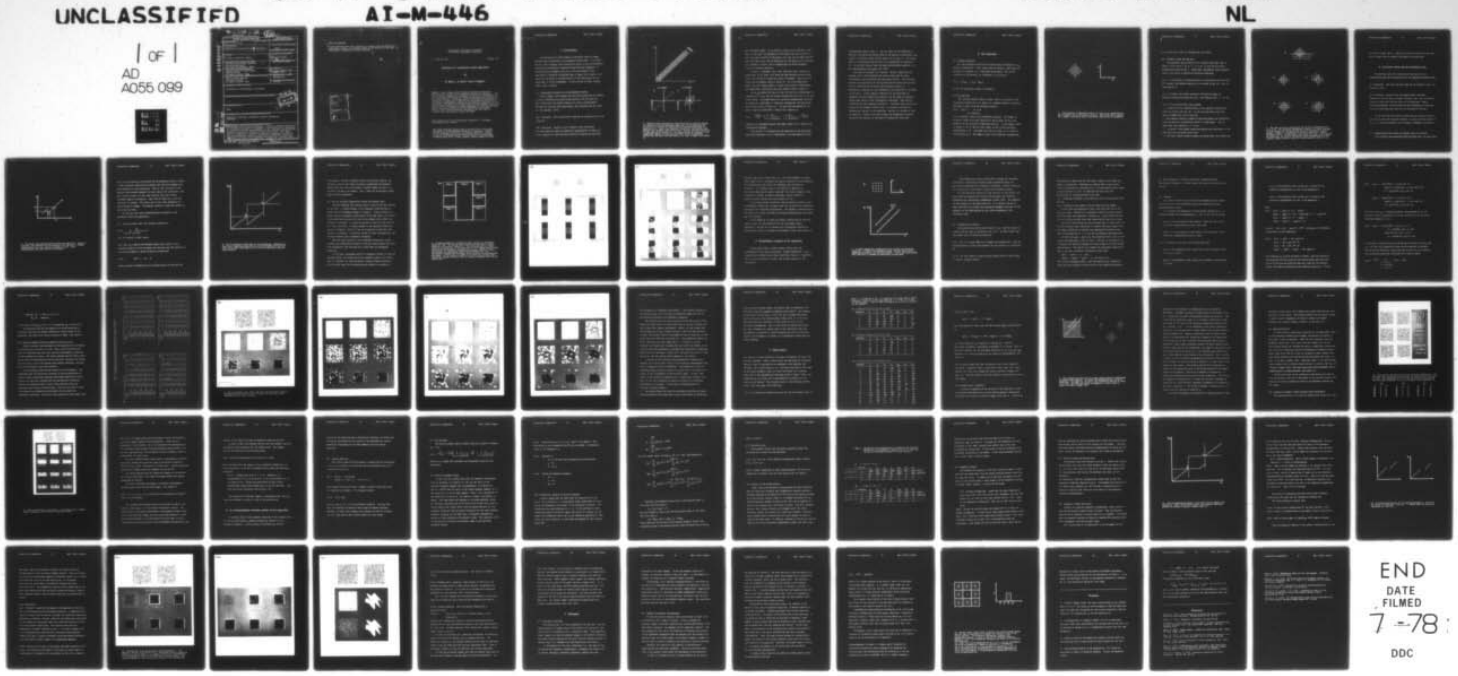
AD-A055 099 MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTE--ETC F/G 20/6
ANALYSIS OF A COOPERATIVE STEREO ALGORITHM,(U)
OCT 77 D MARR, G PALM, T POGGIO N00014-75-C-0643

UNCLASSIFIED AI-M-446 NL

| OF |
AD
A055 099



END
DATE
FILMED
7-78
DDC

FOR FURTHER TRANSLATION

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AI-M-446 AIM 446	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Analysis of a Cooperative Stereo Algorithm	5. TYPE OF REPORT & PERIOD COVERED: MEMO
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): D. Marr, G. Palm & T. Poggio	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0643
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, Massachusetts 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Blvd, Arlington, Virginia 22209	12. REPORT DATE: Oct 77 October 1977
	13. NUMBER OF PAGES: 12 66 P. 65
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, Virginia 22217	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DDC
RETRIEVED
JUN 14 1978
F

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Cooperative Algorithms, convergence, parallel computation, stereopsis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Marr & Poggio (1976) recently described a cooperative algorithm that solves the correspondence problem for steriopsis. This article uses a probabilistic technique to analyze the convergence of that algorithm, and derives the conditions governing the stability of the solution state. The actual results of applying the algorithm to random-dot steriograms are compared with the probabilistic analysis. A satisfactory mathematical analysis of the asymptotic behavior of the algorithm is possible for a suitable choice of the parameter values and loading rules, and again the actual performance

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A055099

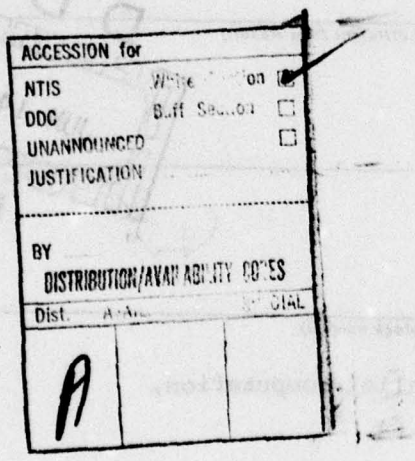
AD No.
DDC FILE COPY

407 483

BLOCK 20 CONCLUDED

of the algorithm under these conditions is compared with the theoretical predictions. Finally, some problems raised by the analysis of this type of "cooperative" algorithm are briefly discussed.

AD A020 668



Massachusetts Institute of Technology
Artificial Intelligence Laboratory

A. I. Memo No. 446 October 1977

Analysis of a cooperative stereo algorithm

by

D. Marr, G. Palm and T. Poggio****

SUMMARY: Marr & Poggio (1976) recently described a cooperative algorithm that solves the correspondence problem for stereopsis. This article uses a probabilistic technique to analyze the convergence of that algorithm, and derives the conditions governing the stability of the solution state. The actual results of applying the algorithm to random-dot stereograms are compared with the probabilistic analysis. A satisfactory mathematical analysis of the asymptotic behaviour of the algorithm is possible for a suitable choice of the parameter values and loading rules, and again the actual performance of the algorithm under these conditions is compared with the theoretical predictions. Finally, some problems raised by the analysis of this type of "cooperative" algorithm are briefly discussed.

**Max-Planck-Institut fur Biologische Kybernetik, 74 Tubingen 1, Spemannstrasse 38, Germany.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.

1 Introduction

The extraction of stereo-disparity information from two images depends upon establishing a correspondence between them. In a recent article, Marr & Poggio (1976) analyzed the nature of the correspondence computation and derived a cooperative algorithm that implements it. Although several examples were given of the performance of the algorithm on random-dot stereograms (Marr & Poggio 1976, figures 3-6), space did not permit a thorough analysis of the fixed points of the algorithm, or of its convergence. In this article, we shall examine these issues in detail.

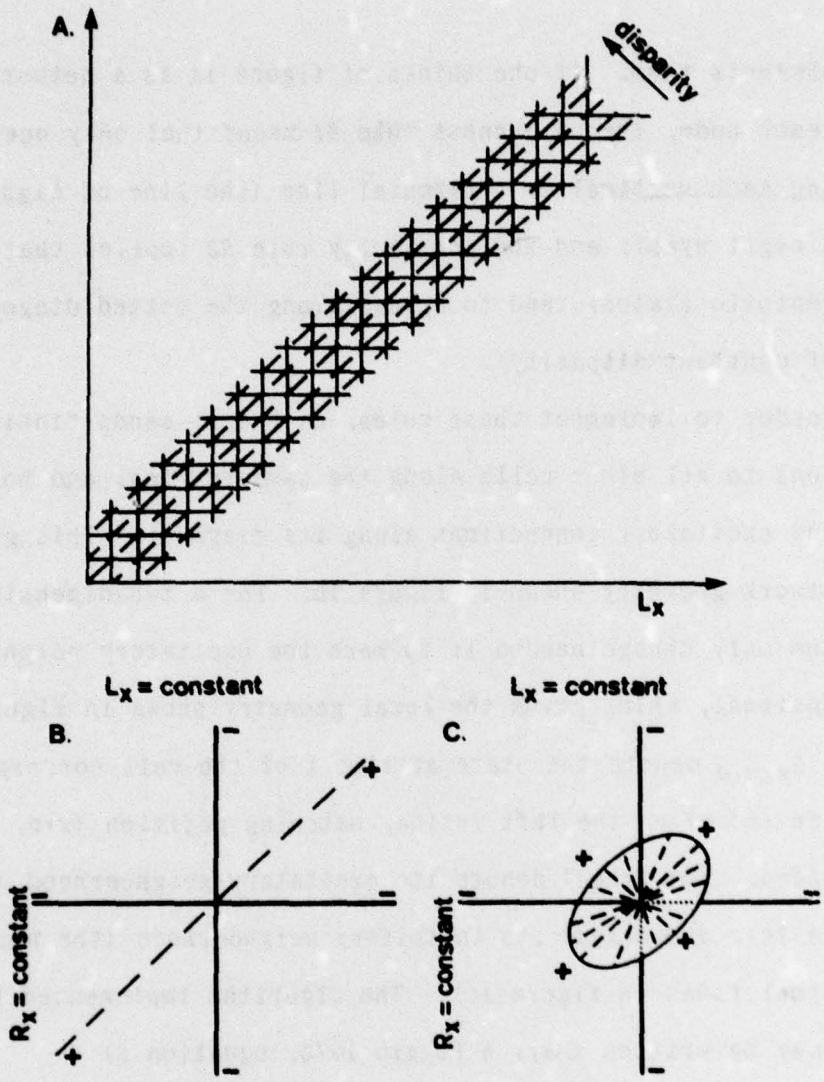
1.1 Computational structure of the correspondence problem

Marr & Poggio (1976) argued that the stereo problem may be reduced to that of matching two primitive descriptions, one from each eye. They showed that the central problem is to find a correspondence between the left and right descriptions, that satisfies the two rules (p. 284 and Marr, 1974):

(R1) Uniqueness: Each item from each image may be assigned at most one disparity.

(R2) Continuity: Disparity varies smoothly almost everywhere.

By constructing an explicit geometrical representation of these two rules (figure 1c), they were able to derive a cooperative algorithm



1. Figure 1a shows the explicit structure of the two rules *R1* and *R2* for the case of a one-dimensional image, and it also represents the structure of a network for implementing the algorithm described by equation *1.1.1.* Solid lines represent "inhibitory" interactions, and dotted lines represent "excitatory" ones. 1b gives the local structure at each node of the network 1a. This algorithm may be extended to two-dimensional images, in which case each node in the corresponding network has the local structure shown in 1c. (Marr & Poggio 1976 figure 2).

that implements them. If one thinks of figure 1a as a network, with a cell at each node, the uniqueness rule *R1* means that only one cell is "on" along each vertical or horizontal line (the line of sight from the left and right eyes); and the continuity rule *R2* implies that solutions (its asymptotic states) tend to spread along the dotted diagonals (lines of constant disparity).

In order to implement these rules, each cell sends "inhibitory" connections to all other cells along the same vertical and horizontal lines, and excitatory connections along its diagonal. This gives the local network geometry shown in figure 1b. For a two-dimensional image, the only change needed is to make the excitatory neighborhood two-dimensional, which gives the local geometry shown in figure 1c.

Let $C_{x,y;d}$ denote the state at time t of the cell corresponding to coordinate (x, y) on the left retina, matching position (x+d, y) on the right retina. Let $S(xyd)$ denote its excitatory neighborhood (the disc in figure 1c), and $O(xyd)$ its inhibitory neighborhood (the horizontal and vertical lines in figure 1c). The algorithm implemented by the network may be written (Marr & Poggio 1976, equation 2)

$$1.1.1 \qquad C^{(t+1)}_{x,y;d} = \sigma\left\{ \sum_{x',y',d' \varepsilon S(x,y,d)} C^{(t)}_{x',y';d'} - \varepsilon \sum_{x',y',d' \varepsilon O(x,y,d)} C^{(t)}_{x',y';d'} + C^{(0)}_{x,y;d} \right\}$$

where σ is a threshold function that takes values 0 or 1, and ε is an "inhibition" constant.

This article is concerned with the properties of the algorithms defined by equation *1.1.1* or, equivalently, with the behavior of the

corresponding network (fig. 1). The two inputs to the algorithm or network, from which the initial state of the network is determined, are usually two matrices whose entries consist of 0's and 1's. The second matrix is constructed from the first by x-translations of regions of it. As we shall discuss later the algorithm defined by equation 1.1.1 has some analogies with games like "life".

The plan of the paper is as follows: Section 2 describes the loading rules, which determine the initial state from the input stereograms, and also defines the algorithm precisely. The relations between the fixed points of the algorithm and the states that satisfy the two conditions *R1* and *R2* are then discussed (section 3). A probabilistic approach to the convergence of the algorithm is outlined in section 4. Actual computer simulations of the algorithm are compared with the probabilistic analysis, and the range of parameter values that yield a "nice" convergence is discussed. Some special situations are also analyzed (section 5). A suitable (and restrictive) choice of the parameter values in eq. 1.1.1 allows a satisfactory mathematical analysis of the algorithm: section 6 is devoted to such an approach. Finally, we briefly discuss the mathematical problems raised by the analysis of this type of "cooperative" algorithm.

2 The algorithm

2.1 Loading conditions

Let the positions on the left and right retinas be denoted by $L_{x,y}$ and $R_{x,y}$ respectively. These arrays take the values 0, indicating the absence of a feature, or 1, indicating the presence. The initial condition of the network, for stereogram L, R is given by

2.1.1 $$C^{0}_{x,y;d} = L_{x,y} \cdot R_{x+d,y}$$

within the appropriate range d of disparity.

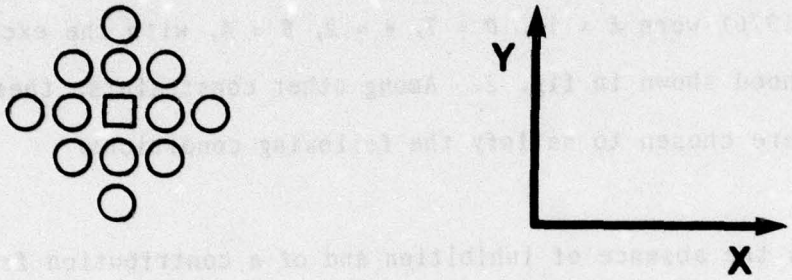
2.2 The algorithm

The relation between states at times t and t+1, is given by the recurrence relation eq *1.1.1*, where σ is a sigmoid function in general, and here is taken to be the threshold function

2.2.1 $\sigma(u) = 1$ if $u \geq \theta$,

$= 0$ otherwise.

ϵ is a constant, known as the "inhibition constant". The number of disparity layers d we shall denote by D, and we shall let N be the diameter of the excitatory neighborhood $S(x,y,d)$. In the example shown in figure 2, $N = 5$, and the total number of cells in an excitatory neighborhood is 13. The number less the cell itself is 12, which we shall denote by E. The number of cells in an inhibitory neighborhood



2. The excitatory neighborhood (figure 1c) used in our implementation has a diameter of 5, and contains 13 cells. The central cell, marked by a square, receives at most 12 excitatory inputs from its neighbours.

of a given cell is $2D - 2$, excluding the cell itself.

