |
| 2 | 5.5 | 4 |
| 3 | 5.5 | 3.5 |
| 4 | 6 | 3 |
| 5 | 6.5 | 2 |
| 6 | 7.5 | 1 |
| 7 | 9.5 | 0 |
| 8 | Sat | 0 |

Figure 3.




| t | P(A) | P(B) | P(C) | P(D) |
|---|---|---|---|---|
| 1 | 6 | 5 | 6 | 5 |
| 2 | 6.5 | 4.5 | 6.5 | 4.5 |
| 3 | 7.5 | 3.5 | 7.5 | 3.5 |
| 4 | 9.5 | 1.5 | 9.5 | 1.5 |
| | Sat | 0 | Sat | 0 |

Figure 4.

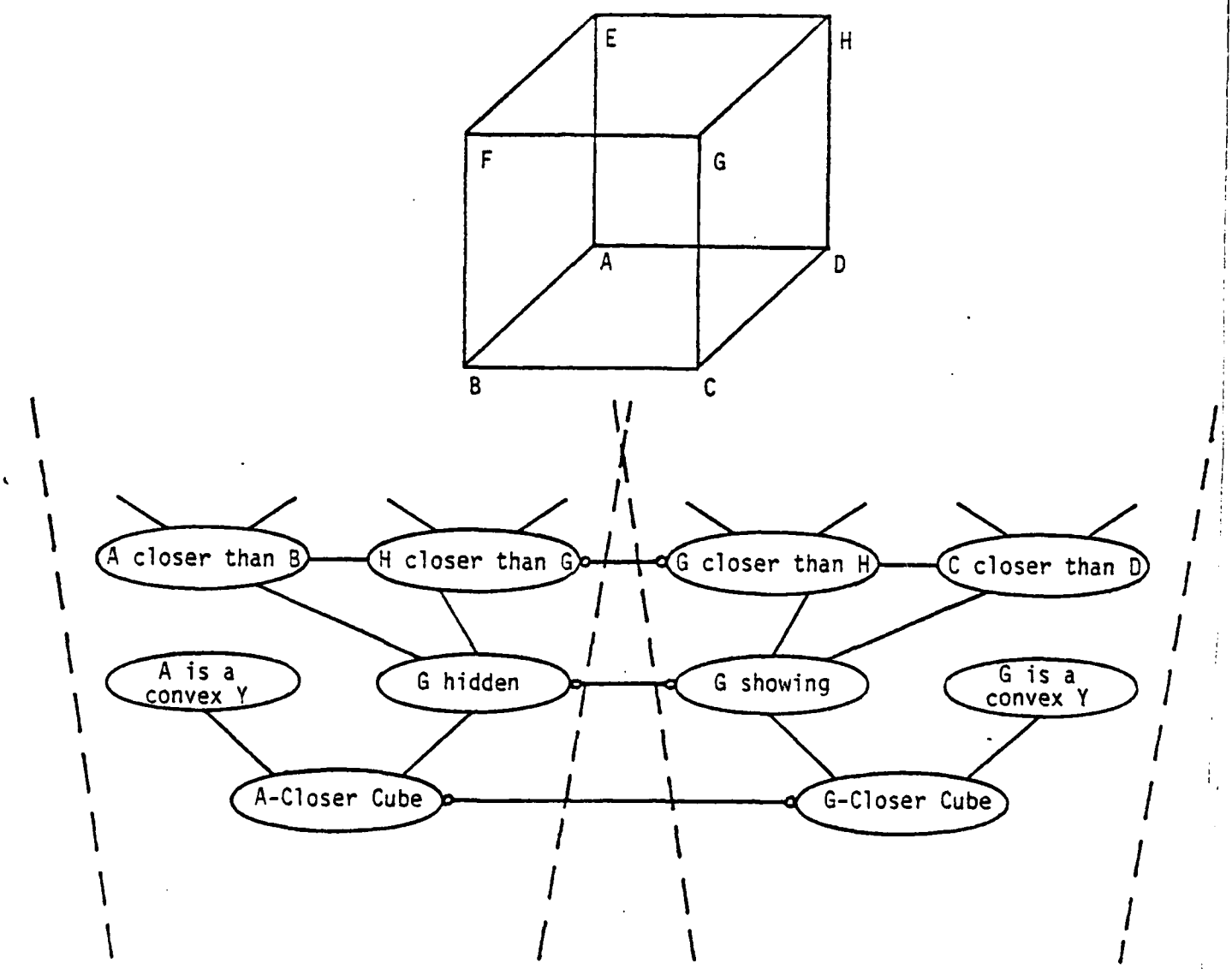



Figure 5: Network Fragments for the Two Readings of the Necker Cube.

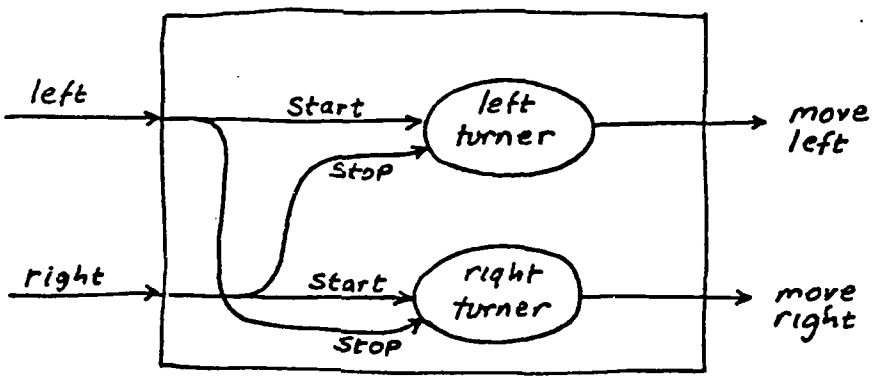

left
Start
left turner
move left
Stop
right
Start
right turner
move right
Stop

Figure 6.

