Simple computer models for hierarchical processing applied to sensory receptive fields

by

Eduardo Carlos Solessio

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Major: Biomedical Engineering

Iowa State University
Ames, Iowa

1988

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

It is an accepted fact that perceptual information undergoes preprocessing and is transformed as it passes from the periphery of the nervous system to the higher centers in the brain. Apparently, one of the objectives of preprocessing is to decrease redundancies and enhance features in the afferent information, so as to decrease the amount of processing required at the brain. The brain interprets the arriving information and associates it, (it is believed), with certain classes or concepts.

Neurons are the processing units in the nervous system. Neurons have intrinsic properties that characterize their responses to stimuli; but neurons are not independent units. Neurons interconnect and communicate extensively with each other. Thus, when a stimulus is applied, many neurons are activated. Each neuron receives input from other neurons in the system, integrates the converging information, and produces a response governed by its own intrinsic properties. Processing of the information depends on the neurons' properties and on their interconnections.

The purpose of the study was to describe and discuss possible properties and interconnections of neurons in an idealized sensory network and to evaluate their functional significance in the processing of topographic information.

The properties and interconnections proposed were based on physiological evidence obtained from classical literature. Assumptions were introduced where there was a lack of physiological evidence.

A mathematical model was built based on the quantitative description of neuron properties and interaction characteristics. The model was made up of units (which represented the neurons of the nervous system) that responded to a stimulus according to a 'power law' transfer function. The units were arranged in levels (that represented the synaptic stations in the nervous system) with lateral interactions among units in the same level and convergent and divergent interconnections to units in other levels. The number of units decreased at each of the successively higher levels of the hierarchically organized model. The implemented model had a hierarchical organization and a pyramidal structure.

A computer program was written and run to evaluate the model. Two model versions were tested. First, a one dimensional model with linear and piecewise linear units was tested. Then, a two dimensional model, which was an extension of the one dimensional model, was also tested.

The time invariant receptive fields of units in the one dimensional model were evaluated. The receptive fields were

characterized by an on-center response for units with self-inhibition and an off-center response for units with self-excitation. In addition, when the piecewise linear units were introduced, adaptive receptive fields resulted. The range of the inhibitory response of the units was limited resulting in fields with large inhibitory regions and decreased disinhibitory effects. The adaptive response was most noticeable when the response to a step function was tested.

It was demonstrated that a transient response with respect to time can be obtained from units with sustained response as a result of feedback mechanisms. External interconnections, such as self-feedback, can serve the purpose. Thus, the response with respect to time of a unit can be governed by the interconnections of the external network. The transient response of a unit combined with a nonsymmetric receptive field yielded a unit with a response that was sensitive to the direction of motion of the stimulus.

Finally, a time invariant two dimensional model was tested. This model proved sensitive to spatial features of the input stimulus. Edges were enhanced in the first hierarchical level and sensitivity to edge orientation was observed in the second and third levels. Corners were

detected in the last hierarchical level, and were represented in the response map as peaks surrounded by moderate activity.

CHAPTER 2: LITERATURE REVIEW

The mathematical model built was based on a number of physiological facts found in the classical literature. Most of the information was obtained from studies of cats and monkeys which, like man, are species in the higher end of the evolutionary ladder. The purpose of this chapter is to present the information that was used in synthesizing the model.

The somatic and visual modalities were considered as primary sources of physiological evidence. These are topographic modalities with similar functional and organizational characteristics in their pathways from the sensing surface to the higher centers. The first objective of this review is to highlight those features that are common to both modalities and might therefore be basic components in the processing of sensory information in the afferent nervous system.

The concept of receptive fields is introduced and tactile as well as visual receptive fields at different levels of the afferent pathway are described. Proposed models for receptive fields by Rodieck, Marr and others are discussed. At the cortical level, the work by Hubel and Weisel is presented. Finally, the studies by Hartline and Ratliff concerning the eye of the horseshoe crab are

discussed along with the reasons why their findings were applied to the present model. As shall be seen later, lateral inhibition may play an important role in the determination of the receptive field and the Hartline-Ratliff equations appear to correctly describe the quantization and distribution effects of such inhibitory interactions.

Some generalities in the somatic and visual pathways can be deduced after observing the anatomical constitution of both afferent systems. According to Kandel and Schwartz [1981], the pathway for the somatic sense begins with a receptor sensitive to a specific submodality, for example a Meissner corpuscle that mediates superficial touch. Large myelinated fibers from these receptors in the skin enter the spinal cord via the dorsal roots and branch upward to synapse in the medulla with cells in the dorsal column nuclei (the gracile and cuneate nuclei). Second order sensory neurons in the dorsal column nuclei cross the midline in the medulla and ascend the brainstem on the opposite side and form synapses with cells in the ventral posterior lateral nucleus of the thalamus. The third order neurons in the thalamus send axons to the primary and secondary somatic sensory regions in the cortex of the brain.

The peripheral anatomical path for visual information can be described as follows. The photoreceptors (cones or rods) synapse on bipolar cells in the outer plexiform layer of the retina. The bipolar cells transmit the information to the ganglion cells with which they synapse at the inner plexiform layer. The ganglion cells, in turn, connect to either the lateral geniculate nucleus of the thalamus or to the superior colliculus (an important center for the control of head, neck, and ocular movements). Finally, the neurons originating in the lateral geniculate nucleus project to the striate cortex.

At each synaptic nucleus, the cell that projects out of the nucleus (relay cell) to the next nucleus (relay station) receives synaptic input from many afferent fibers, each afferent fiber terminating on many cells. Thus, convergence and divergence are characteristic interactions at the relay stations.

In addition to the relay cells, the input fibers also activate interneurons that can be either inhibitory or excitatory. In the visual system the horizontal cells make lateral inhibitory interconnections at the outer plexiform layer, while the amacrine cells are interneurons at the inner plexiform layer of the retina. Both are examples of local interneurons that contribute to the transformation of

information transmitted to the brain. In the touch sense the local interneurons at synaptic nuclei produce two types of inhibition; presynaptic and postsynaptic. Inhibitory interactions are general and are encountered repeatedly in the relay stations of both sensory systems.

One may deduce from the above description, that for the two sensory modalities the information is passed from one nucleus to the next one along relay neurons. In some nuclei the information may be transformed by the relay station interneurons and local interconnections and is then sent to the next processing station. This type of processing is representative of hierarchically organized systems.

Having described the general connectivity characteristics of the touch and vision systems, it is necessary to examine the processing properties of the neurons themselves. Neurons may be functionally described in terms of their input-output relationship. This relation has been determined experimentally by applying a controlled voltage across the neuron's membrane and measuring the resulting elicited response frequency [Stevens, 1966]. In general, the results show that the slope of the relation is progressively smaller for increasingly intense stimuli and that the response is negligible for stimulus intensities below a critical threshold value. A qualitatively similar

curve is obtained for receptor response to stimulation of the proper modality. The curve has been expressed in terms of a power law [Stevens, 1971],

$$\psi = K(I-I_0)^n \qquad (2.1)$$

for $I \geq I_0$ and $\psi = 0$ otherwise,

with,

ψ: firing rate

I: applied intensity

K: constant value

I_0: threshold

n: exponential.

The exponential value depends on the modality, experimental conditions, etc. In general the value of n varies within 0.2 and 1.0 s as to fit the compression of the response curve in the upper range of stimulus intensities (saturation effects).

We can assume that a receptor is a special type of neuron that shares the power law input-output relationship but differs in that its depolarization arises through the transduction or transformation of external signals into neural activity rather than through the summation of the activity of the converging neurons.

More centrally located neurons often present spontaneous activity. They often have an intrinsic or

resting firing rate which is different from zero. Stimuli may cause either an increase or a decrease of discharge in these neurons. Threshold in this case is defined as the stimulus magnitude that provoked a deviation in firing from the intrinsic firing rate [Somjen, 1972].

It is interesting to note that for small values of the exponential n, the power law approximates a logarithmic function. Therefore, the classic psychophysical law of Fechner-Weber can be considered a special case of the power law [Stevens, 1971]. This, and the fact that recordings at the first and higher order neurons yield input-output relations that can be expressed in terms of the power law, may have important implications, although it is not yet known how the power law at the neural level relates to the power law at the perceptual level (if, in fact, they are actually related).

To evoke a response in a neuron, the stimulus must be applied within the neuron's receptive field. The receptive field has been defined as 'that area in space where an adequate stimulus evokes a discharge of impulses in the afferent unit' [Mountcastle, 1967]. In the context of the present study, the 'area in space' will be interpreted as the skin surface when referring to the touch sense and as the retina when referring to vision. The concept of

receptive field and its properties is important in that it is representative of the processing faculties of the portion of the afferent system extending from the input surface to the monitored neuron, although it does not elucidate the interconnections and synaptic interactions in the network.

Neurons in the afferent nervous system are not only sensitive to stimulus intensity but to other stimulus factors as well. Other factors that affect the response of a neuron may be: stimulus shape, stimulus orientation, stimulus size, and time factors such as time of stimulation, time between stimulations, and stimulus motion. The number of factors is large and all affect the neuron's response to a larger or smaller degree. Optimum stimuli result in maximum discharge.

The receptive field of ganglion cells in the cat retina was mapped by shining a spot of light on different areas of the retina [Kuffler, 1953]. Two basic receptive field types were found: on-center and off-center. The receptive fields were roughly concentric to the ganglion cell. In an on-center receptive field, light produced an excitatory discharge when applied in the center, whereas, if applied in the surrounding region, inhibition of the discharge with respect to its intrinsic firing rate occurred. The center and surround were antagonistic and tended to cancel each

other when the entire receptive field was uniformly illuminated. The boundaries of the receptive field could not be delineated accurately. They contracted under high background illumination and expanded during dark adaptation. Off-center ganglion cells responded in exactly the opposite way, showing inhibition in the center and excitation in the surround.

Kuffler suggested that the antagonist response in the areas of the receptive fields could result if the connections of the central receptors to the ganglion cell were of one type (either excitatory or inhibitory) while the other type of connection was used for the interconnection of the surrounding receptors to the ganglion cell. He also noted that ganglion receptive fields were overlapping, that is, one spot of light shone in the retina elicited discharge from many ganglion cells. Interestingly enough, overlapping receptive fields projected to neighboring ganglion cells; neurons processing related information were clustered together. This is an important principle of neural organization as will be seen below.

Other models were proposed for cat retinal receptive fields. Some were functional models that applied linear systems techniques. In his model, Rodieck [1965] assumed superposition to be valid. He expressed the response of the

'unit' (the 'units' of the model were representative of neurons in the nervous system) to a moving stimulus as the convolution of the shape of the static receptive field of the unit with a term which was, in turn, the convolution of the first derivative of the shape of the stimulus with the temporal step response. The function he chose for the representation of the receptive field shape was the sum of two two dimensional Gaussian functions, a narrow positive going one and a broad negative going one. These two Gaussian functions did not have a special physical significance; only the overall receptive field shape was considered. The model was consistent with experimental results, but the analytic expressions chosen for the receptive fields and temporal step response were very sensitive in describing the response to moving stimuli at very high or very low velocities (short stimuli in time and long stimuli in time).

A model of directionally sensitive receptor units and their use in the processing of visual motion was proposed [Marr and Ullman, 1981]. These units computed the convolution of the geometrical shape of the receptive field with the image. The shape of the receptive field was obtained from the Laplacian of a Gaussian, $\nabla^2 G$. This differed little from the 'difference of Gaussians' shape,

provided the respective standard deviations were correctly adjusted.

Intensity changes were detected by noting the zero crossings of the convolution of the Laplacian operator with the image,

$$S(x,y,t) = \nabla^2 G * I \qquad (2.2)$$

which Marr suggested corresponded to the response of an X type ganglion cell (sustained response), while the measurement of its partial time derivative,

$$T(x,y,t) = \partial(\nabla^2 G * I)/\partial t \qquad (2.3)$$

corresponded to the response of a Y type cell (transient response).

These components combined to conform a directionally selective zero crossing detector. The receptive field had three components: a sustained on-center X unit, a sustained off-center X unit, and a transient or Y unit. The output of the X units was combined by a logical AND gate. The AND gate detected the presence of a zero located between the two X units. The transient Y unit indicated the direction of motion at the point where the zero had been detected.

The retinal cells, of characteristic on-center and off-center fields, project to the next relay station, the lateral geniculate nucleus (l.g.n.). Therefore, the l.g.n. cells are considered to be in a higher hierarchical level

than the retinal cells. What do the receptive fields of l.g.n. cells look like? The recorded receptive fields at the l.g.n. consisted of three concentric zones: center, antagonistic surround, and synergistic outer surround [Hammond, 1973].

Similarities to cells at the retinal level (on-center and off-center receptive fields) were observed with either sustained responses (X cells) or transient responses (Y cells). Contrary to retinal cells, the surrounds of the l.g.n. fields showed a more powerful antagonism towards their centers.

The receptive field shape was explained in terms of the contribution of the lower converging retinal cells. If each single cell in the l.g.n. received field center input from one or several retinal cells of the same type (all on-center or all off-center), and field surround input from a number of complementary type retinal cells (or same type via inhibitory interconnections), then the retinal surround would give rise to the synergistic outer surround of the l.g.n. receptive fields.

Application of a stimulus (spot or annular) in the synergistic surround of the field resulted in a response having the same characteristics as those obtained for a stimulus applied in the center of the field. For example,

let us assume that that the l.g.n. inhibitory surround was the result of inhibitory interconnections from surrounding on-center retinal cells. When the inhibitory area of the receptive field of a retinal cell was stimulated, the discharge of the retinal cell decreased. Because of the inhibitory nature of the projection of the retinal cell to a l.g.n. cell, a decrease in the activity of the retinal cell resulted in a decrease of the inhibitory input to the l.g.n. cell. As a consequence the response of the l.g.n. cell increased; there was disinhibition.

Excitatory and inhibitory interactions were also found in touch receptive fields. Stimuli delivered on the contralateral forearm of a monkey provoked changes in the discharge rate of central neurons in the post central gyrus of the brain [Mountcastle, 1974]. The cell was excited by stimuli delivered on the preaxial side of the arm and inhibited by stimuli delivered within the surrounding area. Mountcastle suggested that this form of spatial response could be caused by inhibition exerted at every level of the afferent pathway by way of local interneurons.

Other experimentors reported that the receptive fields of cells in the cat visual cortex suggested a degree of complexity at the upper levels that far exceeded anything seen at the lower, more peripherial levels [Hubel and

Weisel, 1962]. A number of functionally different cell types was found, the great majority falling in two groups: simple and complex cells.

The simple receptive fields were radially asymmetric with antagonistic excitatory and inhibitory areas separated by straight lines. The spatial distribution of excitation and inhibition differed from cell to cell, and their effects cancelled when the stimulus consisted of uniform diffuse illumination of the entire receptive field. The most effective stimuli found were slits, edges, and dark bars. Shape, size and position of the stimulus were critical features for the response of the cell. Many of these cells increased their sensitivity to moving stimuli.

Contrary to simple receptive fields, the response of complex cells could not be predicted from maps made with small circular spots. The excitatory and inhibitory regions could generally not be determined. These cells were responsive to spatial features of the applied stimulus such as slits, edges, and dark bars. Stimulus shape and orientation were important factors in determining the response, but, unlike simple cells, stimulus position was no longer a critical factor. The cell fired for several positions of the stimulus within the receptive field. Other variants in the response of complex cells required

additional constraints in the stimulus for improved response [Kuffler et al., 1984]. For example, angles or corners in the stimulus shape as well as direction of motion were important features that resulted in enhanced response of these cells.

It was found that one or more of the receptive field characteristics tended to be shared by neighboring cells [Hubel and Weisel, 1962; Hubel, 1963]. The retinas projected upon the cortex in an orderly fashion. The cortex was divided into discrete columns and within each column the cells shared the same receptive field main axis orientation. These cells were either simple or complex, their receptive fields were situated in the same region of the retina and usually overlapped, although they did not precisely superimpose.

Hubel and Weisel proposed a hierarchical organization for the system. The simple cells received their afferent input from l.g.n. cells, while the simple cells in the same column projected directly to the higher order complex cells.

A similar orderly projection of the tactile receptive fields to the cortex was found [Mountcastle, 1957]. The cortex was organized into narrow vertical columns. Each neuron in a vertical column was activated by the same single submodality: touch, joint position or movement of the

hairs. In addition, the neurons of the same columns had similar receptive fields. These columns can be considered, analogously to columns in the visual cortex, to be functional units that process the information coming from lower levels and projecting it to the association areas where information from all senses is integrated.

The receptive field approach allows the understanding of the behaviour of individual cells, but fails to deal with the problem of the relationship of one cell with its neighbors in the coding of information by cells at the same level. For technical reasons, simultaneous monitoring from a group of cells is very difficult to achieve in practice. Nevertheless, several theories have been put forward.

Erickson [1968] suggested some principles of neural coding common to all sensory modalities. His view opposed the feature detector principle, suggested by Hubel and Weisel, because, according to him, there were simply not enough neurons to represent the entire range of possible stimuli. His hypothesis allowed the possibility of having more neural messages or functions than there were neurons. The information in the nervous system existed in the form of relative amounts of activity across many neurons. That is, the activity of a single neuron did not give by itself any unequivocal information concerning the stimulus quality and

intensity; the activity of each neuron contributed to a 'general pattern of activity'. The function of each neuron could not be given in terms of 'red' or 'vertical edge detector', for a 'red' neuron participated in the 'green' messages and a 'vertical detector' was involved in the representation of lines with other orientations. The activity of a cell was meaningful only in the context of the activity of parallel cells.

This hypothesis was shared by Mountcastle [1967], who proposed the 'frequency profile in a population' to be a neural code. This implied that the discharge frequency of a single unit to input stimulus intensity was extended across large numbers of neurons. The stimulus contour, position, and extent were mapped to neural coordinates as a 'frequency profile' of the neural units.