2.3 Parameter values and some facts

The parameter values chosen for our original algorithm[1] (Marr & Poggio, 1976) were $E = 12$, $D = 7$, $\epsilon = 2$, $\theta = 4$, with the excitatory neighborhood shown in fig. 2. Among other constraints, these parameter values were chosen to satisfy the following conditions:

2.3.1 in the absence of inhibition and of a contribution from the term C^0, straight line borders should fill in as shown in fig. 3a. This is true when $\theta \leq 4$.

2.3.2 straight line borders between two "filled-in" planes at different disparities should not grow. This requires that $4 - 2\epsilon < \theta$.

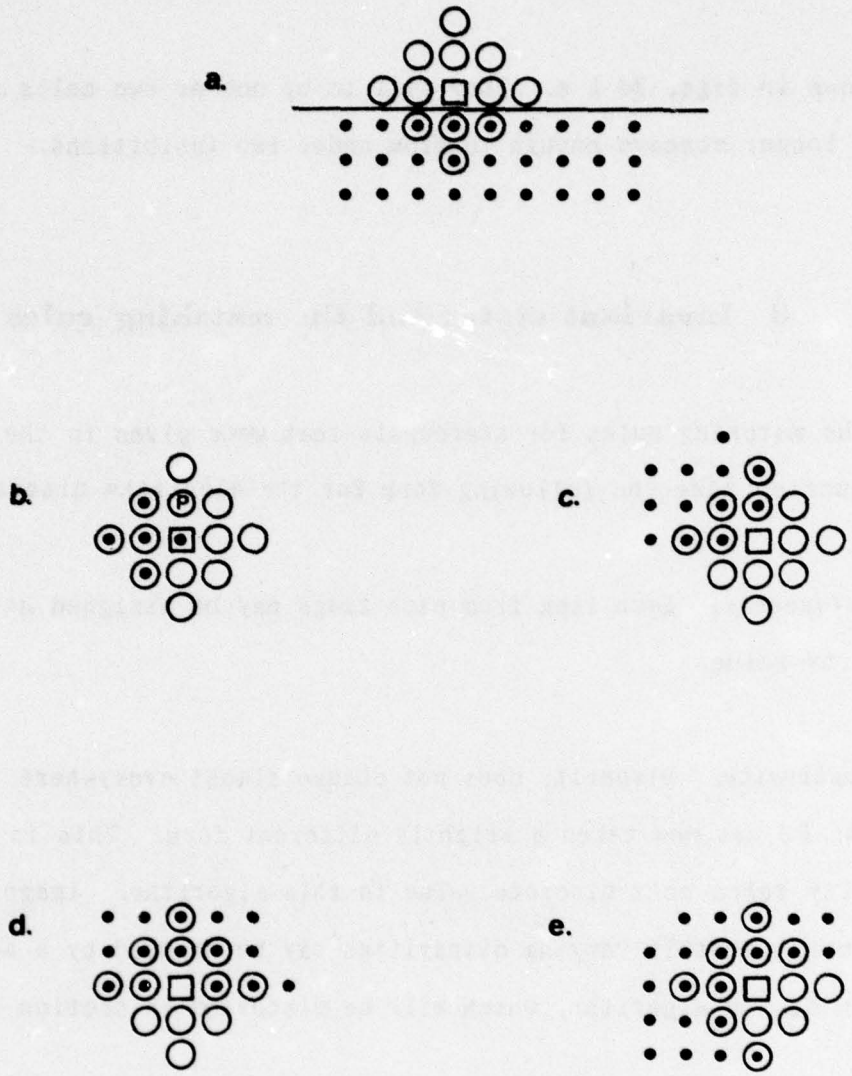
2.3.3 with the particular values chosen:

-- a pattern of five connected points is the smallest configuration that can survive (see fig. 3b). It will not grow unless one other point is added (e.g. at P in fig. 3b).

-- the sharpest convexity capable of surviving against one inhibition, with the help of a contribution from C^0 is a right-angle. Fig. 3c shows that the condition is $6 - \epsilon \geq \theta$.

-- a convex or flat border cannot grow against one inhibition; it can grow only into scattered active cells.

-- the least concave patterns capable of growing under two inhibitions



3. The total excitatory contribution for various configurations of "on" cells. The excitatory neighborhood (figure 2) is shown with open circles, except for the central cell which is indicated by a square because it makes no contribution to the total excitation. With a threshold of 4.0; 3a shows that a flat border will grow in the absence of inhibition, 3b exhibits the smallest stable configuration, 3c the sharpest stable convexity, and 3d & e show concavities that fill in.

are shown in figs. 3d & e. They fill in by one or two cells and then are no longer concave enough to grow under two inhibitions.

3 Invariant states and the matching rules

The matching rules for stereopsis that were given in the introduction take the following form for the algorithm discussed here:

(1) Uniqueness: Each item from each image may be assigned at most one disparity value.

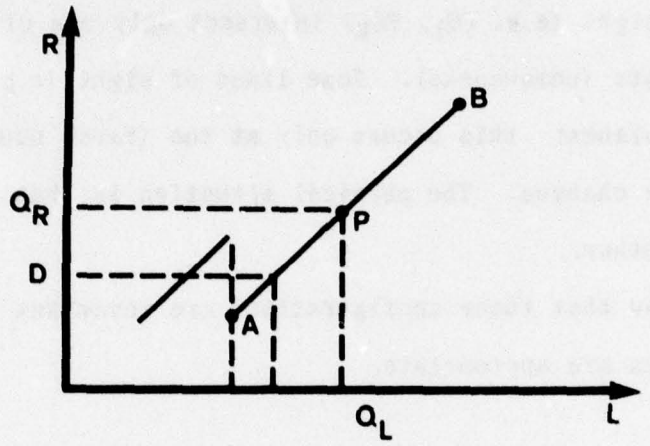
(2) Continuity: Disparity does not change almost everywhere.

Comment: R2 has now taken a slightly different form. This is because disparity takes only discrete value in this algorithm. Images containing smoothly varying disparities may be handled by a modified version of the algorithm, which will be discussed in section 5.

We now show that the states in which these two rules are obeyed are for all practical purposes *invariant*, i.e. they are fixed points of eq. *1.1.1*, and once achieved, do not change in subsequent iterations.

3.1 Configurations that satisfy the matching rules are invariant

The continuity and uniqueness conditions mean that, for each value



4. The solid lines indicate solution planes (*cf* figure 1a). Lines of sight PQ_R, PQ_L intersect solution planes at only one point P, except possibly near the (rare) disparity boundaries like A. Thus configurations that obey rule *R1* are invariant.

of y, a cross-section of the network has the appearance shown in figure 4 (the continuity condition also requires that the active segment has some extension in the y direction). That is, the "on" cells in the network form extended segments like that shown as AB (continuity), and most lines of sight (e.g. PQ_L, PQ_R) intersect only one of these extended segments (uniqueness). Some lines of sight (e.g. to D) may intersect two planes: this occurs only at the (rare) boundaries at which disparity changes. The physical situation is that one surface is obscuring the other.

We show now that these configurations are invariant if the parameter values are appropriate.

(i) Interior points like P are certainly invariant if

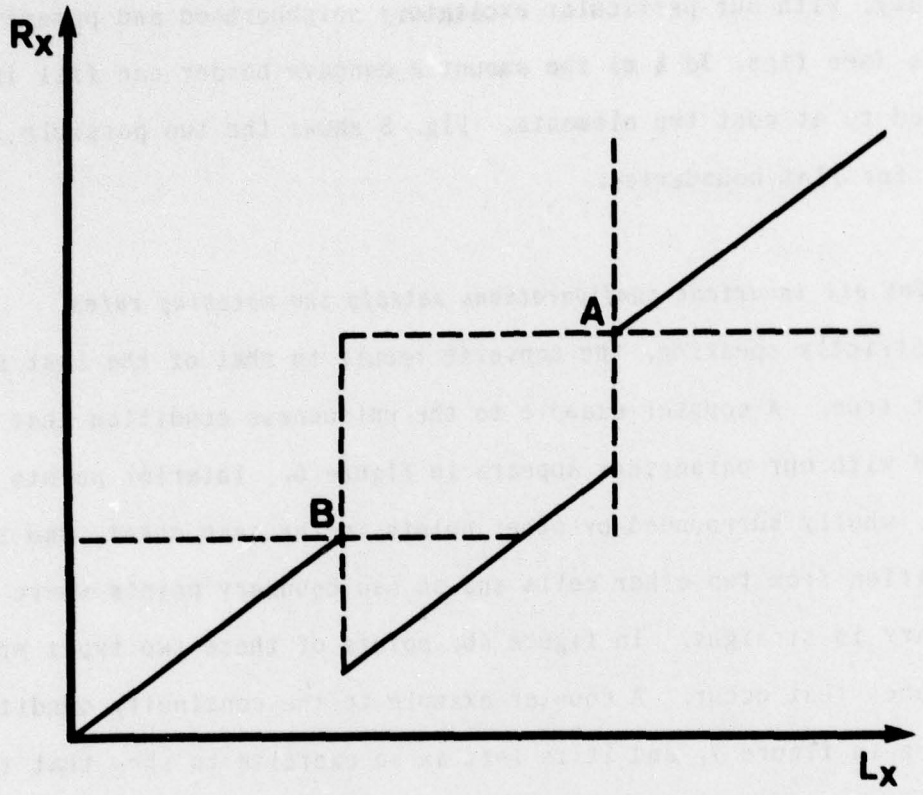
$$3.1.1 \qquad \sum_{x',y',d' \in S(x,y,d)} C^{(t)}_{x',y',d} \geq \theta$$

if P is interior in both x and y.

(ii) Eq. 3.1.1 implies that boundary points like A (fig. 4) on a straight boundary (in the x-y plane) will not grow into the interior of an existing segment at another disparity provided that

$$3.1.2 \qquad \varepsilon/2 + 1 - 2\alpha < \theta$$

Concave pieces of boundaries can in principle grow, but not much for



5. The two possible stable edges for flat boundaries. Depending on the initial conditions, edges can occur that are defined by the line where cells begin receiving one (A) or two (B) inhibitions from the other surface.

two reasons. Firstly, boundaries cannot be everywhere concave, and secondly, with our particular excitatory neighborhood and parameter values (see figs. 3d & e) the amount a concave border can fill in is limited to at most two elements. Fig. 5 shows the two possible stable edges for flat boundaries.

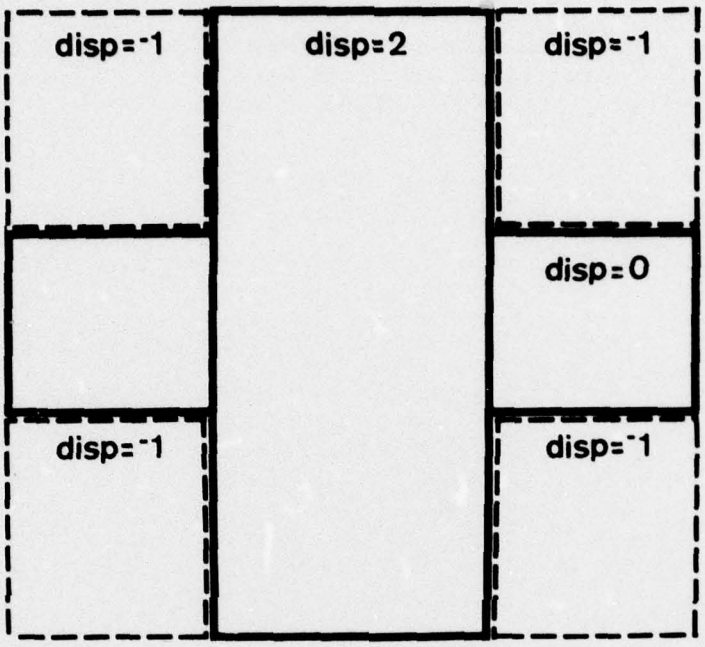
3.2 *Not all invariant configurations sattsfy the matching rules*

Strictly speaking, the converse result to that of the last section is not true. A counter-example to the uniqueness condition that is stable with our parameters appears in figure 6. Interior points of a plane, wholly surrounded by other points in the same sheet, can survive inhibition from two other cells and so can boundary points where the boundary is straight. In figure 6b, points of these two types are the only ones that occur. A counter-example to the continuity condition appears in figure 7, and it is left as an exercise to show that this pattern is invariant. In practice, neither of these configurations can actually develop from a random-dot stereogram.

When the input consists of two stereograms portraying a single surface, the probabilistic analysis of the next section shows that with high probability, the solutions will in fact obey the uniqueness condition.

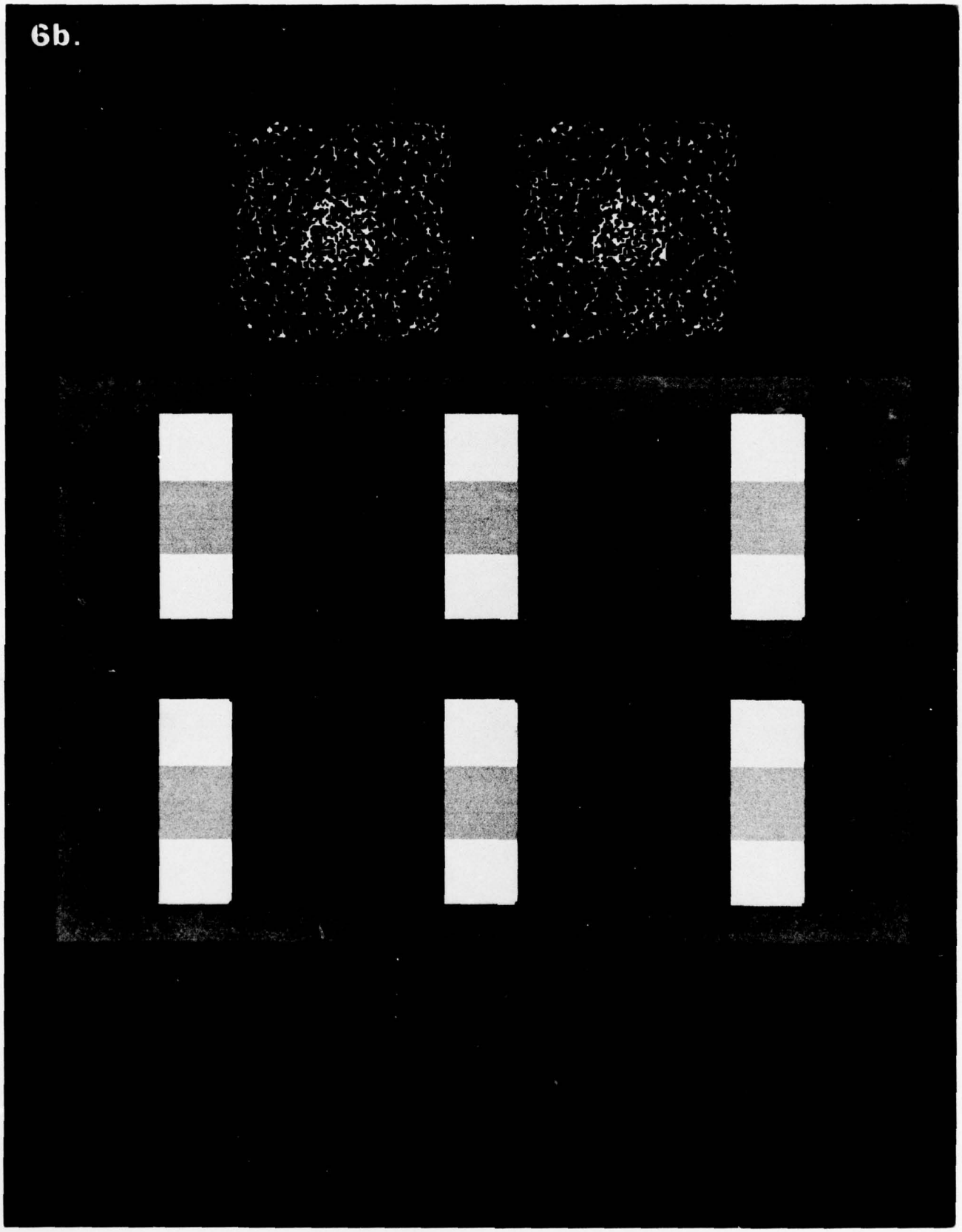
If the input stereograms portray a transparent surface in front of another surface, the algorithm with our parameter values will usually fail to represent the input accurately, tending instead to develop a solution that obeys the two conditions and consists of a mosaic of

