MATNET: A neural network for medial axis transformation

Article *in* Journal of the Chinese Institute of Engineers · September 1993

DOI: 10.1080/02533839.1993.9677551

2 authors, including:

Yung-Sheng Chen
Yuan Ze University
212 PUBLICATIONS **2,372** CITATIONS

SEE PROFILE

The following manuscript was published in

Yung-Sheng Chen and Wen-Hsing Hsu, **MATNET: a neural network for medial axis transformation**, *Journal of the Chinese Institute of Engineers*, Vol. 16, No. 6, 757-771, 1993.

https://www.tandfonline.com/doi/abs/10.1080/02533839.1993.9677551

MATNET: A NEURAL NETWORK FOR MEDIAL AXIS TRANSFORMATION

Yung-Sheng Chen*

Department of Electrical Engineering
Yuan-Ze Institute of Technology
Taoyuan, Taiwan 320, R.O.C.

Wen-Hsing Hsu

Institute of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 30043, R.O.C.
&
Institute of Information Science
Academia Sinica, Taiwan 10529, R.O.C.

Key Words: medial axis transformation, maximal neural network.

ABSTRACT

This paper describes a novel neural network, called MATNET, to perform the medial axis transformation which is often used to extract a stick-figure-like representation from a binary object for pattern analysis or recognition. The MATNET is derived from the structure of the retina, which consists of five neural layers, namely, receptors, horizontal cells, bipolar cells, ganglion cells, and response. In principle, the horizontal cell is implemented for distance computation; the bipolar cell (B-net) and the ganglion cell (G-net) are implemented for calculation of local minimum and local maximum, respectively. The B-net and G-net are concerned with the maximal neural network (Maxnet). The properties of Maxnet are also discussed. Experimental results show that the MATNET performs reasonably.

可實現中心軸轉換之類神經網路設計

陳永盛*

私立元智工學院電機工程學系

許文星

國立清華大學電機工程研究所
中央研究院資訊科學研究所

摘 要

本論文所提出的中心軸轉換之類神經網路是根據人眼視網膜的神經元層而

*Correspondence addressee

設計的。基本上，這些神經元層包括：接受訊號細胞層，水平細胞層，雙極細胞層，和神經節細胞層。在我們所設計之中心軸轉換神經網路中，水平細胞層專司刺激訊號點到非刺激訊號點間之“訊號落差計算”（即是傳統中心軸轉換之“距離計算”），雙極細胞層專司局部訊號最小值計算（即是找出目前刺激訊號點到非刺激訊號點之最近距離），而神經節細胞層則專司局部訊號最大值計算（即是找出屬中心軸刺激訊號點）。整個訊號傳遞過程是由接受訊號細胞層取得訊號，經由水平細胞層平行處理取得所有的“距離”資料，再經由雙極細胞層和神經節細胞層求得最後結果。其中水平細胞層專司水平情報處理，其細胞間的特殊連結設計，使得所有的距離資料可以容易地以平行處理方式一次反應出來。雙極細胞層和神經節細胞層採用著名的最大值類神經網路來實現。基於傳統最大值類神經網路有“對消效應”的缺點，我們提出神經鍵係數的修正克服此種效應。

實驗確定了此模型的可行性。

INTRODUCTION

Thinning or medial axis transformation (MAT) is the most common name for the process of transforming a line-like object from many pixels (picture elements) wide to just a single pixel (i.e., skeleton). A principal reason for developing thinning or MAT is that if two-dimensional binary images consisting of elongated or line-like objects are to be recognized, e.g., alphanumeric printed or handwritten characters, chromosomes, etc., the thickness of the strokes which constitute the objects generally do not contribute to the recognition. Presently, MAT's applications are numerous, e.g., inspection of printed circuit boards, counting of asbestos fibres on air filters, analysis of chromosome shapes, examination of soil cracking patterns, classification of fingerprints, recognition of characters, etc [7,8]. The method of MAT was first introduced by Blum in 1964 to characterize a binary object [1]. Other terms commonly used to describe this process are ''skeletonization'' or ''symmetric transformation'' [17].

Although thinning or MAT facilitates many applications in computer vision (especially for line images), several inherent problems limit the performance of computer algorithms. For instance, the extra short stroke easily generated in the crossing part of the original pattern after thinning process generally needs treatment by a complex merge-deleting algorithm, the noise and distortion generated will prevent correct extraction of various features, and the ''natural'' line images often appearing in the real world are very difficult to treat by thinning or MAT in pattern recognition. To overcome these problems, the major topics described in this paper are focused on the *possible* fundamental operations of processing line images from the viewpoints of neural networks and human visual perception. A neural-like structure called MATNET is newly introduced in this paper to implement the MAT.

For clarity, we begin with the introduction to the basic operations of the MAT in Section 1.1, then proceed to the fundamental structure of the MATNET in Section 1.2.

1. Introduction to the MAT

Fundamentally, the MAT of an image is composed of a set of centers and radii of the *maximal blocks*. Let a binary image Σ be composed of *pattern* S and its *background* $\overline{S}$. The maximal block is generally defined as follows. For a pattern pixel p, a block with a center, say (x,y) of p, and a radius r_p, is defined as a region which contains the pattern pixels. Let S_p be the largest one of such blocks around the pixel p. If there exist no pattern pixels q such that S_q contains S_p, call S_p a maximal block.

The MA of S ($=$*the set of centers of the maximal blocks*) consists of those pixels of S whose *distances* to the nearest boundary of $\overline{S}$ are local maxima. In general, there are three types of distances to be defined:

(1) The *Euclidean distance* between two pixels $p = (x,y)$ and $q = (u,v)$ is

$$d_{(e)}(p,q) = \sqrt{(x-u)^2+(y-v)^2} \quad .$$

(2) The *city block distance* between p and q is

$$d_{(4)}(p,q) = |x-u| + |y-v| \quad .$$

(3) The *chessboard distance* between them is

$$d_{(8)}(p,q) = max(|x-u|, |y-v|).$$

All three of these measures, $d_{(e)}$, $d_{(4)}$, and $d_{(8)}$, are *metrics* [16]. But it is convenient to work with the simpler measures $d_{(4)}$ and $d_{(8)}$ on digital pictures. The illustrations of the MA obtained from the distances $d_{(4)}$

and $d_{(8)}$ are shown in Fig. 1. For $d_{(4)}$ or $d_{(8)}$ distance computation, the *parallel algorithms* are often used. Although the MA can represent the fundamental structure of a picture, it is generally *disconnected,* since the MA is a set of *local distance maxima* [16]. Even though it is disconnected (Note that a thinning algorithm is not concerned here; in general, the thin line is connected [3,4,9].), it possesses inherently the profit of the MA in human visual perception, which is concerned with a *generalization process* of *intuitive pattern recognition* [2,6].

Basically, the major processing steps of the MAT can be roughly divided into three parts: distance computation, calculation of local minimum, and calculation of local maximum [16].

2. The method proposed in this paper

In this paper, we chiefly present a neural network called *MATNET* to implement the MAT based upon several advantages, e.g. parallelly asynchronous nonlogical operations, the capability of fault tolerance, and the convergence of finding multiple goals simultaneously, can all be performed in the neural networking system. The fundamental concern of this implementation is to explore intuitive pattern recognition in human visual perception such as we have studied the intuitive property of perceptual grouping in line segregation [5].

The structure of the MATNET is analogous to that of the retina [14], which lies in the interior of the eyeball. From the viewpoint of the anatomy, the retina consists of several layers of distinct types of cells, as roughly shown in Fig. 2. Distal to the *receptors* (rods and cones) are layers of *horizontal, bipolar,* and *ganglion cells.* The main path of the information flow in the visual system can be roughly described as follows. The light coming through lens of the eye falls on the mosaic of receptor cells in the retina. The receptor cells connect with horizontal cells which sense the stimuli from receptor cells and connect with bipolar cells. And the bipolar cells connect with ganglion cells. It is in these layers that the lateral interactions first occur. The output of the ganglion cells, which is by now an abstract representation of actual light stimulation, travels through the optic nerve to the lateral geniculate and then to the visual cortex.

In accordance with the structure of the retina, we propose an idea to map distance computation into horizontal cells which is based on the three points: (1) *resolution problems,* (2) *generation of horizontal cells,* and (3) *functions of the horizontal cell and its connection.* This will be dealt with in Section 2.

In principle, a set of horizontal cells will respond simultaneously when the corresponding receptor is stimulated. Each set of horizontal cells with respect to a position connects with a bipolar cell which implements the calculation of *local minimum,* i.e. the final distance computation. Such a bipolar cell is called *B-net* in this paper.

Because the MA is a set of local distance maxima, the final layer of the MATNET is used to perform the calculation of *local maxima* and is composed of a set of ganglion cells. Such a ganglion cell is called *G-net* here. Both B-net and G-net are concerned with *maximal neural network* (we call it *classical Maxnet*) [13] and will be described in detail in Section 3. The classical Maxnet has *canceling-effect* i.e., after the final iterations, if more than one maximum is initiated, there will be not any winner appearing on the output of the classical Maxnet. Therefore, we propose a modification of interconnections for the classical Maxnet so that the *modified Maxnet* can be used to derive to B-net and G-net. In Section 3, we

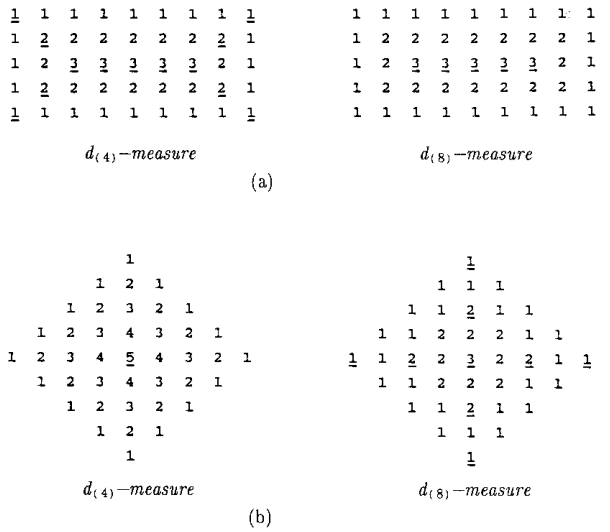


Fig. 1. Distances to S for the points of a rectangle (a) and a diamond (b); the MA points are underlined.

