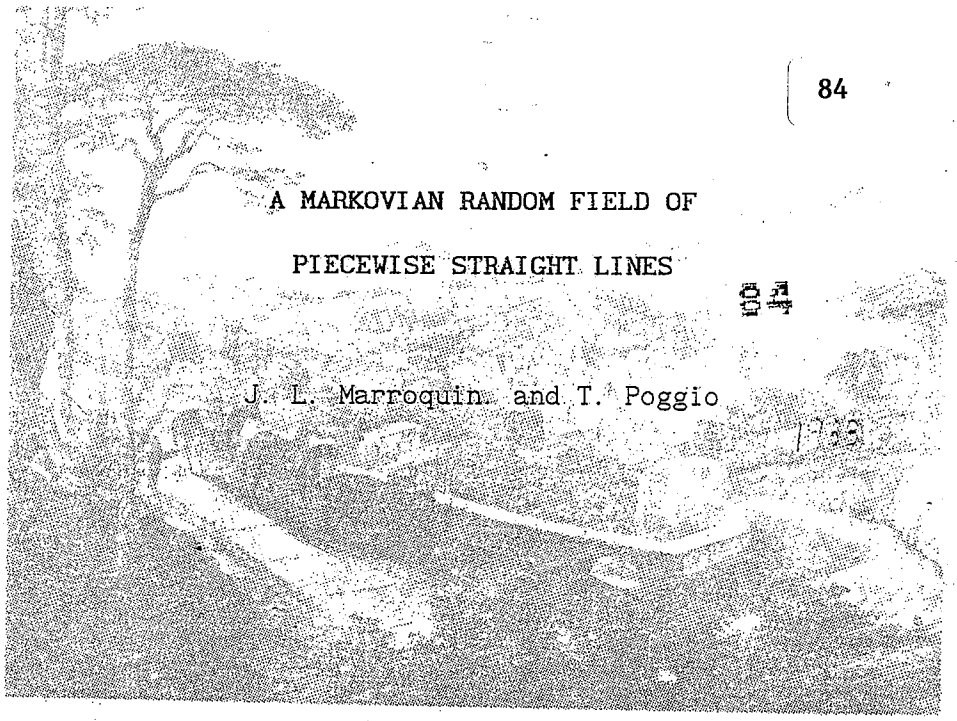
COMUNICACIONES DEL CIMAT



A MARKOVIAN RANDOM FIELD OF PIECEWISE STRAIGHT LINES

J. L. Marroquin and T. Poggio

CENTRO DE INVESTIGACION EN MATEMATICAS

Apartado Postal 402
Guanajuato, Gto.
México
Tels. (473) 2-25-50
2-02-58

"A MARKOVIAN RANDOM FIELD OF PIECEWISE STRAIGHT LINES"

J.L. Marroquin [*] and T. Poggio [**]

[*] Centro de Investigacion en Matematicas
Apdo. Postal 402; Guanajuato, Mexico.

[**] Massachusetts Institute of Technology.
Artificial Intelligence Laboratory.

Abstract : We present the construction of a vector-valued Markovian Random field on a finite lattice, whose equilibrium configurations consist of piecewise straight lines of arbitrary orientations, that uses only nearest neighbor interactions. For certain parameter values, this field presents a form of self organization, in which the lattice is partitioned into regions where particular line directions dominate. We also develop a stochastic cellular automaton (based on the Gibbs Sampler algorithm) that simulates this field. To illustrate the usefulness of this construction for the solution of computational vision problems, we present a simple application: the restoration of images that consist on incomplete contours.

0. Introduction.

A very useful framework for the formulation and solution of inverse (ill-posed) problems in computational vision is Bayesian Estimation theory (Marroquin et. al., 1987). Its use rests on the premise that, in order to solve any such problem (e.g., image restoration and segmentation; surface reconstruction, etc.), one must use prior knowledge about the nature of the solution, and that this knowledge may be expressed in the form of a probability distribution. If this is the case, one may then represent the solution as the minimizer of the expected value (with respect to the posterior measure) of an appropriate error criterion.

A class of probabilistic models that are very convenient for the construction of prior probability distributions, is the class of Markovian random fields (MRF's) on finite lattices (see Geman and Geman, 1984). These models are nice for several reasons:

(i) The global probability distribution on the set of all possible solutions f, may be specified simply by the local interactions (potential functions) between small sets (cliques) of neighbouring elements: the global measure is a Gibbs distribution:

$$P(f) = (1/Z) \exp [- U(f)] \qquad (1)$$

where the "energy" term $U(f)$ is computed as the sum, over all cliques, of the corresponding potential functions (Z is just a normalizing constant). The fact that one has an explicit representation for this global distribution, greatly facilitates the theoretical characterization of the solutions found by this method.

(ii) The behavior of a MRF may be simulated by a regular Markov chain, whose states correspond to global configurations of the field, and whose invariant measure is the corresponding Gibbs distribution. The Markov chain associated with the posterior distribution (which is also Gibbsian) may be implemented by a distributed algorithm (e.g., the Metropolis or Gibbs Sampler algorithms). This algorithm specifies a stochastic cellular automaton from whose evolution one may obtain the optimal solution (see Marroquin, 1985).

In this paper, we describe the construction of a MRF on a square lattice, whose equilibrium configurations correspond to images with long, piecewise straight lines of arbitrary orientations. This field may be used for several purposes; the main application that we have in mind, is to model the properties of the projections of the boundaries of objects, and therefore, to use it as an important component of a system that aims at the integration of several, qualitatively different processes (that analize, for example, intensity edges, stereo, motion, color and texture information) for the purpose of reconstructing such boundaries (see Poggio and Gamble, 1987). In this paper, however, we will only present, for illustrative purposes, a very simple application: the restoration of images that consist of incomplete contours.

The use of MRF's for the explicit modelling of the behavior of contours in an image, was first introduced by Geman and Geman (1984). They used a "line field" (defined on a lattice dual to the

one of the "intensity" field) formed by line segments of four possible orientations (i.e., a 5-state field), and whose function was to decouple adjacent intensity sites in an image restoration task. With small variations, a similar technique has been used by several researchers for other tasks, such as: surface reconstruction from sparse data (Marroquin et. al., 1987); segmentation of textured images (Geman et. al., 1988), etc.

This approach, however, has a serious limitation: if the contour information (or equivalently, the intensity data) is sparse, this line field will not fill large gaps in the contours in an appropriate way, regardless of the value of the parameters. This fact is also reflected in the nature of the sample configurations produced by the "free" line field (i.e., without the coupled intensity data); no matter which parameter values are selected, it is not possible to produce a field of long, straight lines of arbitrary orientations. The reason for this failure is that the only true straight lines that exist in a square lattice are those at 0°, 90° and $\pm 45^{\circ}$; lines in any other direction are really piecewise horizontal, vertical or diagonal (see fig. 1), and the difference between two such orientations cannot be captured by the local nature of the potentials associated with Geman's line field.