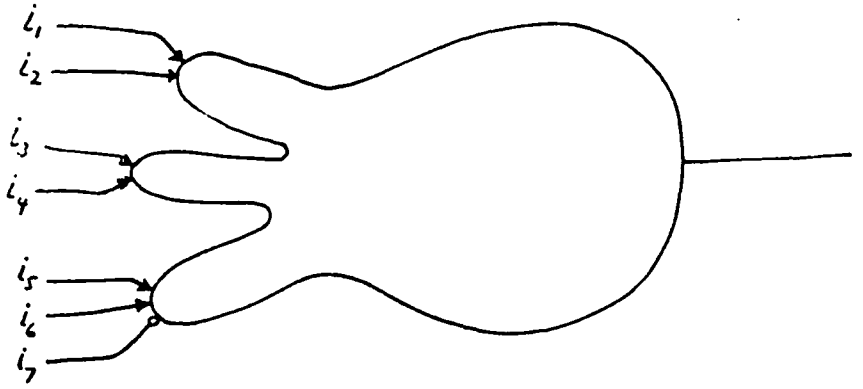



Figure 7 Conjunctive Connections and Disjunctive Input Sites

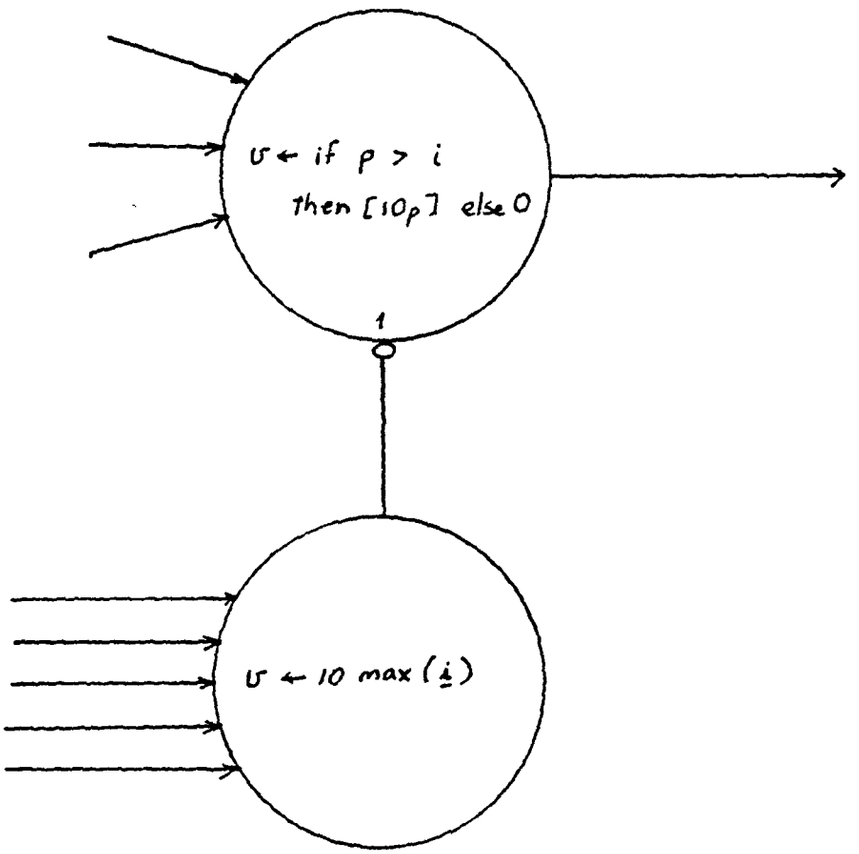
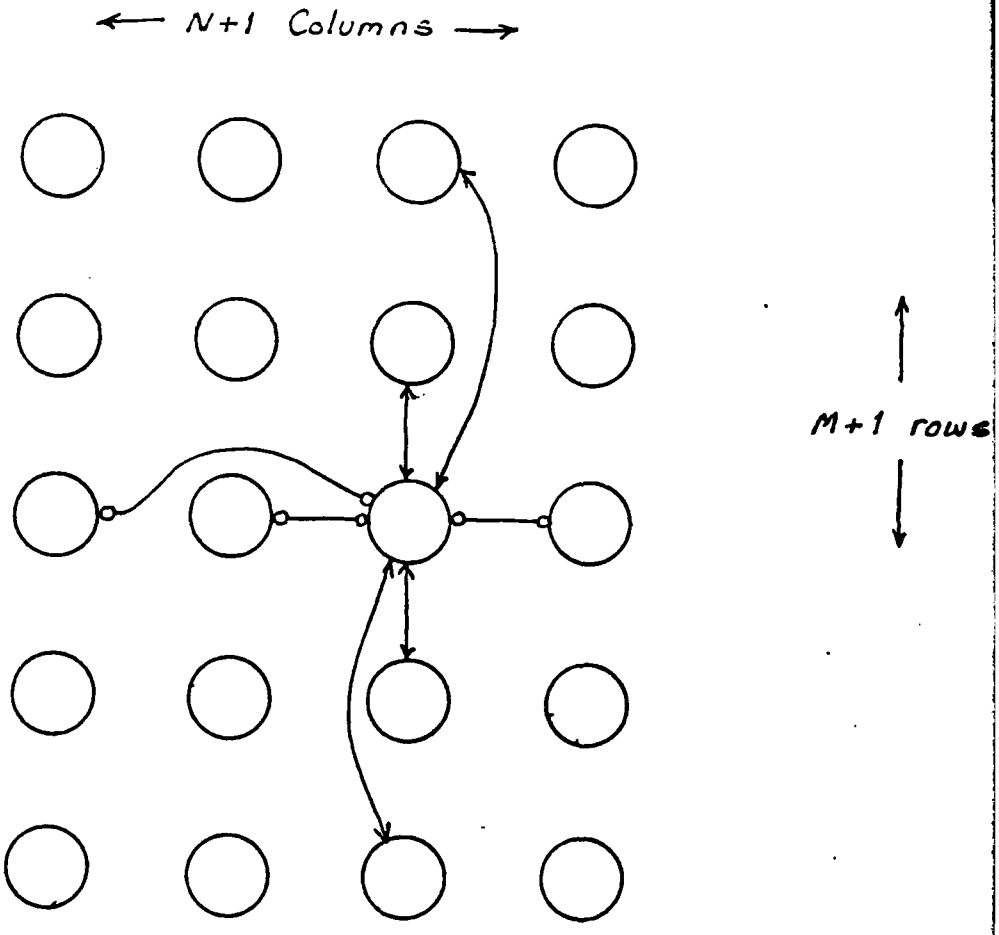


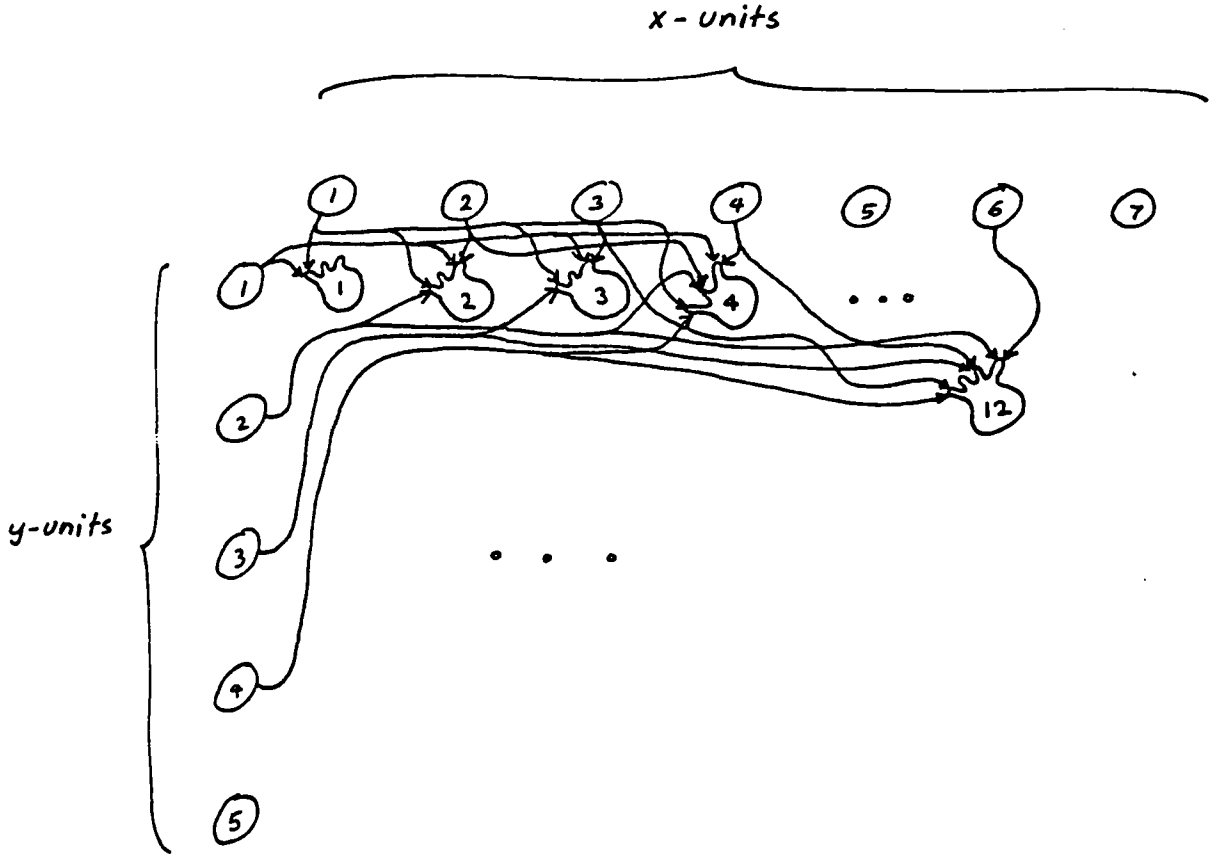

Figure 8. Paired Units for Max WTA




Each unit has N identical inhibitory inputs
and M identical excitatory inputs

Figure 9. Artificially Symmetric Coalition Structure

$$z = f(x,y) = xy$$




z unit :

Figure 10

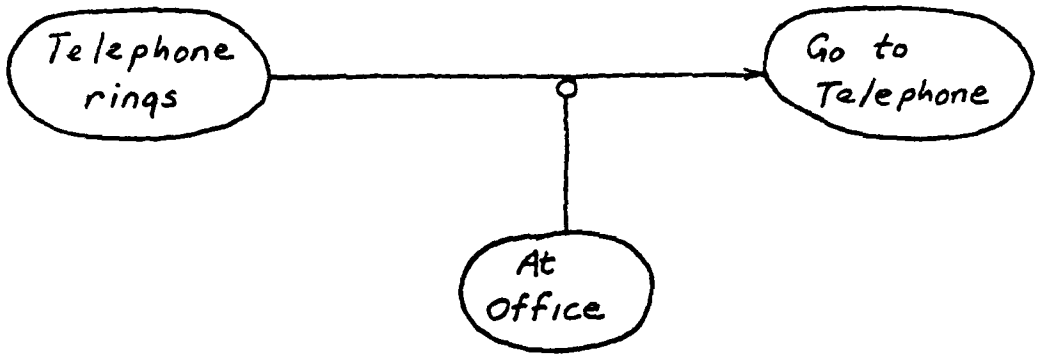



Figure 11. Modifier ($m_j$) on a connection

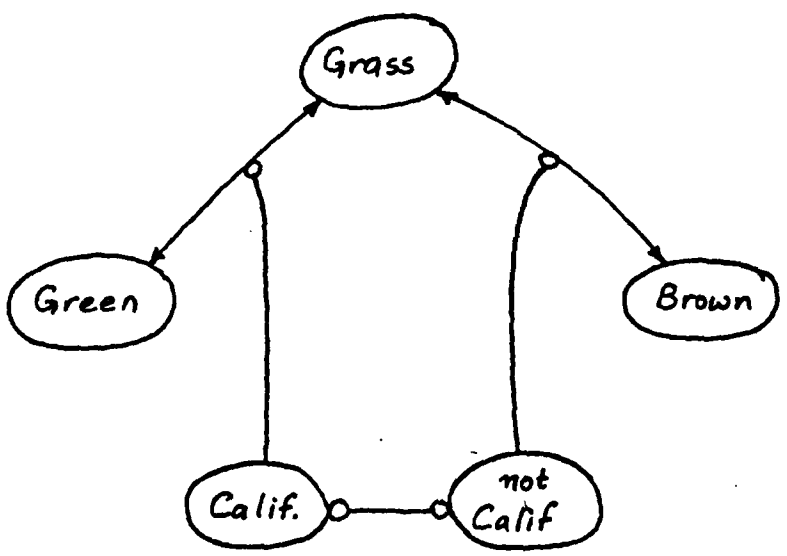



Figure 12. Grass is green connection modified by Calif.