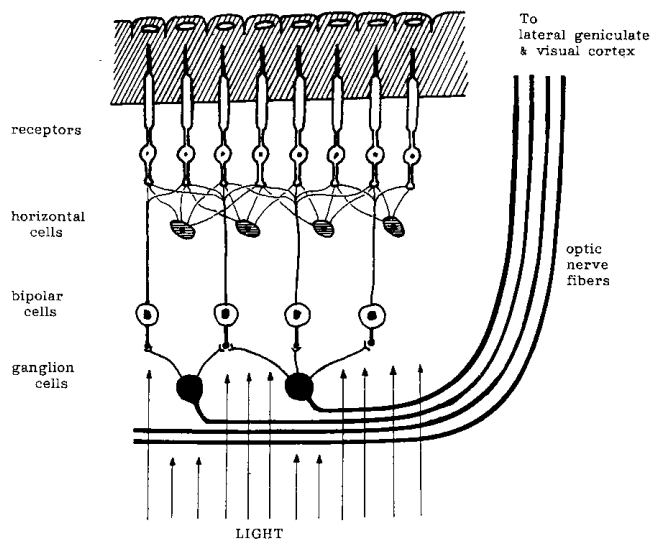


Fig. 2. A schematic diagram of the cells of the retina.

also present the details and carry out some experiments of the MATNET.

Finally, Section 4 gives our conclusions.

MAP DISTANCE TRANSFORMATION INTO HORIZONTAL CELLS

1. Resolution problems

Because the information processed in the retina is extremely complicated and highly parallel, the *resolution* of the visual receptive field should be defined reasonably so that, for each direction of each pixel, the distance computation satisfies the *isometric* property and will not be affected by the use of square grids on the digital picture. Assume that the square array containing $N \times N$ square grids, each denoted by (x,y) with 1×1 units, is the visual receptive field. There are two problems of resolution: (1) *selection of measures* among $d_{(4)}$, $d_{(8)}$, and $d_{(e)}$, and (2) *selection of the number of quantized orientations between 0° and 180°*.

First, in Fig. 3, the relationship of measuring results is $d_{(4)}(\alpha,\beta) > d_{(e)}(\alpha,\beta) > d_{(8)}(\alpha,\beta)$. If we select the measure $d_{(4)}$ (or $d_{(8)}$), the distance computation will be *over-computed* (or *under-computed*). Furthermore, $d_{(8)}(\alpha,\beta) = d_{(8)}(\alpha,\gamma)$ is also very unreasonable! Hence, the measure $d_{(e)}$ is selected for distance computation. To reduce the computation complexity of $d_{(e)}$ and to facilitate the generation of horizontal cells described later, we introduce the following equations: (Note that here $d_{(8)}(p,q) = max(|x-u|, |y-v|)$.)

$$d_{(e)}(p,q) = d_{(8)}(p,q) \cdot V_k , \tag{1}$$

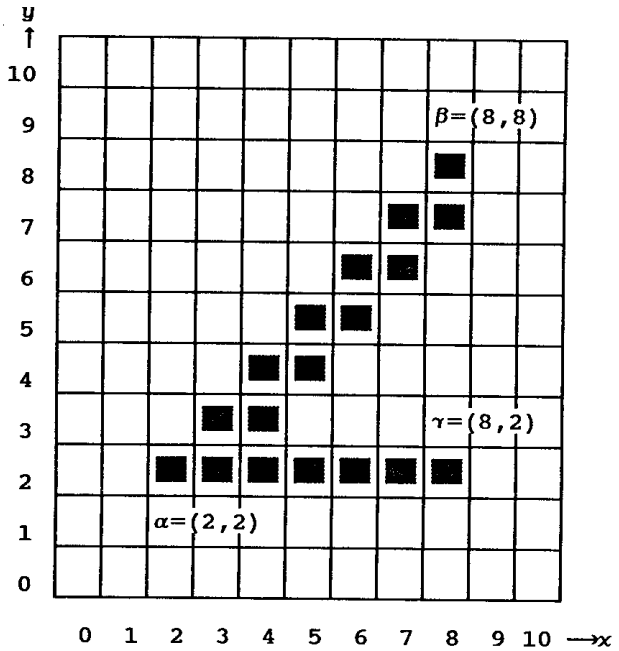
$$V_k = 1/cos(\theta_k) , \tag{2}$$

and

$$\theta_k = tan^{-1}[min(|x-u|, |y-v|) / d_{(8)}(p,q)] . \tag{3}$$

Here, V_k can be imagined as a *cell* charging the voltage which depends on the slope θ_k of the straight line segment between p and q. The measure $d_{(e)}(p,q)$ now can be easily performed by simply summing the voltages (V_k)'s $d_{(8)}(p,q)$ times, since $d_{(8)}(p,q)$ reflects the number of the points between p and q constructing the straight line segment. Hence, if we can build these useful cells and design their connections in advance, then the map of distance computation is constructed completely.

Secondly, we consider the problem of selecting the number of quantized orientations between 0° and 180° (the resolution M). The main concern of this problem is whether or not all the straight lines with the found quantized slopes (orientations) centering on the central point of a given receptive field can cover all the grids of the receptive field. Fig. 4 illustrates the selection of the resolution M. Because the visual receptive field defined is square, we can find a *greatest inner circle* and a *smallest outer circle* intersected with the square. The perimeter of the outer circle ($= \sqrt{2}\,\pi N$) is larger than that of the square ($=4N$), and the perimeter of the inner circle ($=\pi N$) is smaller than that of the square. In addition,



$d_{(4)}(\alpha,\beta) = |2-8| + |2-8| = 12$

$d_{(4)}(\alpha,\gamma) = |2-8| + |2-2| = 6$

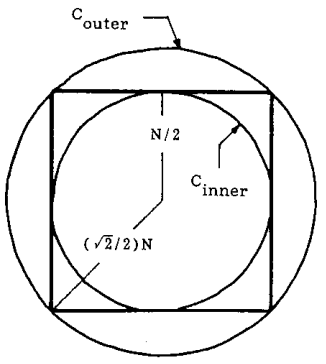
$d_{(8)}(\alpha,\beta) = max(|2-8|, |2-8|) = 6$

$d_{(8)}(\alpha,\gamma) = max(|2-8|, |2-2|) = 6$

$d_{(e)}(\alpha,\beta) = \sqrt{(2-8)^2 + (2-8)^2} = 6\sqrt{2}$

$d_{(e)}(\alpha,\gamma) = \sqrt{(2-8)^2 + (2-2)^2} = 6$

Fig. 3. Illustrations of three measures ($d_{(4)}$, $d_{(8)}$, and $d_{(e)}$). Note that if measure $d_{(4)}$ (or $d_{(8)}$) is used, the distance computation will be overcomputed (or undercomputed).



Perimeter of square = 4N
Perimeter of C_{inner} = πN
Perimeter of C_{outer} = $\sqrt{2}\,\pi$N

Fig. 4. Illustration of selecting the resolution M of the quantized orientations between 0° and 180°. Essentially, the set of all straight lines with found quantized slopes (orientations) centering on the central point of a given receptive field should cover the whole receptive field.

the quantized orientations need only over an angle between 0° and 180°. Since the perimeter of the inner circle is smaller than that of the square, half of it is insufficient to be selected for the resolution M. Hence, the half of the perimeter of the outer circle, i.e., $\sqrt{2}\,\pi N/2$, is chosen here. The resolution M is then fixed by the even integer nearest the value $\sqrt{2}\pi N/2$ and is denoted by $E[\sqrt{2}\pi N/2]$. For example, if N is selected to be 11, then $M = E[(\sqrt{2}\pi \cdot 11)/2] = E[24.435\ldots] = 24$, since the even integer nearest the value 24.435... is 24.

2. Generation of horizontal cells

Distance computation can be explained by the conception of horizontal cell construction. These cells are first grouped into M orientation planes as the ρ-space [18], which is a three-dimensional space where two dimensions are the discretized spatial dimensions of the image and the third dimension is the local, discretized orientation of edges or lines. Fig. 5 shows a diagram of ρ-space. Each orientation plane defined here and shown in Fig. 5 is a set of cells arranged in linear lines juxtaposed over the whole plane. All the linear lines in the kth orientation plane are disjoint pairwise and have the same slope $\frac{k}{M}\pi$. Therefore, Eq. (2) can be replaced by

$$V_k = \begin{cases} 1/\cos\left(\frac{k}{M}\pi\right), & \text{if } \frac{k}{M}\pi \le \frac{\pi}{4}, \\ 1/\cos\left(\frac{\pi}{2} - \frac{k}{M}\pi\right), & \text{if } \frac{\pi}{4} < \frac{k}{M}\pi \le \frac{\pi}{2}, \\ 1/\cos\left(\frac{k}{M}\pi - \frac{\pi}{2}\right), & \text{if } \frac{\pi}{2} < \frac{k}{M}\pi \le \frac{3}{4}\pi, \\ 1/\cos\left(\pi - \frac{k}{M}\pi\right), & \text{if } \frac{3}{4}\pi < \frac{k}{M}\pi \le \pi. \end{cases} \tag{2'}$$

The V_k defined here is thus always positive and ranged in $[1,\sqrt{2}]$. And the calculation of θ_k in Eq. (3) can be ignored.

Consequently, the distance computation in Eq. (1) is then divided into M sub-distance computations, as shown in the following equation:

$$d_k(p,q) = n_k \cdot V_k, \quad k = 0, 1, \ldots, M-1. \tag{4}$$

Here, n_k represents the number of the points between p and q falling on a linear line of the kth orientation plane. Note here that the point p is a pattern pixel and q is a background pixel, respectively, since the desired distance is measured between a given pixel of S and the nearest boundary point of the background $\bar{S}$ along the straight line, as introduced in Section 1.1.