Figure 1. A straight line with slope = 1/3 : at a microscopic level, this line segment is either horizontal or diagonal.

To get the desired behavior, it is necessary to include, in the state of each line element, information about the macroscopic direction in which the straight line is supposed to go. This idea may be implemented by a vector valued MRF (or equivalently, by two coupled, scalar valued fields); the first component (which we will call the "Connection" or "C" field) describes the local connectivity of the line elements, while the second one (the "D" field), specifies the macroscopic direction of the line segment. The values of both components enter into the computation of the potential functions: the macroscopic direction information affects the conditional probabilities associated with the different configurations of the C field.

The Markov chain that simulates the behavior of the field may be constructed in several ways: the best known is probably the Metropolis algorithm (Metropolis et. al., 1953); here, we will use the Gibbs Sampler (Geman and Geman, 1984) , that consists in updating the value of one site of the field at a time (although non-neighboring sites may be updated in parallel), using, to select the new value, the conditional probability distribution, given the values of the field at neighboring sites (this distribution may be readily obtained from the global Gibbs measure of equation 1). A detailed description of the algorithm is given in the appendix. As we mentioned above, from the evolution of the stochastic cellular automaton defined by this algorithm, one may obtain sample configurations of the field, and, by appropriately coupling the observations, if they are present, perform the desired reconstructions. We will now describe the construction of this complex field in detail.

1. The Connection Field.

This is a discrete valued field on a square lattice. The state of each element of the field may take values on a set Q of 21 elements, which describes the permissible connections between neighbors. This set is represented in figure 2.

Figure 2. The set Q of states for the connection field.

The neighborhood of each element consists on its 8 nearest neighbors, and the only cliques with non-zero potentials are those of size 1 and 2 (see figure 3).

Figure 3. (a) The neighborhood of element i. (b) - (f) Cliques with non-zero potentials for the connection field.

1.1. Labeling of Neighbor Interactions.

We consider the following types of neighbor interaction:

i) Indiference ('o') : It has associated a potential value of zero.

ii) Termination ('e') : It represents the interacion of a state connected with state 0.

iii) T junction.

iv) Sharp turn.

v) Straight continuation ('a').

vi) Right turn ('b', 'j', 'r' and 'f').

vii) Left turn ('d', 'k', 'l' and 'g') : Different letters denote different types of turn (see figure 4).

viii) Complete a turn ('c').

ix) Forbidden connection ('-').

Examples for each type of interaction appear in figure 4.

Figure 4. Examples of each type of neighbor interacion for the connection field.

The complete set of labels for all possible neighbor interactions is specified in four 21 X 21 matrices (one for each clique type). The matrices for clique types (b) and (d) are shown in figure 5.

Figure 5. Interaction matrices for clique types (b) (top) and (d) (bottom). The matrices for types (c) and (e) are obtained from symmetry considerations.

These interacion types are used to define the potential functions for the field, by associating a real number with each type,

and with each pair of values of the D component. Before defining these functions, we need to describe the structure of the D field in some detail.

2. Macroscopic Direction Component.

Conceptually, this is a real valued field, whose elements take values in the set $\mathcal{D} = [0,\pi) \cup \{-1\}$. The state of a D element (say, D_i) corresponds to the macroscopic direction of a line (if $D_i \geq 0$), or to an undefined direction (if $D_i = -1$).

Note that, neither every state of the C field ("C state", for short), nor every interaction type for neighboring C states ("C interaction type") are compatible with a given value of the macroscopic direction. In fact, it is possible to divide the interval $[0,\pi)$ into subintervals that are homogeneous, in the sense that there is a unique set of C states and of C interaction types that are compatible with every direction in the subinterval. Once a subinterval is selected, a particular direction may be specified by the relative probability of selecting a straight continuation (C states in [1,4]) versus a right (or left) turn (C states in [5,12]). In table 1 we show the homogeneous subintervals, and the compatible C states and interaction types. Also in this table, we indicate the formula that relates the ratio:

α = Prob. of selecting a turn / Prob. of straight continuation

to the slope m of the macroscopic line.

Table 1 . Homogeneous Subintervals for the Macroscopic Direction.

The columns s t c refer to: "straight continuation", "turn" and "complete a turn" types of C interaction, respectively

#	Subinterval (in terms of slopes)	Compatible C states	Compatible int. types			α
			s	t	c	
0	0	2	a			0
1	(0,1/2]	2,10,11	a	d,l	c	m / (1-2m)
2	(1/2,1)	3,10,11	a	j,c	l	(m-1) / (1-2m)
3	1	3	a			0
4	(1,2]	3,5,8	a	k,c	f	(m-1) / (2-m)
5	(2,∞)	1,5,8	a	b,f	c	1 / (m-2)
6	∞	1	a			0
7	(-∞,-2]	1,6,7	a	d,g	c	-1 / (m+2)
8	(-2,-1)	4,6,7	a	j,c	g	-(m+1) / (m+2)
9	-1	4	a			0
10	(-1,-1/2]	4,9,12	a	k,c	r	-(m+1) / (2m+1)
11	(-1/2,0)	2,9,12	a	b,r	c	-m / (2m+1)

3. Potential Functions.

We will now define the potential functions V_1 and V_2 associated with the (vector valued) states of a single site and a pair of neighboring sites i,j, respectively.

First, we need some notation: Since the C states take values in $\mathcal{Q}$ (figure 2), and the D states in $\mathcal{D}$, we have that V_2 maps $\mathcal{Q} \times \mathcal{D} \times \mathcal{Q} \times \mathcal{D}$ into the reals. Let (q_i, d_i) and (q_j, d_j) be the states of the neighboring sites i,j.; Let $I_{ij} = I(q_i, q_j)$ be the C interaction type corresponding to (q_i, q_j) . A given direction d_i will fall in a unique homogeneous subinterval; one can see from table 1, that for every such subinterval, there is either a single compatible C interaction type (namely, 'a'), or four such types:

one that corresponds to a straight continuation choice, which we call $SC_i = SC(d_i)$; a set of two that correspond to a "turning" choice, which we call $\mathcal{T}_i = \mathcal{T}(d_i)$, and one that corresponds to completing a turn, which we call $CT_i = CT(d_i)$. The set of C states compatible with direction d_i (also from table 1) is denoted by $\mathcal{C}_i = \mathcal{C}(d_i)$. Finally, $\alpha_i = \alpha(d_i)$ denotes the ratio:

$$\alpha_i = Pr(\text{selecting } q \in \mathcal{T}_i) \,/\, Pr(\text{selecting } q = SC_i)$$

which is computed using the appropriate formula from the last column of table 1.