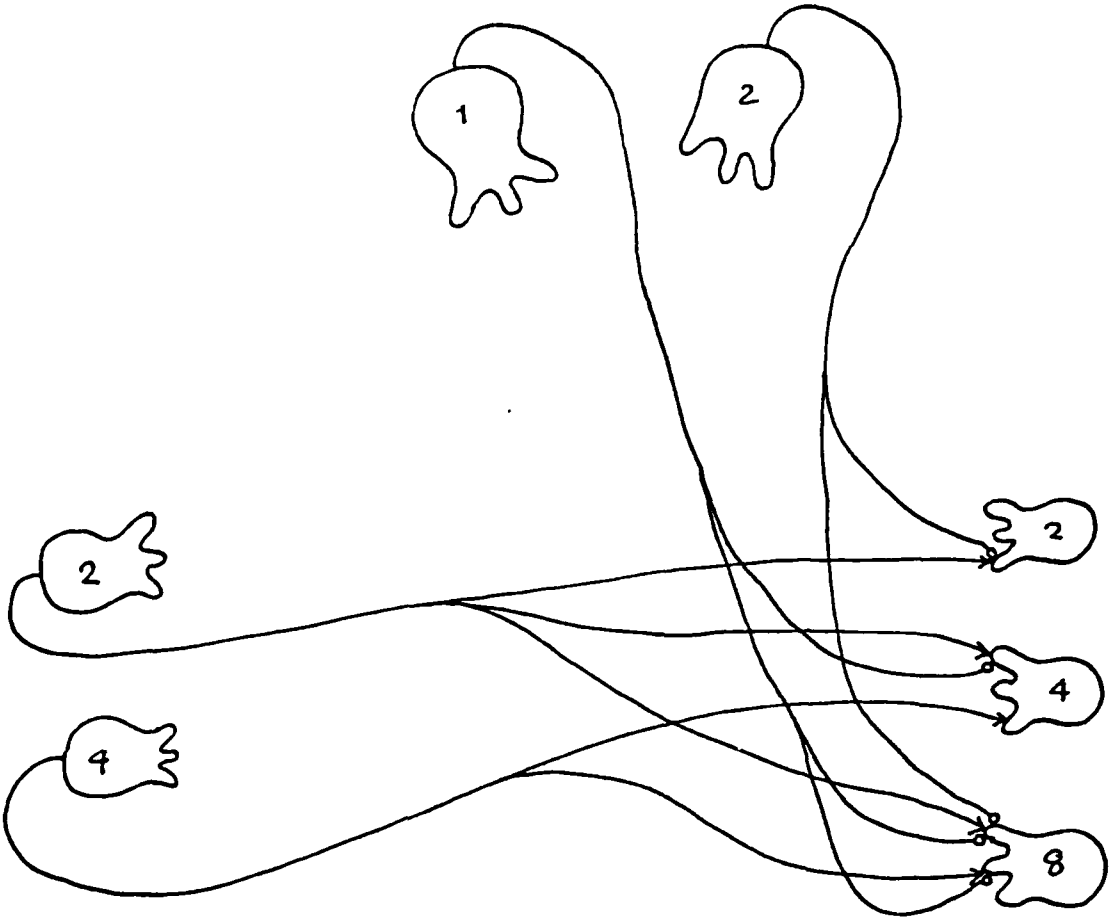



Figure 13 (Only a representative subset of the connections are shown)

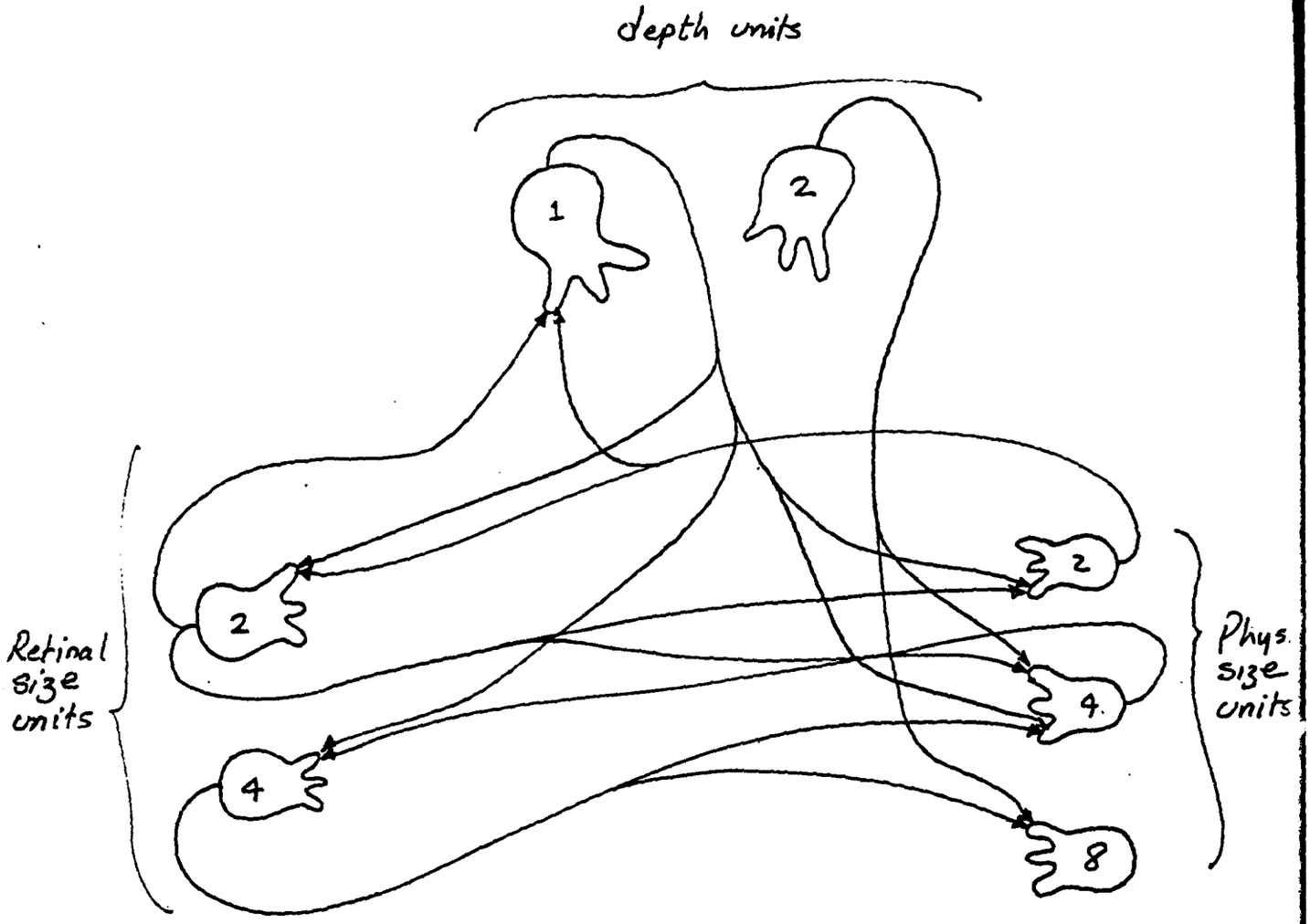



Figure 14 (Only some of many connections shown)