For instance, the visual receptive field is given a size 11×11, i.e., $N = 11$. And we have the resolution $M = 24$ derived and exemplified in Section 2.1. The 24 orientation planes are generated and listed in Fig. 6, where

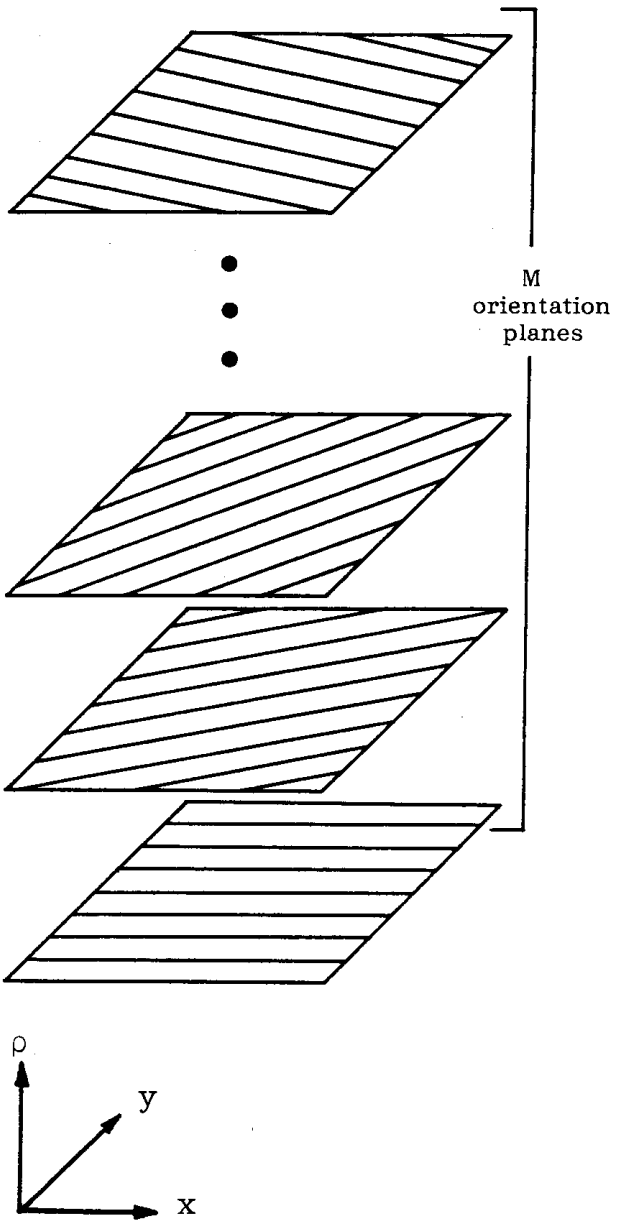


Fig. 5. Diagrammatic view of ρ-space. Here, two dimensions are the discretized spatial dimensions of the image and the third dimension is the discretized orientation of lines.

each element is a horizontal cell.

Because the distance computation will proceed over the directions from 0° to 360°, for each orientation plane, say the kth plane, it can be considered to be composed of two *horizontal cell planes* denoted by H_k^0 and H_k^1. The distance computations are treated by merely counting the number of (n_k)'s with respect to the relative opposite directions. Therefore, Eq. (4) can be restated as:

$$d_k^0(p,q_1) = n_k^0 \cdot V_k, \quad k = 0,1, \ldots, M-1, \tag{4'}$$

$$d_k^1(p,q_2) = n_k^1 \cdot V_k, \quad k = 0,1, \ldots, M-1. \tag{4''}$$

Here, p is a pattern pixel, and q_1 and q_2 are background pixels. These three points are collinear, and p is located

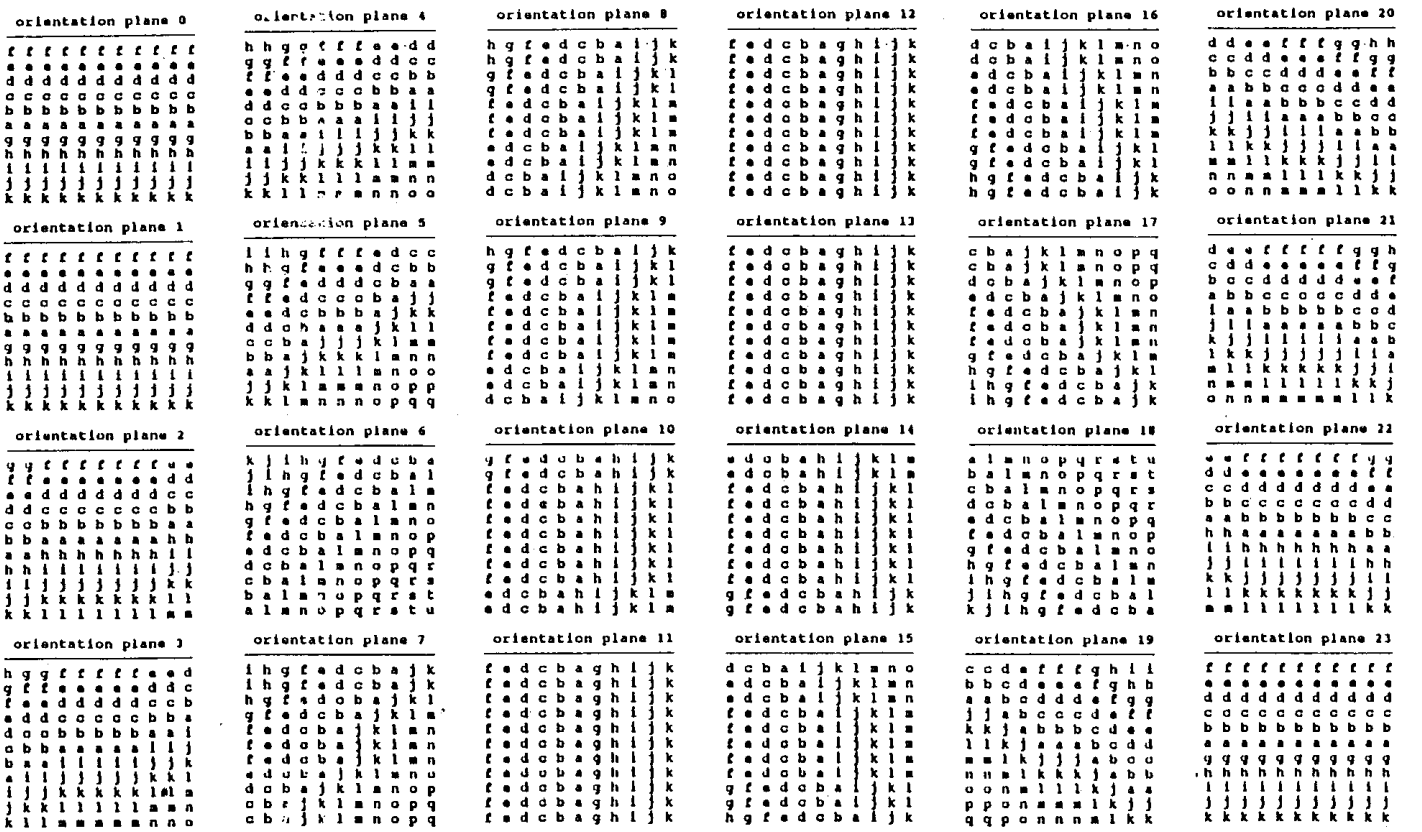


Fig. 6. The size of the visual receptive field is 11×11. There are 24 orientation planes to be generated. Here, for each plane, the cells are arranged in linear lines juxtaposed over the whole plane. The cells in a linear line are marked by the same character, i.e., "a", "b", ... etc.

between q_1 is q_2, as illustrated in Fig. 7. Let $d_k^0(p,q_1)$ be the distance between p and q_1 with the direction from 0° to 180°, and $d_k^1(p,q_2)$ be the distance between p and q_2 with the direction from 180° to 360°, respectively. Such arrangement of horizontal cell planes will facilitate the design of functions of the horizontal cell and its connections described in Section 2.3. Hence, when the number of orientation planes is M, then that of horizontal cell planes is $2M$. For the example shown in Fig. 6, the number of horizontal cell planes is 48.

3. Functions of a horizontal cell and its connections

Figure 8 shows the connections between horizontal cells. Here, the top input plane represents the array of receptors. When a receptor $R(x,y)$ is stimulated, all the corresponding horizontal cells $H_k^0(x,y)$ and $H_k^1(x,y)$, $k=0,1,\ \ldots,\ M-1$ are charged a voltage V_k which is specified by Eq. (2) or (2′), respectively. The voltage is charged from zero volt to the steady V_k which will be held until the stimulus is removed. If receptor $R(x,y)$ is not stimulated, then the following three functions occur:

(1) All the *links* of the corresponding horizontal cells are *open*, i.e., the links $l_k^0(x,y)$ and $l_k^1(x,y)$, $k=0,1,\ \ldots,\ M-1$ are *turned off*. Here, each link between two horizontal cells (illustrated in Fig. 8) is *gated* by the signal line of its corresponding receptor $R(x,y)$.
(2) All the corresponding cells are *discharged* to zero volt.
(3) Those corresponding cells are *grounded internally*.

According to the functions described above, assume that there is a stimulated pattern. For each horizontal cell, we can obtain an *output*, i.e., $d_k^0(x,y) = n_k^0 \cdot V_k$ or $d_k^1(x,y) = n_k^1 \cdot V_k$ depending on the cell being $H_k^0(x,y)$ or $H_k^1(x,y)$. For example, $d_k^0(x,y) = 5V_k$, $d_k^1(x,y) = 2V_k$ in the illustration of Fig. 9. Fig. 10 shows all (n_k)'s (or (d_k)'s) in every horizontal cell plane with the same pattern in Fig. 1a as the stimulus input pattern. Since the desired distance from a pattern pixel (x,y) to the background is the *minimum* of all the distances $(d_k(x,y))$'s, for each (x,y), we have to collect all the output $d_k^0(x,y)$ or $d_k^1(x,y)$, $k=0,1,\ \ldots,\ M-1$ shown in the bottom plane of Fig. 8 and feed them into the B-net described in Section 3.3.