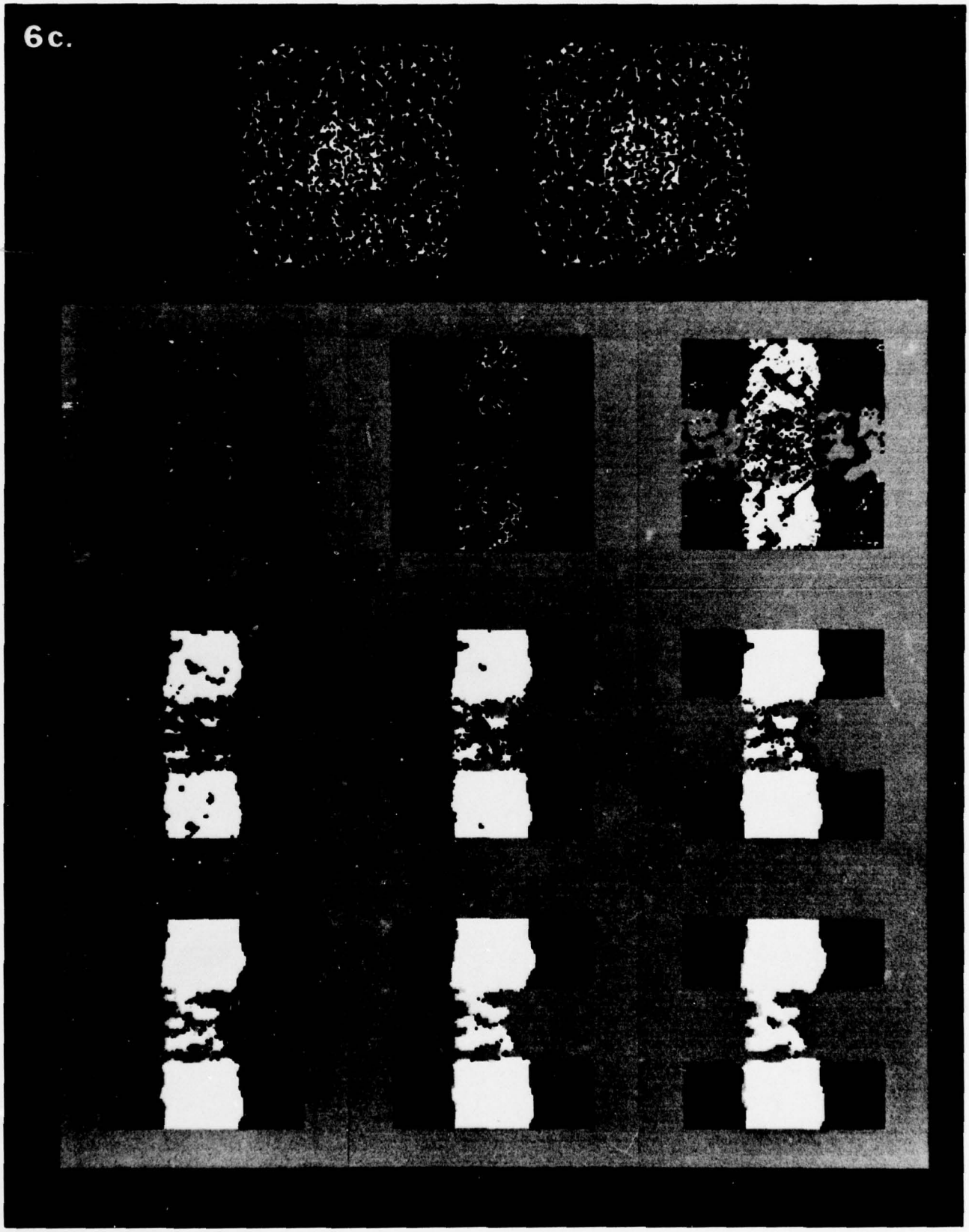
6a.



6. A stable geometrical configuration that violates the uniqueness condition (6a). The central square consists of two planes, one at disparity 2 and one at disparity 0. This configuration is a stable state of the algorithm, in the sense that if it is loaded directly into the network, an invariant configuration is quickly reached in which both planes are represented. Figure 6b demonstrates this. The stereogram is marked Left and Right, and 5 iterations of the algorithm are shown. If the network is loaded in the usual way, however, the algorithm develops a solution that is a mosaic of patches from the two levels (6c).

6b.




6c.

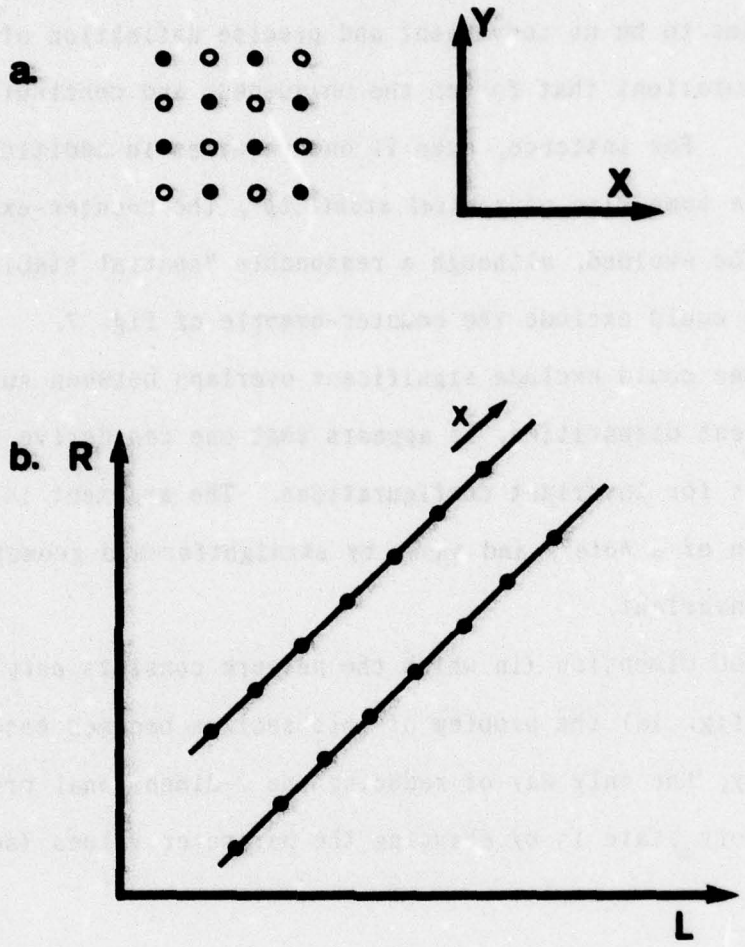
patches from the two levels (fig. 6c). With the parameters we chose, there seems to be no convenient and precise definition of the stability of configurations that forces the uniqueness and continuity of solutions. For instance, even if one requires in addition to invariance some kind of *spatial stability*[2], the counter-example of fig. 6 cannot be avoided, although a reasonable "spatial stability" condition would exclude the counter-example of fig. 7.

If one could exclude significant overlaps between surfaces lying at different disparities, it appears that one can derive the continuity conditions for invariant configurations. The argument is based here on the notion of a *hole*[3], and shows by straightforward geometry that holes are not invariant.

In one dimension (in which the network consists *only* of the part shown in fig. 1a) the problem of this section becomes easier. Apparently, the only way of reducing the 2-dimensional problem to a satisfactory state is by changing the parameter values (see section 6).

4 Probabilistic analysis of the algorithm

We have been unable to obtain general results about the convergence of this type of algorithm. Standard approaches -- e.g. Liapunov-type methods and the usual fixed point theorems -- apparently fail in this situation for reasons that we shall mention in the discussion.



7. A stable geometrical configuration that violates the continuity condition. At each of two disparity values, the "on" cells form a checkerboard pattern, but they are arranged in such a way that neither level can fill in, because of inhibition from the other.

The probabilistic analysis given here, although not completely satisfactory, nevertheless provides useful information about the algorithm's convergence for random-dot stereograms. Strictly speaking its application is restricted to inputs with a random structure.

The idea behind our analysis is that the cells in the network can be divided into populations on which the excitatory and inhibitory inferences are statistically homogeneous (*cf* Marr 1971). Our analysis is very specific to the algorithm of eq. *1.1.1*, because the way in which the cells are divided into populations depends critically on the geometry of the algorithm and on our *a priori* knowledge of its invariant state.

4.1 Assumptions and notation

The algorithm has the structure shown in fig. 1 and the network is loaded from the input as specified by eq. *2.1.1*. We shall assume that the inputs have the following properties.

4.1.1 the 1's in each image occur randomly with probability p, and the autocorrelation of each input sequence (for any given y) is a Kronecker δ.

4.1.2 the input admits a unique solution surface that is large enough to neglect boundary effects.

Condition *4.1.2* means that the left input is equal to the right one, modulo *x*-translation. Condition *4.1.1* implies that in the initial state of the network C, the density of 1's on the solution layer equals ν, and elsewhere it is ν^2. We subdivide the cells into five populations, by classifying them in two ways:

(i) according to whether or not they are a "on" in the initial state C^0, and

(ii) according to the number of active inputs from the images.

We draw both the populations 0 and 1 from cells that lie on the solution layer; population 0 is defined to receive no active inputs from the image, and population 1 receives two. Notice that there are no cells in the solution layer that receive exactly one active input.

The other three populations that we define refer to cells that lie off the solution layer; population 11 receives two active inputs from the image, population 10 receives one, population 00 receives none. The five populations (0, 1, 11, 10, 00) are exclusive and exhaustive.

We denote by $p_0(t)$, $p_1(t)$, etc. the probability that a cell in the respective population is "on" at time t. The goal of our analysis is to express the values of the $p_\pi(t)$ in terms of $p_\pi(t-1)$ for the various populations π. This allows us to examine the convergence numerically, and we say that a solution is achieved at time T when

$$p_0(t) = p_1(t) = 1, \quad \text{and}$$

$$p_{00}(t) = p_{10}(t) = p_{11}(t) = 0, \quad \text{for every } t \geq T.$$

The critical assumption here is that the quantity $p_\pi(t)$ completely describes the structure of active cells in the respective population

π. This assumption is true for the initial iteration and only approximate thereafter. We shall discuss this point at the end of the section.

4.2 Formulae

The state of a cell (x,y,d) at time $(t+1)$ depends upon the number of active cells in its excitatory $S(x,y,d)$ and inhibitory $O(x,y,d)$ neighborhoods at time t.

If we denote the populations to which the cell belongs by π, (π running through the five populations 0, 1, 00, 11, 01), let us define:

$e_\pi(r)$ to be the probability that exactly r cells are "on" in the excitatory neighborhood $S(x,y,d)$ at time t and

$i_\pi(r)$ to be the probability that exactly r cells are "on" in the inhibitory neighborhood $O(x,y,d)$ at time t.

It is convenient to introduce some further quantities:

$q_S(t)$ is the probability that a given cell on the "solution" plane is active at time t.

$q_W(t)$ is the probability that a given cell elsewhere in the network is active.

$q_-(t)$ is the probability that a given cell is active in the inhibitory neighborhood of a cell in the population 0.

$q_+(t)$ is the probability that a given cell is active in the inhibitory neighborhood of a cell in the population 1.

Then:

4.2.1
$$q_s(t) = p_0(t).(1 - v) + p_1(t).v$$
$$q_w(t) = p_{00}(t).(1 - v)^2 + p_{01}(t).2v(1 - v) + p_{11}(t).v^2$$
$$q_-(t) = p_{00}(t).(1 - v) + p_{01}(t) . v$$
$$q_+(t) = p_{11}(t).v + p_{10}(t).(1 - v)$$

Writing $B(n, f; m) = {}_mC_n.f^n(1 - f)^{m-n}$, where ${}_mC_n$ is the binomial coefficient, we have immediately

4.2.2
$$e_1(r) = e_0(r) = B(r, q_1(t); E)$$
$$i_1(r) = B(r, q_+(t); 2D - 2)$$
$$i_0(r) = B(r, q_-(t); 2D - 2)$$
$$e_{11}(r) = e_{00}(r) = e_{10}(r) = B(r, q_w(t); E)$$

The remaining i_π are more difficult to obtain, since the inhibitory contributions to cells lying off the solution plane come from cells lying on the solution plane *and* from cells lying off the solution plane, and these two populations obey different statistics. In fact

4.2.3 $$i_{11}(r) = [p_1(t)]^2.B(r-2, q_+(t); 2D-4) + 2p_1(t).(1-p_1(t)).B(r-1, q_+(t); 2D-4) + [1-p_1(t)]^2.B(r, q_+(t); 2D-4)$$

$$i_{00}(r) = [p_0(t)]^2.B(r-2, q_-(t); 2D-4) + 2p_0(t).(1-p_0(t)).B(r-1, q_-(t); 2D-4) + [1-p_0(t)]^2.B(r, q_-(t); 2D-4)$$

The final case i_{10} is especially awkward, because along one of the inhibitory lines the probability of a cell being "on" is q_+ and along the other diagonal it is q_-.

4.2.4 $$i_{10}(r) = \Sigma \{p_1(t)B(k-1, q_+(t); D-2) + (1-p_1(t))B(k, q_+(t); D-2)\} \cdot \{p_0(t).B(r-k+1, q_-(t); D-2) + (1-p_0(t))B(r-k, q_-(t); D-2)\}$$

We now need to relate the $p_\pi(t+1)$ and the $p_\pi(t)$ in terms of the e_π and i_π. For each cell population we know the distributions of incoming excitation and inhibition, and we know that a cell will be on whenever the excitations exceed the inhibitions by at least θ. Hence

4.2.5 $$p_\pi^{t+1} = \sum_{\substack{n = \theta \text{ to } E \\ m = 0 \text{ to } 2D-2 \\ n - \varepsilon m \geq \theta_\pi}} e_\pi^t(n) \cdot i_\pi^t(m)$$

$$\text{where } \theta_\pi = \theta - 1 \quad \text{for } \pi = 1, \pi = 11$$
$$\theta_\pi = \theta \quad \text{otherwise}$$

If the input term $C^0_{x,y,d}$ of eq. *1.1.1* is neglected, $\theta_\pi = \theta$ for all π.