Leibovic [1982] analyzed the way information was transmitted from one processing level (source level) to a second processing level (target level). For his discussion, Leibovic considered a number of features of the nervous system organization. He assumed that a layered arrangement of cells represented a processing level, that convergence and divergence existed between layers, that the number of target cells was smaller that the number of source cells, and that target cells had overlapping on-center receptive

fields. The receptive fields were defined as the difference of two Gaussian functions (similar to Rodieck's model).

From his studies Leibovic concluded that broad receptive fields reproduced the stimulus pattern more closely than narrow receptive fields. The narrow receptive fields generated an oscillating response wherever the input presented irregularities. Apparently, much of the information was preserved and the global patterns were transmitted from level to level.

Lateral inhibition is a phenomenon commonly observed at each synaptic relay or nucleus of the afferent nervous system. Lateral inhibitory interconnections at any of the nuclei have been found to be of several types: presynaptic, postsynaptic, recurrent, etc. It is believed that lateral inhibition plays an important role in the processing of the information on its way to the higher centers.

A step towards understanding the function of lateral inhibition in the processing of sensory information was achieved by Hartline [1969] in his studies of the visual receptors (ommatidia) of horseshoe crabs. As a result of his experiments, Hartline determined the existence of lateral interactions among the receptors. He described the interactions as being inhibitory, recurrent, and decreasing with increasing distance separation of the interacting

ommatidia in the eye. He expressed these interactions in terms of a set of simultaneous equations now known as the Hartline-Ratliff equations.

Ratliff [1965] emphasized the similarity in the neural response of horseshoe crab with a perceptual phenomenon called the Mach bands. He observed that wherever the stimulus intensity curve of a stimulated surface had a concave or a convex flexion, then the neural activity was higher or lower, respectively, than its surroundings. The same type of deviations at the flexion regions had been detected at the perceptual level in humans. Therefore, Ratliff suggested that probably both the neural and perceptual processes could be linked in some degree (analogous to the power law situation presented above).

Naturally, there is a significant difference between human perception and the horseshoe crab nervous system, but, perhaps, as Ratliff inferred, some principles that are true for the simple interaction in the eye of the horseshoe crab can prove useful in elucidating the very complex features of neural physiology of highly developed animals.

CHAPTER 3: METHODS AND RESULTS

This chapter describes the mathematical model built, the computer programs created to implement the model and the results obtained. The information is organized in sections which tend to follow the course of the work during the research. Each section consists of

- presentation of the model version,
- implementation of the computer program,
- results obtained.

The reason for choosing a modular structure for this chapter is twofold. First, the model complexity increases in the successive sections as innovations are introduced. Most innovations are based on the results of former versions. Second, examples of several functional properties are easier to illustrate with simple models.

In this chapter the one dimensional model is presented first. The general assumptions upon which the model is based are introduced before moving on to the linear and nonlinear models. Time effects are considered in a section dealing with direction selectivity. A two dimensional model, which is an extension of the one dimensional model, is described last. In this last part of the chapter emphasis is made on model structure, system solution, and results; not on concepts and assumptions outlined in the introduction of the one dimensional model.

One Dimensional Model

A one dimensional model was built to

- develop the notation and system solution that could later be applied to the two dimensional model,
- study the effects of the assumptions made,
- observe the response to parameter changes.

The model structure consisted basically of equidistant units arranged in one dimension, without ring connections at the ends. Arrays were interconnected in series/succession giving rise to hierarchical organization. Interconnections within arrays and between arrays are described below.

In this section the linear and nonlinear models are presented. Solutions for both systems, as well as the corresponding program implementation are described. The results shown include receptive fields and step responses.

Implementation of the mathematical model

The mathematical model implemented was based on physiological evidence present in classical literature and in assumptions of three different natures: simplifying, generalizing and complementary. The following is a brief description of the physiological facts and assumptions included in the model.

Processing unit The model was made up of linear and continuous processing units or cells interconnected with each other to build the processing network. The linear transfer function used was a simplification of the power law suggested by Stevens [1971]. The power coefficient was assumed equal to one and the threshold ignored. A second, nonlinear model, with piecewise linear units was developed also. A unitary value was assigned to the power coefficient, but the threshold was not ignored. Since it was a static model, the units were assumed to have a nonadaptive response with respect to time.

Convergence layer This was an array of equidistant processing units (target units) characterized by the interconnections to the units in the preceding layer (source units). The response of the target units was determined by the addition of the weighted activity of the source units as shown in Figure 3.1. Mathematically,

$$M_i = \sum_m w_{\mu(i),m} I_m \qquad (3.1)$$

for m = 1 to N, with,

M_i: response of target unit i at convergence layer,

I_m: activity of unit m at source layer,

$w_{\mu(i),m}$: Gaussian weighting factor from source unit m to target unit i, whose position projection to the source layer is $\mu(i)$,

N: number of units in the source layer.

It was assumed that the spatial weighting function was decreasing with distance. This was expressed mathematically using a Gaussian spatial distribution function centered on the position of the target unit,

$$w_{\mu(i),m} = e^{-(\mu(i)-m)^2/2\sigma^2_{con}} \qquad (3.2)$$

with,

σ_{con}: standard deviation for convergence weighting distribution.

The standard deviation was a model parameter.

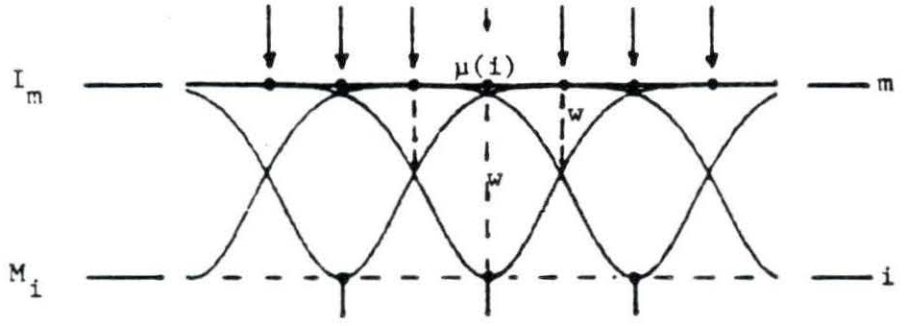

FIGURE 3.1. Convergence layer interconnections

The purpose of this layer was to represent the functional role of the relay cells that interconnect the subsequent relay stations [Kandel and Schwartz, 1981]. It may be observed in Figure 3.1 how several units at level m contributed to the response of one single unit at level i, there was convergence of inputs. Similarly, it may be observed how the overlapping weighting functions resulted in

divergence: one single source unit at level m contributed to the response of several target units at level i. The number of target units received their input from a larger number of source units yielding a pyramidal structure [Levine, 1985; Leibovic, 1982]. This assumption was an extension of what is observed in the retina, where the number of light receptors is larger than the number of ganglion cells.

Lateral inhibition layer This was an array of equidistant units characterized by interconnections within the array. In this layer, the output or response of each unit was fed back to its own input and to neighboring units as shown in Figure 3.2. The lateral interconnections were inhibitory [Kandel and Schwartz, 1981], and their influence decreased with distance. Mathematically, this was expressed in terms of the Hartline-Ratliff equations derived from the lateral interconnections in the eyes of horseshoe crabs [Hartline, 1969],

$$O_i = f(M_i, K_{(i,p)}, O_p) \qquad (3.3)$$

with,

O_i: output of unit i,

M_i: input of unit i,

$K_{(i,p)}$: lateral interconnection factor from output of unit p to input of unit i,

f: a linear or piecewise function of M_i, $K_{(i,p)}$,

and O_p.

A Gaussian spatial distribution function was used to assign values to the lateral interconnection coefficients [Ratliff, 1965],

$$K_{(i,p)} = e^{-(i-p)^2/2\sigma^2_{inh}} \qquad \text{for } p \neq i \qquad (3.4)$$

$$K_{(i,p)} = SI \qquad \text{for } p = i, \qquad (3.5)$$

SI: self-feedback coefficient,

σ_{inh}: standard deviation for lateral weighting distribution.

The standard deviation was a model parameter. An arbitrary value was assigned to the self-feedback coefficient, another model parameter.

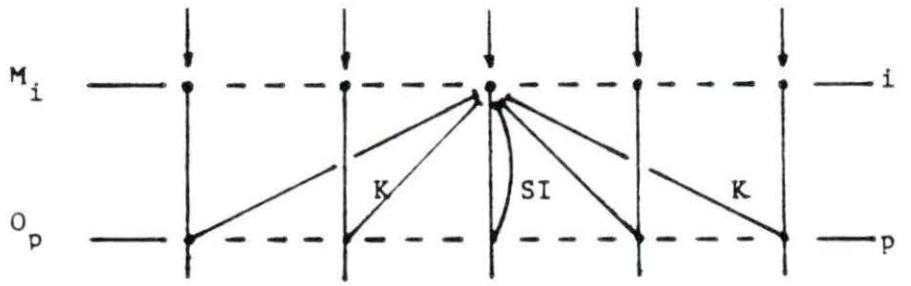

FIGURE 3.2. Lateral inhibition layer interconnections

The purpose of this layer was to represent the processing that takes place at the relay stations. Several types of lateral interaction have been reported at the relay stations (synaptic nuclei) [Kandel and Schwartz, 1981; Mountcastle, 1974]. The Hartline-Ratliff equations, which

describe recurrent inhibitory interactions, were used as an approximation to the many processes performed at the relay stations.

To account for the intrinsic firing rate displayed by many neurons at the higher levels of the nervous system [Somjen, 1972], it was assumed that,

$O_i > 0$ represented an increased discharge,

$O_i < 0$ represented inhibited discharge, whereas

$O_i = 0$ represented the intrinsic discharge of the neuron.

Processing level A processing level consisted of a convergence layer followed by a lateral inhibition layer. Processing levels were used as modular blocks whose organization and successive interconnection determined the model's structure. The processing characteristics of each level depended on the values assigned to the layer parameters. Figure 3.3 shows the arrangement of a processing level.

Thus, every processing level represented a set of relay neurons and the relay station where they terminated. For example, the first processing level represented the first order neurons and the first relay station, the second processing level represented the second order neurons and the second relay station, etc.

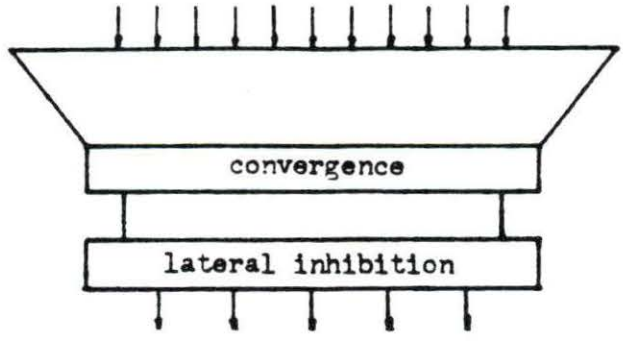

FIGURE 3.3. Structure of a processing level

Model structure The model consisted of processing levels interconnected in series. The information flowing out of a certain level P was simultaneously the input to level P+1. Level P+1 processed the information and relayed it to level P+2, and so on. This structure is typical of hierarchical systems.

A system with N hierarchical levels was built with N successive processing levels. Figure 3.4 shows a model with N=3 hierarchical levels. Information flows from the broad base towards the narrow tip of the pyramidal structure.

Linear Model

A linear model was the first tested. This initial version of the model allowed me to observe the response of a hierarchically organized system and weigh the significance of some of the model's parameters in the system's response.

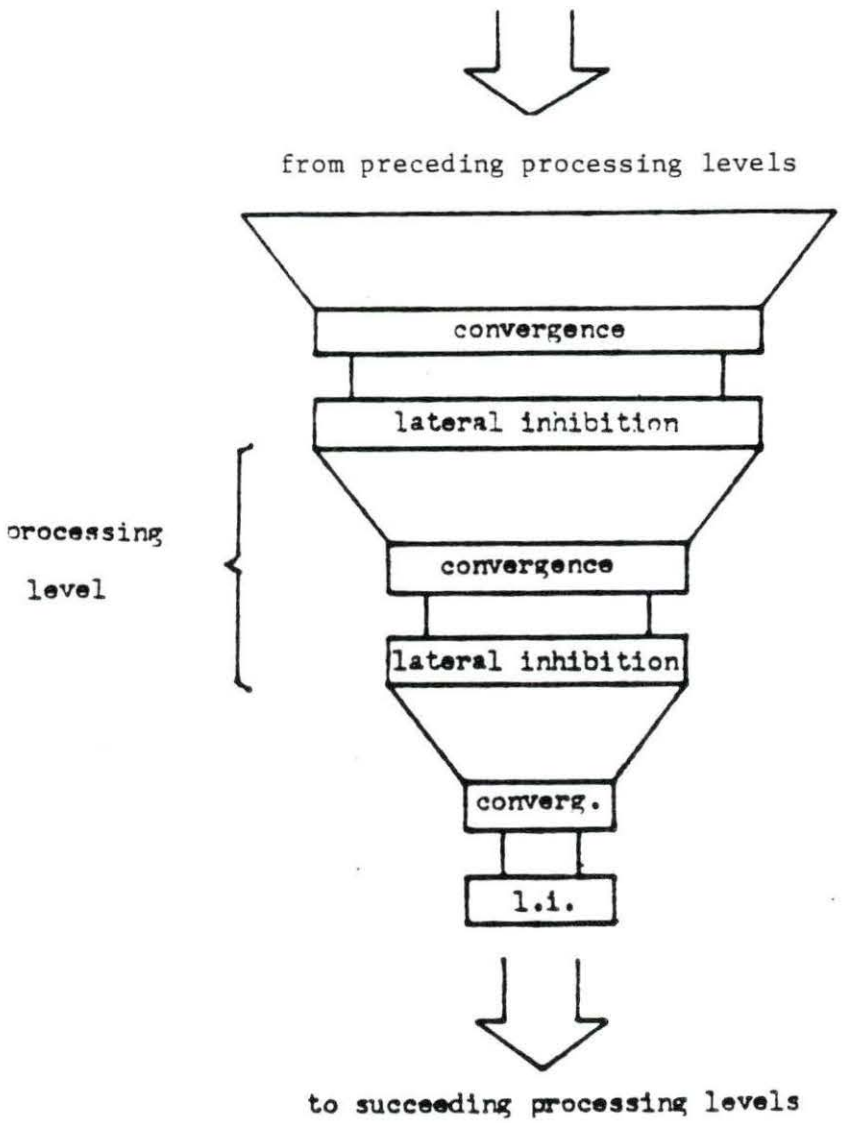

FIGURE 3.4. General model structure

Linearity was defined by assuming a linear response of all units in the model. This implied no change in the convergence equations, but resulted in the following equations for the lateral inhibition layer (see fig. 3.2),

$$O_i = M_i - \Sigma K_{(i,p)} O_p \qquad (3.6)$$

which in matrix notation was written as,

$$[O] = [M] - [K][O] \qquad (3.7)$$

with,

[O] : output vector,

[M] : input vector, and

[K] : lateral coefficients matrix.

Note that row i of [K] contained the inhibitory coefficients from all outputs to unit i. The system solution was obtained through,

$$([U]+[K])[O] = [M] \qquad (3.8)$$

$$[A][O] = [M] \qquad (3.9)$$

$$[O] = [Ainv][M] \qquad (3.10)$$

where,

[U] : identity matrix, and

[Ainv] : system transfer function.

Note that row i of [Ainv] contained the direct contribution coefficients from every unit in the array to the output of unit i. In matrix notation,

$$[Ainv] = [C_{i,j}] \quad \text{for } i=1,N \text{ and } j=1,N \qquad (3.11)$$

and,

$$[R_i] = [C_{i,j}] \quad \text{for } j=1,N. \qquad (3.12)$$

The solution of the system was accomplished by determining the inverse matrix [Ainv]. This matrix of size

N·N (where N was the number of elements in the lateral inhibition layer considered), was a sparse, diagonally-symmetric matrix. Edge effects were observed on the borders of the matrix but were negligible so long as N was much greater than σ_{inh}. Thus, the transfer function was interpreted as being representative of a convolution operation. The solution of the system was obtained either by applying equation 3.10 or through the convolution of the input with $[R_c]$, the central row of [Ainv],

$$[O] = [M]*[R_c]. \qquad (3.13)$$

Finding the inverse of a matrix when N was a large number was very demanding for the minicomputer PDP 11-23 used in terms of time and memory. Things got even worse for a two dimensional model with N·N units in the inhibitory array, for which [A] was a $N^2 \cdot N^2$ matrix.

One approach to solve this problem was windowing. It was noticed that the influence of lateral inhibition decreased with distance. Two units that were sufficiently apart could be considered as noninteracting. Therefore, a small matrix [A'] of size N'·N' was implemented. This matrix represented only the significant interactions with respect to a central unit. The inverse of [A'] was calculated yielding [A'inv]. $[R'_c]$, the middle row of [A'inv], was essentially equal to $[R_c]$, the middle row of

[Ainv]. The vector $[R'_c]$ was used as a window and the output solved as the convolution of the input with the window,

$$[O] = [M]*[R'_c]. \qquad (3.14)$$

Accurate results were obtained if N' was much greater than σ_{inh}.

A model with the following parameters was built and run:

Input: 81 elements

First level: 41 units

Second level: 21 units

$\sigma_{con} = 1$; $\sigma_{inh} = 2$; SI = 0.3.

Computer programs in Fortran language were written and run on a minicomputer. These programs solved the system using the windowing technique and the Gauss-Jordan method to compute inverse matrices. Receptive fields and step responses were determined for systems with one and two hierarchical levels. The role of self-feedback in the system response was also analyzed.

Receptive fields Receptive fields of central units at the first and second levels were calculated. The receptive fields were determined by shifting a point input along the input array and mapping the corresponding output of the observed unit. See Appendix A for program listing.