THE MATNET

Because the final distance computation is to select the local minimum from the set of all distances over all

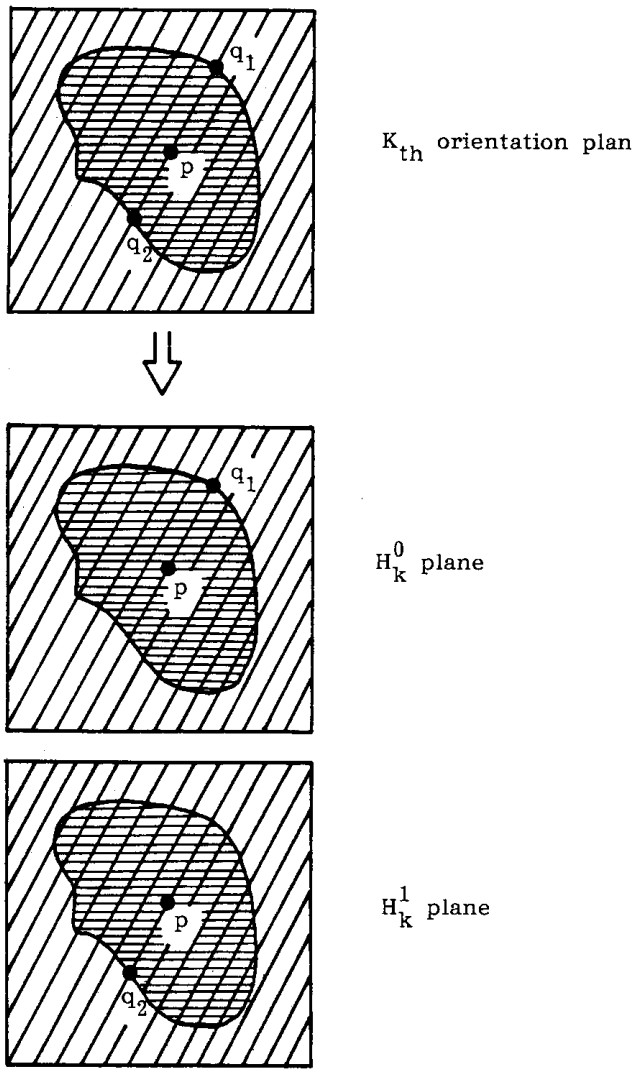


Fig. 7. An orientation plane is composed of two horizontal cell planes so that the distance measures of two mutually opposite directions for a pixel P can be performed (mapped) by two cell planes, respectively.

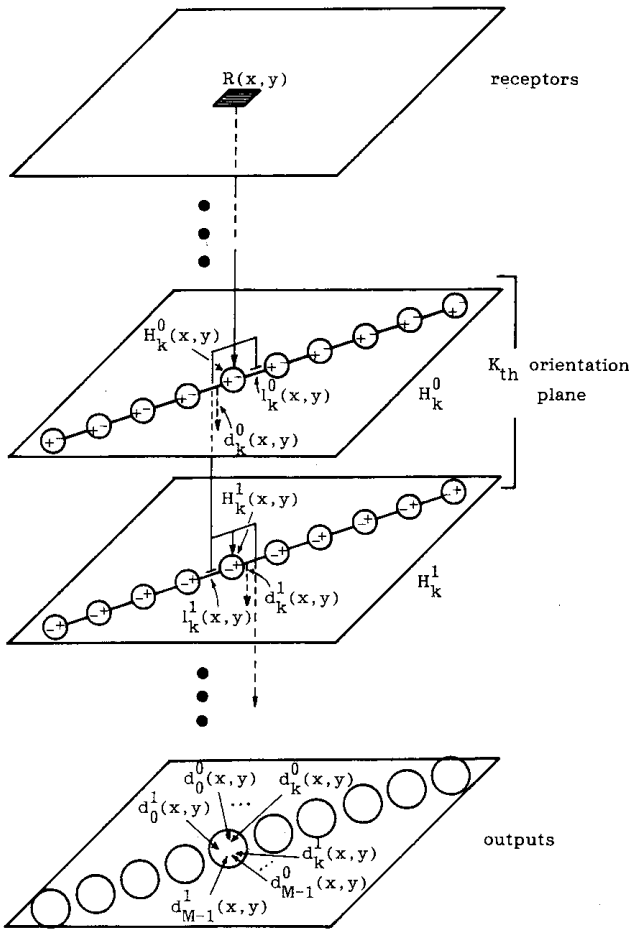


Fig. 8. Diagram of the layer of horizontal cells. It consists of an input plane, 2M horizontal cell planes, and an output plane. The connections between horizontal cells are also shown here.

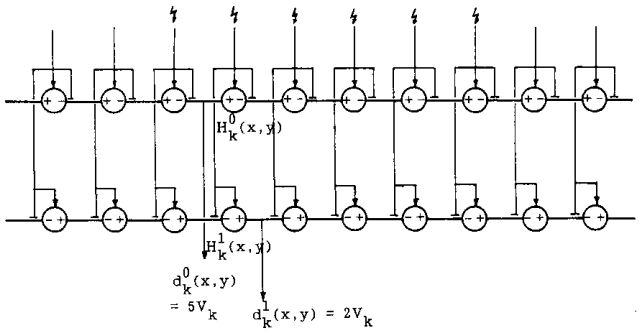


Fig. 9. Illustration of distance computations to map a given kth orientation line consisting of a series of horizontal cells. Here, "y" denotes a stimulus pattern pixel there. And V_k is given by Eq. (2').

the specified directions for each pattern pixel (x,y), and because the MA of a picture is a set of the local distance maxima, we shall first introduce the so-called maximal neural network in this section. Then we will present the modified weighting coefficients of interconnections of the Maxnet so that the calculation of local minimum (maximum) used in MATNET can be derived.

1. Classical Maxnet

To select the maximum over the M x_j values, we consider three topologically different neural net structures, which have been developed by Lippmann et al. [13]. They are Brute force, Binary tree, and Maxnet structure. The first two nets use strictly *feed-forward* connections and are relatively large, whereas the third net uses fully *feedback* connections and is less complex. According to the comparison of the three structures in Lippmann et al's report, the Maxnet requires the fewest nodes to pick the maximum value. Hence, the Maxnet should be preferred whenever the deciding factor is the number of nodes required and the slightly delay added by the need to iterate is acceptable. (Note that the Binary tree may be preferred when the deciding factor is the number of interconnections.)

The Maxnet shown in Fig. 11 is motivated by a large number of connections in biological neural nets and

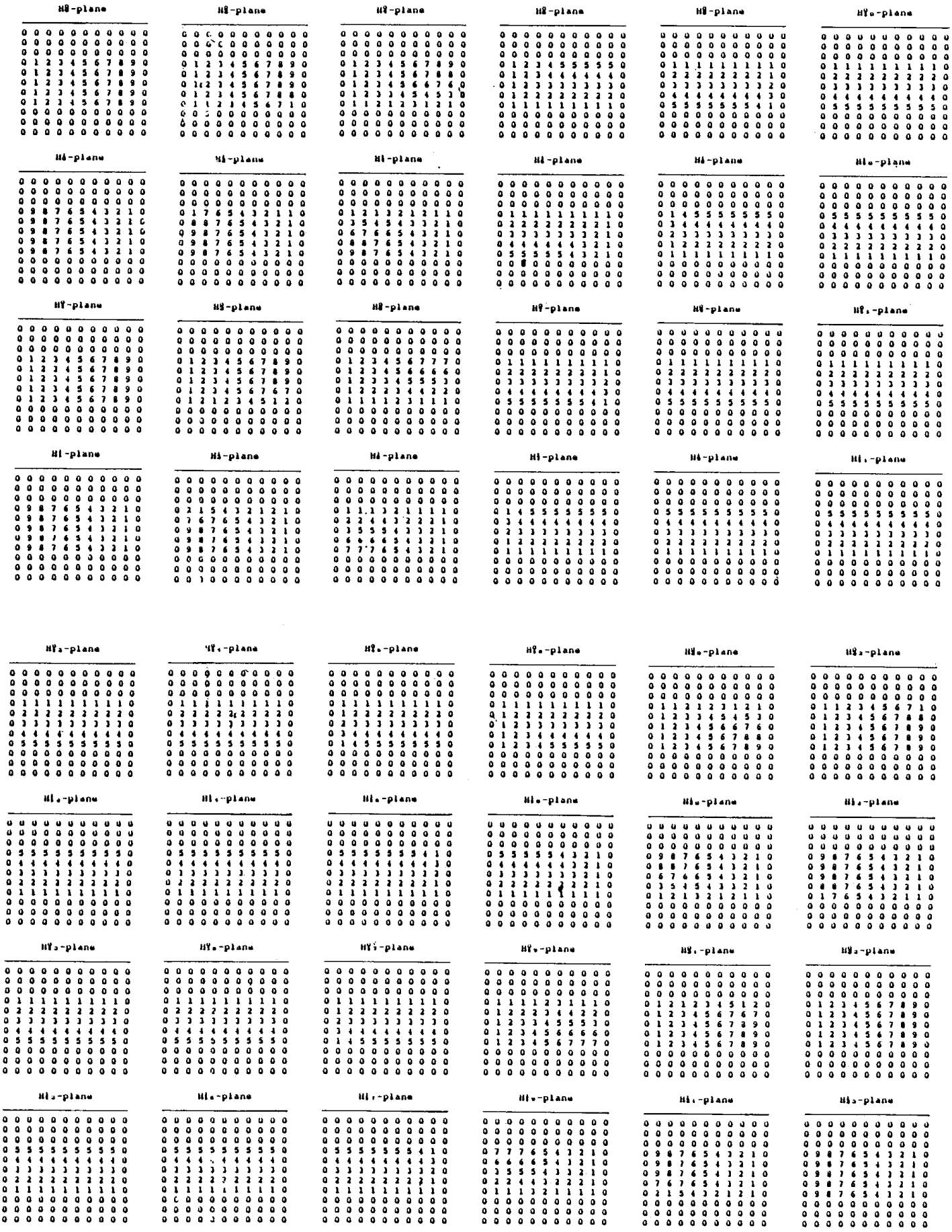


Fig. 10. All (n_k)'s in each horizontal cell plane are produced using the same pattern in Fig. 1a as an input stimulus pattern. The numeral shown in each plane denotes the computed distance (d_k)'s which are referred to Eqs. (4′) and (4″).

by laterally interconnected networks as decribed by Kohonen [12]. Hence it is sometimes called a "winner-take-all" net. Although the Maxnet is similar in structure to the Hopfield net [10], it uses *threshold-logic* nodes instead of *hard-limit* nodes and feeds the output of each node back to its input instead of disallowing this feedback path.