The equations *4.2.6* are too complex to be solved analytically. Numerical solutions were however obtained for various values of the parameters and some of the results are given in table 1 and figure 8.

4.3 Range of parameter values and comparison with actual runs

Figure 8 exhibits the performance of the algorithm for stereograms having densities of from 0.5 to 0.05. Table 1 gives the statistics that were measured from these runs, and also the parameters predicted by the probabilistic theory. The values obtained from the theory match those from the algorithm quite well for the first iteration, but except for the case $\nu = 0.05$, they diverge quite rapidly thereafter, and even this case diverges by the third iteration.

We have already noted the main reason for the discrepancy. The assumption that the statistical structure of various populations is purely random (inside each population and between populations) holds exactly for the first iteration but only approximately thereafter, because the operator of fig. 1c has a local structure which can preserve local clusters of active cells. There are two ways in which this affects our probabilistic description for the second and subsequent iterations. The first is that clusters are more stable than

TABLE 1. The behavior of the algorithm compared with the probabilistic theory of the algorithm, for the stereograms having four different densities that are exhibited in figure 8.

1a. ν = .5, E = 12, D = 7, ε = 2, θ = 4.0 (SQ50%)

Iteration		p_r	p_w	p_0	p_1	p_{00}	p_{01}	p_{11}
1	Algorithm	.46	.07	.93	.01	.29	0	0
	Theory	.47	.087	.92	.01	.35	0	0
2	Algorithm	.70	.04	.43	.97	0	0	.16
	Theory	.62	.02	.26	.97	0	0	.08
3	Algorithm	.90	.01	.99	.81	.04	.001	.001
	Theory	.97	0	.99	.96	0	0	0
4	Algorithm	.98	.002	.96	1.0	0	0	.01
	Theory	1.0	0	1.0	1.0	0	0	0
5	Algorithm	.99	0	1.0	.99	0	0	0
	Theory	1.0	0	1.0	1.0	0	0	0

1b. ν = 0.25, E = 12, D = 7, ε = 2, θ = 4.0 (SQ25%)

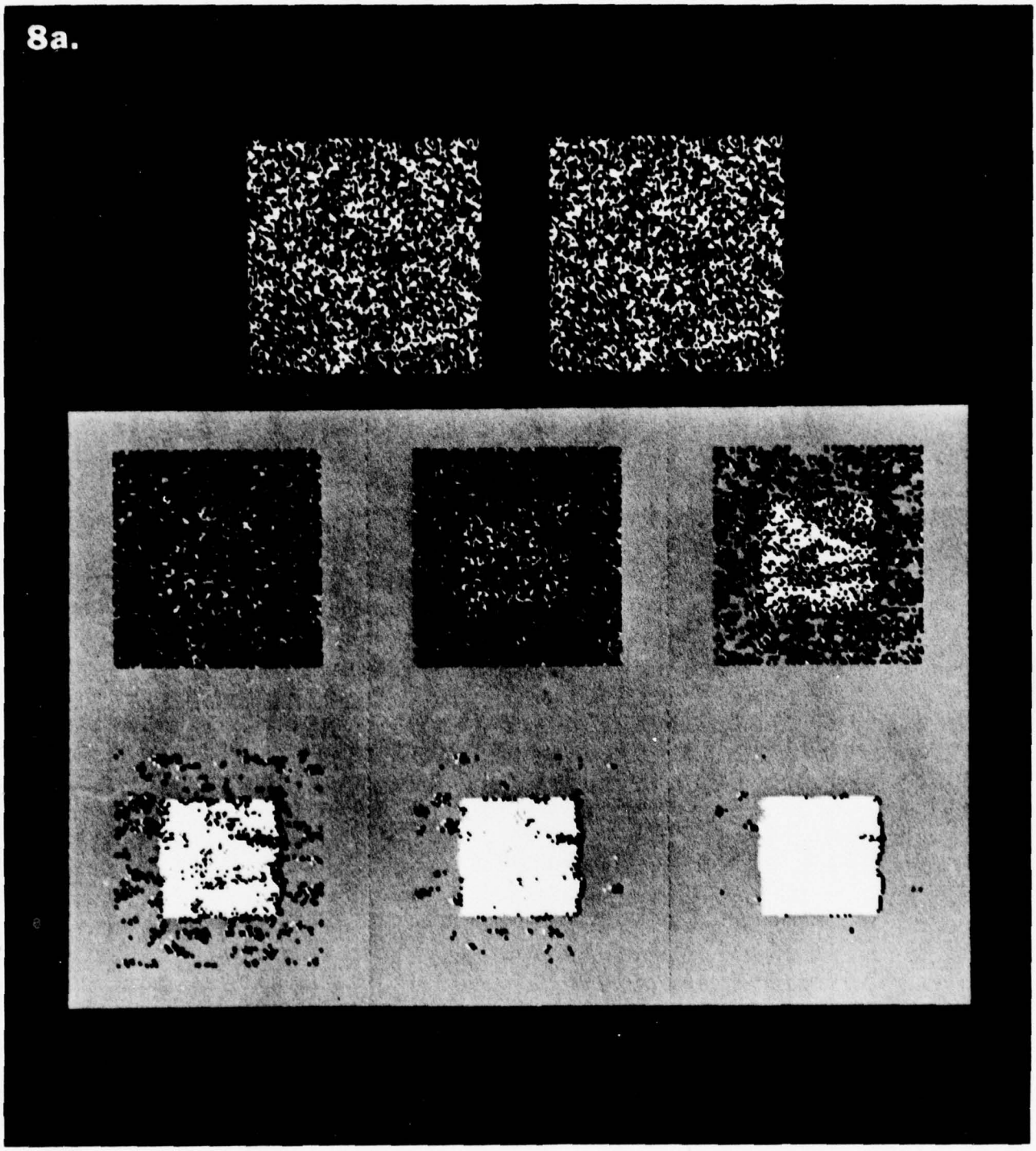
Iteration		p_r	p_w	p_0	p_1	p_{00}	p_{01}	p_{11}
1	Algorithm	.24	.001	.31	.02	.002	0	0
	Theory	.27	.003	.35	.04	.005	0	0
2	Algorithm	.39	0	.26	.81	0	0	0
	Theory	.475	0	.41	.68	0	0	0
3	Algorithm	.58	0	.52	.78	0	0	0
	Theory	.92	0	.90	.97	0	0	0
4	Algorithm	.74	0	.70	.87	0	0	0
	Theory	1.0	0	1.0	1.0	0	0	0

1c. ν = 0.1, E = 12, D = 7, ε = 2, θ = 3.0 (SQ10%)

Iteration		p_r	p_w	p_0	p_1	p_{00}	p_{01}	p_{11}
1	Algorithm	.10	0	.11	.05	0	0	0
	Theory	.11	0	.11	.106	0	0	0
2	Algorithm	.20	0	.15	.72	0	0	0
	Theory	.17	0	.14	.39	0	0	0
3	Algorithm	.36	0	.32	.68	0	0	0
	Theory	.35	0	.32	.61	0	0	0
4	Algorithm	.53	0	.51	.77	0	0	0
	Theory	.86	0	.85	.96	0	0	0
5	Algorithm	.71	0	.69	.83	0	0	0
	Theory	1.0	0	1.0	1.0	0	0	0

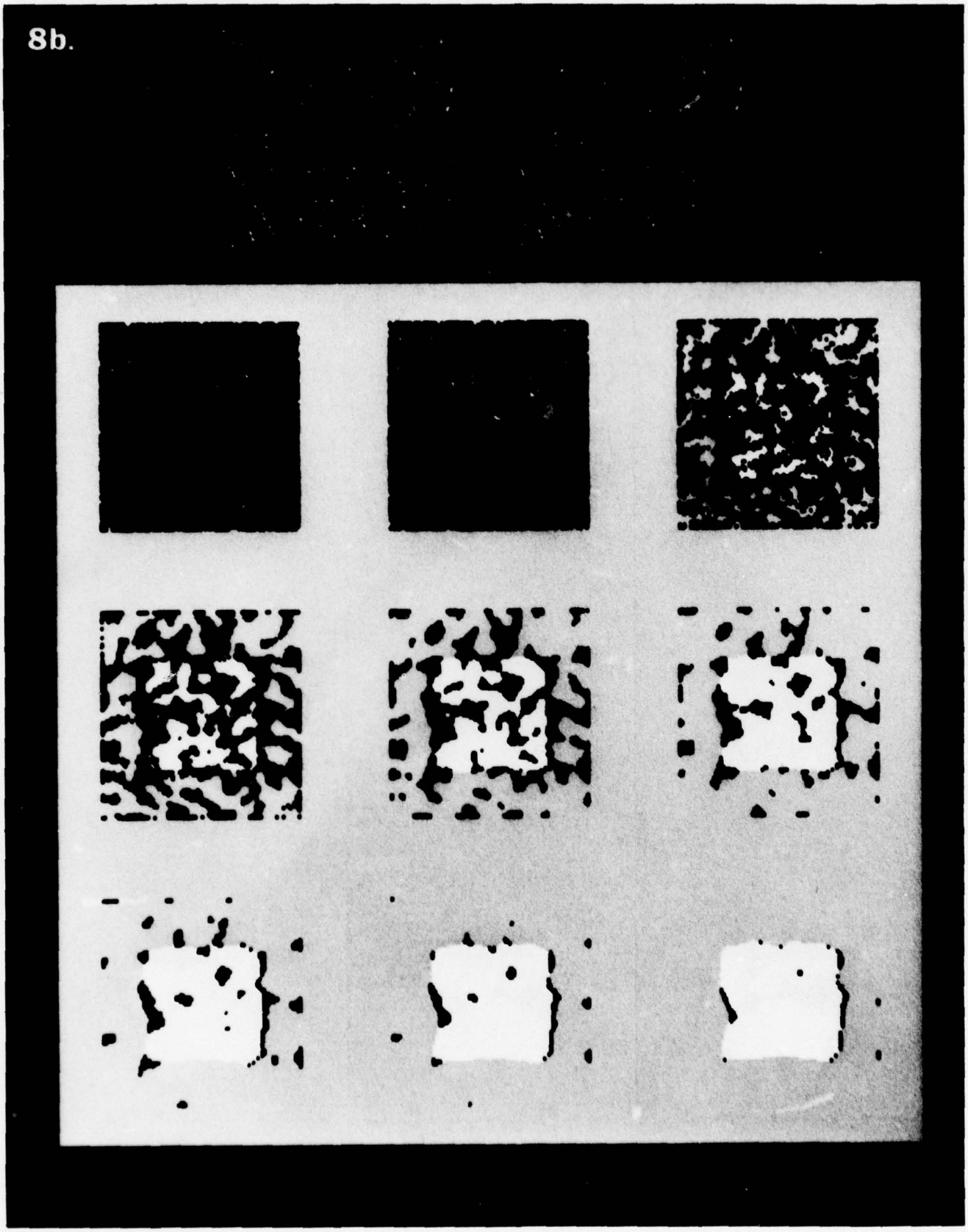
1d. ν = 0.05, E = 12, D = 7, ε = 2, θ = 2.0 (SQ05%)

Iteration		p_r	p_w	p_0	p_1	p_{00}	p_{01}	p_{11}
1	Algorithm	.13	0	.12	.22	0	0	0
	Theory	.13	0	.12	.26	0	0	0
2	Algorithm	.38	0	.35	.88	0	0	0
	Theory	.48	0	.46	.81	0	0	0
3	Algorithm	.67	0	.65	.88	0	0	0
	Theory	1.0	0	1.0	1.0	0	0	0
4	Algorithm	.90	0	.89	.96	0	0	0
	Theory	1.0	0	1.0	1.0	0	0	0
5	Algorithm	.98	0	.98	.98	0	0	0
	Theory	1.0	0	1.0	1.0	0	0	0

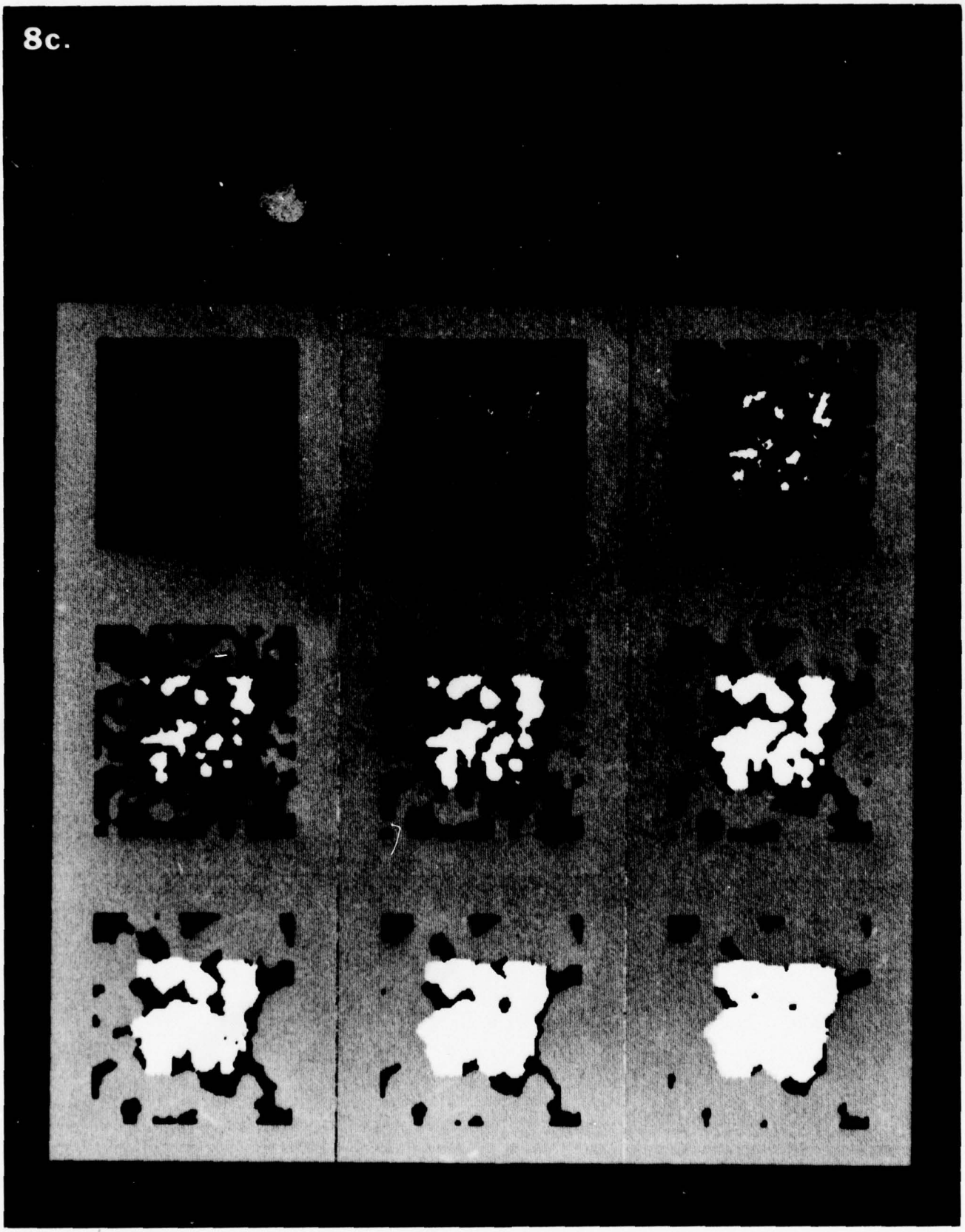


8. The stereograms (Left, Right) and iterations tabulated in table 1. Stereogram densities are 50% (8a), 25% (8b), 10% (8c) and 5% (8d). Parameters are as shown in table 1.