One processing level: Results Results obtained for various self-feedback coefficient values and a Gaussian distribution function at the inhibition layer are presented in the first column of Figure 3.5. The second column of the Figure 3.5 shows the results obtained for a different weighting distribution function at the lateral inhibition layer. The lateral coefficients in this second case were defined as

$$K_{i,p} = (1+\cos(2\pi(i-p)/L)/2 \quad \text{for } |i-p| \leq L/2 \qquad (3.15)$$

with,

$$K_{i,p} = 0 \text{ for } i=p \text{ and } |i-p| > L/2. \qquad (3.16)$$

The first row of Figure 3.5 compares the receptive fields in the absence of self-feedback (SI=0). Note that the receptive fields were significantly different in shape. Row 2 shows the receptive fields when self-inhibition was applied. In this case the self-feedback coefficient of all units in the array was greater than zero. The receptive fields were excitatory in the center and inhibitory in the surrounding area. Row 3 is an example of units with self-excitation. In this situation, the self-feedback coefficient was negative and larger than one in magnitude. No restrictions were imposed on the other units of the array, they could be either self-inhibitory or self-excitatory. It is observed that the receptive fields were inhibitory in the center and excitatory in the surrounding

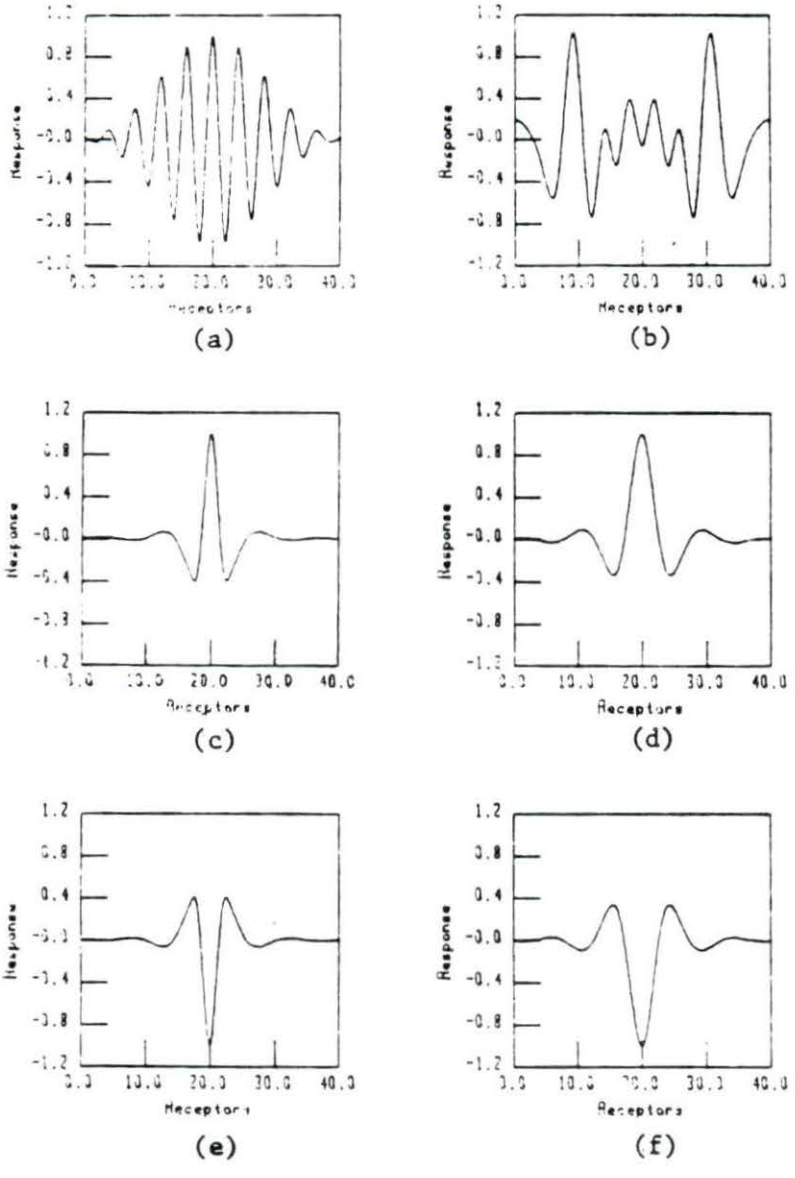

FIGURE 3.5. Effect of feedback on receptive fields [for the following distributions and SI: a) Gaussian, SI=0, b) cosine, SI=0, c) Gaussian, SI>0, d) cosine, SI>0, e) Gaussian, SI<0, f) cosine, SI<0]

area.

Results showed relatively little change in the shape of the receptive fields when self-feedback was introduced. Self inhibition (row 2) resulted in on-center receptive fields, whereas, self-excitation resulted in off-center receptive fields. The receptive fields were similar in shape to the receptive fields reported by Kuffler [1953] for the ganglion cells in the cat's retina. When no self-feedback was applied, the two distributions considered resulted in receptive fields with significantly different shape (row 1).

Two processing levels: Results The receptive field of a model with one processing level (Figure 3.6.a) was compared with the receptive field of a model with two processing levels (Figure 3.6.b).

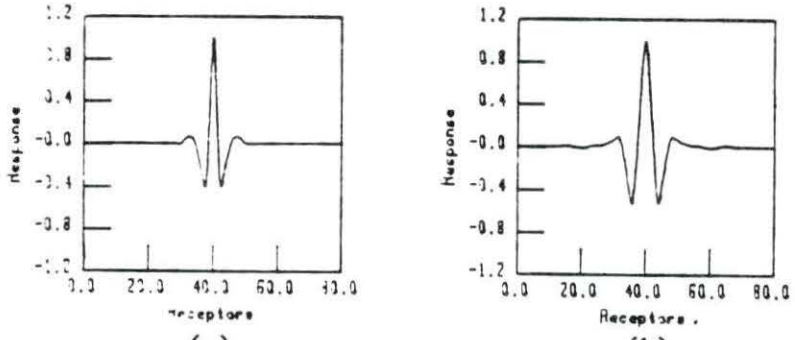

FIGURE 3.6. One dimensional linear receptive fields [for a) unit at level one, b) unit at level two]

It was observed that when the units in the first level had an on-center field, the resulting field for the second level repeated the on-center field shape. The receptive field of the second level covered a larger area than the receptive field of the first level and presented an increased inhibitory magnitude in the surrounding area. These results did not differ greatly from the obsevations made by Hammond [1973] at the lateral geniculate nucleus.

Response to a step input One and two level responses to a step input were studied. The step stimulation consisted of a unit input stimulation along input units 0 to 40 and zero stimulation for units 41 to 80. The output was observed at all 41 output units of the first level and all 21 units of the second level.

Results obtained are presented in Figure 3.7 where it is observed that both systems

- react at the site of the stimulus edge,
- present attenuated response at the region with uniform stimulation,
- present inhibitory response just before the edge.

It is also true that the two level system presents

- an increased response at the edge site,
- a decreased response at the region with uniform stimulation.

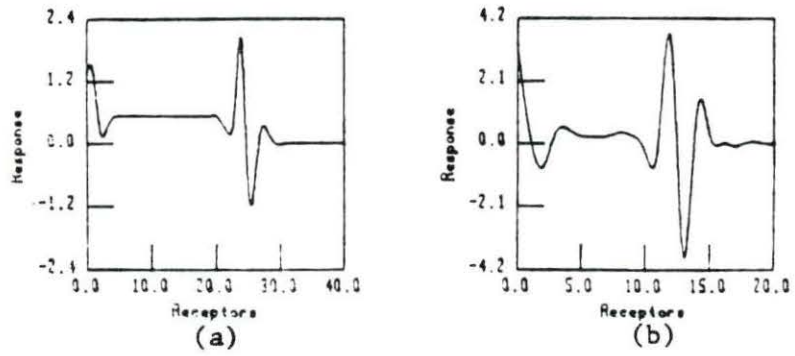

FIGURE 3.7. Linear response to step input [a) at level one, b) at level two]

Nonlinear model

A nonlinearity was introduced in the model at the lateral inhibition layer of the processing levels. The reason why this nonlinearity was included in the model was the fact that negative discharge rates cannot physically occur whenever the sum of inhibitory interactions was greater than the excitatory input. This fact was previously considered for the Hartline-Ratliff equations by imposing the condition that, if the total inhibition was larger than the excitation, then the corresponding response was set to zero [Ratliff, 1965]. A similar assumption was used in this model, but adapted to the notation used. The intrinsic firing rate of a unit had been defined as zero output and inhibited discharge as negative output. Therefore, a negative threshold value, equivalent to null discharge, was included in the model. Thus, whenever the response was

beyond the threshold value, the output was set equal to the threshold value. The following piecewise linear equations resulted for the lateral inhibition layer

$$O_i = (M_i - K_{(i,i)}O_i - \Sigma K_{(i,p)}O_m)_T \qquad (3.17)$$

with,

$$O_i = O_i \quad \text{if} \quad O_i \geq T \qquad (3.18)$$

$$O_i = T \quad \text{if} \quad O_i < T \qquad (3.19)$$

Matrix notation was used although the system could no longer be solved by calculating the inverse of matrix [A] since the system was no longer linear.

A modified version of the Gauss-Seidel iterative method was used to solve the system. The Gauss-Seidel method is generally used to solve systems of simultaneous linear equations. This method assigns arbitrary initial values to the unknowns and calculates the corresponding output values for the first iteration. Next, it starts the second iteration with the unknowns updated as the output values obtained during the first iteration. New output values are obtained and the process continues iterating until the output values of two consecutive iterations differ by less than a pre-established amount.

$$O_i^{r+1} = M_i - K_{(i,i)}O_i^r - \Sigma K_{(i,p)}O_p^r \qquad (3.20)$$

A computational step was added to the Gauss-Seidel method. At the end of each iteration, the resulting output

values were modified according to equations 3.18 and 3.19 above. See program implementation included in Appendix B.

As in the linear model, the shifting window technique was applied to obtain the solution of the system. Some innovations were introduced due to the iterative nature of the solution method. A matrix [A'] of reduced size N'·N' (similar to the matrix used for the linear model) was assigned the significant lateral coefficients with respect to a central unit. The system was solved iteratively for the central unit of the array. The matrix [A'] was then shifted one unit along the input array and the system solved for the new central unit. The process continued for all the units in the array. The effects of the nonlinearities on receptive fields and step responses were evaluated. The results are presented below.

Note that the model parameters remained the same as for the linear system. In running the programs it was observed that the number of iterations decreased for null threshold values and large self-inhibition coefficient values.

Receptive fields The receptive field of a system with one processing level was studied. The model was tested for several threshold values with the results shown in Figure 3.8. Figure 3.8.a shows the response of a system with unlimited inhibition. The receptive field coincided

with that of the linear model. Figure 3.8.b shows the response for a system with a threshold value T=-0.2. It was noticed that inhibition was limited and clamped at the value of T. The inhibited area had increased and disinhibition effects were decreased. Figure 3.8.c shows the response for T=0 (equal to the assumed spontaneous firing frequency rate of the units). Note the virtual absence of inhibition and disinhibition. The negative excursions of the response were demonstrated to be an artifact derived from the interpolating algorithm used to plot the receptive fields. Finally, the response of a linear receptive field whose negative portions of the response were ideally rectified is presented in Figure 3.8.d. It is observed that even though inhibition disappeared, disinhibition did not change. This result is functionally incorrect.

Disinhibition in recurrent networks is a consequence of the indirect interaction among two units across an intermediate unit. Increased activity of a unit as a response to stimulus increases the inhibitory influences on the neighboring intermediate unit. The activity of the intermediate unit decreases, and, as a consequence, so does its inhibitory interaction on the third unit, which is released of inhibitory constraints and increases its firing rate. Thus, disinhibition is mediated through the

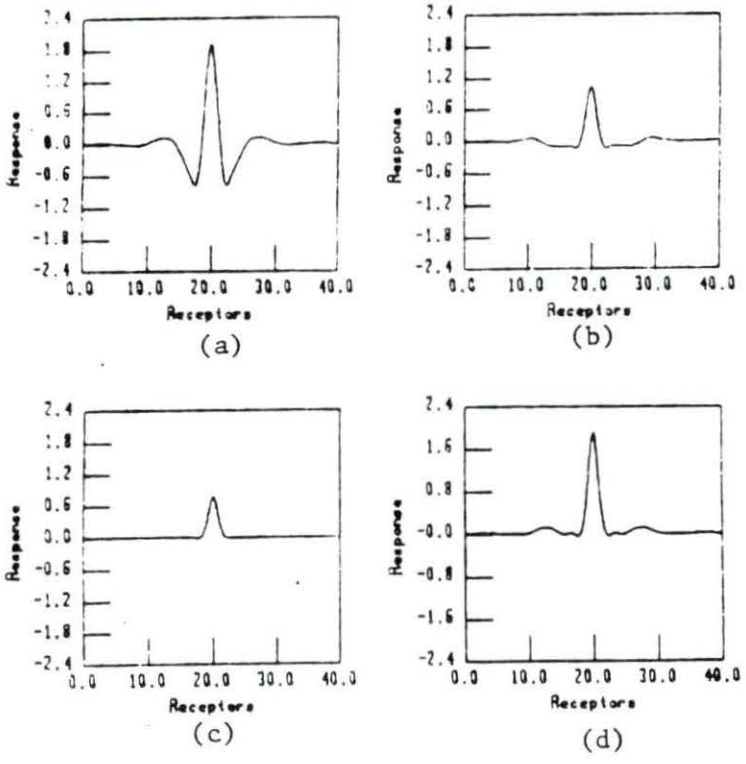

FIGURE 3.8. One dimensional nonlinear receptive fields [for a) unlimited inhibition, b) T=-0.2, c) T=0, d) rectified linear RF]

inhibition of an intermediate unit. Since in Figure 3.8.d the inhibition range of the intermediate unit was cancelled by setting the threshold value equal to the intrinsic discharge rate (T=0), then theoretically, no disinhibition should have taken place.

Response to step input Responses to a step input were studied for systems with one and two processing levels. The results are presented in Figure 3.9.

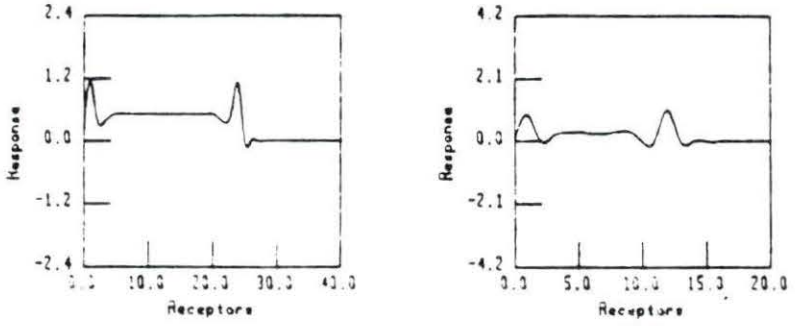

FIGURE 3.9. Nonlinear response to step input [at a) level one, b) level two]

It was observed that both systems

- reacted at the site of the stimulus edge,
- presented attenuated response at the region with uniform stimulation,
- lacked inhibition just before reaching the edge.

It was also noted that the system with two processing levels,

- presented an increased response at the edge site,
- presented a decreased response at the region with uniform stimulation.

The response in the region of uniform nonzero stimulation was the same for both the linear and nonlinear

models. This adaptive response was attributed to the increased firing rates of the units in the region of uniform stimulation. In that region, the gap available between maintained activity and the set threshold increased and resulted in a larger range for inhibition. As a consequence, the model tended to behave linearly.

Linear vs. nonlinear models

Linear and nonlinear models were tested and no important advantages of one model with respect to the other were observed. The most important difference was perhaps, the adaptive response of the nonlinear model. In this section a different type of test is described which contrasted the behaviour of the two systems.

The test consisted of the determination of the receptive field of a central unit located at the convergence layer of the third processing layer, as shown in Figure 3.10.

Results, shown in Figure 3.11.a and Figure 3.11.b, indicated that the peak response was similar for both the linear and the nonlinear models. When the standard deviation of convergence layers was increased, then the nonlinear response was considerably greater than the linear response, as is observed in Figure 3.11.c and Figure 3.11.d. The reason for the improved response was that for the linear

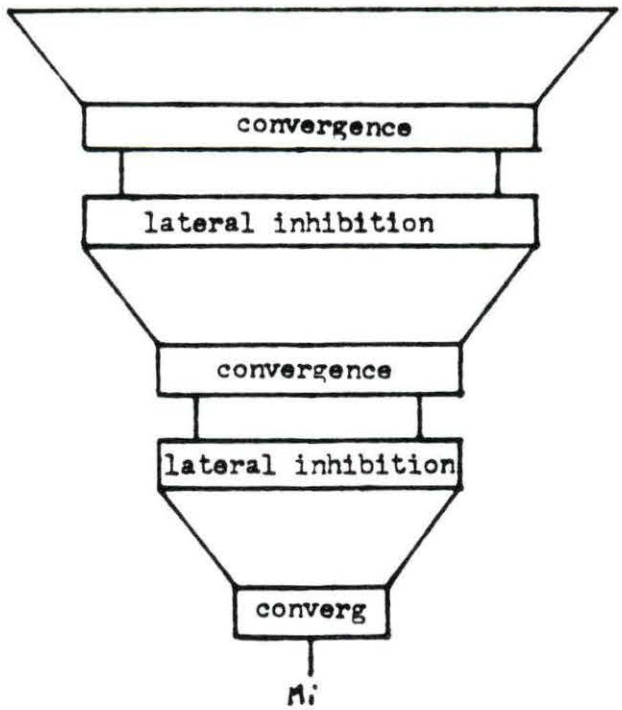

FIGURE 3.10. Linear vs nonlinear test receptor site

model the processing unit in level three averages out the excitatory and the inhibitory portions of the response of level two; increased values of the standard deviation for convergence improved the averaging efficiency. Whereas, for the nonlinear model, there was no averaging out since the inhibitory response of level two was limited by the threshold value set.