For some constants: v_c , v_t , v_e we may now define the single site potential function V_1 as :

$$V_1(q_i,d_i) = \begin{cases} \infty, & \text{if } d_i \geq 0, \text{ and } q_i \notin \mathcal{C}(d_i) ; \\ \ln[1+\alpha_i], & \text{if } d_i \geq 0 \text{ and } q_i \in [1,4] \cap \mathcal{C}(d_i) ; \\ \ln[(1+\alpha_i)/\alpha_i], & \text{if } d_i \geq 0 \text{ and } q_i \in [5,12] \cap \mathcal{C}(d_i) ; \\ 0, & \text{otherwise.} \end{cases}$$

This definition of V_1 serves two purposes: first, it enforces the compatibility between the C and D states of a site (making $V_1 = \infty$ if they are incompatible; note that the C states in [1,12] are the only ones that may have associated a definite direction). Besides, it enforces the growth of a line along a given macroscopic direction, by controlling the relative probability of selecting a straight continuation (i.e., $q \in [1,4]$) versus a "turning" C state ($q \in [5,12]$).

The potential V_2, for cliques of size 2 is defined as:

$$V_2(q_i, d_i, q_j, d_j) = 0, \text{ if } I_{ij} = \text{'o'} ;$$

$$v_e , \text{ if } I_{ij} = \text{'e'} ;$$

$$v_c , \text{ if } d_i = d_j \text{ and}$$

$$q_i, q_j \in \mathcal{C}_i \text{ and } I_{ij} \in \{SC_i\} \cup \mathcal{T}_i ;$$

$$v_c - \ln [(1 + \alpha_i) / \alpha_i] ,$$

$$\text{if } d_i = d_j \text{ and } q_i, q_j \in \mathcal{C}_i \text{ and } I_{ij} = CT_i ;$$

$$v_t , \text{ if } I_{ij} \notin \{\text{'o'},\text{'e'},\text{'-'}\} \text{ and}$$

$$d_k < 0 \text{ and } q_k > 0 , \text{ for some } k \in \{i, j\} ;$$

$$\infty , \text{ otherwise.}$$

This definition of V_2 serves the following purposes:

i) It assigns the values of v_e and 0, respectively, to terminations and "indiferent" C interacions (note that in this cases the potential value is independent of the D component).

ii) If the C interaction corresponds to the continuation of a macroscopic line, it forces the D component to propagate along it, and assigns a value of v_c to V_2 . If the C interaction corresponds to the completion of a turn (i.e., if $I_{ij} = CT_i$), it ensures that $V_1 + V_2 = v_c$.

iii) It assigns a value of v_t to C interacions that correspond to sharp turns or "T" junctions. It also forces, in these cases, the D state of the connecting element to - 1.

iv) It assigns an infinite potential to forbidden interacions ($I_{ij} = \text{'-'}$).

The quantity $\exp [- v_c]$ represents the conditional probability of growth of a macroscopic line, given the states of its neighbors (remember that in the Gibbs sampler algorithm, the probability of accepting a state is proportional to the product of the exponentials of the values of the potentials, for all the cliques to which the element belongs). The quantities:
$\exp [\ln [\alpha_i / (1 + \alpha_i)] - v_c]$ and $\exp [\ln [1 / (1 + \alpha_i)] - v_c]$ represent the conditional probabilities of turning and straight continuation, respectively. Note that their sum is $\exp [- v_c]$, and their quotient is α_i , as it should.

The constants v_e , v_t and v_c are free parameters that control the behavior of the system. As we have mentioned, v_c controls the probability of growth of the lines, so that smaller values of v_c will produce configurations with longer lines; v_e controls both the stability of open contours and the probability that new lines are born, and v_t controls the probability of sharp turns and "T" junctions. These parameters, not only control the dynamic (transient) behavior of the system, but also, since the evolution of the Gibbs sampler automaton is ergodic, they determine the properties of the equilibrium configurations: the relative probability of finding a configuration (C,D) in equilibrium conditions, will be proportional to:

$$\exp [-1 / T \sum_{i,j \,:\, \|i - j\| < 2} V_2(C_i, D_i, C_j, D_j) + \sum_i V_1(C_i, D_i)]$$

Here T (the "temperature") is an additional parameter that is usually introduced to control the overall activity of the system

(note that T affects the relative magnitude of the effective potential values: at low temperatures, the relative differences between these values are exaggerated, while at high values of T, they tend to become uniform).

4. A Practical Implementation.

The continuous valued nature of the D component of the field prevents a direct implementation of the Gibbs sampler algorithm for the generation of sample configurations (and makes the convergence of, say, Metropolis algorithm extemely slow). It is necessary, therefore, to sample the interval $[0,\pi)$, and use a discrete set $\mathcal{D}'$ of allowable directions. This discretization makes sense, since in a finite lattice, one has a finite repertoire of straight lines, anyway. Even with this discretization, a direct implementation of the Gibbs sampler would still be very inefficient. A closer look at the properties of the potential functions, however, indicates that the only pairs $(q,d) \in Q \times \mathcal{D}'$ that have to be considered when updating, say, site i (i.e., the pairs that have associated a finite potential) are the following:

i) Pairs of the form $(q,-1)$, when $q = 0$, $q > 12$, or when q is the connecting element between lines of different orientations.

ii) Pairs of the form (q,d_o), with $q \in \mathcal{C}(d_o)$, which occur when q corresponds to the continuation of a unique macroscopic line of direction d_o which is already active in the neighborhood of i.

iii) Pairs (q,d), where d is selected at random from the set $\mathcal{R}(q)$ = { d : q ∈ $\mathcal{C}$(d) } ∪ {-1}.

In this last case, which occurs when $I(q,f_j) \in \{'o','e'\}$ for *all* j in the neighborhood of i, we note that the value of V_2 is independent of the choice of d; the conditional probability p(q) for the selection of a given C state q in the Gibbs sampler update is in this case :

$$p(q) = (1/Z) \sum_{d\,:\,q \in \mathcal{C}(d)} \exp\left[-\sum_{j \in Ni} V_2(q,C_j)\right] \exp\left[-V_1(q,d)\right]$$

$$= (1/Z)\, W(q) \exp\left[-\sum_{j \in Ni} V_2(q,C_j)\right]$$

where Ni is the neighborhood of site i; Z is a normalizing constant, and the weights

$$W(q) = \sum_{d\,:\,q \in \mathcal{C}(d)} \exp\left[-V_1(q,d)\right]$$

depend only on q, and hence, they may be precomputed (in table 2 we show W(q) for a direction sampling density of 1 sample every 3 degrees; note that for q = 0 or q > 12, W(q) = 1, since the only compatible direction is - 1, and $V_1(q,-1) = 0$).