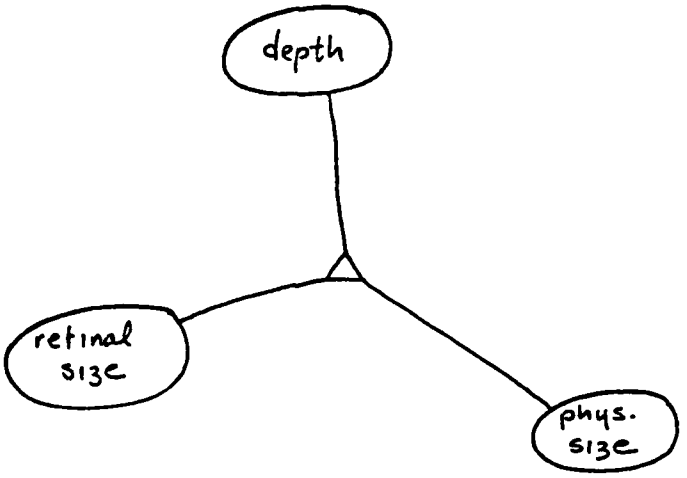



Figure 15 Hinton's notation

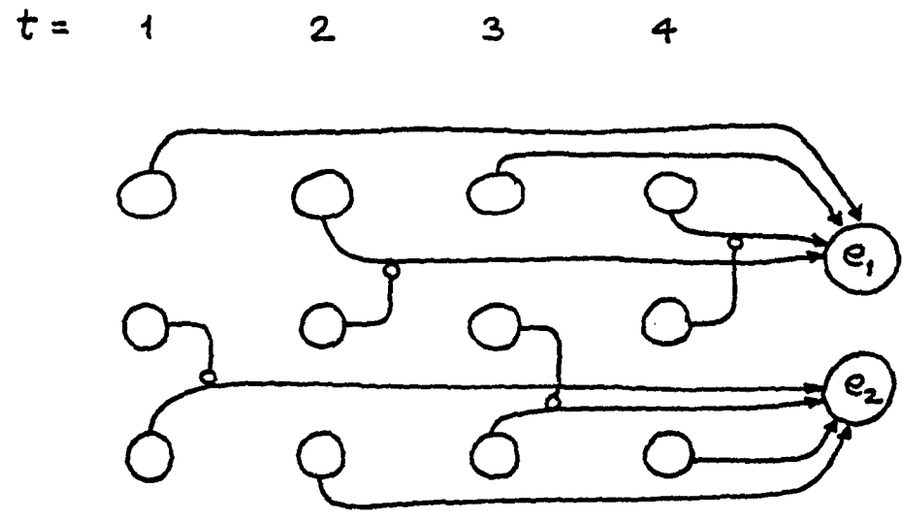



Figure 16 A simple sequencer using modifiers

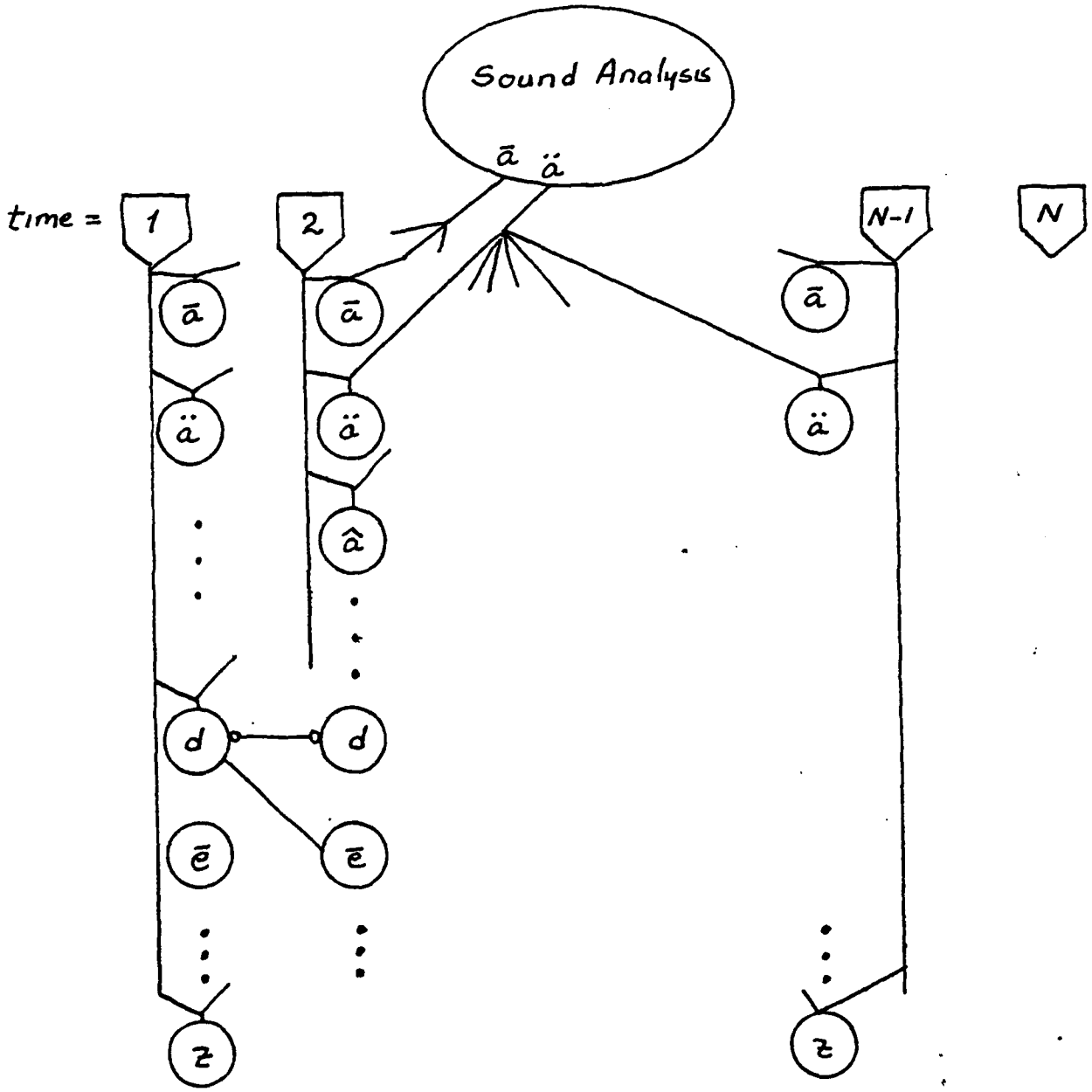



Figure 17 A Phoneme Sequence Buffer