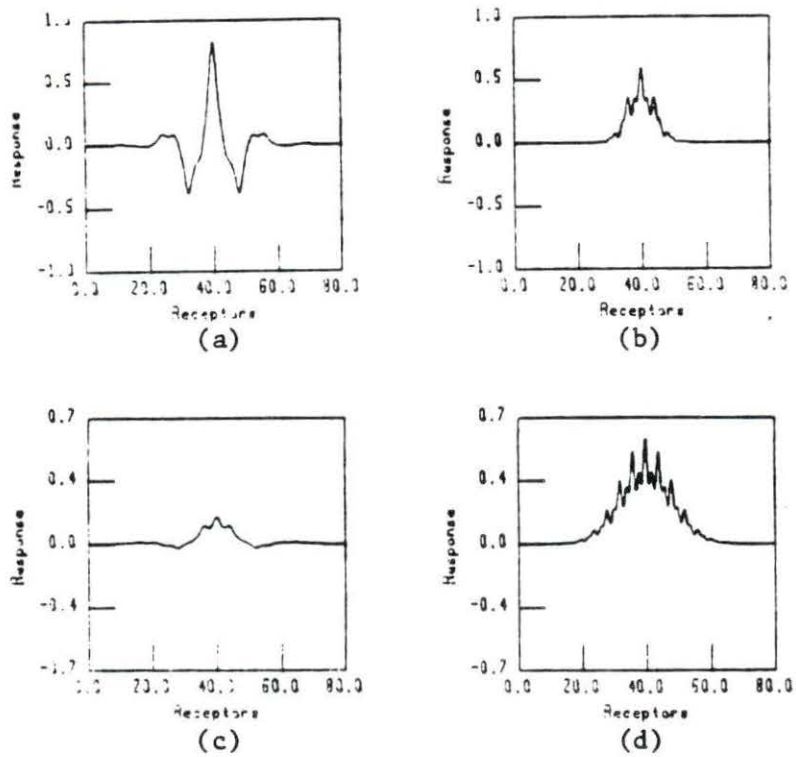

FIGURE 3.11. Comparison of responses [a) linear b) nonlinear c) linear, wide convergence d) nonlinear, wide convergence]

Directional selectivity

This section deals with the implementation of a processing unit sensitive to the direction of stimulus motion.

Directional selectivity was obtained by, a) generating an asymmetric receptive field for the central unit of the

convergence layer indicated in Figure 3.10 and, b) introducing time limitations in the unit's response.

Time is an important factor that had not been considered in the model. Neurons in the nervous system do not present an ideal static response, but have time limitations. For example, neurons may present a sustained response (X cells), a transient response (Y cells), or be sensitive to time between stimuli, to the speed of stimulus motion, etc.

Linear time analysis was applied to approximate the units' time responses. Under this analysis, the units were assumed to behave linearly with respect to time and to have a limited bandwidth, as do physical systems in general. To determine the output, the static response of a unit to a moving stimulus was calculated. Then, linear analysis was applied to the resultant output sequence to determine the time response.

Asymmetric receptive field A nonsymmetric receptive field was obtained by allowing self-inhibition in units 0 to 10 of level two, and self-excitation in units 11 to 20 of the same level. This resulted in on-center receptive fields for the first half of the units in the array and off-center units for the other half. The generated 'static' receptive field is shown in Figure 3.12.

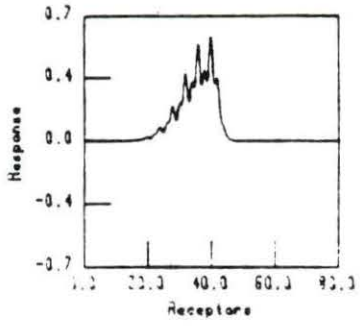

FIGURE 3.12. Static asymmetric receptive field

Figure 3.12 indicates that a stimulus applied on the left half of the receptive field produced an excitatory response of the observed unit, whereas the response was virtually inhibited on the other half of the receptive field. It is important to note that for a point stimulus moved from left to right, the static response sequence of the unit first rose slowly as a result of the low gradient of the receptive field and then fell sharply. For a stimulus moved from right to left, the response sequence showed an initial sharp increment due to the increased gradient in the shape of the receptive field and then decayed slowly as the stimulus moved out of the receptive field.

Dynamic considerations Cells in the lower portions of the afferent nervous system are known to have two classes of response with respect to time: sustained and transient

[Kuffler et al., 1984]. Sustained response has been expressed in terms of a differential equation in time [Amari, 1977]

$$\tau\frac{do(t)}{dt} + o(t) = i(t) \qquad (3.21)$$

whose Laplace frequency transform was a first-order lowpass filter

$$\frac{O(s)}{I(s)} = \frac{1}{1+s\tau} \qquad (3.22)$$

in the s domain.

The transient response was also expressed in terms of a differential equation

$$\tau_2^2\frac{d^2o(t)}{dt^2} + \tau_1\frac{do(t)}{dt} + o(t) = \tau_0\frac{di(t)}{dt} \qquad (3.23)$$

and the frequency response description of this equation corresponded to a bandpass filter,

$$\frac{O(s)}{I(s)} = \frac{s\tau_0}{1+s\tau_1+s^2\tau_2^2} \qquad (3.24)$$

where s was the Laplace operator.

A similar transient response was obtained with a negative feedback system consisting of a sustained response for the direct and feedback paths.

$$\frac{O(s)}{I(s)} = \frac{s+1/\tau_2}{s^2+s(1/\tau_1+1/\tau_2)+(k+1)/(\tau_1\tau_2)} \qquad (3.25)$$

This was true so long as $1/\tau_2$ was negligible with respect to the signal's relevant low frequencies, where

τ_1: time constant for the direct path,

τ_2: time constant for the feedback path, and

k: gain of the direct path.

Then equation 3.25 responded as a bandpass filter within the frequency range of interest.

The Laplace transform is valid for continuous signals that can be transformed into a continuous frequency domain. Since the type of data that resulted from running the model was discrete, then the Laplace transform was not applicable. The Z transform, which is a transform commonly used for the linear analysis of discrete data, was used to determine the time response of the unit.

Thus, the 'feedback' filter in the s domain was mapped to the z domain using the bilineal algorithm. The filter was applied to the static (time invariant) response sequence of the unit to compute the time response. The static response sequence was generated by moving a point stimulus along the input array of the model (conceptually equivalent to the mapping of the receptive field in one dimensional models). Next, the stimulus was moved in the opposite direction (mathematically equivalent to reversing the output order of the response) and the Z transform algorithm applied again.

Figure 3.13.a shows the static response sequence of the unit for a point stimulus moved from left to right (L/R). Figure 3.13.b shows the time response of the unit obtained after applying the Z transform. Figure 3.13.c shows the time response of the unit after rectification of the response beyond the threshold value. Rectification at this level was valid because there were no lateral interactions among the units of a convergence layer. Figures 3.13.d through 3.13.f show the response for a stimulus moved from right to left (R/L).

A close inspection of Figure 3.13.b and Figure 3.13.e indicates that at the velocity of motion simulated, the unit response approximated the gradient of the generated static sequence, in accordance with Marr's and Rodieck's models. The response depended on the direction of motion.

Velocity of motion was an important factor in the determination of the dynamic response. Velocity was controlled indirectly by changing the sampling frequency value in the Z transform coefficients. When the simulated velocity of motion increased, the time response of the unit approximated the integral of the static sequence. Analogously, very slow motion resulted in an attenuated, undifferentiated directional selectivity to stimulus direction of motion. In this case, the signal's relevant

frequencies were negligible with respect to $1/\tau_2$ and most of the energy of the sequence fell under the low frequency, flat portion of the band pass filter.

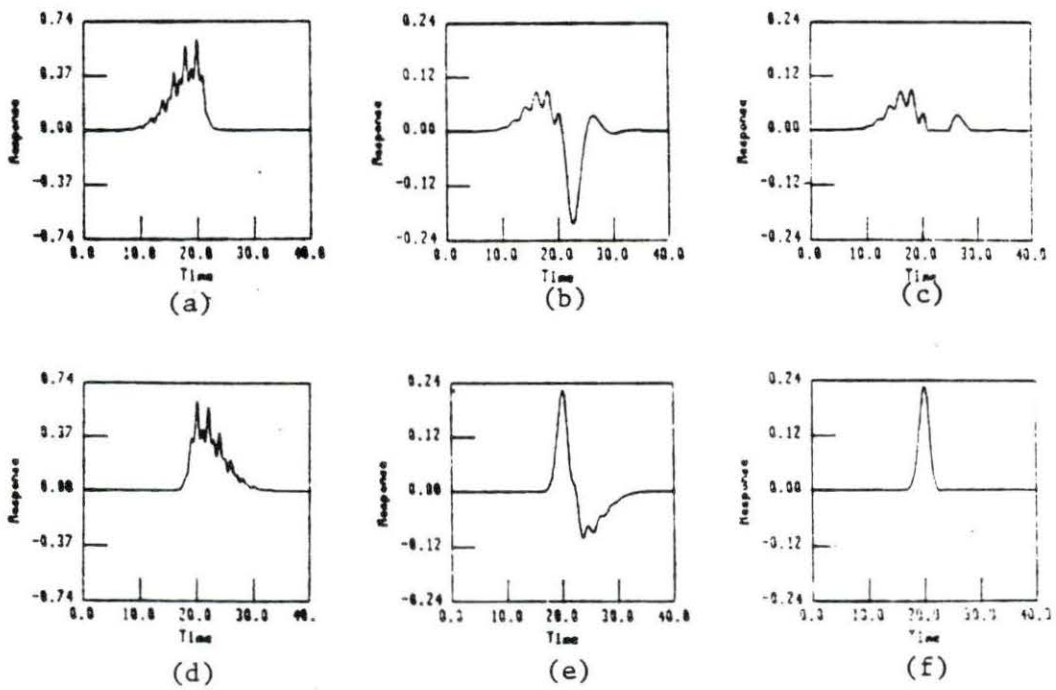

FIGURE 3.13. Direction selectivity response [for a) L/R static, b) L/R time, c) L/R final, d) R/L static, e) R/L time, f) R/L final]

Two Dimensional Model

This section describes the creation and implementation of a two dimensional model. This model was analogous to the one dimensional model, therefore, many of the properties and results obtained for the one dimensional model were extended

to this model. New properties, such as orientation sensitivity, which are inherent to models with more than one dimension are discussed in this section.

The inclusion of a second dimension substantially increased the required amount of processing time and memory necessary to solve the system. On the other hand, models that were limited in size, and were therefore less demanding in terms of computing resources, had their response contaminated by boundary effects. Boundary effects decreased for units located away from the borders (at a distance greater than the unit's receptive field). Thus, it was necessary to build a model with a large number of units and compute the response near the center of the input array. The increased computational demands were partially solved by running the programs in a mainframe computer NAS AS/9160.

This section follows the structure employed for the description of the one dimensional model. The first part describes the new layer equations resulting from the inclusion of the second dimension in the model. It also describes the model structure and processing paths. The following part shows the receptive fields of units in the first and second processing levels. Finally, the response of the system is analyzed.

Implementation of the mathematical model

As mentioned earlier, this model was an analog of the one dimensional model. Therefore, assumptions on which that model were based are also valid for this one. This part deals with the extension of such assumptions to two dimensions.

Convergence layer The convergence layer consisted of a two dimensional array of equidistant target units, where the output of one unit at (x,y) coordinates was given by,

$$M_{(x,y)} = \sum_{mn}\sum w_{\mu(x,y),(m,n)} I_{(m,n)} \qquad (3.26)$$

a two dimensional version of equation 3.1, with,

$M_{(x,y)}$: response of target unit located at coordinates (x,y) of the convergence layer,

$I_{(m,n)}$: activity of source unit located at coordinates (m,n) of the source layer,

$w_{\mu(x,y),(m,n)}$: weighting factor from source unit at coordinates (m,n) to target unit whose projection to the source layer is $\mu(x,y)$.

The weighting spatial function was a two dimensional version of a Gaussian distribution,

$$w_{\mu(x,y),(m,n)} = e^{-((\mu(x)-m)^2/2\sigma_x^2+(\mu(y)-n)^2/2\sigma_y^2)} \qquad (3.27)$$

with,

σ_x: x axis standard deviation, and

σ_y: y axis standard deviation.

In this case we noted the existence of two variables, the standard deviation for the x dimension and the standard deviation for the y dimension. The ratio between these two parameters was important in establishing the shape of the receptive fields, as will be shown later. If both parameters had equal value then radially symmetric receptive fields resulted. If the ratio was different from one, then elongated receptive fields resulted. Receptive fields with various orientations were obtained by using relative axis angle rotation transformations.

Lateral inhibition The lateral inhibition layer consisted of a two dimensional array of equidistant spaced units with lateral interconnections. In this case, it was necessary to express not only the connections from a unit to all other units in the same row, but to all other units in all rows. Thus the output of a unit with spatial coordinates (x,y) was expressed as,

$$O_{(x,y)} = f(M_{(x,y)}, K_{(x,y),(p,q)}, O_{(p,q)}) \qquad (3.28)$$

a two dimensional version of equation 3.2, with,

$O_{(x,y)}$: output of unit located at (x,y),

$M_{(x,y)}$: input of unit located at (x,y),

$K_{(x,y),(p,q)}$: lateral interconnections weighting factor from unit located at coordinates (p,q) to unit with coordinates (x,y),

f: a linear or piecewise linear function of $M_{(x,y)}$,

$K_{(x,y),(p,q)}$, and $O_{(p,q)}$.

The lateral inhibition factors were derived from a Gaussian spatial distribution,

$$K_{(x,y),(p,q)} = e^{-((x-p)^2+(y-q)^2)/2\sigma^2} \qquad (3.29)$$

for $(x,y) \neq (p,q)$,

and,

$$K_{(x,y),(p,q)} = SI \qquad \text{for } (x,y) = (p,q) \qquad (3.30)$$

with,

SI: self-feedback, and

σ: standard deviation for lateral distribution.

Model structure The modular characteristics of the one dimensional model were preserved in the two dimensional extension. This model was constructed by interconnecting the processing levels in series and in parallel. The order in which the modules were put together can be observed in Figure 3.14.

The methodology followed to put the model together was based on Hubel and Weisel's hierarchical organization theory and architectural features reported for the visual cortex of the cat [Hubel and Weisel, 1962; Hubel, 1963]. As mentioned in an earlier chapter, similar features are also found in the touch sense, as reported by Mountcastle [1974]. The model built had a common input processing level followed by a set of parallel paths that converged to the final common

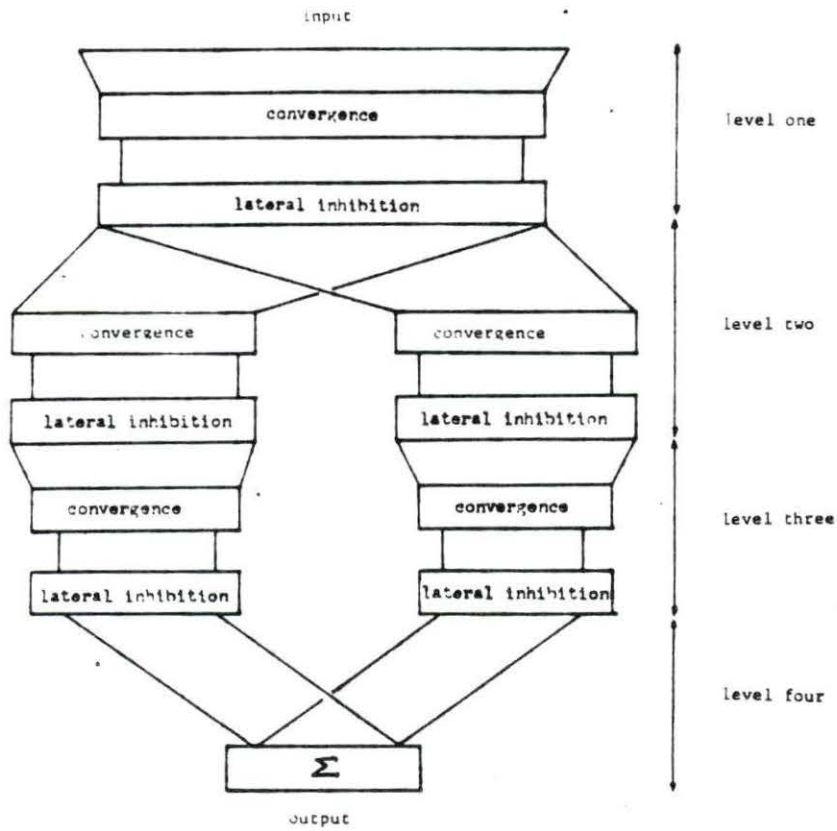

FIGURE 3.14. Two dimensional model structure [a common initial level is followed by parallel columns that converge to the output level]

output level. The first level units had radially symmetric receptive fields similar to those reported at retinal and L.G.N. levels in the cat visual pathway. The second level units had radially asymmetric receptive fields, generated by

assigning unequal values to σ_x and σ_y, that resembled the receptive fields of simple cells. There were four parallel pathways. The receptive fields of each path had been rotated 0°, 45°, 90°, and 135°, respectively.

The second level units projected through symmetric convergence to the third level. The purpose of the third level was to converge input from elongated, similarly oriented receptive fields that were overlapping in space. As a consequence, the third level units had a broad receptive field which was sensitive to the orientation of the stimulus and fairly insensitive to stimulus position. This type of response is representative of complex cells (time effects notwithstanding).

There was an orderly topographic projection from level to level. The input field was represented at each level of each path. Each of the parallel columns processed the same input information according to the particular orientation of the receptive fields of their units. Thus, if we thought of the units at the second level as simple cells and those at the third as complex cells, then each one of the parallel paths was functionally equivalent to the discrete columnar structures in the cortex described by Hubel and Weisel for the visual cortex of the cat [Hubel and Weisel, 1962; Hubel, 1963].