This means that it is possible to select q independently of d, using a weighted Gibbs sampler. After, say, q = r is selected, if r = 0, or r > 12, d is set to - 1; otherwise, d is picked at random from the set $\mathcal{R}(r)$ with distribution

$$P(d) = \exp\left[-V_1(r,d)\right] / \sum_{d' \in \mathcal{R}(r)} \exp\left[-V_1(r,d')\right]$$

Table 2. Weights W(q) for a direction sampling density of 1/3 samples/degree.

q	W(q)
0	1.0
1	5.8
2	5.8
3	5.9
4	5.9
5	9.14
6	8.14
7	8.14
8	9.14
9	9.14
10	8.14
11	8.14
12	9.14
13	1.0
14	1.0
15	1.0
16	1.0
17	1.0
18	1.0
19	1.0
20	1.0

The above remarks mean that it is possible to simulate the behavior of the complete Gibbs sampler algorithm using a cellular automaton that updates its state using a stochastic rule that depends on the total number of occurances of each type of interaction ('t', 'e', 'c', etc.) in the neighborhood of site i, for each $q \in Q$ (the automaton is essentially equivalent to a weighted Gibbs sampler for the appropriate (q,d) pairs, followed, if necessary, by a second Gibbs sampler that uses P(d) for the selection of d; the construction is straight forward, and we omit the details here). With this algorithm it is possible to obtain sample configurations of the field in an efficient way. Its

worst case complexity depends on the size of the set Q plus the maximum size of the set $\mathcal{R}(q)$, which is approximately 0.25 times the cardinality of the set $\mathcal{D}'$.

The efficiency of this implementation may be increased if the quantities : $\exp[-v_e]$, $\exp[-v_c]$ and $\exp[-v_t]$ are precomputed, and if the D component is replaced by a vector formed by the subinterval number and the probability of turning (i.e., instead of D_i, one stores the number N_i of the subinterval to which D_i belongs, and the number $P_i = (\alpha(D_i) / (1 + \alpha(D_i)) \exp[-v_c])$, for each site i. If this is done, the functions V_1 and V_2 reduce to simple table lookup procedures.

5. Macroscopic Order.

The field that we have described has an interesting property : when one generates sample equilibrium configurations of the free field, one observes that, for certain parameter values, the lattice is partitioned into domains, where one particular macroscopic direction dominates all the others (see figure 6; a toroidal lattice was used for this experiment, to avoid the edge effects).

Figure 6. Sample equilibrium configuration of the C component of the field on a toroidal lattice (a black pixel at site i indicates that $C_i > 0$).
Parameter values : $v_c = -0.5$; $v_e = 4$; $v_t = 5$.

Heuristically, one can explain the formation of these patterns as a consequence of the fundamental property of macroscopic systems in thermal equilibrium (Reif, 1965) ; the global equilibrium states (C,D) will be such that the Gibbs free energy F is minimized. At temperature T, F is given by:

$$F(C,D) = U(C,D) - T \ln \Omega\,(U(C,D))$$

where the energy U is:

$$U(C,D) = \sum_{i,j\,:\,\|i-j\|<2} V_2(C_i,D_i,C_j,D_j) + \sum_i V_1(C_i,D_i)$$

and the "degeneracy" $\Omega(U)$ is the total number of possible configurations with energy equal to U (the quantity $\ln \Omega(U)$ is called the "entropy"). If the parameters are selected in such a way, that the formation of macroscopic lines lowers the energy (in particular, if $v_c < 0$), the system in equilibrium will adopt a global configuration that permits the packing of a large number of lines (i.e., patterns with blocks of nearly parallel lines), while retaining a relatively high degeneracy (obtained by shifting these blocks around). A precise explanation is not trivial, and we will not attempt it here; the point that we wish to make is that a line in this field can *induce* its direction to neighboring lines by a mechanism that cannot be predicted by direct extrapolation from the local interactions, but rather, that emerges as a global property of the whole system. We believe that, apart from the practical applications of this property, the detailed study of this phenomenon may have some implications for physics and biology.

The subtlety of the mechanism that produces regions of uniform direction, makes this field very sensitive to cues that indicate preference for a given orientation. This prefered orientation may be induced in the field, for example, by a "flow pattern" (Glass and Perez, 1971), or by the boundaries of regions of elongated shape (see figures 7 and 8), which suggests the important role that this kind of fields may play in the construction of models for the perception of texture, and of oriented patterns.

Figure 7. (a) Elongated region. (b) Equilibrium configuration of the free C field when the growth of lines is restricted to the interior of the shaded region in (a). Parameter values: v_c = -0.5; v_t = 4 ; v_e = 5. (c) Histogram of the values of the D field during 3000 iterations. (d) direction corresponding to the largest peak of the histogram.

Figure 8. (a) "Flow pattern" obtained by superposition of two shifted versions of a set of random dots. (b) Equilibrium configuration of the C field when the observations (a) are coupled to the field using the scheme described in section 6; parameter values : v_c = -0.5 ; v_t = 4 ; v_e = 5 .

Other parameter values produce patterns with different properties.

In figure 9 we show a couple of them. The parameter values that should be selected will depend, of course, on the particular application.

Figure 9. Sample configurations of the C field. Parameter values : (a) $v_c = 0.1$; $v_e = 6$; $v_t = 3$; (b) $v_c = .1$; $v_e = 5$; $v_t = 2$.

---

6. An Application.

As we mentioned in the introduction, the complex field that we have described may be used for a variety of purposes. In this section we present, as an illustration, a simple application: the restoration of an image that consists of incomplete, piecewise straight, closed contours (or equivalently, a model for the formation of the subjective contours that appear when one looks at such an image; see figure 10).

Figure 10. Two images consisting on incomplete contours.

6.1. Coupling the Observations to the (C,D) Field.

The model for the observation process g, is based on the assumption that the observations represent incomplete contours, which means that, although some existing pieces may be missing (which happens with a certain probability), there are no spurious contours present (i.e., the probability that an observation $g_i > 0$, given that $C_i = 0$ is zero). This type of model generates a strong coupling term in the posterior distribution (see Marroquin,

1985, pp 90 - 92), that will force the C field to take a non-zero value (and the D component a compatible value), in all sites where an observed contour is present, in any sample configuration of the field generated by the posterior distribution. Note, however, that since the only thing that we are observing directly is the fact that a contour piece is present or abscent in a given site (i.e., g is a binary field), the compatible (C,D) values must be computed from the observations before the Gibbs sampler automaton is initialized (we will call these computed compatible values (C',D'), the "pseudo-observations").

6.2. Computation of the Pseudo-observations.

The C' component may be obtained directly from g by a non-linear, shift invariant filtering operation: this filter simply looks at configurations of the g field through a moving 3 X 3 window, and looks up the corresponding value for C' in a table. If we number the pixels of a 3 X 3 window in the form shown in figure 11-a, any configuration of the g field within the window may be expressed as a sequence of (at most 8) digits, that indicate the places where $g_i = 1$ (for example, the configuration (6,8,1) is shown in figure 11-b).

Figure 11. (a) Numbering system for the pixels of a 3 X 3 window. (b) Configuration (6,8,1).