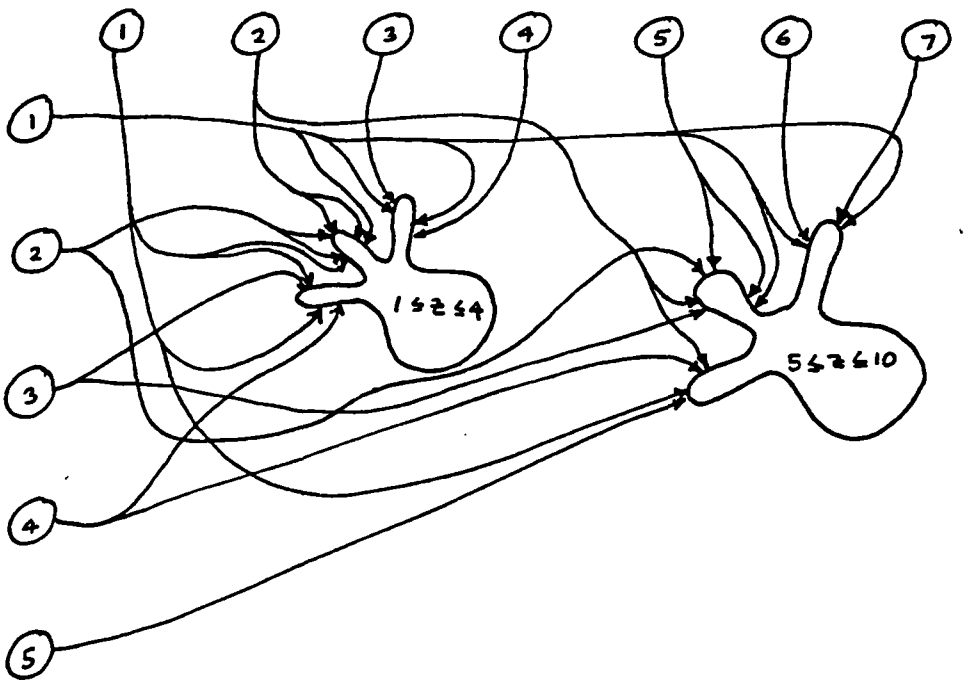



Figure 18 Modified table using less units

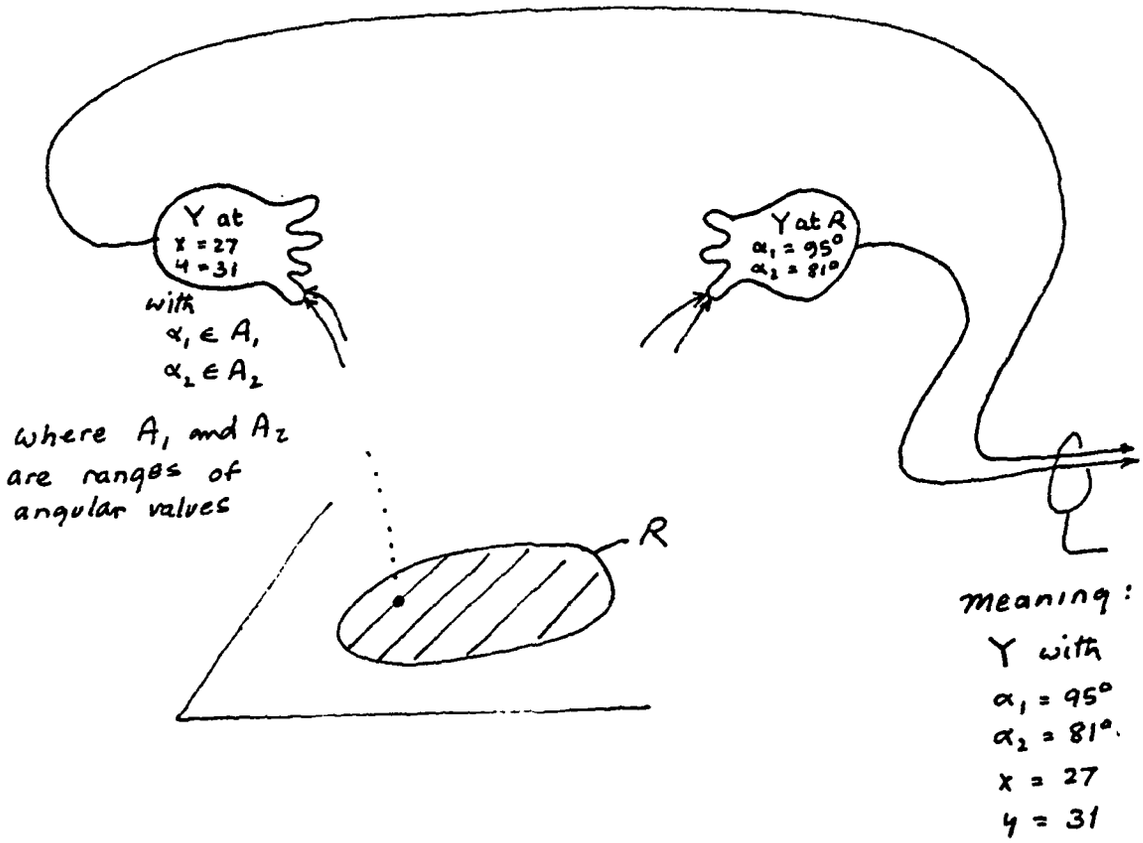



Figure 19

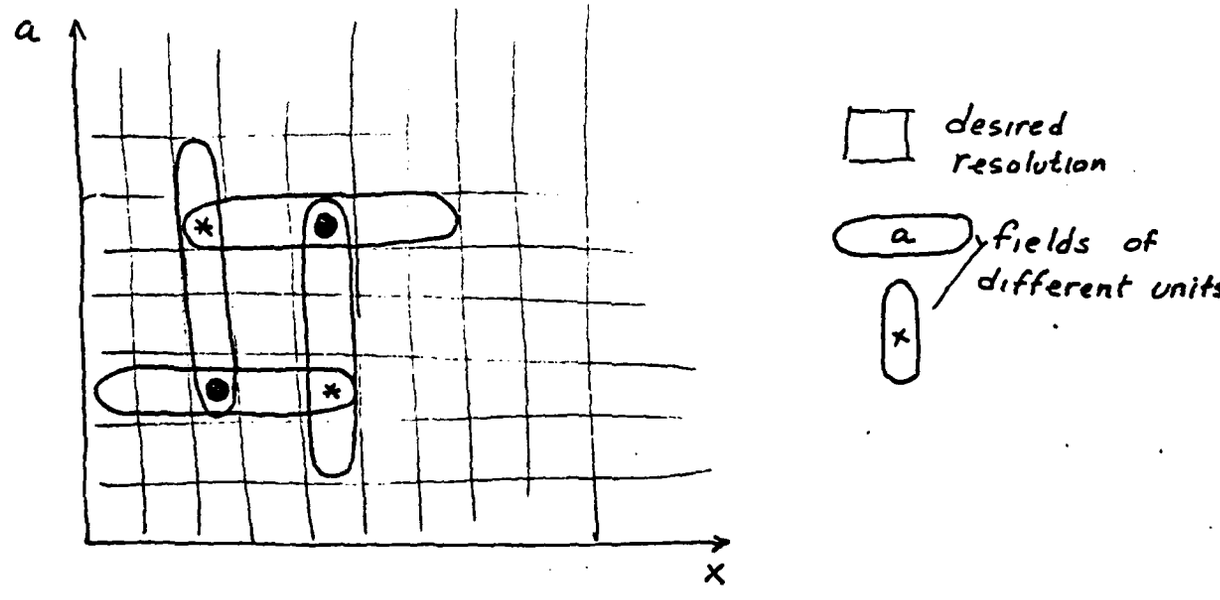



Figure 20 Inputs at (•) cause ghosts at (*)