The fourth level represented what was assumed was the processing role of a 'higher than complex' cell in the cortex. These cells ran radially through the thickness of the cortex and extended laterally as they received information from complex cells in neighboring columns. Thus, they integrated the information from a region of the input field that had been processed by the various paths. In the model, this was achieved by adding point to point the outputs of each path.

Model specifications and solution implementation

The values assigned to the parameters of the two dimensional model are presented in Table 3.1.

TABLE 3.1. Two dimensional model specifications

level	N·N	σ_x	σ_y	σ	SI	type
input	81·81	-	-	-	-	-
first	41·41	1.3	1.3	2.6	1.0	linear
second	21·21	1.3	2.6	2.6	1.0	nonlinear
third	11·11	1.3	1.3	2.6	1.0	nonlinear
fourth	11·11	-	-	-	-	-

The implemented solution was basically a two dimensional version of the one dimensional model. Windowing techniques (discussed previously) were used in both the

convergence and the inhibitory processing layers. A matrix or window was assigned the corresponding Gaussian coefficients and shifted across the input of the convergence layers.

Computation of the coefficients of the lateral inhibition window was a more complex process. In general, to describe the lateral interactions of a two dimensional model, a four dimensional array was necessary. As can be observed in equations 3.29 and 3.30, the lateral coefficient K had four subscripts, which implied the generation of a four dimensional array [K] of size N·N·N·N to represent the interactions between units in all rows and columns. To calculate the solution of the system using matrix algebra, the four dimensional array had to be transformed into a two dimensional matrix. This was accomplished through a change in notation. The units in the input array [M] were no longer labelled according to their spatial coordinates, but were numbered following a line scan transformation into a vector $[M_L]$ of size $N^2 \cdot 1$. Similarly, the lateral interaction array [A] of four dimensions was line scan transformed into a matrix $[A_L]$ of size $N^2 \cdot N^2$, analogous to the one dimensional model, where row i contained the lateral interaction coefficients from all units to unit i. The inverse of $[A_L]$ was computed and the system solution determined as

$$[O_L] = [A_{Linv}][M_L], \qquad (3.31)$$

where $[R_{Li}]$ (the row i of $[A_{Linv}]$), contained the direct contribution coefficients from all inputs to unit i. An inverse line scan transformation on $[O_L]$ was necessary to reorder the output values on to the original two dimensional coordinates notation.

For linear windowing, a reduced matrix $[A'_L]$ of size $N'^2 \cdot N'^2$ was used. This matrix contained the significant lateral interaction coefficients with respect to a central unit. The inverse of the matrix was computed yielding $[A'_{Linv}]$. The middle row $[R'_{Lc}]$ contained only the relevant direct contribution coefficients from neighboring units to the central unit. An inverse scan transformation on $[R_{Lc}]$ reordered the direct contribution coefficients onto a two dimensional window [B], whose convolution with the two dimensional input array yielded the response [O],

$$[O] = [B]*[M]. \qquad (3.32)$$

Windowing for the nonlinear model required the use of the modified Gauss-Seidel subroutine for matrix inversion. This method required the iterative computation of the solution for every unit of the input array. In order to apply $[A'_L]$, the input array [M] of size $N \cdot N$ was line scan transformed into a vector $[M_L]$ of size $N^2 \cdot 1$. The matrix $[A'_L]$, of reduced size $N'^2 \cdot N'^2$, was shifted across the

vector $[M_L]$ and the system solved for the relative central units. These values were stored in $[O_L]$ of size $N^2 \cdot 1$. The process continued until the entire $[O_L]$ had been computed. Finally, the values in $[O_L]$ were relocated onto a two dimensional output array [O] of size $N \cdot N$ through an inverse line scan transformation. See Appendix C for program implementation.

Linear receptive fields

Linear receptive fields were determined for the first and second processing levels. Figure 3.15 shows the receptive field for a central unit at level one. A central excitatory zone with inhibitory surroundings, similar to the on-center receptive field reported by Kuffler [1953] was observed. Disinhibition effects were also noticeable.

Two possible receptive fields for the second level were computed. The first one, Figure 3.16, belonged to a unit with radial symmetry at the convergence layers of levels one and two. An expansion in the extent of the receptive field and an increase in the relative surrounding inhibition with respect to Figure 3.15 were observed. These results were similar to those obtained by Hammond [1973] in his comparison of retinal and L.G.N. responses.

The second case, Figure 3.17, shows a receptive field with radial asymmetry used in the model. Asymmetric

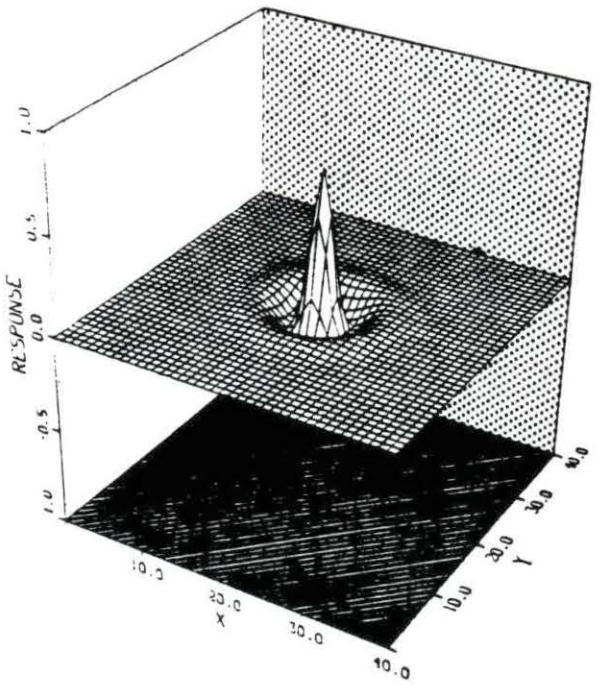

FIGURE 3.15. Receptive field at level one

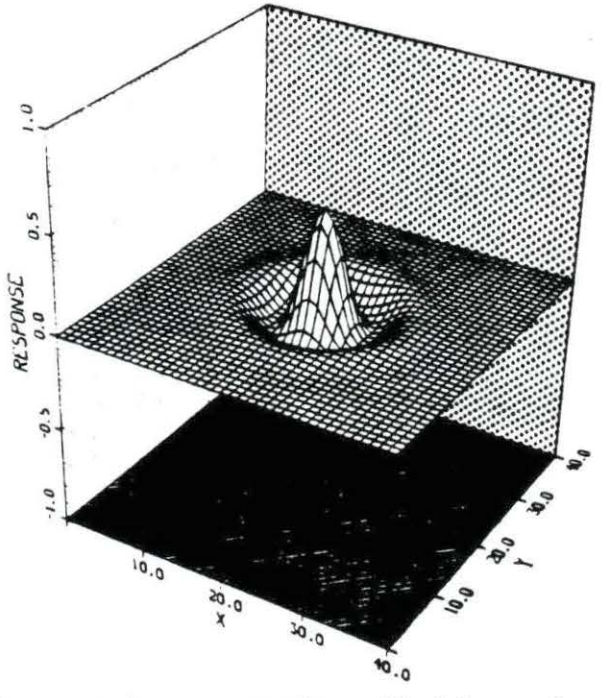

FIGURE 3.16. Symmetric receptive field at level two

convergence distributions were used at level two. As in simple cells, the receptive field had an elongated shape and well defined on and off regions separated by straight lines. In the following section it is shown how units with this type of receptive field responded to different features in the input stimulus.

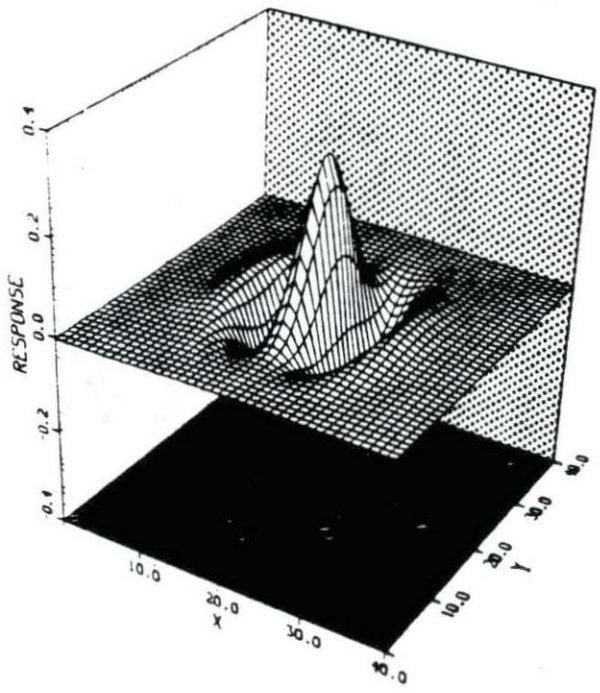

FIGURE 3.17. Asymmetric receptive field at level two

System response

The model's response to simple geometrical shapes was studied. The shapes used were basically squares and triangles of small size with respect to the input array, but

larger than the nonlinear receptive fields. To avoid the edge effects the shapes were located away from the input array boundaries.

The first test was done with a square shape. The square was 40 input units long per side. Figure 3.18 shows the activity in level two when a centered square was applied. The response was obtained from the parallel path with no rotation in the orientation of the receptive field's main axis. There was an increased activity of those units coincident with the shape edges that were parallel to the orientation of the receptive field's main axis.

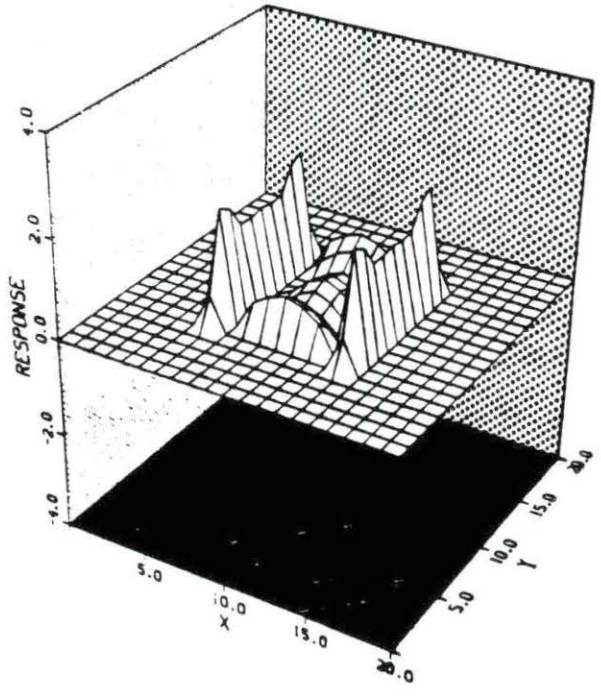

FIGURE 3.18. Level two response to centered square

Figure 3.19 shows the response at processing level three of the same parallel path. A lessened activity in the central portions of the detected edges and central areas was observed.

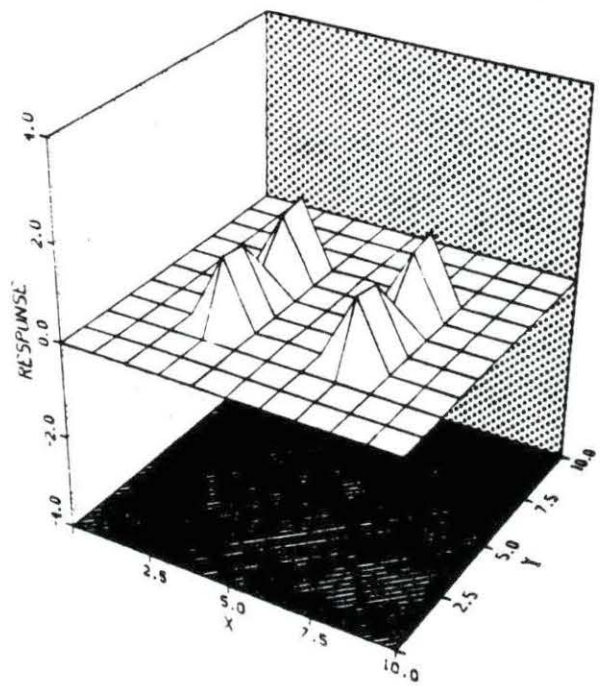

FIGURE 3.19. Level three response to centered square

Figure 3.20 shows the system's response to the centered square. Increased activity coincident with the square's corners is observed. A similar type of response for some complex cells was reported by Hubel and Weisel [1962].

Figure 3.21 shows the response when the square stimulus was shifted four units from the center in the x direction.

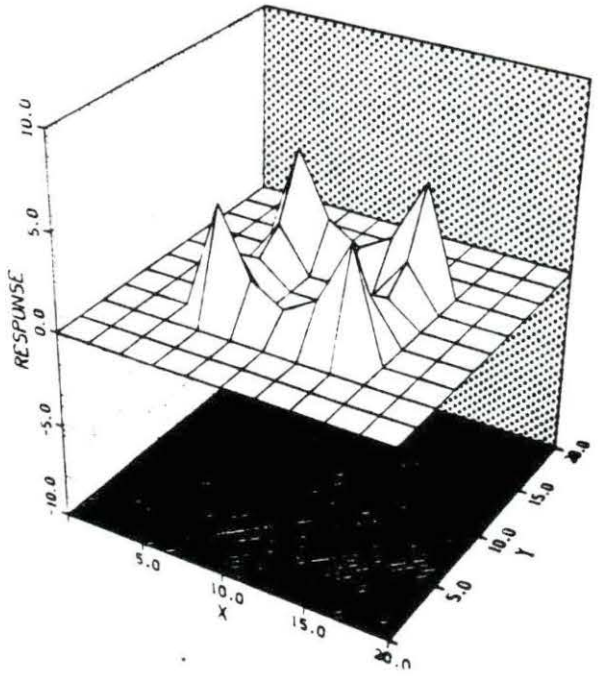

FIGURE 3.20. Response to a centered square

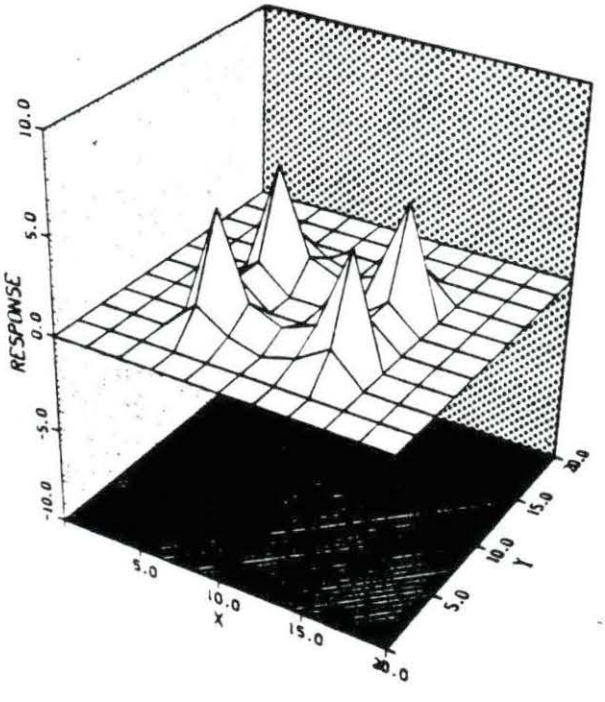

FIGURE 3.21. Response to a shifted square

Figure 3.22 shows the response when the square stimulus was shifted eight units from the center in the x direction.

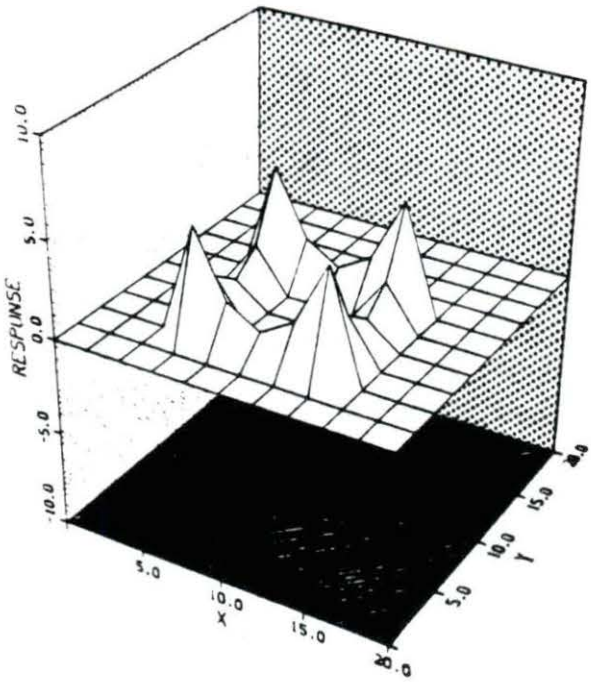

FIGURE 3.22. Response to a further shifted square

A comparison of the results obtained indicates that the presence of corners in the input stimulus was reflected by peaks in the output, whereas the response to uniform (or diffuse) stimului tended to be attenuated. The location of the peaks shifted across the output array when the input stimulus shifted position. Some activity was observed along the lines joining the corners and also in the vicinity of the corners. The amount and distribution of this secondary

activity varied with the stimulus position and no activity was detected outside the projection area of the square stimulus.

Figure 3.23 shows the response to a triangular stimulus. There was an increased activity at corner sites, with the right angle corner having a lower response than the acute angle corners. Perhaps, an explanation can be found at the first level. A unit located at the triangle's right angle corner receives inhibitory input from a relatively larger number of units than does a unit located at an acute corner. Thus, the response tended to rise higher at the acute corners.

It is interesting to note the relative lack of activity along the triangle's diagonal. One possibility was that secondary activity was a local phenomenon of corner activity and thus decreased as the stimulus' corners separated.