Table 3 shows the configurations that correspond to non-zero C' states. Note that only those configurations that might correspond

to macroscopic lines are assigned a C' state; "connecting" configurations (i.e., those that would correspond to values of C' > 12) are assigned a C' value of 0.

Table 3. Configurations of 3 X 3 windows of the g field that correspond to non-zero C' states.

C' state	Configurations
1	(2,6) ; (2,8,6) ; (1,2,6) ; (7,2,6)
2	(0,4) ; (8,0,4) ; (5,0,4) ; (3,0,4)
3	(1,5) ; (8,1,5) ; (0,1,5) ; (6,1,5)
4	(3,7) ; (8,3,7) ; (0,3,7) ; (2,3,7)
5	(1,6) ; (8,1,6) ; (0,1,6) ; (8,0,1,6)
6	(3,6) ; (8,3,6) ; (2,3,6) ; (8,2,3,6)
7	(2,7) ; (8,2,7) ; (0,2,7) ; (8,0,2,7)
8	(2,5) ; (8,2,5) ; (6,2,5) ; (8,6,2,5)
9	(4,7) ; (8,4,7) ; (0,4,7) ; (8,0,4,7)
10	(1,4) ; (8,1,4) ; (0,1,4) ; (8,0,1,4)
11	(0,5) ; (8,0,5) ; (6,0,5) ; (8,6,0,5)
12	(0,3) ; (8,0,3) ; (2,0,3) ; (8,2,0,3)
0	all others.

The computation of D' is complicated by the fact that we pointed out in the introduction, namely, that the macroscopic direction of a line in a square lattice is not a local property. This means that if this computation is to be realized with operators of local support (which is desirable, both for theoretical and practical reasons), some iterative propagation must take place. We will now present a scheme that uses a combination of non-linear filters and deterministic cellular automata.

We said in section 4, that the efficiency of the modified Gibbs sampler automaton is increased if the D component is represented as a vector (N,P), formed by the subinterval number in which D falls, and by the conditional probability of selecting a turning

configuration. This representation is also more convenient for the computation of D' (which becomes (N',P'), as we now show.

The idea is to use a three stage process:

1: Initialize the N' field : This step is implemented with a non-linear filter similar to the one used to obtain C', that acts on C' itself: by examining C' configurations on 3 X 3 windows, one can assign, either a definitive value, if a configuration that corresponds to a unique N' is found (such as the one of figure 12-a) or a tentative one, if the configuration is compatible with more than one value of N' (as in the case of figure 12-b ; tentativeness is signaled by a flag).

Figure 12. The non-linear filter assigns a definitive value N' = 5 to the configuration (a) of C', and a tentative value N' = 0 to (b) (see text).

The table lookup procedure is more complicated than before, because the C' field is not binary. The construction, however is straight forward, and we will not go into the details here.

2: Propagate the N' field : The field obtained in step 1, is used to initialize a deterministic cellular automaton (on an 8-connected lattice), that propagates the definitive values to neighbors that have compatible, tentative states (note that this automaton always converges in a small number of iterations). Upon convergence, the states that remain tentative are made definitive.

3: Initialize and propagate the P' field : The P' field is computed from the fixed state of a deterministic cellular automaton S on an 8-connected lattice. This automaton has access to the final state of the N' field. It is initialized using the rule:

$S_i^{(0)} = 0.5$, if N' is a "turning" state (i.e., if N' > 4) ;

0 , otherwise.

(note that the state of S is continuous valued). The automaton evolves by averaging the value of the state of each site, with that of neighbors that have the same value of N'. This means that the state is propagated along macroscopic line segments, distributing the total "amount of turn" among the elements that form the segment. It is not difficult to see that this automaton will always converge to a unique fixed point S^* (in practice, convergence is signaled when

$$\sup_i \left| S_i^{(t+1)} - S_i^{(t)} \right| < \theta$$

θ being a given tolerance). The conditional probability of a turn is related to S^* by the formula:

$$(\text{Pr. Turn})_i = \exp [- v_c] \, S_i^* / (1 - S_i^*)$$

To summarize : the pseudo-observations (N',P') may be computed from the observations g using a multistage process that involves two non-linear filtering steps, and the convergence of two deterministic cellular automata. It should be noted that the whole

process is quite fast, even on a serial machine (both cellular automata usually converge in less than 15 iterations), and that the algorithms are fully distributed, so that parallel implementations are direct.

6.3. Selection of the Parameter Values.

Before discussing how to select these values, a word of caution is in order: our approach is Bayesian in spirit (in the sense that one expresses prior knowledge in the form of a probability measure), and we believe the a rigorous probabilistic framework, like the one we are using, is useful, because it leads to the design of engines whose behavior (and hence, the results they produce) can be theoretically characterized. We think, however, that one should not go too far in the intent of preserving a rigorous statistical interpretation. In particular, we think that, unless one is in a situation where precise and realistic probabilistic models are available, it makes more sense to *adjust* the parameters of the system, so that it behaves in a desired way, rather than trying to estimate them in a strict statistical sense.

In this particular case, we have three free parameters : v_e, v_t and v_c. As we mentioned in section 3, v_e controls both the stability of open contours, and the probability of new lines being born. Since in the present case we are assuming a priori: first, that the contours should be closed, and second, that the only contours present are those of which the observations are fragments, v_e should be given a very high value, so that in

equilibrium, the only contours that survive are those that complete the observed fragments (we are using $v_e = 20$). v_t controls the probability of occurance of sharp turns or "T" junctions between lines. The value of v_t should also be high, unless one is interested in generating lines that ramificate and change abruptly their directions, which is not the present case (we are using $v_t = 7$) (we have found, in fact, that for the present application, the system is relatively insensitive to variations in the precise values of v_t and v_e, as long as $v_c << v_t << v_e$).

The selection of a value for v_c, which controls the probability of growth of line segments, is more delicate: if v_c is too small, the lines will grow too fast, and the appropriate connections may not be established; if it is too large, the convergence to equilibrium will be too slow. A value of $v_c = 0.2$ has worked very well for the experiments we have performed, but the optimal value may be different for other images.

Note that the same scheme that we have presented may be used to solve other related problems, with a different choice of parameter values; thus, for example, to obtain a field of lines that is organized by a "flow pattern" (as in figure 8), a good choice is: $v_c = -0.5$; $v_t = 4$; $v_e = 5$.

6.4. Experimental Behavior.

In figure 13 we show the result of applying the complete procedure to the incomplete contours of figure 10: first, the

pseudo-observations were computed as described, and then, the modified Gibbs sampler automaton for the coupled system was allowed to reach a stable equilibrium configuration at a fixed temperature T = 1. The values for the parameters were: v_e = 20 ; v_t = 4 and v_c = 0.2 in both cases.