The basic equations of Maxnet are described as

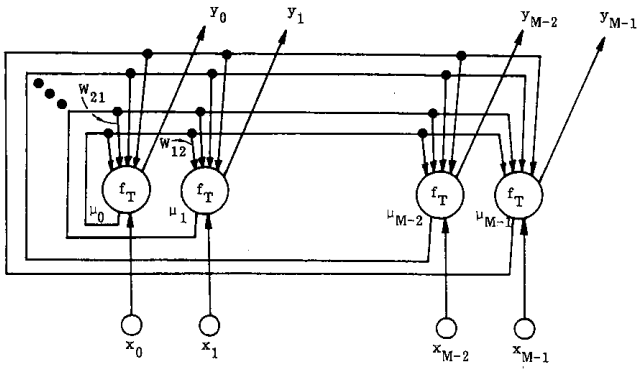


Fig. 11. Maxnet shown here is motivated by the large number of connections in biological neural nets and by laterally interconnected networks as described by Kohonen [12].

node :	$\mu_0(t)$	$\mu_1(t)$	$\mu_2(t)$	$\mu_3(t)$	$\mu_4(t)$	$\mu_5(t)$	$\mu_6(t)$	$\mu_7(t)$
t = 0:	5.000	4.000	7.000	6.000	8.000	3.000	4.000	2.000
t = 1:	1.600	0.500	3.800	2.700	4.900	0.000	0.500	0.000
t = 2:	0.360	0.000	2.780	1.570	3.990	0.000	0.000	0.000
t = 3:	0.000	0.000	2.188	0.857	3.519	0.000	0.000	0.000
t = 4:	0.000	0.000	1.750	0.286	3.214	0.000	0.000	0.000
t = 5:	0.000	0.000	1.400	0.000	3.011	0.000	0.000	0.000
t = 6:	0.000	0.000	1.099	0.000	2.871	0.000	0.000	0.000
t = 7:	0.000	0.000	0.812	0.000	2.761	0.000	0.000	0.000
t = 8:	0.000	0.000	0.536	0.000	2.680	0.000	0.000	0.000
t = 9:	0.000	0.000	0.268	0.000	2.626	0.000	0.000	0.000
t = 10:	0.000	0.000	0.006	0.000	2.599	0.000	0.000	0.000
t = 11:	0.000	0.000	0.000	0.000	2.599	0.000	0.000	0.000

$W_{kj} = -\epsilon$, $(k \neq j, \forall k,j)$ where $\epsilon = 0.1$ $(< 1/8)$

(a)

node :	$\mu_0(t)$	$\mu_1(t)$	$\mu_2(t)$	$\mu_3(t)$	$\mu_4(t)$	$\mu_5(t)$	$\mu_6(t)$	$\mu_7(t)$
t = 0:	5.000	4.000	8.000	8.000	8.000	3.000	4.000	2.000
t = 1:	1.300	0.200	4.600	4.600	4.600	0.000	0.200	0.000
t = 2:	0.000	0.000	3.510	3.510	3.510	0.000	0.000	0.000
t = 3:	0.000	0.000	2.808	2.808	2.808	0.000	0.000	0.000
t = 4:	0.000	0.000	2.246	2.246	2.246	0.000	0.000	0.000
t = 5:	0.000	0.000	1.797	1.797	1.797	0.000	0.000	0.000
t = 6:	0.000	0.000	1.438	1.438	1.438	0.000	0.000	0.000
t = 7:	0.000	0.000	1.150	1.150	1.150	0.000	0.000	0.000
t = 8:	0.000	0.000	0.920	0.920	0.920	0.000	0.000	0.000
t = 9:	0.000	0.000	0.736	0.736	0.736	0.000	0.000	0.000
•								
•								
t = 20:	0.000	0.000	0.063	0.063	0.063	0.000	0.000	0.000
t = 21:	0.000	0.000	0.051	0.051	0.051	0.000	0.000	0.000
•								
•								
t = 30:	0.000	0.000	0.007	0.007	0.007	0.000	0.000	0.000
t = 31:	0.000	0.000	0.005	0.005	0.005	0.000	0.000	0.000
•								
•								
t = 40:	0.000	0.000	0.001	0.001	0.001	0.000	0.000	0.000
t = 41:	0.000	0.000	0.001	0.001	0.001	0.000	0.000	0.000
t = 42:	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
•								
•								

$W_{kj} = -\epsilon$, $(k \neq j, \forall k,j)$ where $\epsilon = 0.1$ $(< 1/8)$

(b)

node :	$\mu_0(t)$	$\mu_1(t)$	$\mu_2(t)$	$\mu_3(t)$	$\mu_4(t)$	$\mu_5(t)$	$\mu_6(t)$	$\mu_7(t)$
t = 0:	9.000	9.000	9.000	9.000	9.000	9.000	9.000	9.000
t = 1:	2.700	2.700	2.700	2.700	2.700	2.700	2.700	2.700
t = 2:	0.810	0.810	0.810	0.810	0.810	0.810	0.810	0.810
t = 3:	0.243	0.243	0.243	0.243	0.243	0.243	0.243	0.243
t = 4:	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073
t = 5:	0.022	0.022	0.022	0.022	0.022	0.022	0.022	0.022
t = 6:	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007
t = 7:	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
t = 8:	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
t = 9:	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
•								
•								
•								

$W_{kj} = -\epsilon$, $(k \neq j, \forall k,j)$ where $\epsilon = 0.1$ $(< 1/8)$

(c)

node :	$\mu_0(t)$	$\mu_1(t)$	$\mu_2(t)$	$\mu_3(t)$	$\mu_4(t)$	$\mu_5(t)$	$\mu_6(t)$	$\mu_7(t)$
t = 0:	5.000	4.000	7.000	6.000	8.000	3.000	4.000	2.000
t = 1:	0.971	0.000	3.195	2.063	4.287	0.000	0.000	0.000
t = 2:	0.000	0.000	2.323	1.050	3.533	0.000	0.000	0.000
t = 3:	0.000	0.000	1.782	0.353	3.126	0.000	0.000	0.000
t = 4:	0.000	0.000	1.372	0.000	2.868	0.000	0.000	0.000
t = 5:	0.000	0.000	1.035	0.000	2.701	0.000	0.000	0.000
t = 6:	0.000	0.000	0.717	0.000	2.576	0.000	0.000	0.000
t = 7:	0.000	0.000	0.414	0.000	2.489	0.000	0.000	0.000
t = 8:	0.000	0.000	0.121	0.000	2.439	0.000	0.000	0.000
t = 9:	0.000	0.000	0.000	0.000	2.424	0.000	0.000	0.000

$W'_{kj} = \frac{1}{M+\frac{k}{M}}$, $(k \neq j, \forall k,j)$ where $M = 8$

(d)

node :	$\mu_0(t)$	$\mu_1(t)$	$\mu_2(t)$	$\mu_3(t)$	$\mu_4(t)$	$\mu_5(t)$	$\mu_6(t)$	$\mu_7(t)$
t = 0:	5.000	4.000	8.000	8.000	8.000	3.000	4.000	2.000
t = 1:	0.611	0.000	3.956	3.941	3.927	0.000	0.000	0.000
t = 2:	0.000	0.000	2.947	2.924	2.901	0.000	0.000	0.000
t = 3:	0.000	0.000	2.257	2.225	2.195	0.000	0.000	0.000
t = 4:	0.000	0.000	1.733	1.693	1.655	0.000	0.000	0.000
t = 5:	0.000	0.000	1.336	1.289	1.243	0.000	0.000	0.000
t = 6:	0.000	0.000	1.036	0.980	0.927	0.000	0.000	0.000
t = 7:	0.000	0.000	0.809	0.746	0.685	0.000	0.000	0.000
t = 8:	0.000	0.000	0.640	0.567	0.498	0.000	0.000	0.000
t = 9:	0.000	0.000	0.514	0.431	0.352	0.000	0.000	0.000
t = 10:	0.000	0.000	0.421	0.327	0.239	0.000	0.000	0.000
t = 11:	0.000	0.000	0.353	0.248	0.149	0.000	0.000	0.000
t = 12:	0.000	0.000	0.306	0.188	0.076	0.000	0.000	0.000
t = 13:	0.000	0.000	0.275	0.142	0.017	0.000	0.000	0.000
t = 14:	0.000	0.000	0.256	0.106	0.000	0.000	0.000	0.000
t = 15:	0.000	0.000	0.243	0.075	0.000	0.000	0.000	0.000
t = 16:	0.000	0.000	0.234	0.046	0.000	0.000	0.000	0.000
t = 17:	0.000	0.000	0.229	0.018	0.000	0.000	0.000	0.000
t = 18:	0.000	0.000	0.227	0.000	0.000	0.000	0.000	0.000

$W'_{kj} = \frac{1}{M+\frac{k}{M}}$, $(k \neq j, \forall k,j)$ where $M = 8$

(e)

node :	$\mu_0(t)$	$\mu_1(t)$	$\mu_2(t)$	$\mu_3(t)$	$\mu_4(t)$	$\mu_5(t)$	$\mu_6(t)$	$\mu_7(t)$
t = 0:	9.000	9.000	9.000	9.000	9.000	9.000	9.000	9.000
t = 1:	1.582	1.565	1.548	1.531	1.516	1.500	1.485	1.471
t = 2:	0.331	0.309	0.287	0.266	0.245	0.226	0.207	0.188
t = 3:	0.127	0.101	0.076	0.052	0.029	0.006	0.000	0.000
t = 4:	0.095	0.066	0.037	0.010	0.000	0.000	0.000	0.000
t = 5:	0.081	0.048	0.016	0.000	0.000	0.000	0.000	0.000
t = 6:	0.073	0.036	0.000	0.000	0.000	0.000	0.000	0.000
t = 7:	0.069	0.027	0.000	0.000	0.000	0.000	0.000	0.000
t = 8:	0.065	0.018	0.000	0.000	0.000	0.000	0.000	0.000
t = 9:	0.063	0.010	0.000	0.000	0.000	0.000	0.000	0.000
t = 10:	0.062	0.002	0.000	0.000	0.000	0.000	0.000	0.000
t = 11:	0.062	0.000	0.000	0.000	0.000	0.000	0.000	0.000