Consistent parameter values were chosen throughout the study so as to simplify the comparison and understanding of the processing functions of the model; although it must be noted that there were parameter combinations that enhanced edge detection better than others, other combinations resulted in an improved attenuation to uniform diffuse stimulation, and yet others were ideal to obtain a smooth response in the direction selectivity output. The important

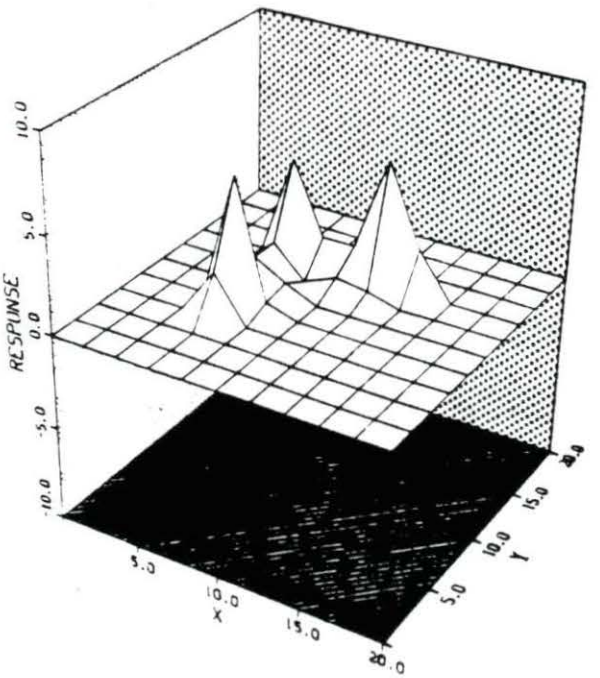

FIGURE 3.23. Response to a triangle

point was that while the structural principles of the model held, there were no significant qualitative changes in the observed response for the range of parameter values tested. An on-center field remained essentially an on-center field, although its extension, inhibition, disinhibition, etc. characteristics could change.

CHAPTER 4: DISCUSSION

The aim of this chapter is to relate the results obtained with the model to some physiological and functional aspects reported in classical literature. It must be remembered that the objective was to integrate in a mathematical model various neurological principles obtained from the classical literature. The model was tested from a functional point of view. Its receptive fields and step responses were determined.

The model implemented represented the lower levels of a standardized topographic modality. As may be inferred from previous chapters, it was a simple model oriented to the hierarchic processing of static information. The model was simple in that it was constrained to basic aspects of neural organization and principles, and was also simple in that it contained a relatively small number of units when compared to the millions of neurons in the nervous system.

Nevertheless, some of the results obtained were interesting in that they partially matched some functional properties reported in the classical literature. Naturally, this results were valid in the limited context in which they were generated and as such it is rather dangerous to extend the results to other systems or to generalize.

The results obtained with the one dimensional model, receptive fields and step responses, are discussed first. Then the possibility of achieving a transient response from a unit as a result of delayed feedback effects is presented. Finally, the two dimensional model and the processing of two dimensional shapes is examined.

On-center and off-center receptive fields were obtained with the one dimensional model. These receptive fields were similar in shape to the cross section of the symmetric receptive fields reported by Kuffler [1953] and Rodieck [1965] for the ganglion cells in the cat's retina. Rodieck [1965] and Marr and Ullman [1981] proposed ad hoc models that resorted to mathematical means with no physical significance to express the shape of the receptive fields. Kuffler suggested that an on-center receptive field originated from excitatory interconnections from the central field receptors to the ganglion cells and inhibitory interconnections from the surrounding field receptors to the ganglion cell. In the proposed model, the excitatory and inhibitory characteristics of the receptive field originated in the systematic interactions of the lateral inhibition layers. It was found that self-feedback may play an important role in the shape of receptive fields. In the first place, it tended to standardize the receptive field

shape; the particular distribution of the lateral interactions (as long as it was monotonically decreasing with distance) was no longer a primary factor in the determination of the receptive field shape. Secondly, the receptive fields were on-center or off-center depending on the nature of the self-feedback connection. Self-inhibition resulted in on-center fields, while self-excitation resulted in off-center fields. The type of receptive field may not necessarily be determined by the intrinsic characteristics of the unit, but may be determined by the particular interconnections of the neural network.

Another interesting feature observed was that on-center and off-center units could be intercalated together in the same array. This implied that if the type of response of the unit was controlled by external connections rather than by intrinsic processes, then it was not necessary to have two independent network arrays differentiated by the nature of the feedback connection. On-center and off-center units could be part of a common network and share the same physiological properties, but have their type of response governed by the nature of the self-feedback connection.

Self-inhibition interactions have been reported by Hartline [1969] in the light receptors of the limulus eye and included in the Hartline-Ratliff equations that describe

postsynaptic recurrent processes. The equations are valid even if we assume that the lateral interactions are mediated through inhibitory interneurons that receive their input from recurrent branches of the postsynaptic relay units. The interneurons terminate on neighboring cells for lateral interactions, and on their own relay unit (to close the loop) for self-inhibition. If the local interneuron was missing, the recurrent branch would terminate directly on its own body, yielding self-excitation. A similar arrangement has been found at the spinal cord level for motor neurons [Kandel and Schwartz, 1981].

An adaptive receptive field resulted from the nonlinearity introduced in the model. The nonlinearity limited the inhibitory response of the unit whenever the inhibitory interactions were larger than the excitatory stimulus (input stimulus and/or intrinsic discharge rate). The on-center and off-center overall characteristics were maintained, but the response of the inhibitory regions of the field depended on the threshold value.

It was noticed that when the threshold value was small and negative allowing a small range for inhibition (as would be the case for a low intrinsic discharge rate of the unit and little background stimulus input) the inhibitory response magnitude was limited to the threshold value and

extended in space over a larger area than in the linear receptive fields. In addition, the disinhibition effects were reduced. When the threshold value was large and negative, allowing a large range for inhibition (as could be the case for high intrinsic discharge of the unit and/or high background stimulus input) the inhibitory magnitude of the response increased and there was a contraction of the inhibitory area of the receptive field similar to that described by Kuffler [1953] for ganglion cells in the cat. There was an increase of the disinhibitory effects on the outer surround of the field. Kuffler did not report disinhibitory effects in his study.

A hierarchical model with two levels was then tested. In this model, the output units of the first level, each with on-center fields, represented the input source for the second level units. The resulting receptive field of a unit in the second level was found to repeat the on-center shape of the first level units with an increased inhibitory magnitude in the surrounding regions, similar to the effects described by Hammond [1973] in his contrast of ganglion with l.g.n. cells in the cat. Disinhibitory effects as reported by Hammond were also observed, but they were not originated by direct projections of on and off center fields as Hammond suggested, but were a result of the lateral interactions at the inhibitory level.

A comparison of the responses at the first and second levels showed that the global patterns of the stimulus were transmitted from one level to another as Leibovic [1982] concluded from his study. The response at stimulus discontinuities was enhanced, whereas the response was attenuated at the regions with uniform diffuse stimulation. It was at the regions with uniform stimulation that the linear and nonlinear models behaved similarly due to the adaptive properties of the nonlinear receptive fields.

At this point no apparent advantage was obvious (except computationally) between the linear and the nonlinear models. The convergence standard deviation had been assigned small values to obtain narrow receptive fields and thus reduce the edge effects in the central regions of the arrays at the higher levels. It was observed how the receptive fields tended to expand in space for units located in successively higher levels. This fact, combined with the pyramidal effect that reduces the number of units at successive levels [Levine, 1985], can greatly distort the expected response.

When broad convergence distributions were allowed in the linear model, the information tended to attenuate as it was transmitted from level to level. The broad convergence distributions averaged out the inhibitory and excitatory

portions of the transmitted information more efficiently than did narrow convergence distributions. Conversely, for the nonlinear model, the inhibition magnitude was limited and therefore the response did not tend to cancel.

It was concluded that the limited inhibition assumption allowed broad or narrow convergence distributions to be used with no risk of averaging out the information as it was transmitted along the successive levels.

Linear time considerations were included in the model and used to demonstrate that the transient response of a unit could be the result of a feedback mechanism. This concept was previously introduced by Hartline [1969]. He suggested that the self-feedback interconnections observed in the ommatidia could be responsible for the transient response of the receptors. He indicated that the delay involved in the transmission of the inhibitory signal through the feedback loop and back to the cell body allowed the output response to reach a peak. When the inhibitory signal arrived back at the cell body, the output decreased and the transient response was initiated.

The approach made was similar to Hartline's, except that the time feedback loop was incorporated in the model at the convergence layer and not at the inhibition layer, for computational reasons. It was found that a unit with

transient response combined with an asymmetric receptive field could be used to determine direction of motion of a stimulus pattern. It is interesting to note that the transient response was obtained by the interconnection of two units with sustained response. Again, as in the case discussed above for on-center and off-center units, it is not necessary to have functionally different units to achieve differential responses. The type of response may be governed by the type of interconnection among units of the same type.

Transient response as a result of feedback may not be restricted to extrinsic means. Feedback processes within a unit may also give rise to an transient response.

It was observed that the time invariant response of the two dimensional model extracted different features at the successive processing levels. General edge enhancement was observed in the first level. Sensitivity to edge orientation was observed in the second and third stages. Corners were detected in the fourth and last level of the model.

Corners were indicated in the response map by peaks surrounded by moderate or secondary activity. The peak response magnitude presented little change with respect to stimulus position, although it proved sensitive to the

stimulus shape corner angles. Acute angles elicited a higher response than did right angles. The secondary activity appeared to contain information about the relative position of the corner. When the stimulus position changed, the magnitude and distribution of the secondary activity changed accordingly. These results showed that information is transformed in passing from the lower processing levels to the higher processing levels.

CHAPTER 5: SUMMARY

The purpose of the study was to describe and discuss possible properties and interconnections of neurons in an idealized sensory network and to evaluate their functional significance in the processing of topographic information.

A mathematical model was built based on physiological evidence obtained from classical literature. Some of the physiological facts included in the model were hierarchical organization of the units, convergence and divergence of the information, and lateral inhibition.

A computer program was written and run to evaluate the model. Two model versions were tested, a one dimensional model and a two dimensional model. For both models, the static receptive fields were mapped and the static response to an input stimulus evaluated.

Results showed typical on-center or off-center receptive fields depending on the nature of the self-feedback connection in the units. Self-inhibition yielded on-center response while self-excitation yielded off-center fields.

Units with piecewise linear transfer functions produced adaptive receptive fields. The characteristics of the receptive fields depended on the inhibitory range allowed by the unit's activity and the threshold value of the piecewise

linear transfer function. The unit's activity was determined as the resultant of the inhibitory interactions and the excitatory input (external and spontaneous).

It was observed that the response of a unit with respect to time may be governed not only by the the intrinsic properties of the unit but also by the type of external interconnections. A transient response was obtained from a unit with sustained response with respect to time when a negative feedback path was introduced.

A unit with transient response combined with a nonsymmetric receptive field yielded a direction sensitive response to moving stimuli.

Finally, a static two dimensional model was tested. This model proved sensitive to spatial features of the input stimulus. Edges were enhanced in the first hierarchical level, sensitivity to edge orientation was observed in the second and third levels. Corners were detected in the last hierarchical level and were represented in the response map as peaks surrounded by moderate activity.

REFERENCES

Amari, S. 1977. A Mathematical Approach to Neural Systems. Pages 67-107 in J. Metzler (ed.). Systems Neuroscience. Academic Press, London.

Erickson, R. P. 1968. Stimulus Coding in Topographic and Non-Topographic Afferent Modalities. Psychological Review, 75:447-465.

Hammond, P. 1973. Contrasts in Spatial Organization of Receptive Fields at Geniculate and Retinal Levels: Centre, Surround, and Outer Surround. Journal of Physiology (London), 228:115-137.

Hartline, H. K. 1974. Visual Receptors and Retinal Interaction. Pages 643-659 in F. Ratliff (ed.). Studies on Excitation and Inhibition in the Retina. The Rockefeller University Press, New York.

Hubel, D. H. 1963. The Visual Cortex of the Brain. Scientific American, 178:477-504.

Hubel, D. H., and T. N. Weisel. 1962. Receptive Fields, Binocular interaction and Functional Architecture in the Cat's Visual Cortex. Journal of Physiology (London), 160:106-154.

Kandel, E. R., and J. H. Schwartz. 1981. Principles of Neuroscience. Elsevier North Holland, Inc., New York.

Kuffler, S. W. 1953. Discharge Patterns and Functional Organization of Mammalian Retina. Journal of Neurophysiology, 16:37-68.

Kuffler, S. W., Nichols, J. G., and A. R. Martin. 1984. From Neuron to Brain. Sinauer Associates, Sunderland, Mass.

Leibovic, K. N. 1982. Principles of Brain Function: Information Processing in Convergent and Divergent Pathways. Pages 91-99 in Pichler F. R. and R. Trapl (eds.). Progress in Cybernetics and Systems Research, Vol. VI. Hemisphere Publishing Corporation, New York.

Levine, M. D. 1985. Vision in Man and Machine. McGraw-Hill, New York.

Marr, D., and S. Ullman. 1981. Directional Sensitivity and its use in Early Visual Processing. Proceedings of the Royal Society (London), Ser. B, 211:151-180.

Mountcastle, V. B. 1957. Modality and Topographic Properties of Single Neuron's of Cat's Somatic Sensory Cortex. Journal of Neurophysiology, 20:408-434.

Mountcastle, V. B. 1967. The Problem of Sensing and the Coding of Neural Sensory Events. Pages 393-408 in Quarton, G. C., Melnechnuk, T., and F. O. Schmidt (eds.). The Neurosciences. Rockefeller University Press, New York.

Mountcastle, V. B. 1974. Medical Physiology, Vol. I. Mosby, St. Louis.

Ratliff, F. 1965. Mach Bands: Quantitative Studies on Neural Netwoks in the Retina. Holden-Day Inc., San Francisco.

Rodieck, R. W. 1965. Quantitative Analysis of Cat Retinal Ganglion Cell Response to Visual Stimuli. Vision Research, 5:583-601.

Somjen, G. 1972. Sensory Coding in the Mammalian Nervous System. Meredith Corporation, New York.

Stevens, C. F. 1966. Neurophysiology: A Primer. Wiley, New York.

Stevens, S. S. 1971. Sensory Power Functions and Neural Events. Pages 226-242 in Lowenstein, W. E. (ed.). Principles of Receptor Physiology. Springer-Verlag, New York.

APPENDIX A

```
C
C       PRO22.FOR
C
C
C       This program calculates the receptive field of a
C       one dimensional, one level, linear system. It uses
C       windowing at both the convergence and LI layers.
C       The receptive field is obtained by shifting a unitary
C       stimulus across the input. The corresponding output
C       is evaluated only for the central receptor.
C
C       Subroutines used:
C
C       MINV: Computes the inverse matrix using Gauss'
C       algorithm
C
C       HGRAPH: Graphics subroutine
C
C       Description of parameters:
C
C       A: Matrix containing lateral inhibition coefficients
C       XI: Input vector
C       XO: Storage of the central receptor output for each
C           position of the input stimulus
C       XH1: Convergence layer output
C       C1: LI layer output
C       B: LI window
C       GAUSCO: Covergence window
C       WOVEC1: Auxiliary vector for inverse matric computations
C       WOVEC2: Auxiliary vector for inverse matrix computations
C       XN:Graphics auxiliary vector
C
C-----------------------------------------------------------------
C
        DIMENSION A(11,11),DIAGA(11),XI(81),XO(81),XN(81)
        DIMENSION B(11),GAUSCO(11)
        DIMENSION XH1(41),C1(41)
        DIMENSION WOVEC1(11),WOVEC2(11)
C
C
        TYPE *,' INPUT standard deviation for convergence'
        ACCEPT *,SIGCON
        TYPE *,' INPUT standard deviation for lateral inhibition'
        ACCEPT *,SIGMA
        TYPE *,' INPUT self feedback coefficient'
        ACCEPT *,SI
C
C-----------------------------------------------------------------
C
C       ** Calculate inhibitory coefficients **
```

```
C
      DO 10 I = 1,11
        DO 10 J = 1,11
          A(I,J) = EXP(-((I-J)/SIGMA)**2/2)
10    CONTINUE
C
C     * Include self feedback effect *
      DO 20 I = 1,11
        A(I,I)= 1+SI
20    CONTINUE
C
C-----------------------------------------------------------------
C
C     * Calculate matrix inverse and load convergence window *
C     * MINV computes inverse and replaces values on A *
C
      D=0
      DO 25 I=1,11
        WOVEC1(I) = 0
        WOVEC2(I) = 0
25    CONTINUE
C
      CALL MINV(A,11,D,WOVEC1,WOVEC2)
C
C     * If value of matrix determinant is 0 then stop *
      IF(D.EQ.0) STOP
C
      DO 30 I=1,11
        B(I) = A(6,I)
30    CONTINUE
C
C-----------------------------------------------------------------
C
C     ** Calculate convergence array **
C
      DO 40 I = 1,11
        GAUSCO(I) = EXP(-((I-6)/SIGCON)**2/2)
40    CONTINUE
C
C-----------------------------------------------------------------
C
      DO 42 I = 1,81
        XI(I) = 0.
        XO(I) = 0.
42    CONTINUE
C
C-----------------------------------------------------------------
c
C     ** Calculate receptive fields **
C
```

```
      DO 1000 KPRIN = 1,81
C
C-----------------------------------------------------------------
C
C     **Clear arrays **
C
      DO 45 I = 1,41
        XH1(I) = 0.
        C1(I)  = 0.
45     CONTINUE
C
C
C-----------------------------------------------------------------
C
C       ** Generate input **
C       * Incremental shift of input stimulus *
C
        IF (KPRIN.GT.1) XI(KPRIN-1) = 0.
        XI(KPRIN) = 1.
C
C-----------------------------------------------------------------
C
C       * First level *
C
C       * Convergence layer *
C
         ICOUNT = 0.
C         * window shifts two units at time across source XI *
         DO 50 I1 = 1,81,2
C           * target XH1 increments one unit at a time *
         ICOUNT = ICOUNT + 1
C
C           * multiply source X1 and widow gausco point to point
C             and then add for target XH1 response *
         DO 50 I2 = 1,11
           I3 = I1+6-I2
*            * avoid operating outside source X! limits *
             IF((I3.LE.0).OR.(I3.GE.82)) GOTO 50
C
             XH1(ICOUNT) = XH1(ICOUNT)+GAUSCO(I2)*XI(I3)
C
50       CONTINUE
C
C       ** Lateral inhibition layer **
C
C       * incremental pointer I1 points at LI unit for which the
C         response is being computed, for RF only the response at
C         central unit (21) is calculated *
        DO 60 I1 = 21,21
C
```

```
C          * calculate convolution of input XH1 with window B *
           DO 62 I2 = 1,11
             I3 = I1+6-I2
C            * avoid computations outside input array *
             IF((I3.LE.0).OR.(I3.GE.42)) GOTO 62
C
             C1(I1) = C1(I1)+B(I2)*XH1(I3)
62         CONTINUE
60       CONTINUE
C
C-------------------------------------------------------------
C
C        * Store output of central unit in XO and repeat for
C          next input position *
         XO(KPRIN) = C1(21)
C
1000     CONTINUE
C
C-------------------------------------------------------------------
C
C        ** graphic output **
         DO 130 I  = 1,81
           XN(I) = I-1
130      CONTINUE
C
         DO 140 I = 1,81
           IF(XO(I).GT.XMAX) XMAX=XO(I)
140      CONTINUE
C
         DO 150 I=1,81
           XO(I)=XO(I)/XMAX
150      CONTINUE
C
         CALL INIPLT(7,4.,4.)
         CALL SCALE(0.,80.,-1.2,1.2)
         CALL AXIS(20.,0.4,'Receptors',9,1,1,'Response',8,1,1)
         CALL SMOOTH(XN,XO,81,0,0,1,0,0)
         CALL ENDPLT
C
END
```