Figure 13. Results of the application of the complete reconstruction algorithm to the incomplete contours of figure 8. Grey pixels correspond to the observed fragments, and black pixels to the reconstructed parts.

Notice that the selection of a high value for v_e causes the equilibrium behavior of the Gibbs automaton to consist, with very high probability, on a constant configuration, formed by closed contours. Since the system is evolving at a fixed temperature, this stable configuration may be interpreted as the MPM (maximizer of the posterior marginals) estimator of the field. This estimator minimizes the expected value (with respect to the posterior distribution) of the total number of errors commited in labeling each pixel as belonging (or not) to a closed contour (see Marroquin et. al., 1987).

The fact that the Gibbs chain converges, with high probability, to a stable state, increases the efficiency of the estimation process, since there is no need for the collection of statistics over a large number of iterations: once the system reaches its stable equilibrium state, this may be interpreted directly as the MPM estimator. The precise form of this stable configuration, however, may depend on the particular (stochastic)

transient path followed by the automaton. The robustness of the estimator may, therefore, be improved by collecting "ensemble" statistics of the configurations obtained with several such paths (note that this may be done in parallel, by simultaneously running an ensemble of similar automata; it may also be possible to introduce a dynamical coupling between these networks, in order to accelerate the overall rate of convergence: the fact that a given element is in a certain state in one of them, should increase the likelihood for the selection of this state in the whole ensemble. We are currently exploring these possibilities).

6.5. Time Varying Parameters.

The considerations of section 6.3 suggest a strategy in which v_c is varied dinamically as the system evolves, so that one gets a more robust behavior, and a faster convergence rate. There are several possible "annealing schedules" $v_c(t)$ (t denotes the iteration number) : it is possible, for example, to design

cyclical schedules that start and finish at a high value, and reach a small value at intermediate times. Another possibility, with which we have obtained very good results, alternates between constant low and high values, in cycles of increasing length (see figure 14)

Figure 14. "Annealing schedule" for parameter v_c (see text).

Each cycle consists thus in a "growing phase" (low v_c), where open lines grow, followed by a "pruning phase" (high v_c), in which all

the lines that remain open, disappear. One starts with short cycles, so that the finer connections between lines are properly established, and then increase the cycle length, so that longer gaps are filled in a reasonable time. Using this strategy, we have obtained results that are indistinguishable from the constant v_c case, in the order of 60 iterations.

7. Discussion.

Let us summarize the main results that we have obtained:

i) We have constructed a vector valued MRF whose equilibrium configurations consist on piecewise straight lines of arbitrary orientations. For certain parameter values, this field presents a form of self organization, in which the lattice is partitioned into regions where particular line directions dominate. This form of organization is very sensitive to (and may be induced by) patterns that display some preferred orientation, such as "flow patterns" or boundaries of elongated regions.

ii) We have designed a stochastic cellular automaton (a modified Gibbs sampler) that efficiently simulates this field.

iii) We have illustrated the use of this engine with a simple application: the reconstruction of incomplete, piecewise straight contours.

All the algorithms that we have presented are fully distributed,

which makes attractive their implementation in fine-grained parallel machines (such as the "Connection Machine", Hillis, 1985). Such implementations should greatly increase their computational efficiency and their practical value.

There are some straight forward extensions that may be done to increase the applicability of this construction: one may, with some minor modifications, define the (C,D) (or equivalently, the (C,N,P)) field on a lattice dual to the one where an associated "intensity field" is defined (see Geman and Geman, 1984), and use it for image restoration and segmentation (Geman et. al., 1988), as well as for surface reconstruction (Marroquin et. al., 1987) ; it is also possible to define more complex coupling schemes that involve several, qualitatively different types of observations, an application that is of particular interest to us. An interesting possibility is the construction of MRF's whose equilibrium configurations are piecewise *smooth* contours. We believe that this may be achieved with a scheme similar to the one presented here, with a more complex updating mechanism for the D component. This, however, remains very much an open problem.

Appendix. The Gibbs Sampler Algorithm.

Consider the vector valued MRF f, defined on a lattice L. Suppose that the state of each element f_i takes values on a finite set $Q = \{q_1, \ldots, q_N\}$. Suppose also, to simplify the notation, that the only cliques with non-zero potentials are of size one and two, and let $V_1(\cdot)$ and $V_2(\cdot,\cdot)$ denote the potential function associated with each clique configuration. If there are observations present (i.e., if we are using the posterior distribution to simulate the field), there will be, in the energy function, a coupling term $\Phi_i(f_i)$ for each site i (the form of these coupling functions will depend on the probabilistic model for the observations). The posterior distribution is:

$$P(f ; g) = (1/Z) \exp\left[-\sum_i \sum_{j \in N_i} V_2(f_i, f_j) - \sum_i (\Phi_i(f_i) + V_1(f_i)) \right]$$

where N_i is the neighborhood of site i (in our case, $N_i = \{j : \|i-j\| < 2\}$).

The Gibbs sampler algorithm (Geman and Geman, 1984), consists in updating the state of the sites of L, one at a time (although non-neighboring sites may be updated in parallel), using the conditional probability distribution for the states, given the states of its neighbors, for the selection of the updated value. Suppose that the state of site i is being updated. The conditional probability distribution w(q) is :

$$w(q) = \exp [-u_i(q)] \; / \; \sum_{r \in Q} \exp [-u_i(r)]$$

$$\text{where} \quad u_i(q) = \sum_{j \in Ni} V_2(q, f_j) + \Phi_i(q) + V_1(q) \quad , \; q \in Q$$

The updating algorithm is as follows:

1: put $a_0 = 0$;

2: for k = 1 to N, compute :

$a_k = a_{k-1} + V'_2(q_k) \Phi'_i(q_k) \cdot \prod_{j \in Ni} V'_2(q_k, f_j)$

where $V'_2(q,r) = \exp [-V_2(q,r)]$; $V'_1(q) = \exp [-V_1(q)]$ and $\Phi'(q_i) = \exp [\Phi(q_i)]$.

3: compute a pseudo-random number r with uniform distribution in the interval $[a_0, a_N]$.

4: find k such that $r \in (a_{k-1}, a_k]$, and put the new value of

$f_i = q_k$.

Note that the finite size of Q implies that the functions V'_1 and V'_2 can always be implemented as table lookup procedures. The interaction function Φ' , also takes, for the strong interaction that we are considering in section 6, a simple form :

$\Phi'_i(q) = 0$, if there is a pseudo-observation in site i, and q is incompatible with it.

1, otherwise.

References.

Geman S. and Geman D. (1984), "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images" IEEE Trans. on PAMI, 6, 721 - 741.

Geman D., Geman S. and Graffine C. (1988), "Locating Texture and Object Boundaries" Preprint.

Glass L. and Perez R. (1971) "Perception of Random Dot Interference Patterns" Nature 246, 360.

Hillis D. (1985), "The Connection Machine". MIT Press, Cambridge, MA.