$W'_{kj} = \frac{1}{M+\frac{k}{M}}$, $(k \neq j, \forall k,j)$ where $M = 8$

(f)

Fig. 12. (a)–(c) illustrate the behaviour of the (classical) Maxnet with the weight defined in Eq. (6). Note that the canceling effect described in the text appears in (b) and (c). (d)–(f) illustrate the behaviour of the (modified) Maxnet with the weight defined in Eq. (6′). There is no canceling effect to appear.

follows: Input values are applied at time zero through input nodes on the bottom of Fig. 11. This initializes output nodes for each node at time zero ($\mu_j(0)$) to the input values:

$$\mu_j(0) = x_j , \quad j = 0,1, \ldots, M-1. \tag{5}$$

Since the weight W_{kj} from node k to node j is defined as follows:

$$W_{kj} = \begin{cases} 1, & k = j \\ -\epsilon, & k \neq j, \epsilon < \dfrac{1}{M}, \end{cases} \quad k,j = 0,1, \ldots, M-1. \tag{6}$$

Each node inhibits all other nodes with a value equal to the node's output multiplied by a small negative weight ($-\epsilon$). Each node also feeds back to itself with unity gain. The Maxnet then iterates to find maximum via the following equation:

$$\begin{aligned} \mu_j(t+1) &= f_T \left[\sum_{k=0}^{M-1} W_{kj}\mu_k(t) \right] \\ &= f_T \left[\mu_j(t) - \epsilon \sum_{\substack{k=0 \\ k\neq j}}^{M-1} \mu_k(t) \right], \\ & \qquad j = 0,1, \ldots, M-1 . \end{aligned} \tag{7}$$

Here, f_T denotes the threshold logic as described below:

$$f_T(\alpha) = \begin{cases} \alpha, & \text{if } \alpha > 0 , \\ 0, & \text{if } \alpha \leq 0 . \end{cases} \tag{8}$$

After convergence, only the output node corresponding to the maximum input will have a non-zero value. This value will, in general, be less than the original (time zero) value of that node. The output values of the Maxnet are thus simply the node output values after convergence:

$$y_j = \mu_j(\infty) , \quad j = 0,1, \ldots, M-1. \tag{9}$$

In Lippmann et al's report, the Maxnet will always converge, which had been proved, when the inhibitory weight W_{kj} $(k \neq j)$ is given by $\epsilon < \frac{1}{M}$, as described in Eq. (6). The convergence means that the node outputs stop changing, and only the output of one node corresponding to the maximum input is positive.

Figures 12(a)-(c) illustrate the behaviour of the Maxnet. Although the convergence is slower when the peak value across nodes in the Maxnet is less distinct, the average number of iterations also does not grow strongly with the number of nodes in the Maxnet [13].

In Figs. 12(b) and (c), if, after convergence, those output nodes corresponding to the identical maximum inputs have no non-zero values, then selection of the maximum over M x_j values fails. The reason is that all the weights W_{kj} $(k \neq j)$ are set to the same value, which results in *canceling effect*. In next sub-section, we will present the modification of weights (W_{kj})'s so that the problem of cancelling effect disappears. The Maxnet introduced in this sub-section is thus called the *classical Maxnet*.

2. Modification of interconnections

To avoid the canceling effect and make Maxnet still convergent, two conditions for designing the modified weights (W'_{kj})'s should be considered: (1) $| W'_{kj} | \leq \frac{1}{M}$ $(k \neq j)$ and (2) $W'_{kj} \neq W'_{lj}$ $(k \neq j,\ l \neq j,\ k \neq l)$. Hence, Eq. (6) can be replaced as:

$$W'_{kj} = \begin{cases} 1 , & k = j , \\ -\dfrac{1}{M + \dfrac{k}{M}} , & k \neq j , \end{cases} \quad k,j = 0,1, \ldots, M-1. \tag{6'}$$

From this equation, we can observe two facts: (1) $W'_{ki} = W'_{kj}$, $(k \neq i,\ k \neq j,\ i \neq j)$, and (2) $| W'_{ki} | > | W'_{lj} |$, $(k \neq i,\ l \neq j,\ k < l,\ i = j$ or $i \neq j)$. With the modified weights, even though more than one maximum is initiated, selection of the maximum over the M x_j values will always succeed. The proof outlined below guarantees that there is only one positive output corresponding to one of the identical maximum inputs. In this proof, we should prove its distinctive property and the unique maximum appeared at the final iteration.

First, we prove the distinctive property defined that two ouputs $\mu_i(t)$ and $\mu_j(t)$ will become zero or distinctive eventually. Let $\mu_i(t)$ and $\mu_j(t)$ be the identical nonzero values initially given, since the identical situation is our main concern. Now we consider $\mu_i(t+1)$ and $\mu_j(t+1)$,

$$\begin{aligned} \mu_i(t+1) &= f_T \left[\sum_{k=0}^{M-1} W'_{ki}\mu_k(t) \right] \\ &= f_T \left[\mu_i(t) + W'_{ji}\mu_j(t) + \sum_{\substack{k=0 \\ k\neq i \\ k\neq j}}^{M-1} W'_{ki}\mu_k(t) \right], \end{aligned}$$

$$\begin{aligned} \mu_j(t+1) &= f_T \left[\sum_{k=0}^{M-1} W'_{kj}\mu_k(t) \right] \\ &= f_T \left[\mu_j(t) + W'_{ij}\mu_i(t) + \sum_{\substack{k=0 \\ k\neq i \\ k\neq j}}^{M-1} W'_{kj}\mu_k(t) \right]. \end{aligned}$$

Since the third term in the bracket of the equations $\mu_i(t+1)$ and $\mu_j(t+1)$ are the same due to $W'_{ki} = W'_{kj}$ $(k \neq i,\ k \neq j,\ i \neq j)$ depending on the fact (1) observed previously, they can be reduced as follows.

$\mu_i(t+1) = f_T[V_1+V]$ and $\mu_j(t+1) = f_T[V_2+V]$,

here

$$V_1 = \mu_i(t) + W'_{ji}\mu_j(t),\ V_2 = \mu_j(t) + W'_{ij}\mu_i(t),$$

$$\text{and}\quad V = \sum_{\substack{k=0\\k\neq i\\k\neq j}}^{M-1} W'_{ki}\mu_k(t) = \sum_{\substack{k=0\\k\neq i\\k\neq j}}^{M-1} W'_{kj}\mu_k(t).$$

According to the values of V_1, V_2, and V, there are three cases to be generated and discussed.

Case 1. $\mu_i(t+1) \neq \mu_j(t+1)$. Since $\mu_i(t) = \mu_j(t)$ defined previously are nonzero values which can be maximal or not and the values of V_1+V and V_2+V are positive then we can compare the terms of V_1 and V_2.

$$|W'_{ji}| > |W'_{ij}|, \quad \text{if } i > j,$$

$$|W'_{ji}| < |W'_{ij}|, \quad \text{if } i < j,$$

Moreover, (W'_{kj})'s, $k \neq j$, are negative. Therefore, we have the following inequalities:

$$\mu_i(t+1) < \mu_j(t+1), \quad \text{if } i > j,$$

$$\mu_i(t+1) > \mu_j(t+1), \quad \text{if } i < j.$$

Case 2. $\mu_i(t+1) = \mu_j(t+1) = 0$. The case happens in the situation when both V_1+V and V_2+V become non-positive values. In this case, we first consider that there exists at least a maximum for $\mu_i(t)$ and $\mu_j(t)$. Assuming here that $\mu_i(t)$ is a maximum, then we have a set of inequalities that

$\mu_i(t) > \mu_k(t)$, $\forall\ k,\ k \neq i$ and $k \neq j$,

$\mu_i(t) \geq \mu_j(t)$, the equality holds when they are both the maximum.

Therefore, we have

$$(M-1)\mu_i(t) > \sum_{\substack{k=0\\k\neq i}}^{M-1} \mu_k(t), \text{ and then}$$

$$\mu_i(t) > \sum_{\substack{k=0\\k\neq i}}^{M-1} \frac{1}{M-1}\mu_k(t).$$

Since $\dfrac{1}{M-1} > \dfrac{1}{M+\dfrac{k}{M}}$, $\forall\ k$, then

$$\mu_i(t) > \sum_{\substack{k=0\\k\neq i}}^{M-1} \frac{1}{M+\dfrac{k}{M}}\mu_k(t), \text{ and by Eq. (6') then}$$

$$\mu_i(t) + \sum_{\substack{k=0\\k\neq i}}^{M-1} W'_{ki}\mu_k(t) > 0.$$

Hence, we obtain an obvious conclusion, i.e., when $\mu_i(t)$ is a maximum, it is impossible to conduct the result of $\mu_i(t+1)$ to zero.

If $\mu_i(t)$ and $\mu_j(t)$ are the identical maximum, in accordance with the above derivations, they will not become zero at $t+1$; and by the Case 1, they will be mutually distinctive at $t+1$. Furthermore by the transfer function f_T [Eq. (8)], all the values of μ_i's at $t+1$ should be reduced. Hence, for Case 2, if $\mu_i(t)$ and $\mu_j(t)$ are non-maximum, they will become zero at $t+1$ or before the unique maximum is obtained.

Since outputs occurring in this case become zero before the termination of iterations, they will not affect the following process.

Case 3. $\mu_i(t+1) = \mu_j(t+1) \neq 0$. This case happens when $V_1 = V_2$, i.e., when the following relationship between $\mu_i(t)$ and $\mu_j(t)$ holds.

$$\mu_i(t) = [(1-W'_{ji}) / (1-W'_{ij})]\mu_j(t).$$

However, in the next iteration, say, at $t+2$, the results of $\mu_i(t+2)$ and $\mu_j(t+2)$ will still be mutually distinct by Case 1 or will be zero values by Case 2. Note that $\mu_i(t)$ and $\mu_j(t)$ are non-maximal nonzero values. Accordingly, we complete the first part of the proof, showing the distinctive property.