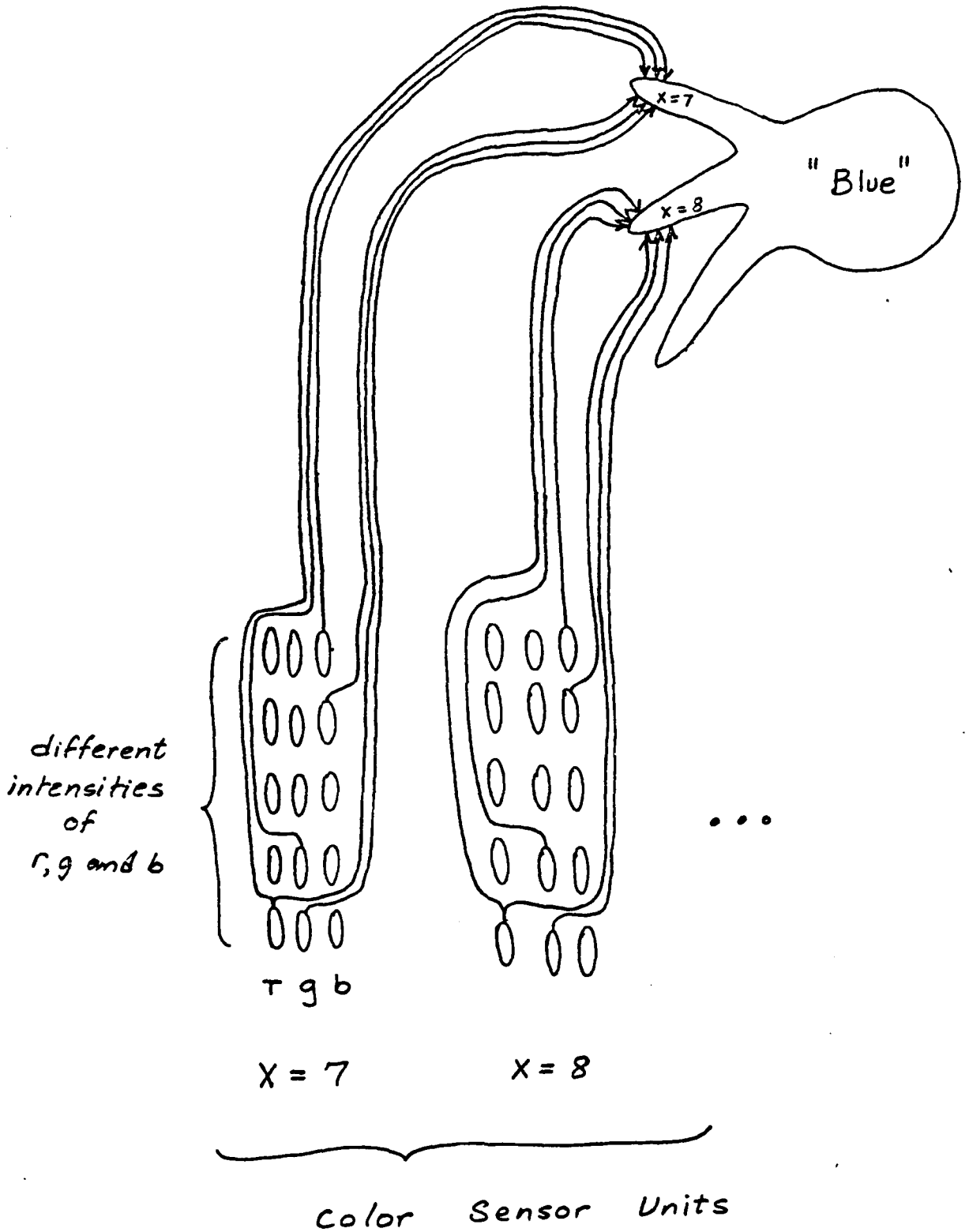



Figure 21. Feature measurement: a unit which responds to a range of blue

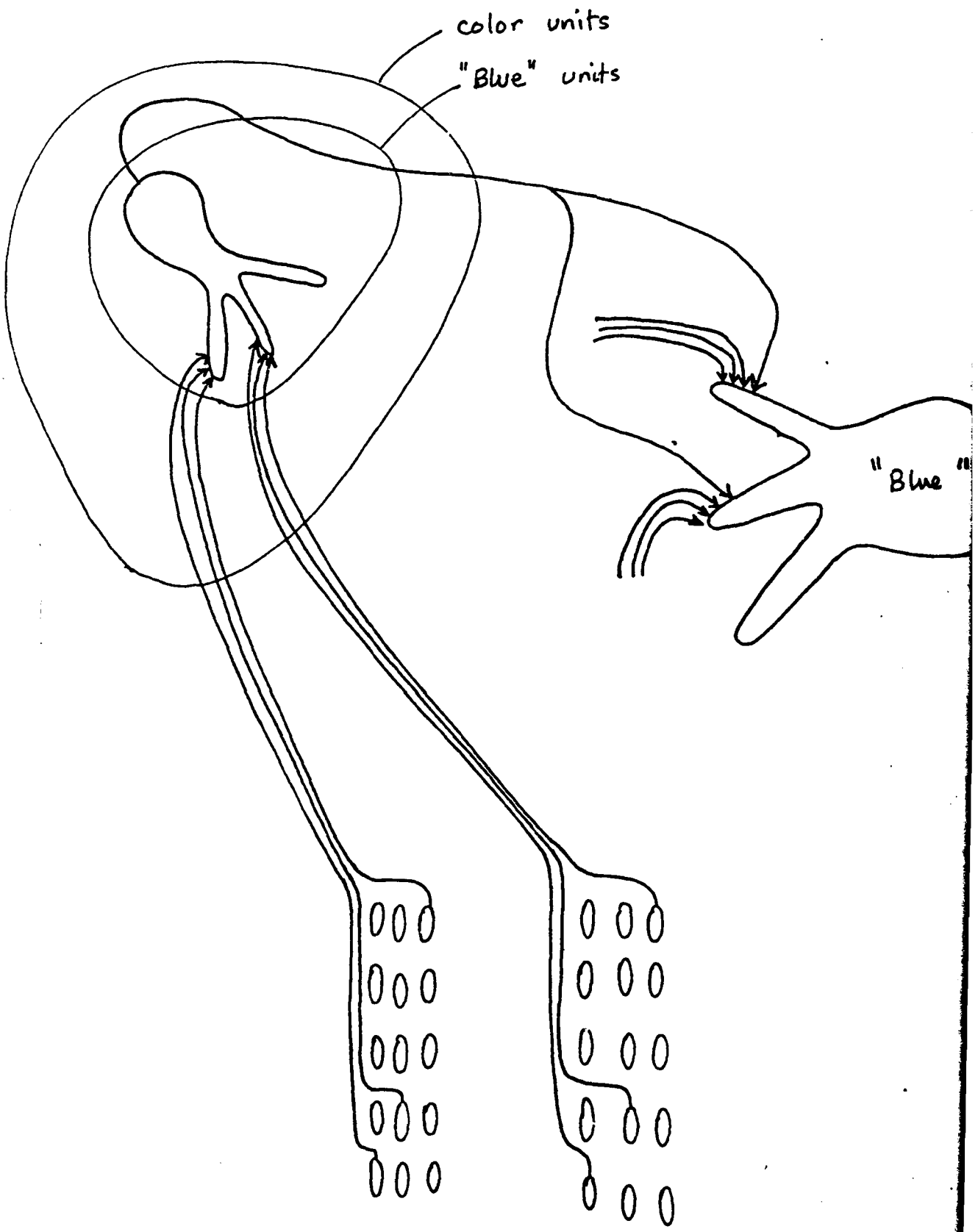



Figure 22 Tuning the "Blue" unit with a precise value unit

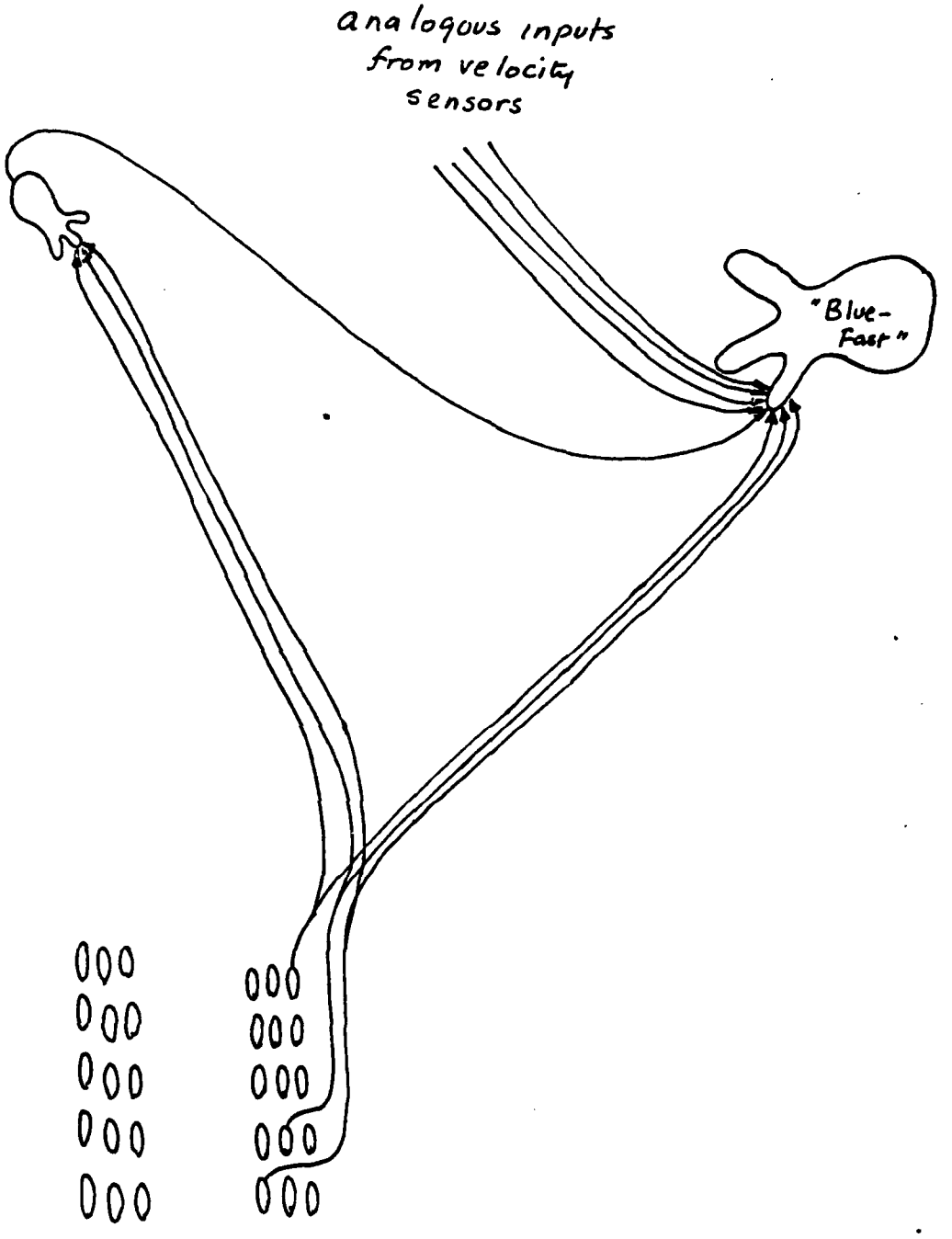



Figure 23. A BF Unit