Marroquin J.L. (1985), "Probabilistic Solution of Inverse Problems" Tech. report 860. MIT, Artificial Intelligence Lab.

Marroquin J.L., Mitter S. and Poggio T. (1987), "Probabilistic Solution of Ill-Posed Problems in Computational Vision". J. Am. Stat. Asoc. 82:397 pp 76-89.

Metropolis N., Rosenblueth A., Rosenblueth M., Teller A. and Teller E. (1953) "Equation of State Calculations by Fast Computing Machines" J. Phys. Chem. 21, 1087.

Poggio T. and Gamble E. (1987), "Visual Integration and Detection of Discontinuities. The Key Role of Intensity Edges" MIT AI-Memo 970

Reif (1965) "Fundamentals of Statistical and Thermal Physics" McGraw Hill.

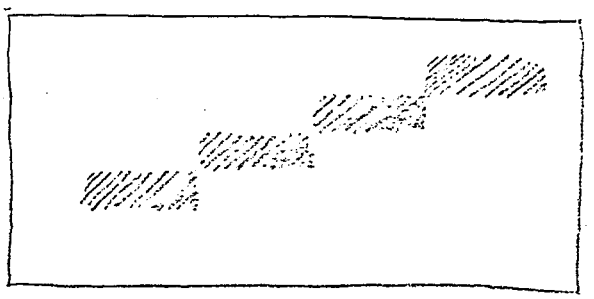


Fig 1

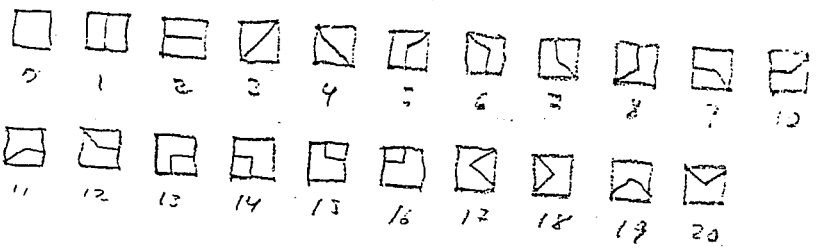


Fig 2

Fig 3

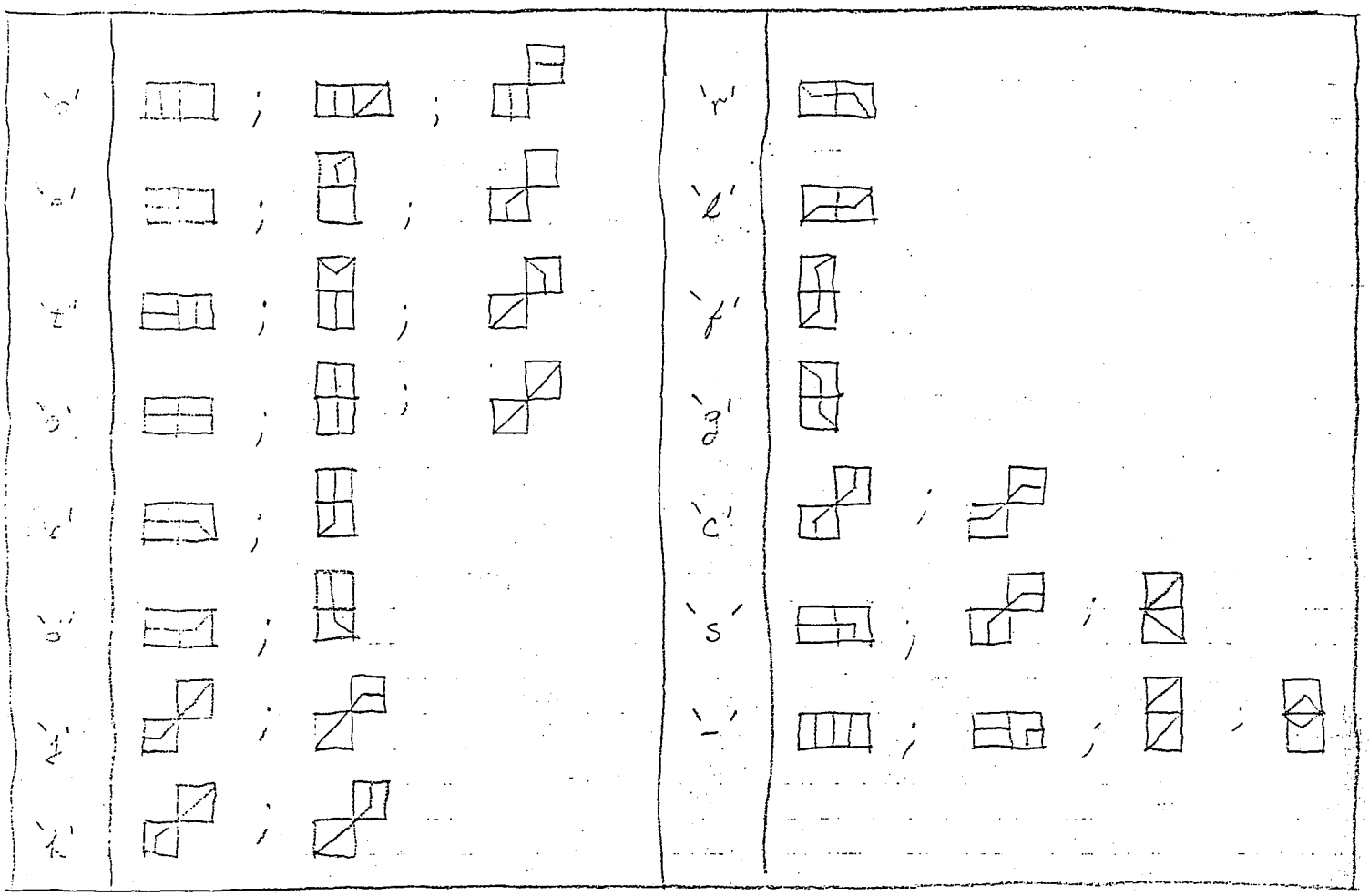


Fig 4

Fig 5.