Now we continuously prove that the unique maximum can be obtained after the termination of iterations. Assume that both x_1 and x_j are given maximal inputs. Hence, we have $\mu_i(0) = x_i$ and $\mu_j(0) = x_j$ initially. First, by Case 1, we have $\mu_i(1) \neq \mu_j(1)$. And according to the derivations and considerations of Cases 2 and 3, we will have a unique maximum appear eventually on μ_i or μ_j depending on $i < j$ or $i > j$, respectively. If $x_i > x_j > 0$, and x_i is the given maximum input, we similarly have $\mu_i(0) = x_i$ and $\mu_j(0) = x_j$ initially. By Cases 2 and 3, we also will have a unique maximum appear eventually on μ_i.

In general, systems having organized asymmetry can exhibit oscillation and chaos [11]. Since the distance from equilibrium and nonlinearity may both be sources of order capable of driving a system to an ordered configuration. It possesses a highly nontrivial connection between the order configuration and the stability [15]. The weights designed in our modified Maxnet are not symmetric but are subtly changed *in order*. Accordingly, they form an ordered configuration as the properties shown in Cases 1 and 2 to support the convergency. These complete the whole proof.

From the above proof, we observe a property that after convergence, the final positive value will always appear on the node addressed the least number among those

nodes initiated with identical maximum values. (From a physical standpoint, such design of the modified weights in Eq. (6′) can be regarded as a kind of mutually different length among neuronal dendrites or axons. If further discussions proceed from the psychological viewpoints, the so-called vigilance factor which can check on cluster allocation should be involved.)

Figures 12(d)-(f) illustrate the behaviour of Maxnet with the modified weights. We call this net the *modified Maxnet*, which will facilitate the construction of the B-net (for local minimum) and G-net (for local maximum) described in the next sub-section.

3. B-net and G-net

Figure 13 shows a structure of B-net, which is conceptually based on the function of the *bipolar* cell in the retina shown in Fig. 2. The main function introduced here is to collect all the outputs of horizontal cells, i.e., $d_k^0(x,y)$ and $d_k^1(x,y)$, $k = 0,1, \ldots, M-1$, for the pattern pixel (x,y) and to decide the *minimum* value of this collection. In this structure, V_{max} is a specified value which is larger than any value of $d_k^0(x,y)$ or $d_k^1(x,y)$ for all k. The original input value is *subtracted* from V_{max} so that the original minimum becomes the maximum, and vice versa. Therefore, after performing these subtracted inputs on the modified Maxnet, the final non-zero output node, i.e., y_{max}, will *mark* the original minimum input.

Because our concern is to obtain the *distance* between any pixels in patterns S and background $\overline{S}$, the output $D(x,y)$ shall show the original minimum distance value, i.e.

$$D(x,y) = \min \{ d_k^0(x,y) \text{ or } d_k^1(x,y) \mid k = 0,1, \ldots, M-1 \} . \tag{10}$$

This equation can be replaced as:

$$D(x,y) = g_T \left[\sum_{i=0}^{2M-1} z_i \right] , \tag{11}$$

$$z_i = sw_{y_i} [d_k^p(x,y)] , \tag{12}$$

where $i = 2k+p$, $k = 0,1, \ldots, M-1$, $p = 0$ or 1. Here g_T denotes the threshold logic as defined in Eq. (8). And the *switching function* (sw) is defined as below:

$$sw_\beta [\alpha] = \begin{cases} \alpha, & \text{if } \beta > 0 , \\ 0, & \text{if } \beta = 0 . \end{cases} \tag{13}$$

Since the modified Maxnet will always converge, after convergence, only one output node will reveal positive value. And only the corresponding z_i will have a corresponding input value $d_k^P(x,y)$, whereas the other z_i values are all zeros. Hence, $D(x,y)$ is eventually reduced to

$$D(x,y) = g_T [d_k^P(x,y)] = d_p^P(x,y), \tag{14}$$

where node y_{2k+p} reveals positive value y_{max}.

All the above equations are easily mapped into the B-net structure shown in Fig. 13. Fig. 14 shows the processing result of these B-nets using the outputs of horizon-

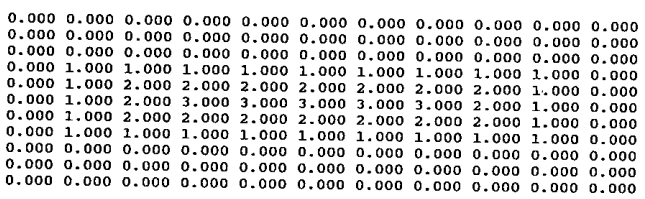


Fig. 14. The result of distance computation is obtained by using the proposed B-nets. The inputs are the outputs of horizontal cells in Fig. 10.

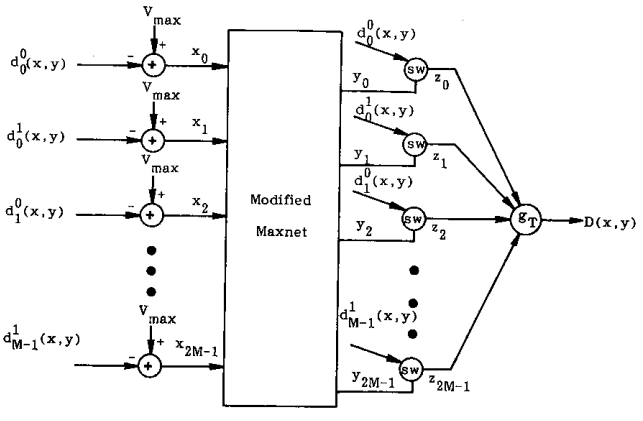


Fig. 13. The structure of B-net. The main function of B-net is to collect all the outputs of horizontal cells and to decide the minimum value of this collection. See text for details.

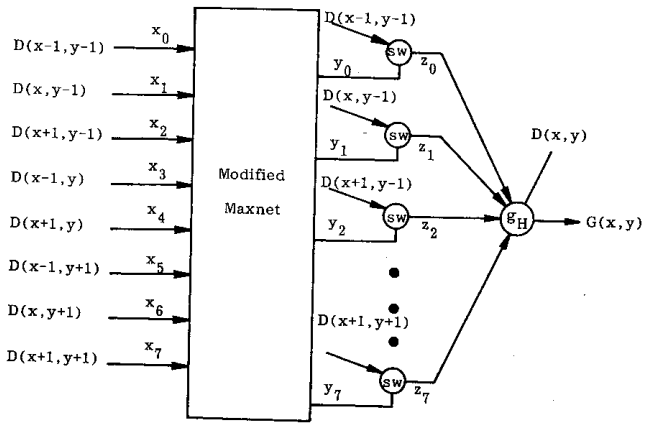


Fig. 15. The structure of G-net. The main function of G-net is to collect the distances D's of neighbors of pixel (x,y) and to decide whether or not the maximum of this collection is larger than $D(x,y)$.

tal cells in Fig. 10 as inputs.

Analogously, the structure of the G-net shown in Fig. 15 is conceptually based on the function of the *ganglion* cell in the retina shown in Fig. 2. Because the visual receptive field is arranged as a square array, there merely exist *eight* neighbors for each pixel (x,y). They are denoted by $(x-1,y-1)$, $(x,y-1)$, $(x+1,y-1)$, $(x-1,y)$, $(x+1,y)$, $(x-1,y+1)$, $(x,y+1)$, and $(x+1, y+1)$. The main function of G-net is to collect the distances D's of the neighbors of pixel (x,y) and to decide whether or not the *maximum* of this collection is larger than $D(x,y)$. If the maximum is larger than $D(x,y)$, then $D(x,y)$ is not a local maximum and $G(x,y)$ has no response (i.e. $G(x,y)=0$.) Otherwise, the $D(x,y)$ is a local maximum and $G(x,y)$ has response (i.e. $G(x,y)=1$.)

Basically, the structure of the G-net is similar to that of the B-net, except that the *adders* in the B-net are removed and the threshold logic g_T in the B-net is replaced by a hard limit g_H. The g_H is defined below:

$$g_H(\alpha) = \begin{cases} 1, & \text{if } \alpha \le D(x,y), \\ 0, & \text{if } \alpha > D(x,y). \end{cases} \quad (15)$$

Figure 16 shows the processing result of these G-nets using the outputs of the B-nets in Fig. 14 as inputs. The result shown in Fig. 16 is the so-called MA of the given picture shown in Fig. 1a.

4. Summary of the MATNET

Up to now, all the details of the MATNET have been described in Section 2 (*horizontal cells* corresponding to *distance computations*) and in this section (B-net and G-net corresponding to the calculation of *local minimum* and *local maximum*, respectively). The ensemble of the MATNET is thus completely drawn in Fig. 17. The right side of this figure denotes the corresponding anatomic parts in the retina shown in Fig. 2 and the left side denotes the basic operations in the MAT.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Fig. 16. The processing result is obtained by using the proposed G-net. It is the MA of the given picture shown in Fig. 1a.

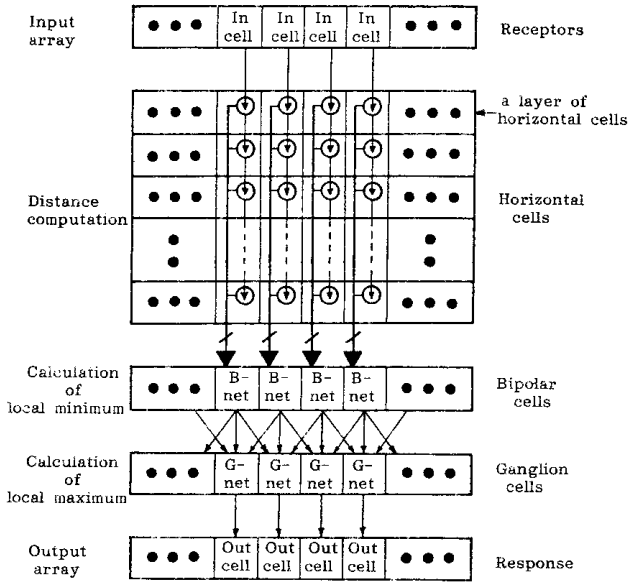


Fig. 17. The ensemble of the presented MATNET. The corresponding anatomic parts in the retina shown in Fig. 2 and the basic operations in MAT are also noted in this figure.