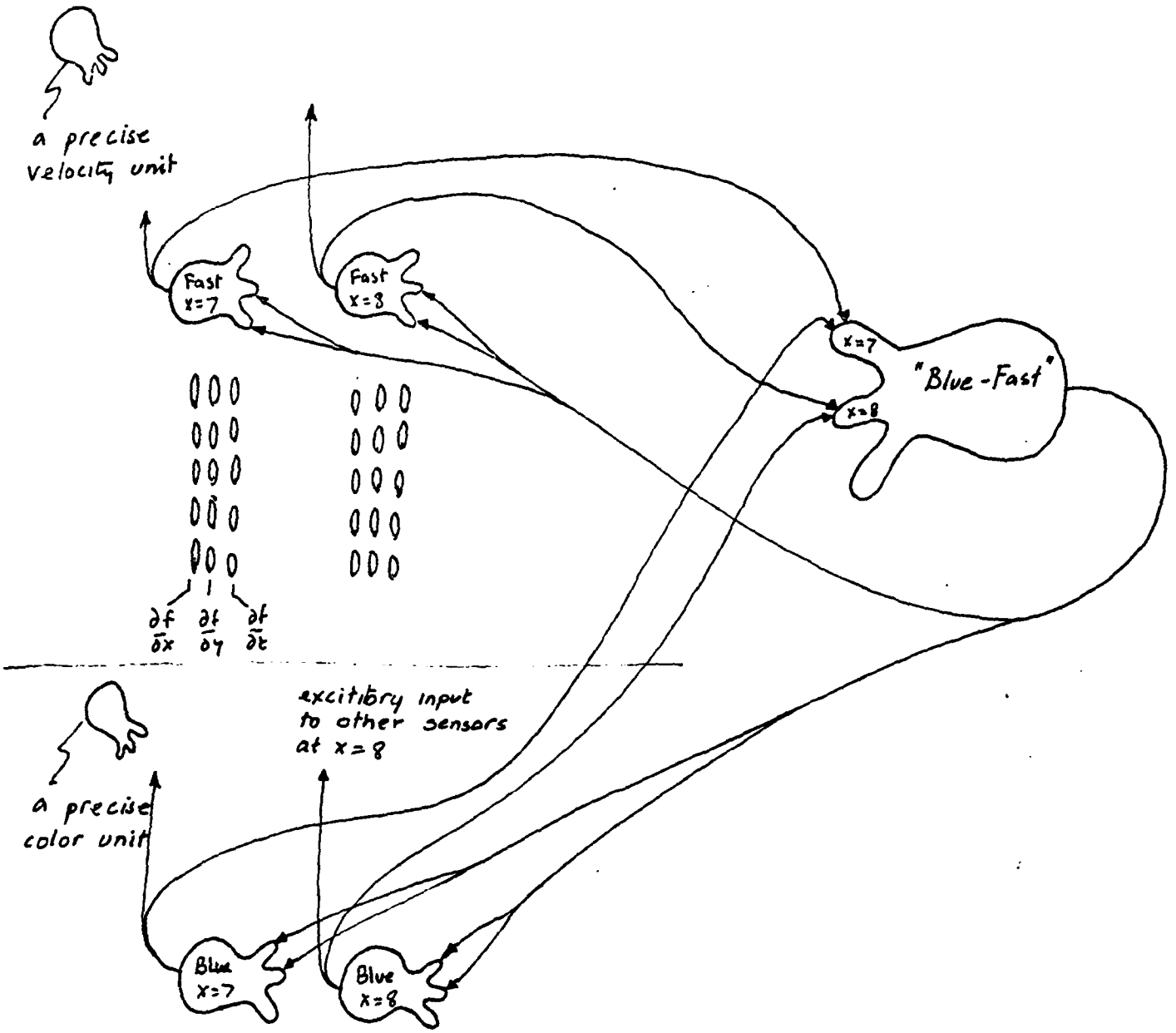



Figure 24 A Blue-Fast unit (see following figure for connection details) showing spatial context units

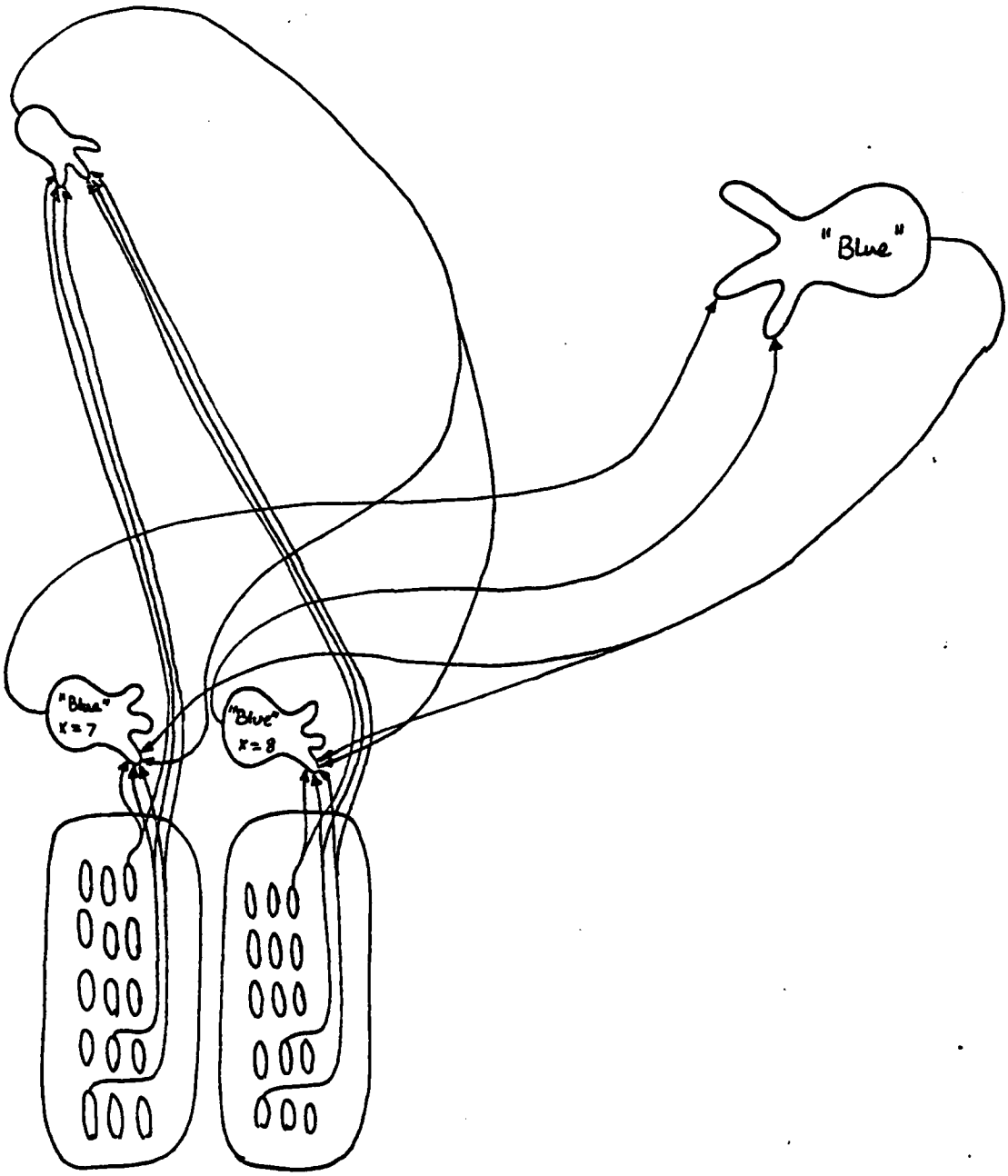



Figure 24a

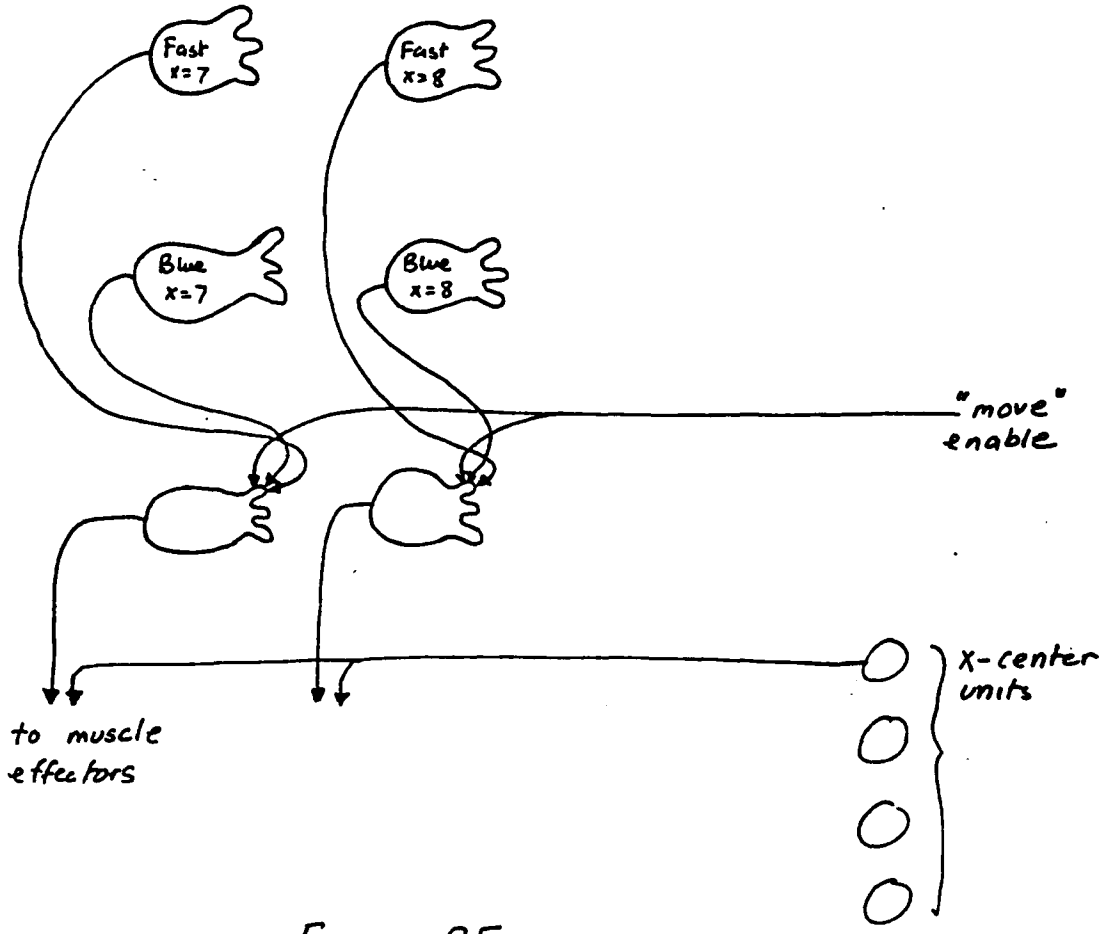

Fast
x=7
Fast
x=8
Blue
x=7
Blue
x=8
"move"
enable
to muscle
effectors
x-center
units