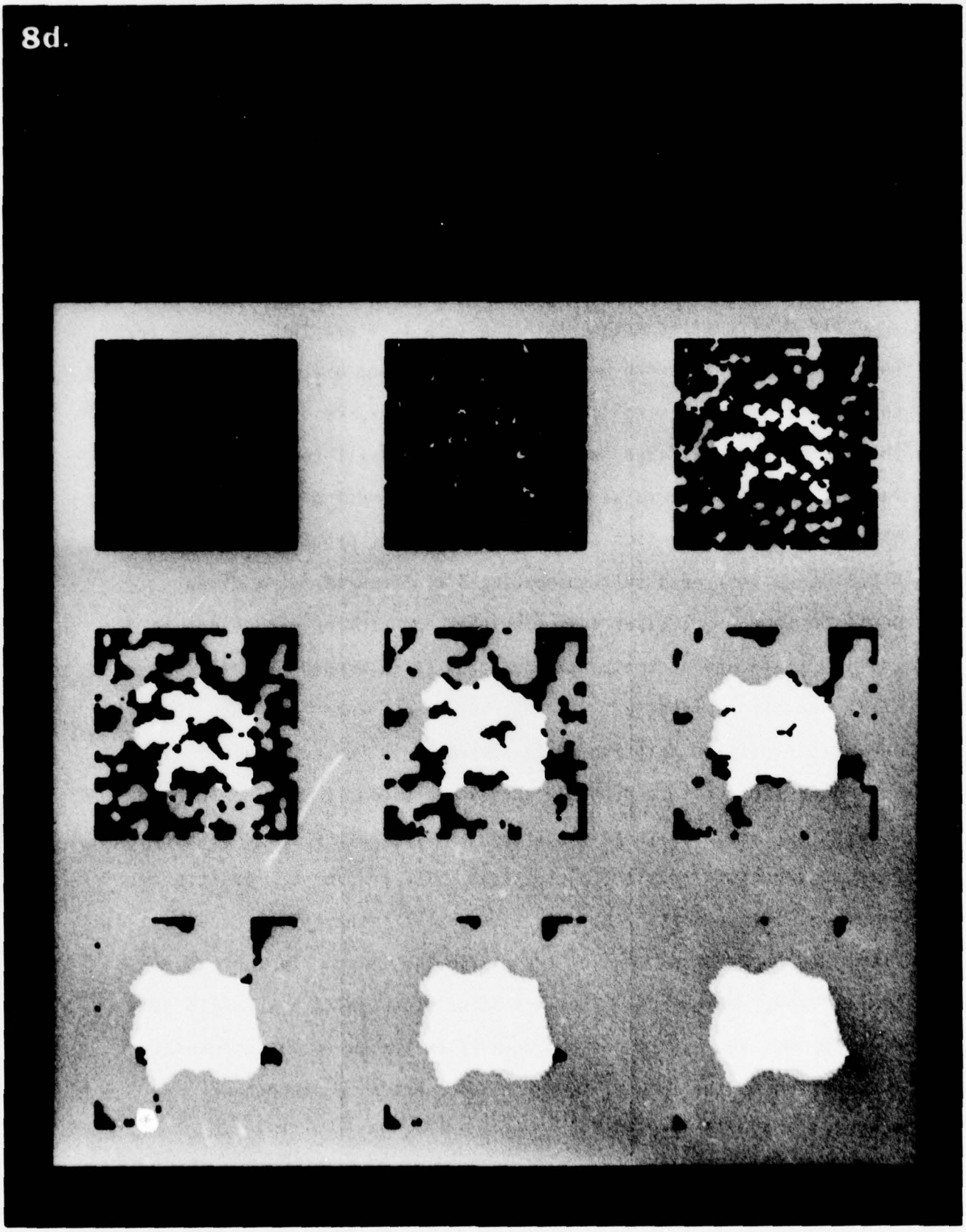
8b.



8c.



8d.



the assumption of randomness would predict. Thus clusters forming on the solution layer will in certain circumstances change the rate of convergence predicted by the randomness assumption.

The difficulties arise where clusters form off the solution layer. These will again tend to be more stable than our analysis assumes, but their effect acts against convergence. However, we shall argue that the probability of large "wrong" clusters is small for most patterns. In fact, the typical value of the probability that a wrong cell is "on" after the first iteration lies around 0.1. The probability (after the first iteration) of a self-supporting 3 x 3 cluster at a given position in a wrong layer (assuming that the cluster was absent in the initial state and accepting the oversimplified assumption of randomness after the first iteration) is about 10^{-9}, and hence less than 10^{-4} that one exists off the solution plane somewhere in the network.

A cluster of this size may survive permanently, because every element in it has at least 6 cells in its excitatory neighborhood, and this is enough to resist 1 inhibition). The probability of this or something larger arising by chance is so small that if it occurs it is likely to be a consequence of the particular image. In fact, some small "wrong" patches do sometimes occur (inspect Marr & Poggio 1976 fig. 5d) but such instances can usually be traced to an accidental correlation in the image. In this sense, extended patches are "correct" solution regions.

The second effect that leads to discrepancies between the theory and the behavior of the algorithm is also a side-effect of clustering,

since as well as being stable, the clusters tend to concentrate "on" cells more than the randomness assumption would predict. For example, at iteration 2 of the case $\nu = 0.25$ (fig. 8b), although the overall density of ones on the solution plane is about 0.39, it is far from true that each cell can expect to find $0.39\mathcal{E}$ "on" cells in its excitatory neighborhood. Cells in the filled in regions have almost all their neighbors on, whereas those in the interstices have none. Convergence is achieved by a growth outwards that fills in the blank regions, but although it is steady, it is necessarily slower than the theory predicts.

5 Observations

5.1 There is a wide latitude in the range of parameters for which the network converges. Table 2 shows firstly the wide range in stereogram density ν that is tolerated by our parameters (with fixed θ), and secondly, for a fixed value of ν ($\nu = 0.5$) gives some idea of the range of the other parameter values for which the network will converge. Note that in the implementation described by Marr & Poggio (1976), the threshold was not fixed, but was determined by the density of "on" cells in the network. This allowed solution to the matching problem over a very wide range of dot densities.

5.2 Let us define the probability that a cell on the solution layer is

TABLE 2. The algorithm of eq. 1.1.1 converges for a wide range of control parameters. Tables 2a, b show convergence for $\nu = 0.5$ and $\nu = 0.1$ with the same parameters. Table 2c shows convergence for an entirely different set of parameters.

2a. $\nu = 0.5$, E = 12, D = 7, E = 2, $\theta = 3.0$

Iteration	p_r	p_w	p_0	p_1	p_{00}	p_{10}	p_{11}
1	.50	.15	.98	.026	.61	0	0
2	.57	.13	.15	.997	0	0	.54
3	.69	.039	.995	.39	.16	0	0
4	.97	.007	.935	1.0	0	0	.029
5	1.0	0	1.0	1.0	0	0	0

2b. $\nu = 0.1$, E = 12, D = 7, E = 2, $\theta = 3.0$

Iteration	p_r	p_w	p_0	p_1	p_{00}	p_{10}	p_{11}
1	.11	0	.11	.106	0	0	0
2	.17	0	.14	.39	0	0	0
3	.35	0	.32	.62	0	0	0
4	.86	0	.85	.96	0	0	0
5	1.0	0	1.0	1.0	0	0	0

2c. $\nu = 0.5$, E = 2, D = 7, E = 0.5, $\theta = 1.0$

Iteration	p_r	p_w	p_0	p_1	p_{00}	p_{10}	p_{11}
1	.40	.11	.75	.058	.43	0	.010
2	.55	.23	.11	.99	.004	0	.90
3	.45	.083	.78	.11	.32	0	.006
4	.59	.20	.20	.99	.003	0	.80
5	.51	.063	.82	.20	.24	0	.009
6	.65	.17	.32	.99	.003	0	.66
7	.62	.042	.87	.36	.15	.001	.014
8	.76	.11	.54	.97	.002	0	.43
9	.82	.021	.94	.71	.053	.002	.026
10	.94	.027	.88	.995	0	0	.11
11	.995	.009	.996	.995	.001	0	.031
12	1.0	.004	1.0	1.0	0	0	.014
13	1.0	.002	1.0	1.0	0	0	.007
14	1.0	0	1.0	1.0	0	0	.003

"on" at time t to be

$$p_r(t) = v.p_1(t) + (1 - v)p_0(t)$$

and the probability that a cell off the solution layer is on at time t as

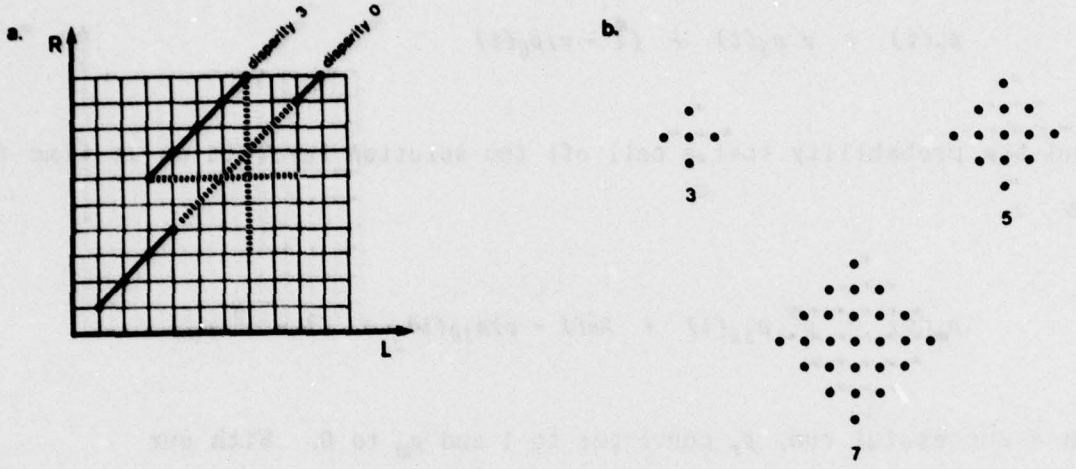
$$p_w(t) = v^2.p_{11}(t) + 2v(1 - v)p_{10}(t) + (1 - v^2)p_{00}.$$

In a successful run, p_r converges to 1 and p_w to 0. With our particular parameters, convergence is monotonic if it occurs. This is not true, however, for the individual quantities p_1. p_0. p_{11}. p_{10}. p_{00}, neither is it true of p_r and p_w for all values of the parameters (see table 2).

5.3 We have already seen that the sharpest local corner capable of resisting 1 inhibitory input is about 90^o or more, hence thin, sharp regions will tend to be rounded off locally (see Marr & Poggio 1976 fig. 5c). The exact shape of the input pattern is preserved only up to this limit.

5.4 Minimum size vs. disparity

A natural consequence of the structure of the algorithm is that the minimum resolvable area of a small pattern against a background increases with disparity (see Marr & Poggio 1976, fig. 6). We give an



9. The minimum resolvable area of a small pattern against a background increases with disparity. To prevent the background from filling in completely, the length of the patch in the x-direction must be at least $d + 2$ (9a). 9b shows the circles of diameters 3, 5 and 7 used in figure 6 of Marr & Poggio 1976.

estimate of the dependence of minimum patch size on disparity difference. Consider a section for some fixed y of the network (fig. 9). Assume that the patch and background regions are filled in. The condition for growth at a point (x, y, d) under 1 inhibition is that the number of "on" cells in an excitatory neighborhood should be not less than $\theta + \epsilon - C^0 = 5$ or 6, depending on the initial conditions. From fig. 3 we see that flat or convex regions will not grow whereas concave regions will. Hence our small patch will not tend to grow, whereas the background will spread until stopped by two inhibitions. We see from fig. 9a that to prevent the background from filling in completely (which would subsequently destroy the patch because convex borders cannot survive two inhibitions), the length of the patch in the x direction must be at least $d + 2$. This condition must hold for at least three adjacent lines aligned in the y direction. Fig. 6 of Marr & Poggio (1976) illustrates the approximate validity of this relation. Fig. 9b shows the sizes of circles of diameter 3, 5, and 7 used in the input for that figure. These precise patterns do not necessarily emerge in the appropriate layer of the network because of the random nature of the borders. The circle of diameter 3 contains no 3 x 3 subset and therefore does not survive at any disparities. The circle of diameter 5 contains one 3 x 3 square and survives as expected at disparity 1; it also survives, apparently accidentally, at disparity 2, but not at disparity 3. The circle of diameter 7 contains one 5 x 5 square and thus survives at disparity 3.

A trivial consequence of this analysis is that horizontal stripes

(parallel to the x axis) are in general more stable than vertical ones (parallel to the y axis). The minimum thickness for horizontal stripes is about 3 and is independent of disparity whereas the minimum thickness for vertical stripes is about $d + 2$ (see fig. 10).