Several MAs of capital letters of the English alphabet, which is used in Govindan and Shivaprasad's thinning study [9], obtained by the MATNET are shown in Fig. 18, where "–" denotes an unresponsive point of the corresponding pattern pixel and "*" denotes a responsive point of the MA.

Although the MA of a picture does not have the connecting property, it plays an important role for designing a generalization process of intuitive pattern recognition in human visual perception [2,6]. Moreover, if the MATNET is designed in analog hardware architecture, the stimulation-response will act fast. This is because the neural networks possess the emergent collective computational ability [10]. Since such MATNET is active all the time, if the receptors of the MATNET are fixedly stimulated by a stimulus pattern, then after (fast) convergence, the response of the MATNET will always display a stable MA of the given pattern. Of course, if the input pattern is not fixed, the MA response will be unstable, due to the time delay resulting from the convergence time and the variation of the stimulus pattern.

CONCLUSIONS

In this paper, we present a neural network called MATNET to implement the medial axis transformation which can be applied to a novel neural network called I-net to perform the intuitive pattern recognition in human visual perception [2,6]. The structure of the MATNET is analogous to that of the retina which lies in the interior of the eyeball. It consists of five layers, namely, receptors, horizontal cells (for distance computation), bipolar

cells (for calculation of local minimum), ganglion cells (for calculation of local maximum), and response. The bipolar and ganglion cells are implemented by the so-called B-net and G-net, respectively, which are concerned with the maximal neural network (Maxnet). The properties of Maxnet have also been discussed. After the ordered modification of interconnection weights in the neural network, the problem of canceling effect is overcome. Experiments have shown that the MATNET can reasonably perform the task of medial axis transformation.

ACKNOWLEDGEMENT

This work was supported by the National Science Council of the Republic of China, Research Contract No. NSC79-0408-E007-12.

NOMENCLATURE

$D(x,y)$	the minimum distance value
$d_{(4)}(p,q)$	city block distance between two pixels p and q
$d_{(8)}(p,q)$	chessboard distance between two pixels p and q
$d_{(e)}(p,q)$	euclidean distance between two pixels p and q
$d_k(p,q)$	distance between p and q on the kth orientation plane computed by n_k times V_k
f_T	a threshold logic
g_H	a hard limit
g_T	a threshold logic
$H_k(x,y)$	a horizontal cell
$l_k(x,y)$	the link of the corresponding horizontal cell $H_k(x,y)$

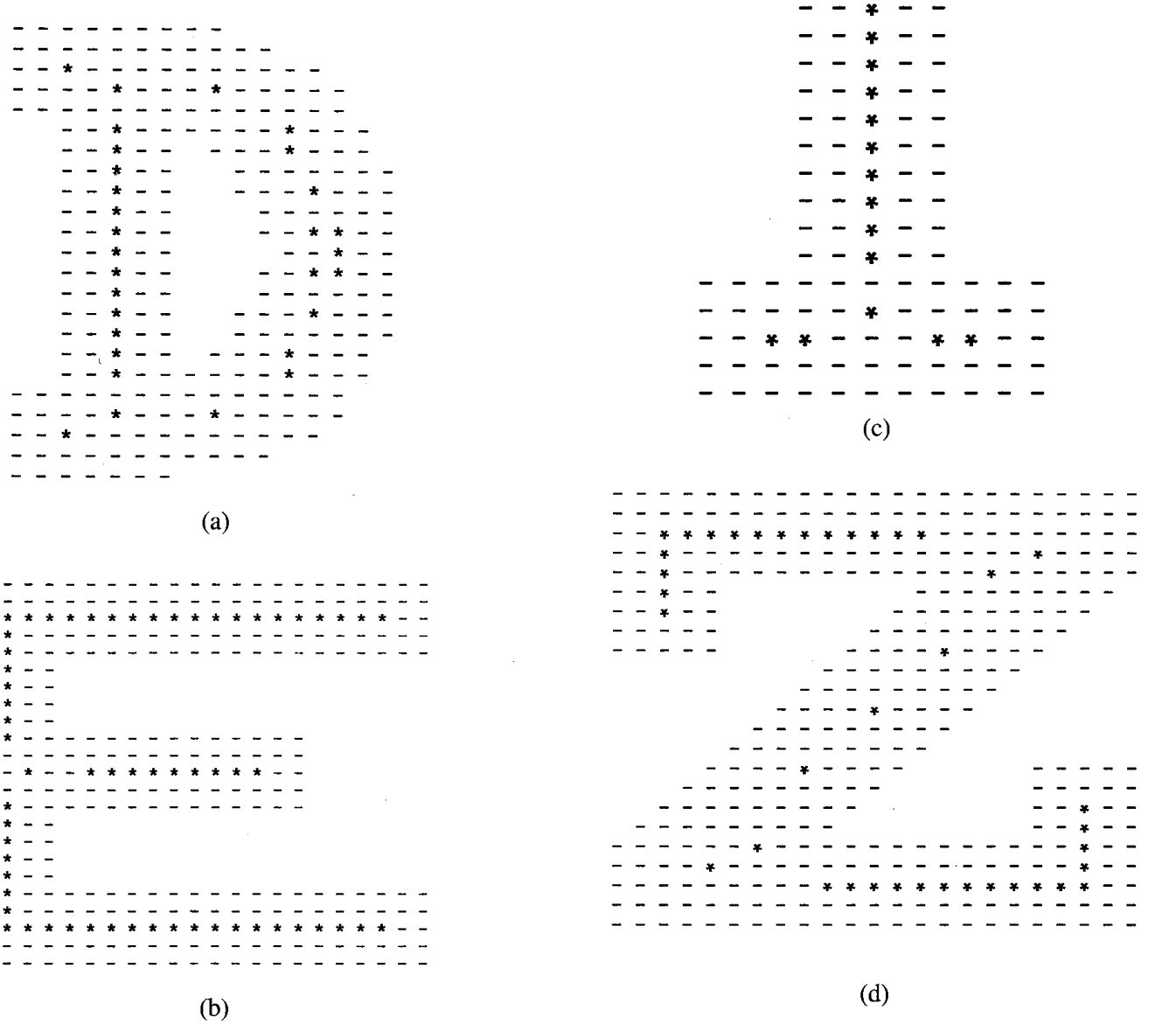


Fig. 18. The MAs of four capital letters of the English alphabet are obtained by the MATNET and denoted by "*"s.

n_k the number of the points between p and q falling on a linear line of the kth orientation plane

V_k a cell charging the voltage which depends on the slope θ_k

w_{kj} the weight from node k to j

Greek symbols

θ_k the slope of the straight line segment between p and q

μ_j the jth output node of the Maxnet

REFERENCES

1. Blum, H., "A Transformation for Extracting New Descriptors of Shape," in *Models for the Perception of Speech and Visual Form,* W. Whaten-Dunn, Ed., MIT Press, Cambridge, MA, pp. 362-380, Proc. of Meeting (1964).
2. Chen, Y.S., *On the Study of Line-Image Processings via the Computer Algorithm and Human Visual Perception,"* Ph.D. Thesis, National Tsing Hua University, Hsinchu, Taiwan, R.O.C. (1989).
3. Chen, Y.S. and W.H. Hsu, "A Systematic Approach for Designing 2-Subcycle and Pesudo 1-Subcycle Parallel Thinning Algorithms," *Pattern Recognit.,* Vol. 22, pp. 267-282 (1989).
4. Chen, Y.S. and W.H. Hsu, "A 1-Subcycle Parallel Thinning Algorithm for Producing Perfect 8-Curves and Obtaining Isotropic Skeleton of the L-Shape Pattern," *IEEE Proc. of Computer Vision and Pattern Recognition,* San Diego, California, USA, pp. 208-215 (1989).
5. Chen, Y.S. and W.H. Hsu, "An Interpretive Model of Line Continuation in Human Visual Perception," *Pattern Recognit.,* Vol. 22, pp. 619-639 (1989).
6. Chen, Y.S., J.L. Lin and W.H. Hsu, "Implementation on the Concept of Intuitive Human Pattern Recognition," *Proc. of the 3rd International Conference on Visual Search,* Natingham, UK (1992).
7. Chu, Y.K. and C.Y. Suen, "An Alternate Smoothing and Stripping Algorithm for Thinning Digital Binary Patterns," *Signal Process.,* Vol. 11, pp. 207-222 (1986).
8. Davies, E.R. and A.P.N. Plummer, "Thinning Algorithms: A Critique and a New Methodology," *Pattern Recognit.,* Vol. 14, pp. 53-63 (1981).
9. Govindan, V.K. and A.P. Shivaprasad, "A Pattern Adaptive Thinning Algorithm," *Pattern Recognit.,* Vol. 20, pp. 623-637 (1987).
10. Hopfield, J.J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. Natl. Acad. Sci. USA,* Vol. 79, pp. 2554-2558 (1982).
11. Hopfield, J.J. and D.W. Tank, "Computing with Neural Circuits: A Model," *Science,* Vol. 233, pp. 625-633 (1986).
12. Kohonen, T., *Self-Organization and Associative Memory,* 2nd edition, Springer-Verlag, New York (1984).
13. Lippmann, R.P., B. Gold and M.L. Malpass, "A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification," *MIT Lincoln Laboratory Technical Report* TR-769 (1987).
14. Morgan, C.T. and R.A. King, *Introduction to Psychology,* 3rd edition, McGraw-Hill, New York (1966).
15. Nicolis, G. and I. Prigogine, *Self-Organization in Nonequilibrium Systems,* John Wiley & Sons, New York (1977).
16. Rosenfeld, A. and A.C. Kak, *Digital Picture Processing,* 2nd edition, Vol. 2, Academic Press, New York (1982).
17. Smith, R.W., "Computer Processing of Line Images: A Survey," *Pattern Recognit.,* Vol. 20, pp. 7-15 (1987).
18. Walters, D., "Selection of Image Primitives for General-Purpose Visual Processing," *Comput. Vision Graphics Image Process,* Vol. 37, pp. 261-298 (1987).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: Dec. 12, 1991
Revision Received: July 20, 1992
and Accepted: Sep. 30, 1992