Figure 25

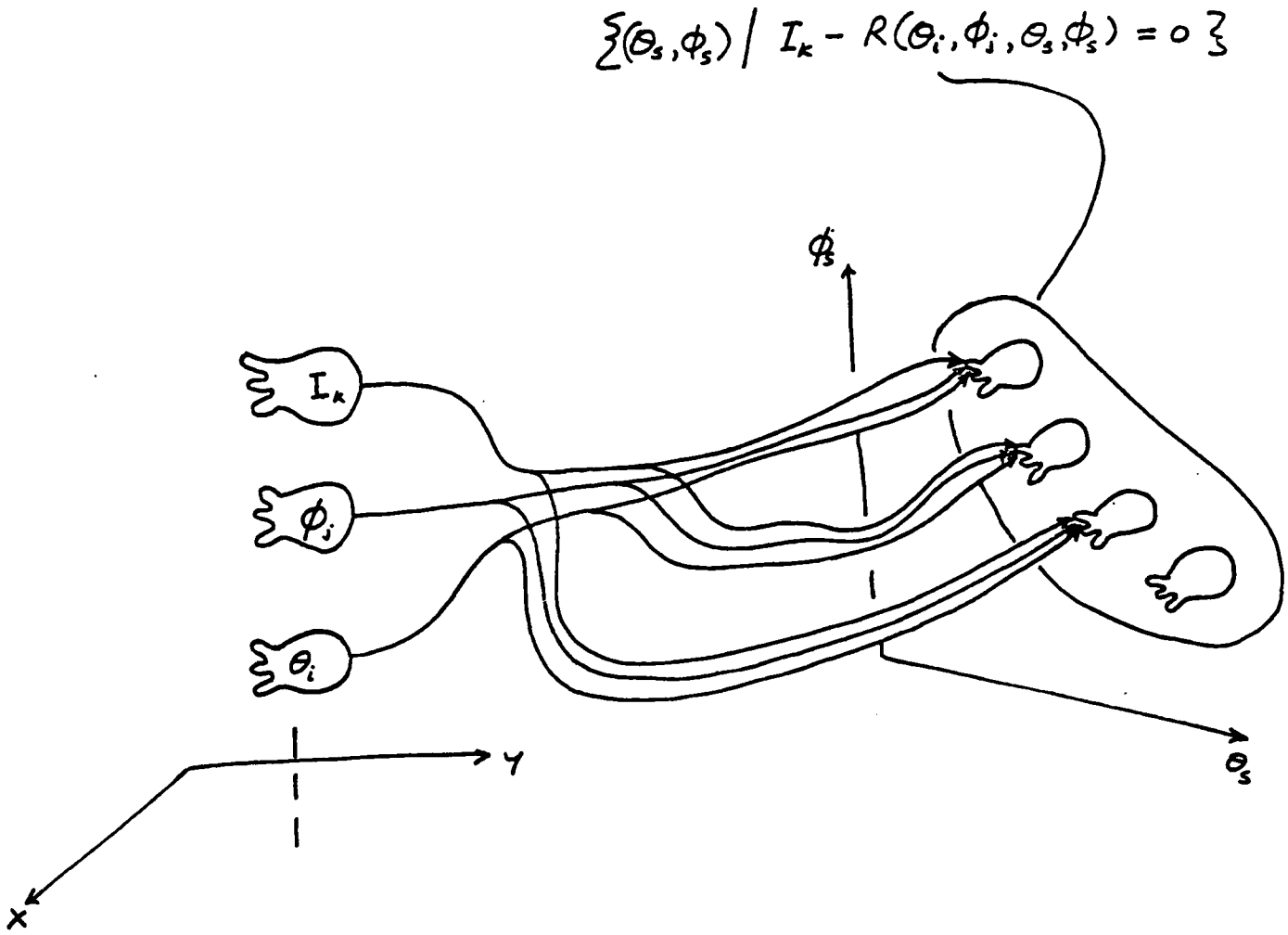
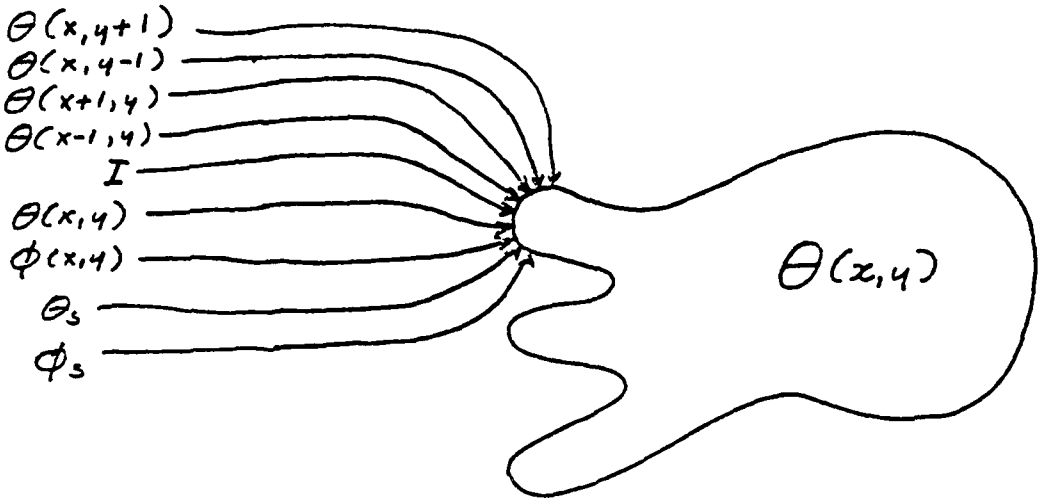
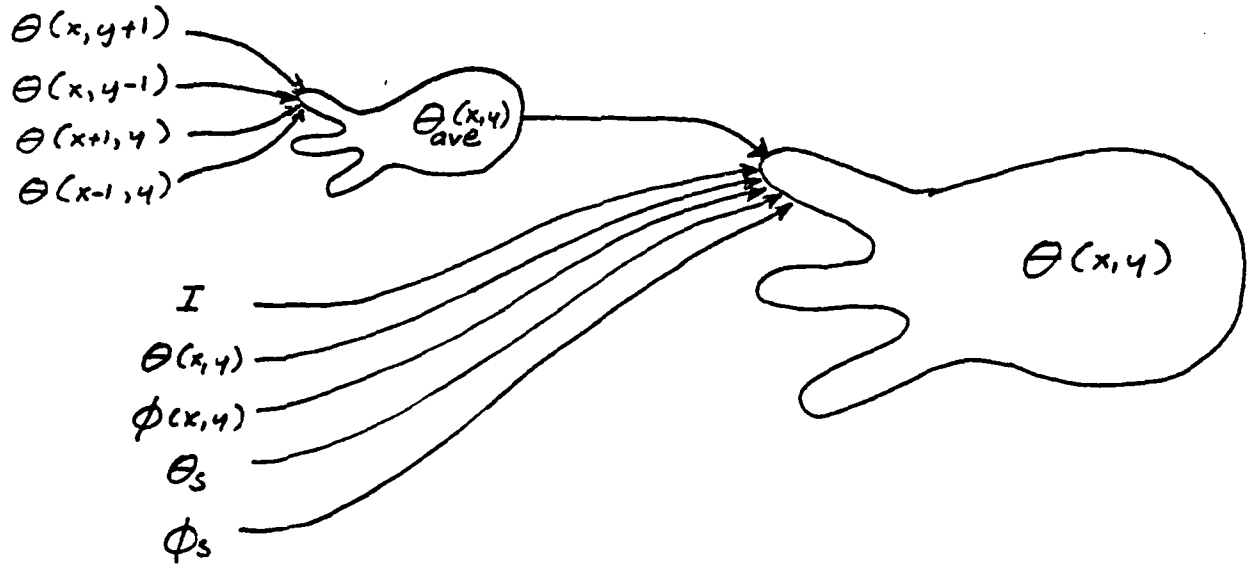



Figure 26 A portion of the connections used to detect illumination angle




a.




b.

Figure 27 . Alternate connection schemes for computing surface orientation