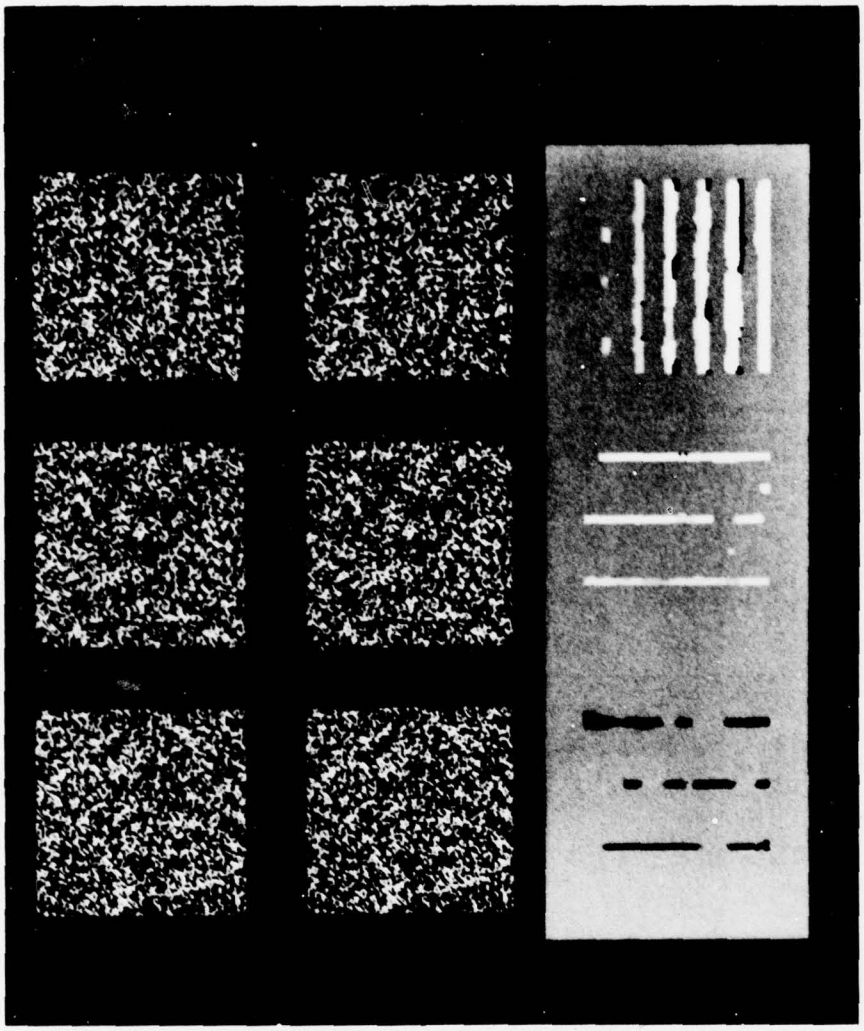
5.5 *Uncorrelated areas*

If there exists a sufficiently large area in the input where there is no correlation between the two images, the network will detect it (see figs. 5 and 6 of Science). After the first iteration (with our parameter values and $\nu = 0.5$) only a few cells remain "on" in the uncorrelated region, but provided the region is sufficiently large they will receive no inhibition from the surrounding more organized layers. Hence those cells that are on may act as germs for small regions that have become stable by the time the surround encroaches upon them, *e.g.* fig. 5d of Marr & Poggio (1976). Relatively small ($<< d$) uncorrelated areas probably have to develop stable platelets to survive (see fig. 6d of Marr & Poggio 1976), and large uncorrelated areas decompose into a random mosaic of stable platelets (see figure 11).

Uncorrelated areas can be recognized as such during the read-out from the network, when the 1's that appear in the solution found by the network are used to establish an explicit correspondence between the two images.

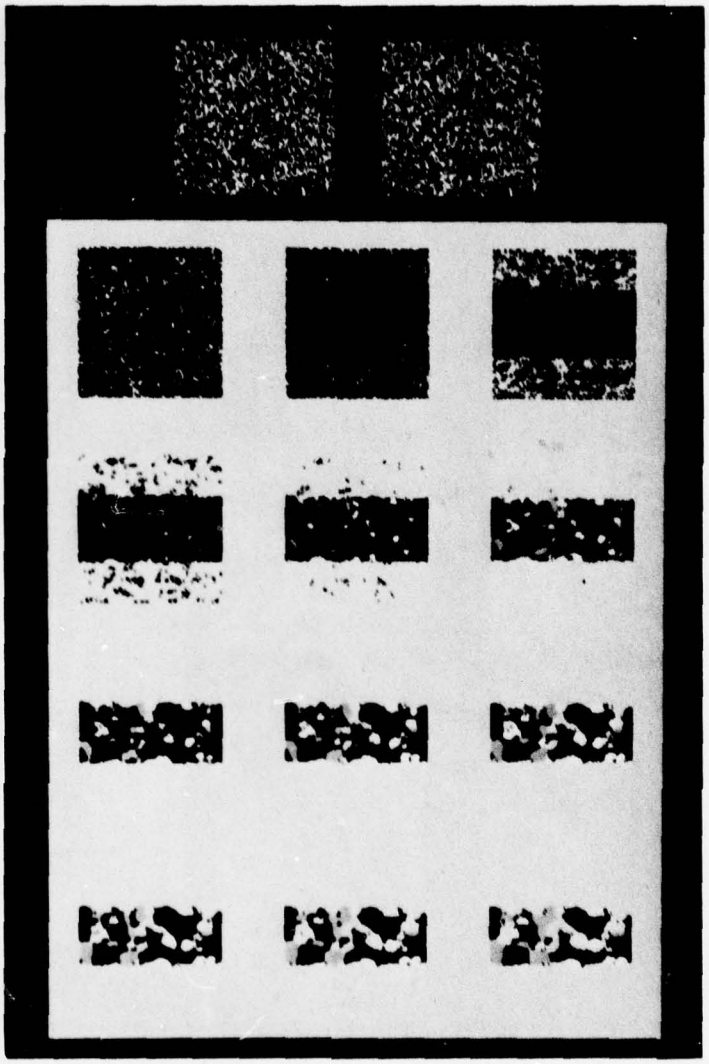
5.6 *Extension to images in which disparity varies continuously*

The algorithm of eq. *1.1.1* with the loading rules of eq. *2.1.1* can



10. Thin vertical and horizontal stripes of various disparities. The left and right stereograms are shown with the stable network solution to them. The stereograms are 100 by 100, and consist of stipes with the following coordinates (x or y), thicknesses s and disparities d:

	10a			*10b*			*10c*	
x	s	d	y	s	d	y	s	d
15	2	-1	15	2	-1	15	2	+1
30	3	-1	30	3	-1	30	3	+1
45	3	-2	45	2	-2	45	2	+2
60	4	-2	60	3	-2	60	3	+2
75	4	-3	75	2	-3	75	2	+3
90	5	-3	90	3	-3	90	3	+3



11. The central band is uncorrelated. It decomposes into a random mosaic of patches, each of which is eventually stable.

deal only with images having discrete disparity values. The disparity in natural images commonly varies continuously. There are two approaches to this problem. One is to incorporate the representation of continuous values directly into the algorithm, and the other is to use the same algorithm, but with special rules for loading it and for interpreting its final state.

The first approach would clearly lead to a considerably different algorithm, perhaps more along the lines of the networks studied by Wilson & Cowan (1973), (see also H. R. Wilson 1977). Such an algorithm could not be treated within the framework of this article.

The second approach does not require any changes in the analysis of the algorithm itself. One could, for exampe, define the loading conditions as follows:

Let Δ be the disparity attached to a possible correspondence between items in the left and right images. For integral d,

5.6.1 If $d - \eta \leq \Delta < d + \eta$, load the cell corresponding to disparity level d in the network.

For surfaces whose disparity does not oscillate too much or too densely, the value $\eta = 0.5$ will lead to satisfactory results. The final state of the network establishes a correspondence between items in the left and right images, but their associated disparity is read not from the network (*i.e.* d) but directly from the input (i.e. Δ). Confusions may of course arise in the correspondence established by the

network if the value of d spans the disparity range too coarsely.

In order to deal with surfaces that are less well-behaved, one can incorporate some hysteresis into the loading rules. The loading process then consists of the following steps:

5.6.2 Load cells according to *5.6.1* with $\eta = 0.3$ (say).

5.6.3 Moving across the image (x, y) in a spatially ordered way, if possible match (x, y, Δ) was not loaded by *5.6.2*, adopt the following procedure:

Let d^- = Integral part of Δ, $d^+ = 1 + d^-$. Examine (x, y) neighborhoods of (x, y, d^-) and of (x, y, d^+) in the network as it is loaded so far. Assign the current match to that d whose neighborhood contains more loaded cells, if one of them does. Else load this point according to *5.6.1* with $\eta = 0.5$.

This process will load most images in satisfactory way, and the read-out procedure is similar to that of the previous case.

6 A mathematically tractable version of the algorithm

A suitable choice of the parameter values and of the loading rules of the algorithm allows a complete mathematical analysis of its asymptotic behavior. In this section we introduce this "strict"

version of the algorithm and we characterize rigorously its properties. The actual performance of this version of the algorithm for various random dot stereograms will be then compared with the original algorithm.

6.1 Loading conditions

The initial state C^0 of the network is loaded from the stereograms L, R in a way similar to the previous case but according to eq. *6.1.1* (instead of eq. *2.1.1*).

6.1.1 $$C^0_{x,y;d} = L_{x,y} \cdot R_{x+d,y}$$

where $1.1 = 0.0 = 1, \quad 1.0 = 0.1 = 0.$

This loading rule can be easily extended to cases in which more than two features are present. It is enough to define

6.1.2 $$f_i \cdot f_j = \delta_{ij},$$

where f_i and f_j, $(i \neq j)$ are two different features. The case when only two features are present clearly poses the hardest matching problem. We shall later compare this loading rule with the original eq. *2.1.1* and discuss their relative merits for real images.

6.2 *The algorithm*

The relation betwen states at times t and $t+1$ is given by (compare eq. *1.1.1*)

$$6.2.2 \quad C^{t+1}_{x,y;d} = \sigma\left\{\inf\left[\sum_{x',y',d' \in S(x,y,d)} C^{t}_{x',y';d'} \, , \, H\right] - \varepsilon \sum_{x',y',d' \in O(x,y,d)} C^{t}_{x',y';d'}\right\}$$

where H is a number that represents the "saturation" value for the excitation.

6.3 *Choice of parameter values*

In this case the loading rules lead, for random dot stereograms with two features, to a density of 1 for the "on" cells on the "correct" diagonal segments and, correspondingly, to a density of $\nu^2 + (1 - \nu)^2$ for the "on" cells on the "wrong" diagonal segments (ν is the density of 1's in the input images). When $\nu = 0.5$, the density of the wrong cells is also 0.5; for smaller or larger ν the density is higher. The idea behind this approach is to choose parameter values for the first iteration that "kill" most of the "wrong" cells (and of course some of the "right" ones); from the second iteration on, the parameter values are such to ensure "filling-in" of the right diagonal segments, allowing, at the same time, a satisfactory mathematical analysis of the evolution of the network's state. This approach, which is carried out in the next two sections, leads to the following parameter values:

6.3.1 S and O are as in fig. 2, $D = 7$ and $N = 5$ as before. Self-excitation is now included but the C^0 term is omitted. We therefore write $E = 13$ instead of 12.

6.3.2 Iteration 1:

$H = 13$ (so that the *inf* operation can be neglected)

$\epsilon = 0.2$

$\theta = 10.75$

6.3.3 Second and subsequent iterations:

$H = 7$

$\epsilon = 4.0$

$\theta = 3.5$

6.4 Probabilistic analysis of the first iteration

We shall assume that the inputs have the properties *4.1.1* and *4.1.2*. As in section 4.1, we distinguish several populations of cells which are homogeneous with respect to the interaction structure: the populations are again denoted by 0, 1, 11, 10, 00 according to their respective inputs from the two images (see section 2), and p_0, p_1, etc. denote the probability that a cell in the respective population is "on" after the first iteration. In this case the formulae for the solution layer are:

$$p_1 = \sum_{i=0}^{\frac{13-\theta}{\varepsilon}} {}_{12}C_i \cdot \nu^i (1-\nu)^{12-i}$$

$$p_0 = \sum_{i=0}^{\frac{13-\theta}{\varepsilon}} {}_{12}C_i \cdot (1-\nu)^i \cdot \nu^{12-i}$$

For the "wrong" layers (writing $\mu = \nu^2 + (1-\nu)^2$), the formulae are

$$p_{11} = \sum_{k=0}^{12} {}_{12}C_k \, \mu^k (1-\mu)^{12-k} \sum_{i=0}^{\frac{k+1-\theta}{\varepsilon}-2} {}_{10}C_k \, \nu^i (1-\nu)^{10-i}$$

$$p_{00} = \sum_{k=0}^{12} {}_{12}C_k \, \mu^k (1-\mu)^{12-k} \sum_{i=\phi}^{\frac{k+1-\theta}{\varepsilon}-2} {}_{10}C_k (1-\nu)^i \nu^{10-i}$$

$$p_{10} = \sum_{k=0}^{12} {}_{12}C_k \mu^k (1-\mu)^{12-k} \sum_{i=0}^{\inf\left(\frac{k-\theta}{\varepsilon}-2\right),5} {}_5C_i \nu^i (1-\nu)^{5-i} \cdot \sum_{j=0}^{\inf\left(\frac{k-\theta}{\varepsilon}-2\right)-i,5} {}_5C_j (1-\nu)^j \nu^{5-j}$$

Therefore the probability that a cell in the solution layer is "on" after the first step is

$$p_r = p_1 \cdot \nu + p_0 \cdot (1-\nu)$$

and the probability that a cell off the solution layer is "on" after the first step is

$$p_w = \nu^2 p_{11} + 2\nu(1-\nu)p_{10} + (1-\nu)^2 p_{00}.$$

These equations can be used to find suitable parameter values. The parameters given in the previous section yield the values for p_r and p_w

shown in table 3.

6.5 Equivalent rules

The parameter values from the second iteration on imply the following main "rules" for the algorithm:

6.5.1 one "on" cell in the inhibitory neighborhood always suffices kill an "on" cell

6.5.2 without inhibition, at least three excitatory "on" cells are needed for "survival" of an "on" cell and four for its "birth".

6.6 Analysis of the second iteration

Table 3 gives the densities p_r and p_w after the first iteration. Only for the first iteration can a probabilistic analysis provide a reliable estimate of the density of "on" cells on the solution surface. As in our earlier analysis (table 1), it becomes unreliable for the second iteration, because clusters of "on" cells can be expected to form in the solution layer (see figure 14 below). Rule *6.5.1* implies, however, that "wrong" clusters will disappear after the second iteration, unless they consist of at least four elements. Moreover, these elements must in practise be very close together for each to support the other three. In addition, according to rule *6.5.2*, none of them can lie in the inhibitory neighborhood of other "on" cells (for

TABLE 3. The behavior of the mathematically tractable of the algorithm, together with the probabilistic theory of the first iteration, for the two stereograms exhibited in figures 14a and b.

3a. $\nu = 0.5$, E = 13, D = 7

ε	θ	H	Iteration	p_r	p_w	p_0	p_1	p_{00}	p_{10}	p_{11}	
0.2	10.75	13.0	1	.9998	.0017	.9998	.9998	.0023	.0011	.0023	Theory
			1	.99	.0003	.99	.98	.0008	0	.0004	Algorithm
4.0	3.75	7.0	2	.99	0	.99	.99	0	0	0	
			3	1.0	0	1.0	1.0	0	0	0	

3b. $\nu = 0.25$, E = 13, D = 7

ε	θ	H	Iteration	p_r	p_w	p_0	p_1	p_{00}	p_{10}	p_{11}	
0.2	10.75	13.0	1	.976	.0078	.968	1.0	.0039	.0015	.082	Theory
			1	.96	.0002	.95	1.0	.002	.0003	.002	Algorithm
4.0	3.5	7.0	2	.98	0	.98	.97	0	0	0	
			3	1.0	0	1.0	1.0	0	0	0	

instance on the solution layer where the density of on cells is relatively high, see table 3). We argue that the probability of such situations is very small (actually much smaller than in the case considered in section 4.4). If this occurs it can be attributed to an accidental correlation in the images. In this sense extended clusters are in fact "right" solution regions.

6.7 Asymptotic analysis

The probabilistic analysis of the first iteration (table 3) shows that one can assume that, from the second iteration onwards, there are no wrong "on" cells. It remains now to show that the density of "on" cells on the solution layer is high enough to allow asymptotic filling-in of the "right" surfaces. We prove the following:

6.7.1 Filling-in Proposition. Assume that (at some iteration n) there are no "on" cells off a given layer (diagonal), and that the density of "on" cells on this layer exceeds 0.4375 = 7/16. Then, in the asymptotic configuration, there are no "off" cells on this layer.