APPENDIX B

```
C
C       PROGRAM: PRO53.FOR
C
C
C       This program calculates the response of a one dimensional,
C       two level, piecewise linear system to an arbitrary input.
C       It uses windowing technique at both the convergence layer
C       and L.I. layer. A linear response can be obtained if the
C       chosen threshold value THRES is large and negative.
C
C       Subroutines used:
C
C       MIO: Solves the nonlinear system by iterations.
C
C       HGRAPH: Graphics subroutine.
C
C       Description of parameters:
C
C       A: Matrix containing lateral inhibition coefficients
C       DIAGA: Vector with A's diagonal elements for iterative
C       solution
C       XI: Input vector
C       XHI: Convergence layer output at level 1
C       C1: Lateral inhibition layer output at level 1
C       XH2: Convergence layer output at level 2
C       C2: Lateral inhibition layer output at level 2
C       GAUSCO: Convergence window
C       XHAUX: LI auxiliary vector loaded with segment of input
C       vector on which LI windowing is computed
C       CAUX: LI auxiliary vector with output from LI windowing
C       XN: Graphics auxiliary vector
C
C-----------------------------------------------------------------
C
        DIMENSION A(11,11),DIAGA(11),XI(81),XN(21)
        DIMENSION GAUSCO(11)
        DIMENSION XH1(41),C1(41),XH2(21),C2(21)
        DIMENSION XHAUX(11),CAUX(11)
C
C
        TYPE *,' INPUT Standard deviation for convergence'
        ACCEPT *,SIGCON
        TYPE *,' INPUT Standard deviation for lateral inhibition'
        ACCEPT *,SIGMA
        TYPE *,' INPUT Self feedback coefficient'
        ACCEPT *,SI
        TYPE *,' INPUT Threshold value'
        ACCEPT *,THRES
        TYPE *,' INPUT Stimulus limits'
        TYPE *,' LOW LIMIT, UPPER LIMIT'
```

```
        ACCEPT *,KLOW,KHIGH
C
C-------------------------------------------------------------------
C
C       ** Load inhibitory weighing coefficients **
C
        DO 10 I = 1,11
          DO 10 J = 1,11
            A(I,J) = EXP(-((I-J)/SIGMA)**2/2)
10      CONTINUE
C
C       * Prepare matrix for iterative subroutine *
        DO 20 I = 1,11
C         * Include self feedback coefficient *
          DIAGA(I) = 1.+SI
       A(I,I)=0.
20      CONTINUE
C
C-------------------------------------------------------------
C
C       ** Load convergence weighing coefficients on window **
C
        DO 40 I = 1,11
          GAUSCO(I) = EXP(-((I-6)/SIGCON)**2/2)
40      CONTINUE
C
C-------------------------------------------------------------
C
        DO 42 I = 1,81
          XI(I) = 0.
42      CONTINUE
C
C-------------------------------------------------------------
c
C       ** Input stimulus generation **
C
        DO 1000 KPRIN = KLOW,KHIGH
          XI(KPRIN)=1.
1000    CONTINUE
C
C-------------------------------------------------------------
C
C       **Clear arrays **
C
          DO 45 I = 1,41
            XH1(I) = 0.
            C1(I)  = 0.
45        CONTINUE
C
          DO 46 I= 1,21
```

```
            XH2(I) = 0.
            C2(I)  = 0.
46        CONTINUE
C
C-------------------------------------------------------
C
C        ** First level **
C
C       * Convergence layer *
C
C
          ICOUNT = 0.
C       * Window shifts two units at a time across source XI *
          DO 50 I1 = 1,81,2
C           * Target XH1 increments one unit at a time *
            ICOUNT = ICOUNT + 1
C
C           * Multiply source XI and window GAUSCO one to one
C       and then add for target XH1 response *
            DO 50 I2 = 1,11
              I3 = I1+6-I2
C             * Avoid operating outside source XI limits *
              IF((I3.LE.0).OR.(I3.GE.82)) GOTO 50
C
                XH1(ICOUNT) = XH1(ICOUNT)+GAUSCO(I2)*XI(I3)
C
50          CONTINUE
C
C       ** Lateral inhibition layer **
C
C       * Incremental pointer I1 points at LI unit for which
C         the response is computed and indirectly
C         determines segment of input XH1 to be loaded
C         onto XHAUX *
        DO 60 I1 = 1,41
          KOUNT=0
C
C         * Clear XHAUX auxiliary array *
          DO 61 K=1,11
            XHAUX(K)=0.
61        CONTINUE
C
C         * Load the relative segment of input XH1 on to
C           XHAUX auxiliary array *
          DO 62 I2 = 1,11
            I3 = I1+6-I2
            KOUNT=KOUNT+1
            IF((I3.LE.0).OR.(I3.GE.42)) GOTO 62
C
              XHAUX(KOUNT)=XH1(I3)
```

```
62          CONTINUE
C
C           * Calculate system solution *
            CALL MIO(A,XHAUX,CAUX,DIAGA,11,THRES)
C           * Save only central receptor response *
            C1(I1)=CAUX(6)
C
60        CONTINUE
C
C
C-------------------------------------------------------------
C
C         ** second level **
C
C         * Convergence layer *
C
C
          ICOUNT = 0
C         * Window shifts two units at a time across source C1 *
          DO 80 I1 = 1,41,2
C           * Target XH2 increments one unit at a time *
            ICOUNT = ICOUNT+1
C
C         * Multiply source C1 and window GAUSCO one to one
C            and then add for target XH2 response *
          DO 80 I2 = 1,11
            I3 = I1+6-I2
C           * Avoid operating outside source C1 limits *
            IF((I3.LE.0).OR.(I3.GE.42)) GOTO 80
C
              XH2(ICOUNT) = XH2(ICOUNT)+GAUSCO(I2)*C1(I3)
C
80        CONTINUE
C
C         ** Lateral inhibition layer *
C
C         * Incremental pointer I1 points at LI unit for which
C           the response is computed and inderectly
C           determines the segment of input Xh1 to be
C           loaded onto XHAUX *
          DO 90 I1 = 1,21
            KOUNT=0
C
C           * Clear XHAUX auxiliary array *
            DO 91 K=1,11
              XHAUX(K)=0.
91          CONTINUE
C
C           * Load relative segment of input XH2 onto
C             XHAUX auxiliary array *
```

```
            DO 92 I2 = 1,11
              I3 = I1+6-I2
              KOUNT=KOUNT+1
              IF((I3.LE.0).OR.(I3.GE.22)) GOTO 92
C
                XHAUX(KOUNT)=XH2(I3)
92          CONTINUE
C
C           * Calculate system solution *
            CALL MIO(A,XHAUX,CAUX,DIAGA,11,THRES)
C           * Save only central receptor response *
            C2(I1)=CAUX(6)
90        CONTINUE
C
C-------------------------------------------------------------
C
C         ** Graphic output **
C
          DO 130 I  = 1, 21
            XN(I) = I-1
130       CONTINUE
C
C         * Normalize output *
          DO 140 I = 1,21
          IF(C2(I).GT.XMAX) XMAX=C2(I)
140       CONTINUE
C
          XMAX=XMAX*1.2
C
          CALL INIPLT(7,4.,4.)
          CALL SCALE(0.,20.,-XMAX,XMAX)
          CALL AXIS(5.,XMAX/2,'Receptors',9,1,1,'Response',8,1,1)
          CALL SMOOTH(XN,C2,21,0,0,1,0,0)
          CALL ENDPLT
C
          TYPE *,'SIGMA=',SIGMA,'SIGCON=',SIGCON
          TYPE *,'SI=',SI,'THRES=',THRES
          END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE MIO(C,XIN,XOUT,DIAG,N,T)
C
C
      DIMENSION C(N,N),XIN(N),XOUT(N),DIAG(N)
      ITER = 1
      EPS  = 1E-5
C
C
CC      DO 5 I = 1,N
CC        XOUT(I) = 0.
CC5     CONTINUE
C
99    BIG = 0.0
      DO 100 I = 1,N
        SUM = 0.
        DO 10 J = 1,N
          SUM = SUM+C(I,J)*XOUT(J)
10      CONTINUE
C
      TEMP = (XIN(I)-SUM)/DIAG(I)
C
      IF(TEMP.LT.T) TEMP = T
C
      RESID = ABS(TEMP-XOUT(I))
      IF(RESID.GT.BIG) BIG = RESID
      XOUT(I) = TEMP
100   CONTINUE
C
      IF(BIG.LT.EPS) RETURN
      IF(ITER.GT.150) GOTO 200
C
      ITER = ITER+1
      GOTO 99
C
200   TYPE *,'MORE THAN 150 ITERATIONS'
      RETURN
      END
```