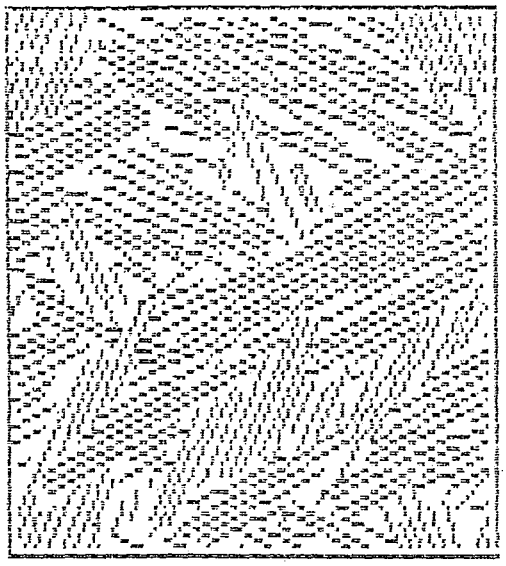
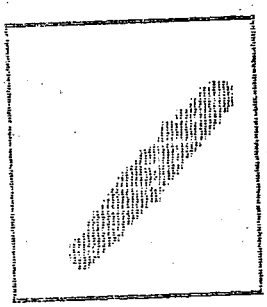
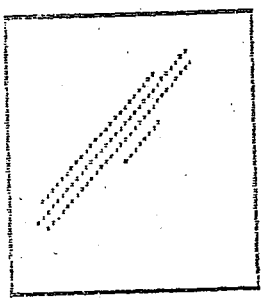


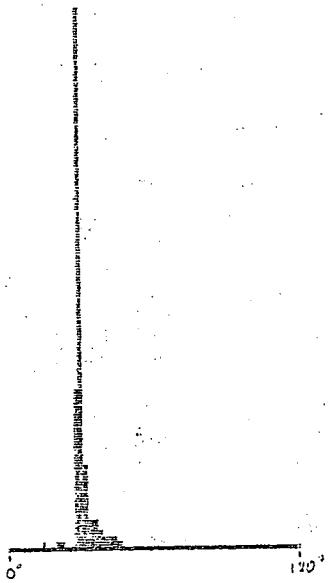
Fig 6.



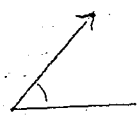
(a)



(b)

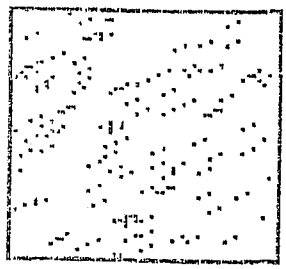


(c)

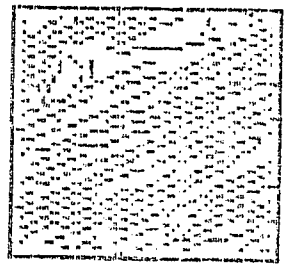


(d)

Fig 7.

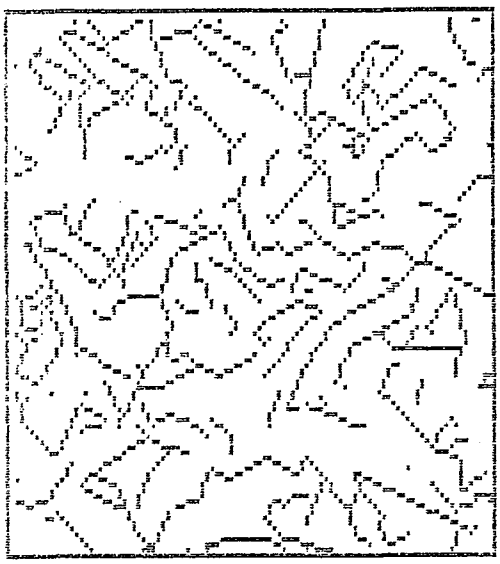


(a)

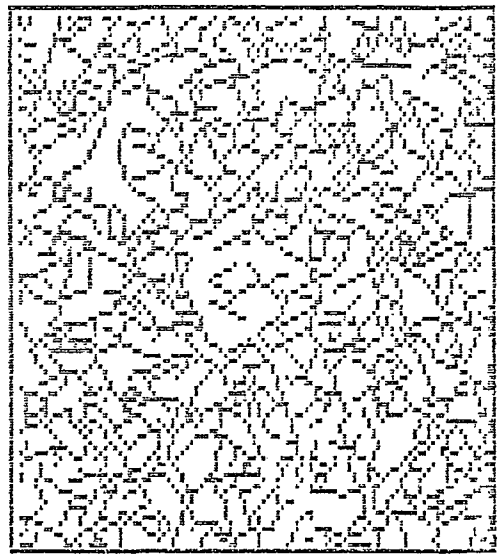


(c)

Fig 8

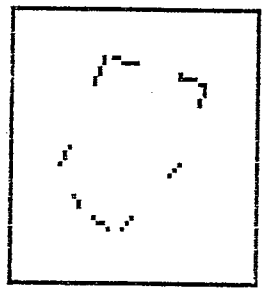


(a)

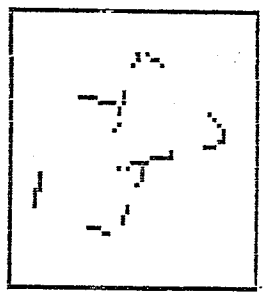


(b)

Fig 9.

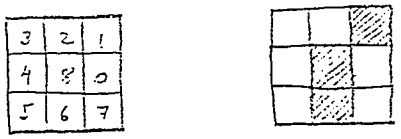


(a)



(b)

Fig 10.

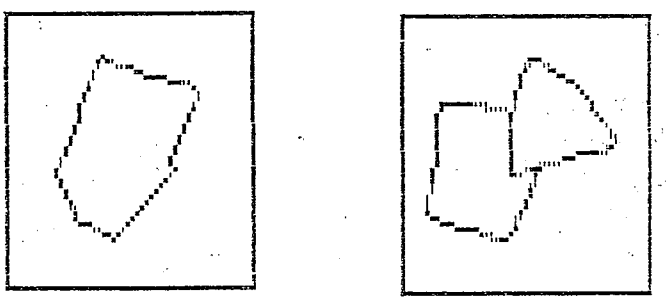


(a) (b)

Fig. 11

(a) (b)

Fig 12.



(a) (b)

Fig 13.

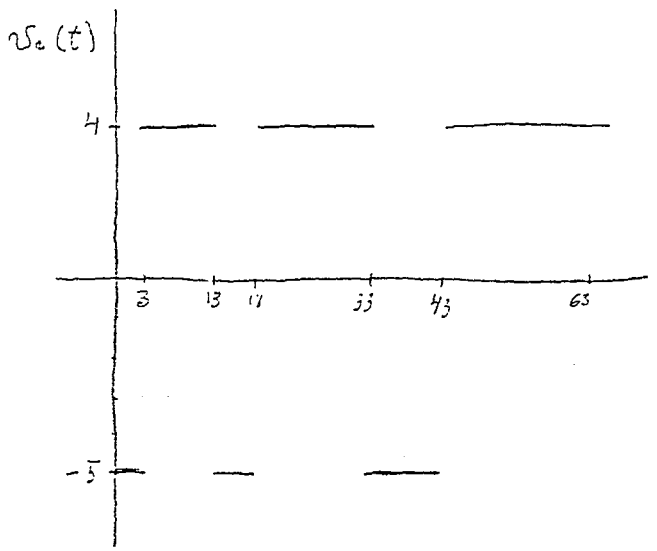


Fig 14