Proof: Divide the solution plane into squares of 4 by 4 cells (we neglect boundaries). At least one of these squares must contain 8 "on" cells, for, otherwise, every square would contain at most 7 "on" cells yielding a density of at most 7/16, in contradiction with the hypothesis. This square will fill up with "on" cells. (This can be

seen by examining the various possible ways in which the 8 cells can be distributed, and we leave it as an exercise for the reader). Starting from this square, the whole plane will asymptotically be filled by "on" cells (since, by hypothesis, no inhibitory cells need be considered).

6.8 Invariant states and matching rules

The matching rules were defined in section 3. States that satisfy the matching rules with the present parameter values are shown in fig. 12. In view of the rules *6.5.1* and *6.5.2*, the following clearly hold:

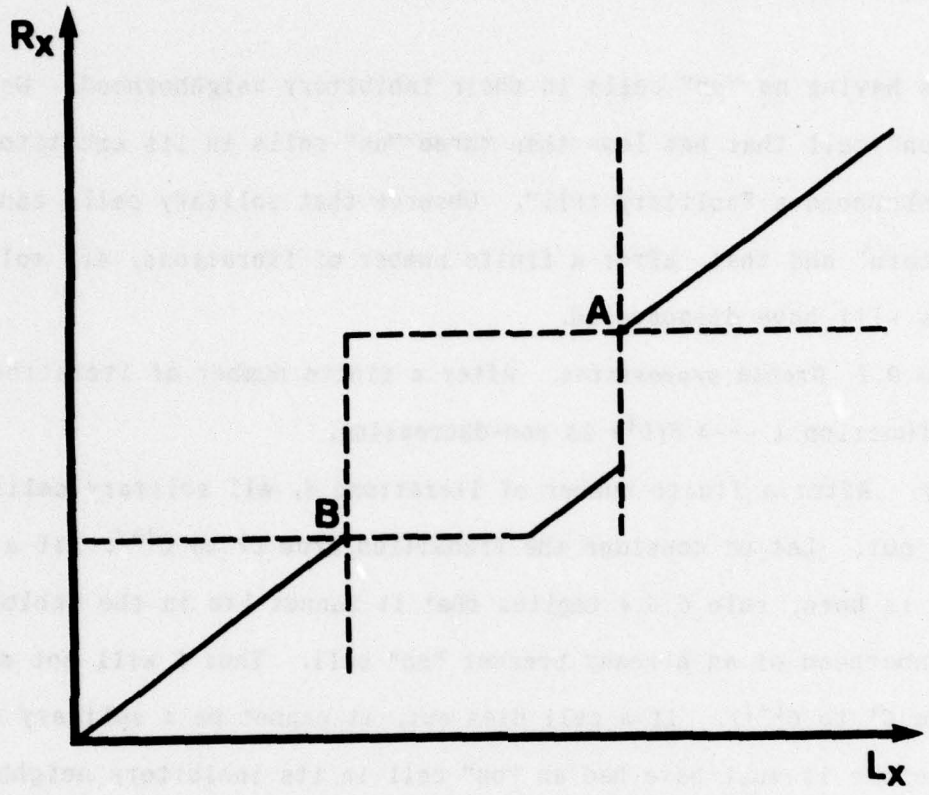
i) Configurations that satisfy the matching rules (fig. 12) are invariant.

ii) Conversely, invariant configurations clearly have to obey the uniqueness condition (because of *6.5.1*). The probabilistic analysis of the second step, together with the "filling-in" proposition *6.7.1*, ensures in practise that there will be no holes[2] in the asymptotic invariant configurations.

6.9 Asymptotic Liapunov description

Besides the invariant asymptotic configuration, limit cycles of the type described in figure 13 may also occur. Thus the previous description of asymptotic invariant states is not complete. We provide here an asymptotic analysis in terms of a Liapunov-like function which also encompasses such non-invariant states.

For a given state C^i, we define $F(C^i)$ to be the number of "on"



12. With the modified parameters, cells cannot survive against one inhibition. Hence stable states satisfy the uniqueness condition, because no overlap is possible (compare figure 5).

cells having no "on" cells in their inhibitory neighborhood. We call an "on" cell that has less than three "on" cells in its excitatory neighborhood a "solitary cell". Observe that solitary cells can never be "born" and that, after a finite number of iterations, all solitary cells will have disappeared.

6.9.1 Growth proposition. After a finite number of iterations, the function $i \dashrightarrow F(C^i)$ is non-decreasing.

Proof: After a finite number of iterations i, all solitary cells have died out. Let us consider the transition from C^i to C^{i+1}. If a new cell is born, rule *6.6.1* implies that it cannot lie in the inhibitory neighborhood of an already present "on" cell. Thus F will not decrease (from C^i to C^{i+1}). If a cell dies out, it cannot be a solitary cell. Therefore it must have had an "on" cell in its inhibitory neighborhood at iteration i. Thus F will not decrease.

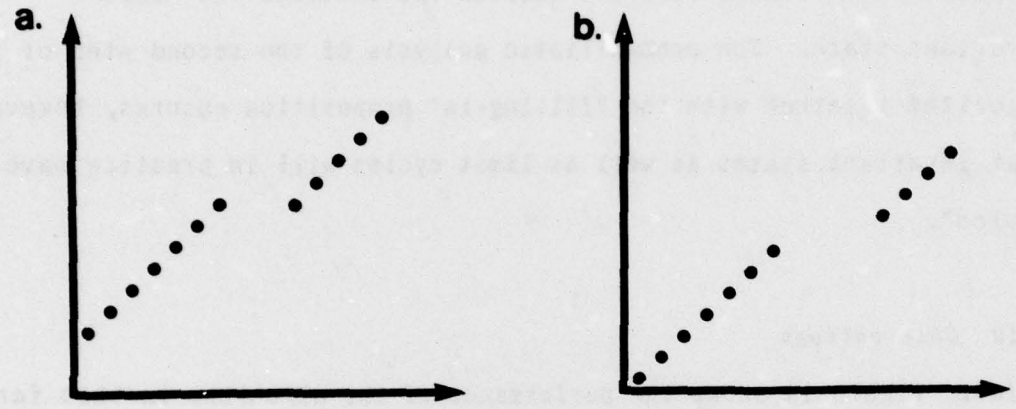
The growth of F adequately describes the filling-in process, respecting at the same time the "uniqueness" matching rule.

The growth proposition implies that:

6.9.2 For any initial configuration C^0, the limit $Lim\ F(C^i) = F(C)$ exists (since F is bounded above by the number of cells in one layer).

6.9.3 After a finite number of iterations, $F(C^i)$ remains constant.

Thus the asymptotic behavior of the system is characterized by the



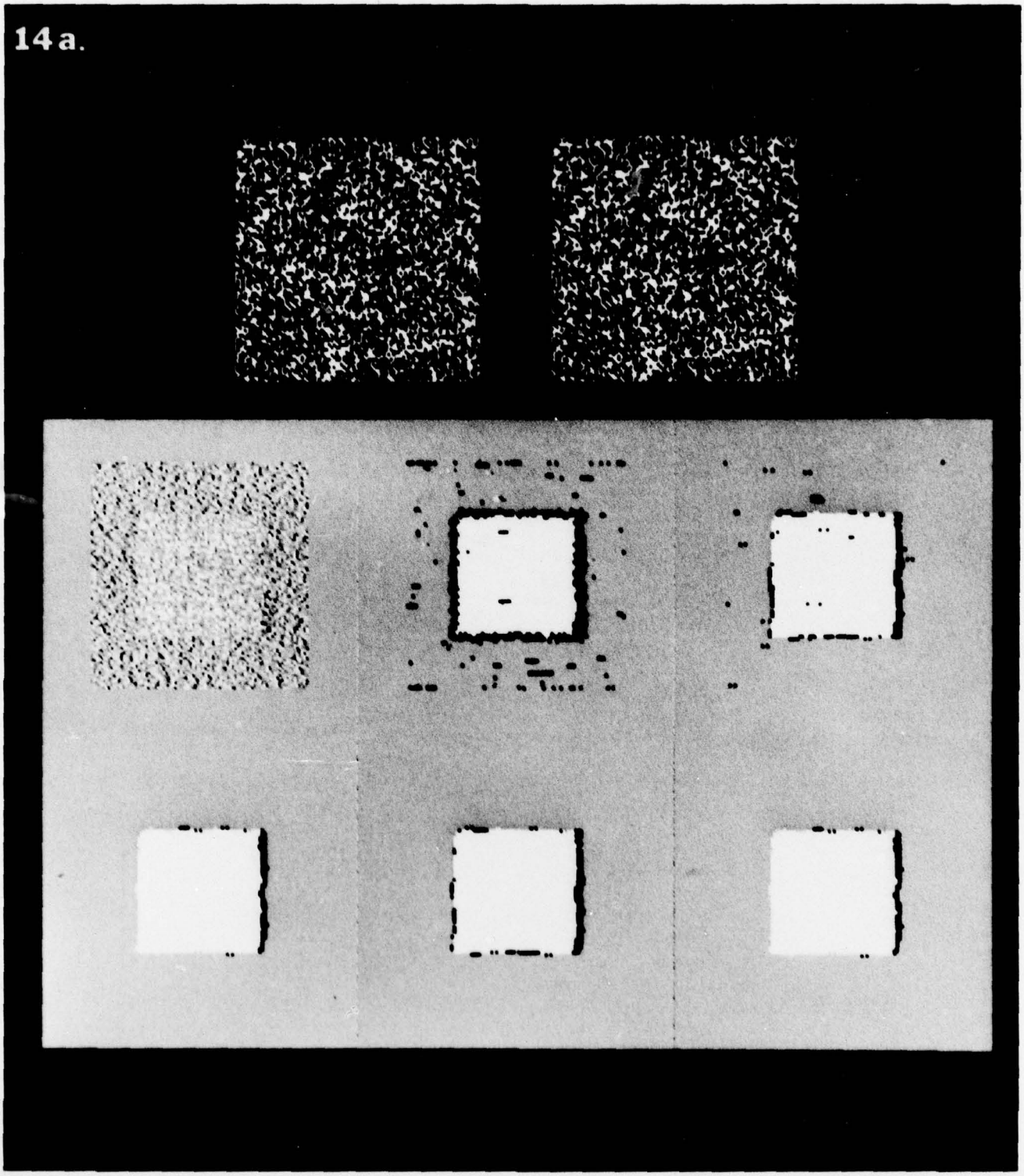
13. An oscillating solution with the modified parameters. The state 13a occurs at iterations i, $i+2$, $i+4$, ..., whereas state 13b occurs at iterations $i+1$, $i+3$, $i+5$, ...

following: Apart from invariant solutions, only those cycles can (asymptotically) occur for which *F* remains constant. This is a strong restriction on the possible asymptotic oscillatory states, for it means that they have to be of the type shown in fig. 13. The growth proposition by itself does not exclude for instance the "zero" invariant state. The probabilistic analysis of the second step of the algorithm together with the "filling-in" proposition ensures, however, that invariant states as well as limit cycles will in practice have no "holes".

6.10 Observations

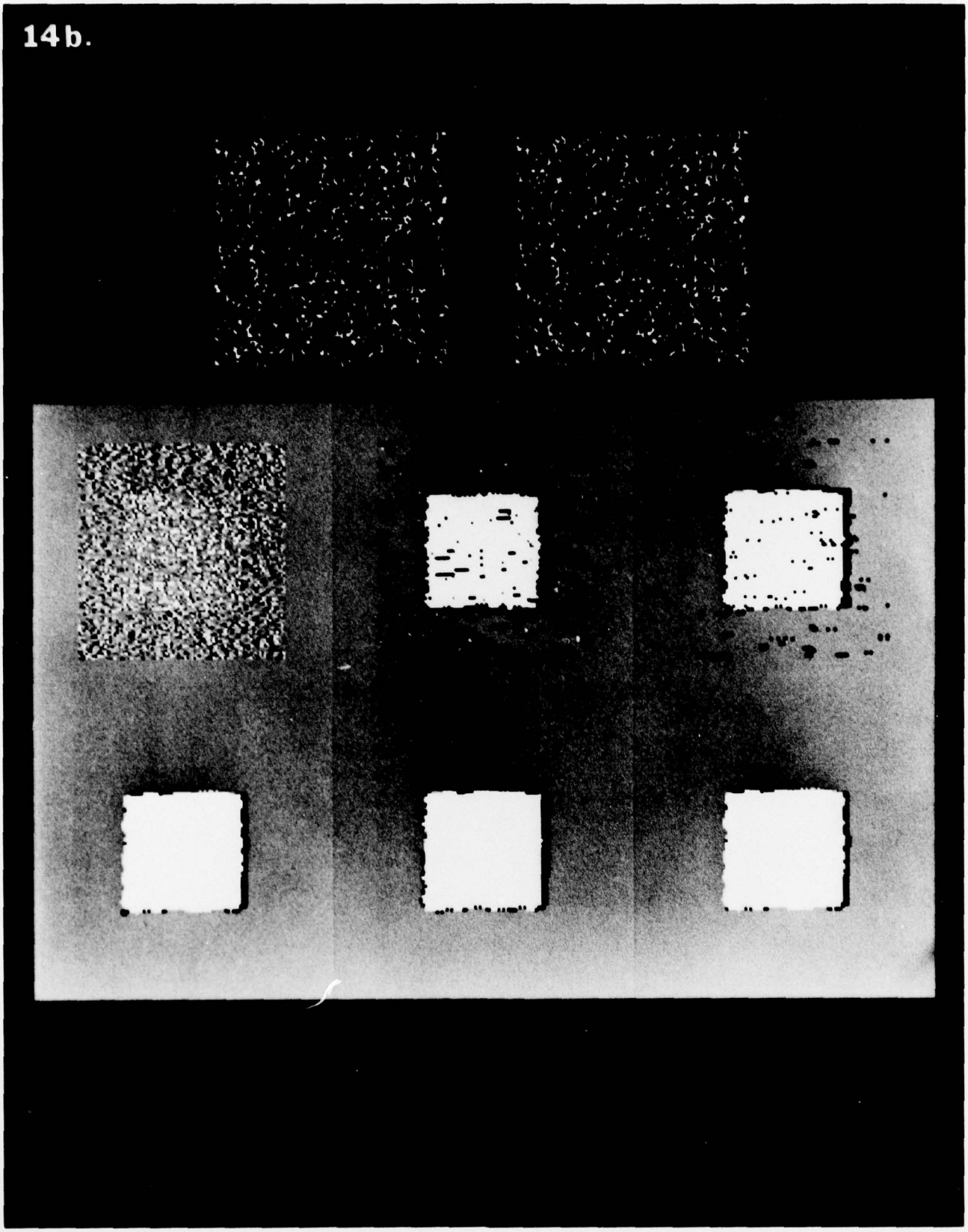
6.10.1 Figure 14 shows the performance of the algorithm in this form for a few different patterns and pattern densities. A comparison with figure 8 reveals that the type of "strategy" for achieving a successful matching is different: Firstly, wrong cells are drastically eliminated at the expense of losing many right cells, and then filling-in of the surviving surfaces takes place. This contrasts with the more complicated "strategy" revealed in figs. 3 - 6 of Marr & Poggio 1976. It is remarkable how, while the basic structure of the algorithm remains the same, a change of parameter values and loading conditions can bring about so deep a change in the algorithm's behavior.