APPENDIX C

```
C
C
C     PROGRAM: R2DNL
C
C     This program calculates the response of a two dimensional
C     linear model to an arbitrary input. Linear processing is
C     assumed at the first level, while a piecewise solution is
C     applied at other levels. Windowing techniques are
C     used for both the convergence and LI layers. A linear
C     response can be obtained if the chosen threshold value
C     THRES is large and negative.
C     The final output values are stored in file R2DNL.DAT and
C     can be plotted using program GRAR2D.
C
C     Subroutines used:
C
C     CLR: Clears array
C
C     CLRIO: Clears input and output arrays to a level
C
C     CONV: Calculates the response of the convergence layer
C
C     CONVAN: Adds point to point the output from each
C     individual parallel path
C
C     DIACLR: Prepares matrix A for iterative subroutine.
C     Loads main diagonal elements of matrix A on to DIAGA
C     and makes matrix A diagonal elements equal to zero
C
C     GENIND: Generates the LI coefficients
C
C     GENINP: Generates system input
C
C     INH: Calculates the piecewise linear response of a
C     lateral inhibition layer without using windowing
C     techniques
C
C     INHL: Calculates the linear response of a LI layer
C     using windowing techniques
C
C     INHN: Calculates the piecewise linear response of
C     a LI layer
C
C     LOADA: Loads the LI coefficients on matrix A,
C     includes the self inhibition effects
C
C     LOADB: Loads the elements in the middle row of
C     matrix A on to linear LI window
C
C     LOGAUS: Loads convergence window
```

```
C
C     MIO: Calculates the solution of a system of piecewice
C     linear equations using an iterative method
C
C     MINV: Calculation of matrix inverse using Gauss method
C
C     WRIFIL: Writes output file
C     on to window B
C
C     WRIXTO: Saves the output of each parallel path
C
C
C     Description of parameters:
C
C     GEN: Four dimension array for LI coefficients
C     computation
C     A: LI matrix
C     DIAGA: Vector loaded with A diagonal elements for
C     iterative computations
C     AINV: Inverse LI matrix
C     B: Linear LI window
C     WOVEC1: Auxiliary vector for the computation of inverse
C     matrix
C     WOVEC2: Auxiliary vector for the computation of inverse
C     matrix
C     GAUS1: Symmetric convergence window
C     GAUS2: Asymmetric convergence window
C     XTOT: Storing array for the output of each parallel
C     path
C     XI: Input array
C     XHi: Convergence layer array for level i
C     Ci: LI layer array for level i
C     Ni: Number of units per side of array at level i
C     SIGMA: LI standard deviation
C     SIGCO1: Convergence standard deviation
C     SIGCO2: Convergence standard deviation
C     SI: Self feedback value
C     THRES: Threshold value
C
C-------------------------------------------------------------
C
C     Initialization
C
      DIMENSION GEN(11,11,11,11)
      DIMENSION A(121,121),DIAGA(121)
      DIMENSION AINV(121,121),B(11,11),WOVEC1(121),WOVEC2(121)
      DIMENSION GAUS1(11,11),GAUS2(17,17)
      DIMENSION XTOT(11,11,4)
      DIMENSION XI(81,81)
      DIMENSION XH1(41,41),C1(41,41)
```

```
      DIMENSION XH2(21,21),C2(21,21)
      DIMENSION XH3(11,11),C3(11,11)
      DIMENSION XH4(11,11)
C
      N1=81
      N2=41
      N3=21
      N4=11
C
      SIGMA   = 2.0
      SIGCO1  = 1.3
      SIGCO2  = 2*SIGCO1
      SI      = 1.0
      THRES   = 0.0
C
      DO 10 I=1,121
        WOVEC1(I)=0.
        WOVEC2(I)=0.
10    CONTINUE
C
C-----------------------------------------------------------
C
C     Main Program
C
C     Initialization of Matrices and associated arrays
C
C     * Generate LI coefficients *
      CALL GENIND(GEN,SIGMA)
C     * Load LI coefficients on matrix A *
      CALL LOADA(GEN,A,SI,11,121)
C     * Load LI coefficients on matrix AINV *
      CALL LOADA(GEN,AINV,SI,11,121)
C     * compute inverse of matrix and store solution in AINV *
      CALL MINV(AINV,121,D,WOVEC1,WOVEC2)
C     * load LI linear window with middle row of AINV
      CALL LOADB(AINV,B)
C     * prepare A and DIAGA for iterative subroutine *
      CALL DIACLR(A,DIAGA,121)
C
C
C     * load symmetric convergence window *
      CALL LOGAUS(GAUS1,SIGCO1,SIGCO1,11,6,0.)
C
C
C     * clear input and output arrays *
      CALL CLR(XI,N1)
      CALL CLR(XH4,N4)
C
C     Response
C
```

```
C     * generate input *
      CALL GENINP(XI)
C
C     * calculate output first level *
      CALL CLRIO(XH1,C1,N2)
      CALL CONV(XI,XH1,GAUS1,11,6,N1,N2)
      CALL INHL(XH1,C1,B,N2)
C
C     * calculate the output of parallel paths with RF main
C       axis orientation given by ANG, output is stored at
C       XTOT *
        DO 20 KONT=1,4
          ANG=(FLOAT(KONT-1))*45.
          CALL CLRIO(XH2,C2,N3)
          CALL CLRIO(XH3,C3,N4)
C         * convergence asymmetric window oriented according
C           to ANG value *
          CALL LOGAUS(GAUS2,SIGCO1,SIGCO2,17,9,ANG)
C         * calculate level two *
          CALL CONV(C1,XH2,GAUS2,17,9,N2,N3)
          CALL INHN(XH2,C2,N3,A,DIAGA,THRES)
C         * calculate level three *
C         * no windowing is used at this LI layer due to
            the small size of the LI layer *
          CALL CONV(C2,XH3,GAUS1,11,6,N3,N4)
          CALL INH(XH3,C3,11,121,A,DIAGA,THRES)
C
C         * save parallel path output *
          CALL WRIXTO(C3,XTOT,KONT)
20      CONTINUE
C
C     * compute level four output *
      CALL CONVAN(XTOT,XH4)
C     * write file *
      CALL WRIFIL(XH4,N4)
C
      STOP
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE CLR(X,N)
C
C     This subroutine clears two dimensional arrays
C
      DIMENSION X(N,N)
C
      DO 10 I=1,N
      DO 10 J=1,N
        X(I,J)=0.
10    CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE CLRIO(AUXI,AUXO,N)
C
C     This subroutines clears the contents of two arrays:
C     AUXI and AUXO.
C
C     Description of parameters:
C
C     AUXI: array
C     AUXO: array
C     N: number of elements per side of the array
C
C-------------------------------------------------------------

      DIMENSION AUXI(N,N),AUXO(N,N)
C
      DO 10 I=1,N
      DO 10 J=1,N
         AUXI(I,J)=0
         AUXO(I,J)=0.
10    CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE CONV(XI,XH,GAUS,K1,K2,N1,N2)
C
C     This subroutine calculates the response of a
C     convergence layer using the windowing technique. The
C     window is shifted along the source array in steps of
C     two units at a time, and the point to point product of
C     the window elements with their corresponding source
C     array elements is performed and added up to determine
C     the target unit's response. The results are stored in
C     neighboring units of the target array, giving rise to
C     the pyramidal structure of the model.
C
C     Description of parameters:
C
C     XI: source array
C     XH: target array
C     GAUS: convergence window
C     N1: number of units per side of source array
C     N2: number of units per side of target array
C     K1: number of units per side of convergence window
C     K2: middle of convergence window
C
C-----------------------------------------------------------
C
      DIMENSION XI(N1,N1),XH(N2,N2),GAUS(K1,K1)
C
      IXCONT=0
C     * window shifts two steps at a time in x direction
C       across the source array *
      DO 10 IX1=1,N1,2
C
C     * target pointer increments one unit at time in x
C       direction *
      IXCONT=IXCONT+1
      IYCONT=0
C
C     * window shifts two steps at a time in y direction
C       across the source array *
      DO 10 IY1=1,N1,2
C       *target pointer increments one unit at time in y
C        direction *
         IYCONT=IYCONT+1
C
C        * multiply point to point the source with the window
C          and add for target response *
         DO 10 IX2=1,K1
            IX3=IX1+K2-IX2
         DO 10 IY2=1,K1
            IY3=IY1+K2-IY2
```

```
C
C            * avoid operating outside of source limits *
             IF((IX3.LE.0).OR.(IX3.GE.(N1+1))) GOTO 10
             IF((IY3.LE.0).OR.(IY3.GE.(N1+1))) GOTO 10
C
          XH(IXCONT,IYCONT)=XH(IXCONT,IYCONT)+
     @           GAUS(IX2,IY2)*XI(IX3,IY3)
C
10     CONTINUE
C
       RETURN
       END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE CONVAN(XTOT,XH)
C
C     This subroutine adds point to point the contents of
C     all four 11*11 arrays in XTOT
C
C     Description of parameters:
C
C     XTOT: input array
C     XH: output array
C
C-------------------------------------------------------------
C
      DIMENSION XTOT(11,11,4),XH(11,11)
C
      DO 10 I=1,11
      DO 10 J=1,11
      DO 10 K=1,4
C
         XH(I,J)=XH(I,J)+XTOT(I,J,K)
C
10    CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE DIACLR(A,DIAGA,N)
C
C     This subroutine transfers the contents in the main
C     diagonal of matrix A to vector DIAGA and clears
C     the matrix's diagonal
C
C     Description of parameters
C
C     A: LI matrix
C     DIAGA: vector
C     N: dimension of A and DIAGA
C
C-------------------------------------------------------------
C
      DIMENSION A(N,N),DIAGA(N)
C
      DO 10 I=1,N
         DIAGA(I)=A(I,I)
         A(I,I)  =0.
10    CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE GENIND(GEN,SIGMA)
C
C     This subroutine generates the LI coefficients using a
C     four dimensional array. The first two subscripts
C     indicate the row and column location in the LI array
C     where the lateral connection originates and the last
C     two subscripts indicate the row and column of the
C     array to which the lateral connection projects.
C
C     Description of parameters used:
C
C     GEN: Four dimensional array for systematic generation
C     of LI coefficients
C     SIGMA: LI standard deviation
C
C-------------------------------------------------------------
C
      DIMENSION GEN(11,11,11,11)
C
      DO 10 IUX=1,11
      DO 10 IUY=1,11
        DO 10 IX=1,11
        DO 10 IY=1,11
C
          GEN(IUX,IUY,IX,IY)=EXP(-((IX-IUX)**2+(IY-IUY)**2)
     @          /(2*(SIGMA**2)))
C
10      CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE GENINP(XI)
C
C     This subroutine generates the input pattern. This is
C     done by setting the loop indexes with the appropriate
C     values.
C
C     Description of parameters:
C
C     XI: Input array on which the pattern is generated
C
C-----------------------------------------------------------------

      DIMENSION XI(81,81)
C
      DO 20 I=21,61
      DO 20 J=21,61
C
      XI(I,J)=1
C
20     CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
```

```
C
      SUBROUTINE INH(XH,C,N1,N2,A,DIAGA,T)
C
C     This subroutine calculates the piecewise response of
C     a LI layer. It does not use the windowing technique
C     since it is assumed to operate with arrays whose
C     side length is comparable to the source's; thus saving
C     computation time. To solve the system a line scan
C     transformation is performed with the array elements.
C     This arrangement allows the application of an
C     iterative method based on the Gaus-Seidel algorithm.
C
C     Description of parameters:
C
C     A: matrix with the LI coefficients and zero for the
C     elements in the main diagonal
C     DIAGA: contains original elements in main diagonal
C     of A
C     XH: input array
C     C: output array
C     XIN: auxiliary array used store the values of XH
C     after the 'line scan' transformation
C     XOUT: auxiliary array with output values after
C     iterative solution of the system
C     N1: number of units per side in LI layer
C     N2: number of units per side in matrix A and
C     vector DIAGA
C
C-----------------------------------------------------------
C
      DIMENSION A(N2,N2),DIAGA(N2),XH(N1,N1),C(N1,N1)
      DIMENSION XIN(441),XOUT(441)
C
C     * line scan transformation of input array *
      DO 5 IX=1,N1
      DO 5 IY=1,N1
            AUX=N1*(IX-1)+IY
         XIN(AUX)=XH(IX,IY)
C        * clearing the output array to start first
C          iteration (may be commented out) *
         XOUT(AUX)=0.
5     CONTINUE
C
      ITER=1
C     * resolution coefficient *
       EPS=1E-04
C
C     * start iteration *
99    BIG=0.0
      DO 100 I=1,N2
```

```
          SUM=0.
          DO 10 J=1,N2
            SUM=SUM+A(I,J)*XOUT(J)
10      CONTINUE
C
C       * solve for variable I *
        TEMP=(XIN(I)-SUM)/DIAGA(I)
C       * compare with threshold and change if necessary *
        IF(TEMP.LT.T) TEMP=T
C       * determine the maximum difference in all present
C         output values with respect to values in the previous
C         iteration *
        RESID=ABS(TEMP-XOUT(I))
        IF(RESID.GT.BIG)BIG=RESID
        XOUT(I)=TEMP
C
100     CONTINUE
C
        IF(BIG.LT.EPS) GOTO 200
C       * stop if more than 250 iterations are made *
        IF(ITER.GT.250) STOP
        ITER=ITER+1
        GOTO 99
C
C       * once resolution specifications are met, apply
C         inverse 'line scan' transformation and place output
C         values in their corresponding 2 D array positions
200     DO 20 I=1,N1
        DO 20 J=1,N1
          AUX=(I-1)*N1+J
          C(I,J)=XOUT(AUX)
20      CONTINUE

        RETURN
C
        END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE INHL(XH,C,B,N)
C
C     This subroutine calculates the response of a LI layer
C     with linear units. The windowing technique is used.
C
C     Description of parameters used:
C
C     XH: input array
C     C: output array
C     B: window with transfer elements
C
C-------------------------------------------------------
C
      DIMENSION B(11,11)
      DIMENSION XH(N,N),C(N,N)
C
C     * increment pointer that indicates section of input
C       array to be processed by window *
      DO 10 IX1=1,N
      DO 10 IY1=1,N
C
C     * point to point multiplication and addition of
C       window B elements with section of input array
C       contents *
      DO 10 IX2=1,11
         IX3=IX1+6-IX2
      DO 10 IY2=1,11
         IY3=IY1+6-IY2
C
C        * avoid operating outside input array *
         IF((IX3.LE.0).OR.(IX3.GE.(N+1))) GOTO 10
         IF((IY3.LE.0).OR.(IY3.GE.(N+1))) GOTO 10
C
            C(IX1,IY1)=C(IX1,IY1)+B(IX2,IY2)*XH(IX3,IY3)
C
10    CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------
```

```
C
      SUBROUTINE INHN(XH,C,N,A,DIAGA,THRES)
C
C     This subroutine calculates the piecewise linear
C     response of a LI layer. The windowing technique is
C     used. A section of the input array is systematically
C     picked and stored in a window array that transfers the
C     data to the iterative subroutine MIO. This subroutine
C     solves the system based on the Gauss-Seidel algorithm.
C     The solution for the central unit of the input window
C     is returned and saved. The following section of the
C     input array is picked and the process continues.
C
C     Description of parameters:
C
C     A: matrix with LI coefficients and zero for the
C     elements in the main diagonal
C     DIAGA: contains original elements in main diagonal
C     of A
C     XH: input array
C     C: output array
C     XHAUX: auxiliary array to store the portion of the
C     input array that is being processed by the LI
C     iterative window
C     C1111: variable with solution value for window's
C     central unit
C
C     Subroutines used:
C
C     MIO: solves the piecewise system using a modified
C     Gaus-Seidel iterative method. In doing so, it must
C     first rearrange its input array XHAUX through a
C     'line scan' transformation.
C
C-----------------------------------------------------------
C
      DIMENSION A(121,121),DIAGA(121)
      DIMENSION XH(N,N),C(N,N)
      DIMENSION XHAUX(11,11)
C
C
      C1111=0.
C
C     * start window shift *
      DO 10 IX1=1,N
      DO 10 IY1=1,N
C
C        * clear auxiliary array before every window shift*
         DO 20 KX=1,11
         DO 20 KY=1,11
```

```
            XHAUX(KX,KY)=0.
20        CONTINUE
C
C       * load corresponding section of input array on to
C         the auxiliary array *
        DO 30 IX2=1,11
           IX3=IX1+6-IX2
        DO 30 IY2=1,11
           IY3=IY1+6-IY2
C
           IF((IX3.LE.0).OR.(IX3.GE.N+1)) GOTO 30
           IF((IY3.LE.0).OR.(IY3.GE.N+1)) GOTO 30
C
              XHAUX(IX2,IY2)=XH(IX3,IY3)
C
30      CONTINUE
C
C       * solve for windowed section *
        CALL MIO(A,XHAUX,C1111,DIAGA,THRES)
C
C       * save solution for central unit of windowed
C         section *
        C(IX1,IY1)=C1111
C
10      CONTINUE
C
        RETURN
        END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE LOADA(GEN,A,SI,N1,N2)
C
C
C     This subroutine loads the LI coefficients stored in
C     GEN on to a matrix A. Now, the array units are labelled
C     according to a 'line scan' numbering array and not
C     according to their coordinates in the array. Thus,
C     in matrix A, an element in row X and column Y refers
C     to the lateral connection from unit Y to unit X.
C     The self feedback effect is also included.
C
C     Description of parameters:
C
C     GEN: Four dimension array with LI coefficients
C     A: Matrix loaded with reordered coefficients
C     SI: Self feedback coefficient
C     N1: Dimension of GEN
C     N2: Dimension of A
C
C-----------------------------------------------------------
C
      DIMENSION GEN(N1,N1,N1,N1),A(N2,N2)
C
C     * load A from GEN according to line scan
C       transforamtion *
      I=0
      DO 10 IUX=1,N1
      DO 10 IUY=1,N1
C
         I=I+1
         J=0
C
        DO 10 IX=1,N1
        DO 10 IY=1,N1
C
           J=J+1
C
           A(I,J)=GEN(IUX,IUY,IX,IY)
C
10    CONTINUE
C
C     * self feedback effect *
      DO 20 I=1,N2
        A(I,I)=1.+SI
20    CONTINUE
C
      RETURN
      END
C
```

```
C-------------------------------------------------------------
```

```
C
      SUBROUTINE LOADB(A,B)
C
C     This subroutine loads the contents of the middle row
C     of matrix A on to matrix B. In doing so, an 'inverse
C     line scan' transformation that arranges the B matrix
C     elements to be in line with the corresponding input
C     unit for point to point window computations.
C
C     Description of parameters:
C
C     A: system transfer function
C     B: array for linear windowing at LI layer
C
C-------------------------------------------------------------
C
      DIMENSION A(121,121),B(11,11)
C
      DO 10 I=1,11
      DO 10 J=1,11
        B(I,J)=A(61,11*(I-1)+J)
10    CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE LOGAUS(GAUS,SIGCOX,SIGCOY,K1,K2,ANG)
C
C     This subroutine loads the convergence window GAUS with
C     the weighing coefficients resulting of applying a
C     2 dimensional gaussian distribution with standard
C     deviations SIGCOX and SIGCOY on the respective axis.
C     Orientation transformations are applied to rotate
C     the distribution's main axis an angle as indicated
C     by ANG.
C
C     Description of parameters:
C
C     GAUS: Convergence window
C     SIGCOX: Standard deviation for x axis
C     SIGCOY: Standard deviation for y axis
C     ANG: Rotation of main axis
C     K1: Dimension of window
C     K2: Center of window
C
C-------------------------------------------------------------
C
      DIMENSION GAUS(K1,K1)
C
C     * convert to radians *
      ALPHA=ANG*3.141591/180
C
      DO 10 IX=1,K1
      DO 10 IY=1,K1
C
C     * rotation transformations *
      XR=(FLOAT(IX-K2))*COS(ALPHA)+(FLOAT(IY-K2))*SIN(ALPHA)
      YR=(FLOAT(IY-K2))*COS(ALPHA)-(FLOAT(IX-K2))*SIN(ALPHA)
C
C     * calculate coefficients *
        GAUS(IX,IY)=EXP(-(((XR)/SIGCOX)**2+
     @      ((YR)/SIGCOY)**2)/2)
C
10    CONTINUE
      RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE MIO(A,XHAUX,C1111,DIAGA,T)
C
C     This subroutine calculates the solution of a system
C     of piecewise linear equations using an iterative
C     algorithm. The method is based on a modified version
C     of the Gaus-Seidel iterative solution, with a step
C     added at the end of each iteration to account for the
C     threshold effect. The subroutine also rearranges the
C     input data array through 'line scan' transformation
C     into a one dimensional vector, thus maintaining
C     coherence with the form in which the LI coefficients
C     are ordered in matrix A. The solution value of the
C     central unit is returned to the calling program.
C
C     Description of parameters:
C
C     A: matrix with LI coefficients and zero for the
C     elements in the main diagonal
C     DIAGA: contains original elements in main diagonal
c     of A
C     XHAUX: auxiliary array with section of input array
C     to be processed
C     XIN: auxiliary vector which stores input values
C     after the 'line scan' transformation
C     XOUT: auxiliary vector on which the solution values
C     of the system are stored
C     C1111: variable with solution value of central unit
C
C-------------------------------------------------------------
C
      DIMENSION A(121,121),DIAGA(121),XHAUX(11,11)
      DIMENSION XIN(121),XOUT(121)
C
C     * line scan transformation *
      DO 5 IX=1,11
      DO 5 IY=1,11
            AUX=11*(IX-1)+IY
          XIN(AUX)=XHAUX(IX,IY)
C         * may be commented out to reduce the number of
C           iterations in uniform input regions by using
C           the output values as starting approximations
C           for the following iterative solution *
          XOUT(AUX)=0.
5     CONTINUE
C
      ITER=1
C     * set resolutiion coefficient *
      EPS=1E-04
C
```

```
99     BIG=0.0
       DO 100 I=1,121
         SUM=0.
         DO 10 J=1,121
           SUM=SUM+A(I,J)*XOUT(J)
10     CONTINUE
C
C      * solve for variable I *
       TEMP=(XIN(I)-SUM)/DIAGA(I)
C      * compare with threshold and change if necessary *
       IF(TEMP.LT.T) TEMP=T
C      * determine maximum difference of all present output
C        values with respect to values in the previous
C        iteration *
       RESID=ABS(TEMP-XOUT(I))
       IF(RESID.GT.BIG)BIG=RESID
       XOUT(I)=TEMP
C
100    CONTINUE
C
       IF(BIG.LT.EPS) GOTO 200
C      * stop if more than 250 iterations *
       IF(ITER.GT.250) STOP
       ITER=ITER+1
       GOTO 99
C
C      * when resolution specifications are met, return
C        solution value for central unit of array *
200    C1111=XOUT(61)
       RETURN
C
       END
C
C-------------------------------------------------------------
```

```
C
C       SUBROUTINE MINV
C
C       PURPOSE
C          INVERT A MATRIX
C
C       USAGE
C          CALL MINV(A,N,D,L,M)
C
C       DESCRIPTION OF PARAMETERS
C          A - INPUT MATRIX, DESTROYED IN COMPUTATION AND
C              REPLACED BY RESULTANT INVERSE.
C          N - ORDER OF MATRIX A
C          D - RESULTANT DETERMINANT
C          L - WORK VECTOR OF LENGTH N
C          M - WORK VECTOR OF LENGTH N
C
C       REMARKS
C          MATRIX A MUST BE A GENERAL MATRIX
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C          NONE
C
C       METHOD
C          THE STANDARD GAUSS-JORDAN METHOD IS USED. THE
C          DETERMINANT IS ALSO CALCULATED. A DETERMINANT
C          OF ZERO INDICATES THAT THE MATRIX IS SINGULAR.
C
C     .....................................................
C
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
C
C        .....................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS
C        DESIRED, THE C IN COLUMN 1 SHOULD BE REMOVED FROM
C        THE DOUBLE PRECISION STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,D,BIGA,HOLD,DABS
C
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION
C        STATEMENTS APPEARING IN OTHER ROUTINES USED IN
C        CONJUNCTION WITH THIS ROUTINE.
C
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE
C        MUST ALSO CONTAIN DOUBLE PRECISION FORTRAN
C        FUNCTIONS.  ABS IN STATEMENT 10 MUST BE CHANGED TO
C        DABS.
C
```

```
C        ..................................................
C
C        SEARCH FOR LARGEST ELEMENT
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT
C        IS CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
```

```
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
```

```
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
C
C-------------------------------------------------------------
```

```
C
      SUBROUTINE WRIFIL(XO,N)
C
C     This subroutine writes out a file with the system
C     output values.
C
C     Description of parameters:
C
C     XO: output array
C
C-----------------------------------------------------------------
C
      DIMENSION XO(N,N)
C
        WRITE(6,100)((XO(I,J),J=1,N),I=1,N)
100     FORMAT(11F10.5)
C
      RETURN
      END
C
C-----------------------------------------------------------------
```

```
C
      SUBROUTINE WRIXTO(C3,XTOT,KONT)
C
C     This subroutine loads the contents of array C3 on the
C     layer of XTOT indicated by KONT
C
C     Description of parameters:
C
C     C3: input array
C     XTOT: storing array
C     KONT: pointer
C
C-------------------------------------------------------------
C
      DIMENSION C3(11,11),XTOT(11,11,4)
C
      DO 10 I=1,11
      DO 10 J=1,11
        XTOT(I,J,KONT)=C3(I,J)
10    CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------
```