6.10.2 Because of the rules of the present algorithm, especially rule *6.5.1*, the sharpest corner capable of surviving (of course under no inhibitions) is limited only by the need for an "on" cell to have at

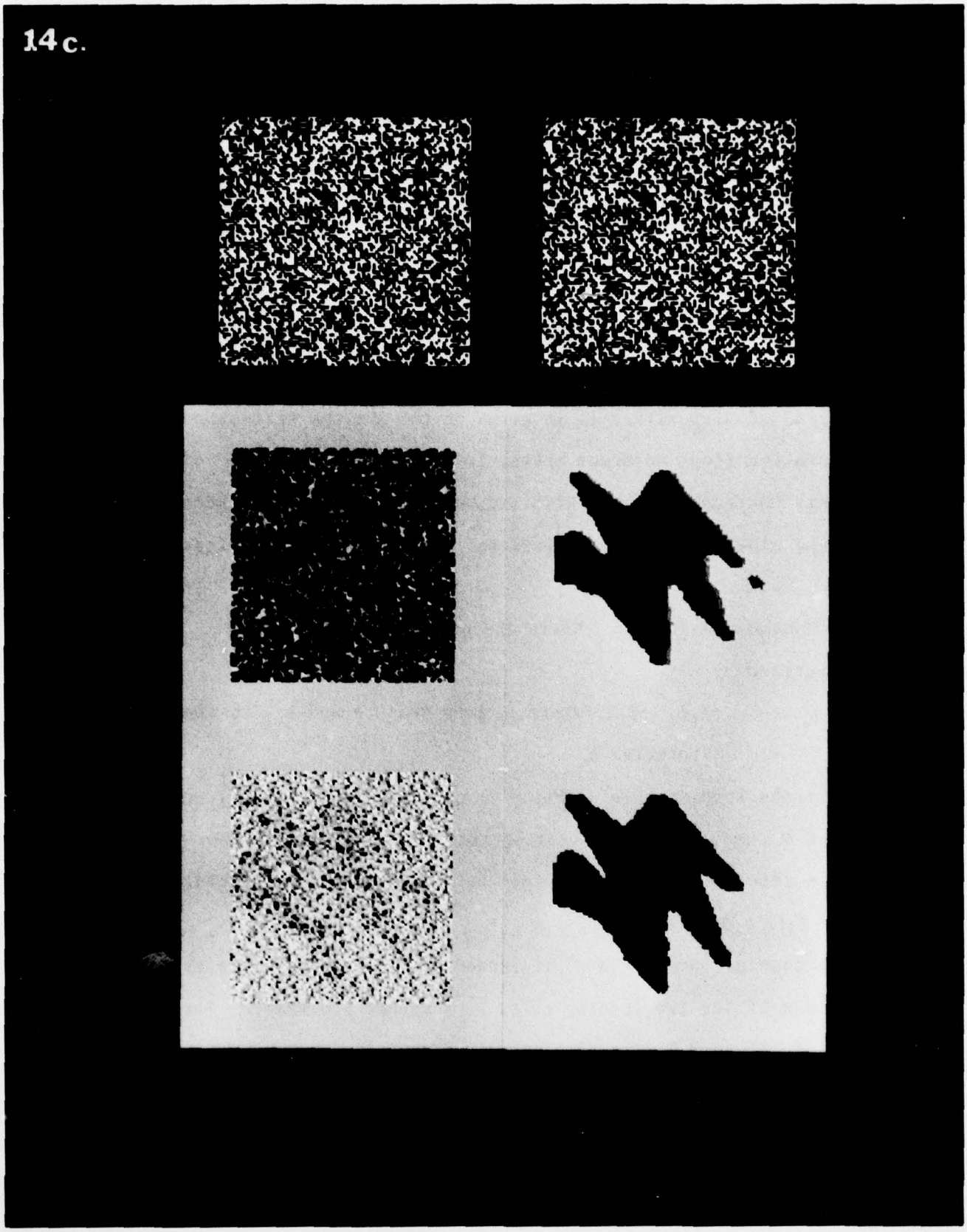


14. The behavior of the algorithm with modified parameters. The densities are 50% (14a) and 25% (14b). The parameters are as stated in section 6.3, and in table 3. 14c compares the two sets of parameters on a stereogram of a star that contains arms of various angles. The original parameters tend to give a more accurate final configuration.

14b.



14c.



least three excitatory neighboring cells. This allows a 45 degree corner.

6.10.3 Minimum size vs. disparity. Again because of rule *6.5.1*, the minimum resolvable area of a small pattern against a background does not depend on disparity. It is given by the minimum self-supporting configuration (four adjacent cells, from rule *6.5.2*).

This contrasts sharply with the property discussed in section *5.4*, where the minimum size has a characteristic dependence on disparity.

6.10.4 Loading conditions. While the present loading rule is characterized by

6.10.4.1 $f_i f_j = \delta_{ij}$, where f_i is a feature and δ_{ij} is the Kronecker δ,

the previous loading rule (section 2) can also be characterized by equation *6.1.2*, with the convention that δ_{ij} is also zero when either i or j are zero. In other words the "null" feature has a special status ($f_0 f_j = f_j f_0 = 0$, all j).

In case of two-valued (0,1) random dot stereograms, the choice of either one of the two loading rules is somewhat arbitrary. For densities around 0.5, the straight equation *6.10.4.1* seems to make more sense, since the black and white dots play equivalent roles. This is not clear, however, at very low densities (nor at very high ones).

In the case of natural images, more than two feature types have to be used (for instance, lines and edges at various orientations). In

this case, however, not every point is labelled with a corresponding feature; the absence of any feature at a given point is a common event. The null feature seems to have a basically different role from the other features. These arguments clearly support the loading conditions used in the first part of the paper (see Marr & Poggio 1976). It is clear, on the other hand, that both loading conditions may work. For both, an increasing number of "feature types" implies of course an increasingly better algorithm convergence. The choice between them depends in the end on the typical feature densities that one wants to deal with. For natural images, quantitative estimates have only recently become possible (Marr 1976, 1977).

7 Discussion

7.1 Alternative algorithms

The algorithm eq. *1.1.1* can be modified in various ways. One can adopt alternative loading rules for the network as in section 6, and one can vary the parameters over a substantial range. Such apparently minor changes can cause considerable changes in the network's behavior, but often without changing the end result (see for instance section 6), because they still implement the same computational constraints.

If the geometry of the local interactions (i.e. the shape of the excitatory and inhibitory neighborhoods) is changed, the network will in general implement a different computation, because the local

constraints will have changed. If only the parameter values are changed, our analysis (section 3) may still apply. If the geometry is changed, our analysis will in general become irrelevant.

Interestingly, for a specific stereogram density, a non-iterative version of our algorithm can recover disparity satisfactorily (see fig. 14a iteration 1). John Fairfield (personal communication) suggested an algorithm in which (1) excitation is summed independently within each disparity layer, and (2) for each position, one selects only the most excited of the cells in the different disparity layers. This algorithm performs well for the case $\nu = 0.5$.

7.2 Comments on analyzing such operations

We find the style of analysis that we were forced to adopt to be unsatisfactory for a number of reasons. Firstly, although our arguments appear to provide a qualitatively accurate description of the algorithms's behavior, the arguments are not completely rigorous. The main reasons for this lie in the difficulty of assessing the validity of the randomness assumptions that are necessary for the probabilistic analysis; and, to a lesser extent, in the need to examine a number of special cases in order to establish the stability of various solutions.

Secondly, our analysis is very specific to the particular algorithm and the particular parameters. This style of proof cannot lead to any general results about the convergence of such operators.

In order to overcome the first of these problems one can follow

the approach of section 6. The price one pays is that the analysis is valid for a narrower parameter range, which happens not to include the original parameter values (see Marr & Poggio 1976). The difficulty with the assumption of randomness arises because of the constant spatial structure of the operator Ξ (eq. (1) and fig. 2c of Marr & Poggio 1976). It should perhaps be noted that this objection does not apply to the similar analysis given by Marr (1971) of a cooperative associative memory algorithm because there the local operator had a variable and essentially random structure.

The second of these difficulties seems to be inherent in the nature of this type of cooperative algorithm. No general approach is at present available. Standard approaches[4] that we have tried have failed up to now. The flavor of the difficulties is the following. A configuration that is stable may be perturbed by changing a large number of cells without affecting its asymptotic state, provided that the perturbed cells are well scattered and interior. On the other hand, one fixed point of the algorithm can be shifted into another by perturbing only a few cells, provided that they have a suitable configuration. Thus, the usual distance between two configurations, namely the number of cells having different states, does not reflect the behavior of the algorithm. Therefore, the problem seems to be how to incorporate the geometry of the interactions into the metric distance between configurations.

It seems unlikely that one can construct a useful general theory of algorithms of the form

$$7.2.1 \quad C^{n+1} = \sigma\{L(C^n)\},$$

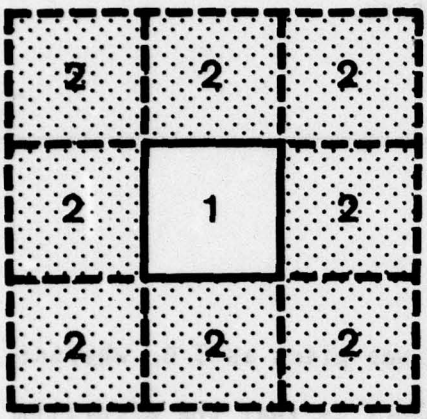
where L is a linear operator on the vector C, and σ is a nonlinear (coordinate-wise) function. J. H. Conway's game "Life" can, for example, be written this way (see fig. 15) and with an appropriate input pattern is Turing universal (unpublished result discovered independently by J. H. Conway and R. W. Gosper).

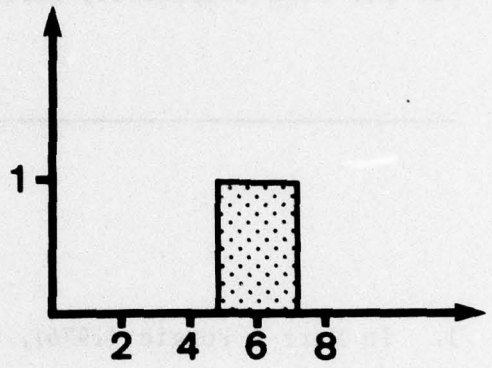
This suggests that theories of this type of algorithm must take due account of the structure of the input data and will probably be restricted to very specific forms of eq. *7.2.1*.

A mathematical understanding of the behavior of eq. *7.2.1* would represent a breakthrough of rather general importance. Cooperative phenomena similar to those which can be described by eq. *7.2.1* are important in physics (Haken 1977, Kawasaki 1972, K. G. Wilson 1975), in development (Mostow 1975), and in biology (Eigen 1971, Marr 1971, Richter 1976).

Furthermore, such a theory might also allow one to synthesize in a standard way cooperative algorithms of the form of eq. *7.2.1* from an analysis of the constraints on a computation.

Acknowledgements: We thank K. P. Hadeler and W. Reichardt for interesting discussions, Karen Prendergast for preparing the illustrations, and the Mathlab Group for allowing us to run the probabilistic theory in MACSYMA, the M.I.T. symbolic algebraic





15. Conway's game "life," which is played on an infinite plane square lattice, may be represented in a manner very similar to that of our stereo operator. The excitatory neighborhood, together with appropriate weights, is shown in 15a, and the threshold function appears in 15b. This combination reproduces the rules of life exactly, and these are;

(1) A cell will die at generation $n+1$ if < 2 or > 3 of its 8 neighbors are alive at generation n (death by starvation or overfeeding).

(2) A cell with exactly 2 living neighbors at generation n will be alive at generation $n+1$ if and only if it is alive at generation n.

(3) A cell with exactly 3 living neighbors at generation n will be alive at generation $n+1$.

manipulation system. Pirooz Vatan helped with MACSYMA programming. *Science* kindly gave permission for the reproduction of figure 1. D. M. thanks the Max-Planck Institut fur Biologische Kybernetik in Tubingen for its kind hospitality during his visit there.

Footnotes

1. In Marr & Poggio (1976), the value of θ was given as 3.0, whereas here it is 4.0. The reason for the discrepancy is that the algorithm used to produce the stereograms for that article essentially used the condition $> \theta$, whereas here, we use the condition $\geq \theta$.

2. A configuration is "spatially stable" if it is in some sense invariant under small perturbations (for instance each active point can be required to belong to a 3 x 3 neighborhood of points with the same disparity).

3. There is a *hole* in the network for a given y if there exist two intersecting lines of sight neither of which contains an "on" cell.

4. The continuous version of the algorithm eq. 1.1.1 cannot be described in terms of a potential dynamics. In fact the dynamical system

$$C = \sigma\{\Xi C\} - C = f(C) \qquad (\sigma \text{ a "smooth" threshold})$$

does not admit a scalar potential function $V(C)$ such that

$$[f(C)]_{x,y;d} = \partial V(C)/\partial C_{x,y;d}.$$

A necessary condition for this to be true is that

$$\frac{\partial}{\partial C_{x,y;d}} f_{x'y'd'}(\underline{C}) = \frac{\partial}{\partial C_{x',y';d'}} f_{xyd}(\underline{C}), \quad \text{all } x,y,d,x',y',d'$$

This is not true in general, because of the nonlinearity σ (consider the case in which xyd and $x'y'd'$ are on the same disparity layer and are reciprocally excitatory).

References

Eigen, M. (1971) Self-organization of matter and the evolution of biological macromolecules. *Naturwissenschaften 10*, 465-523.

Haken, H. (1977) *Synergetics.* Heidelberg: Springer-Verlag.

Kawasaki, K. (1972) Kinetics of Ising models. In *Phase transitions and critical phenomena, Vol 2, Eds. Domb & Green, pp443-501. New York: Academic Press.*

Marr, D. (1971) Simple memory: a theory for archicortex. *Phil. Trans. R. Soc. B 262*, 23-81.

Marr, D. (1974) A note on the computation of binocular disparity in a symbolic, low-level visual processor. *M.I.T. A.I. Lab. Memo 327.*

Marr, D. (1976) Early processing of visual information. *Phil. Trans. R. Soc. B 275*, 483-524.

Marr, D. (1977) Representing visual information. *AAAS 143rd Annual Meeting, Symposium on Some Mathematical Questions in Biology*, February, (in the press). Also available as *M.I.T. A.I. Lab. Memo 415.*

Marr, D. & Poggio, T. (1976) Cooperative computation of stereo disparity. *Science 194*, 283-287.

Mostow, (1975) *Mathematical models for cell rearrangement.* Newhaven, Conn: Yale University Press.

Richter, P. H. (1976) The network idea and the immune response. In: *Theoretical Immunology*, Eds. G. I. Bell, A. S. Perelson & G. H. Pimbley. New York: M. Dekker.

Wilson, H. R. (1977) Hysteresis in binocular grating perception: contrast effects. *Vision Res.* (in the press).

Wilson, H. R. & Cowan, J. D. (1973) A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik 13*, 55-80.

Wilson, K. G. (1975) The renormalization group: Critical phenomena and the Kondo problem. *Reviews of Modern Physics 47*, 773-840.