131-145

C-CALCULUS: AN OVERVIEW

E. R. Caianiello, A.Gisolfi, S.Vitulano

Scuola di Perfezionamento in Scienze Cibernetiche e Fisiche
Istituto di Scienze dell'Informazione
Facoltà di Scienze - Università di Salerno -

A CALCULUS OF PARTITION

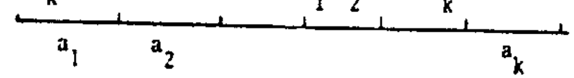
1) We propose to present some applications to Pattern Recognition of a "mathematical game" started some years ago by one us (Caianiello, 1973), which as named "C-calculus" for reasons which will be reminded in the sequel. We wish to state forthwith that it is simpler in principle than ordinary arithmetics; various fields can be envisaged in which it might prove of use: e.g. manifold topology, integration theory, fuzzy sets (where it might provide a natural tool for numerical computation), measure theory in physics, data-base structures, neural models, etc. This, we hope, will be apparent to the reader; we must restrict ourselves here only to the specific field of interest in the present context. We shall endeavour to keep language and arguments as plain as the subject really is; recourse to abstract formalism is often a disguise more convenient to the author than to the reader. We bagin therefore by reminding the game with which it all started. Take any integer positive numbers, and apply to them the rules of arithmetics, with the restrictions that only the direct operatios, sum and multiplication, be allowed, the inverse ones, substraction and division, forbidden; define furthermore the sum and the product of any two digits as follows

$$a + b = \max(a,b)$$
$$a \times b = \min(a,b) \qquad (1)$$

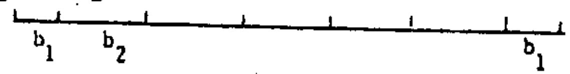
We may thus "multiply" any two such numbers, e.g. 736 and 491

```
  736 ×        491 ×
  491 =        736 =
  ----         ----
   111           461
  736           331
 434           471
 -----         -----
 47461         47461
```

We find that multiplication (and addition) thus defined are always commutative for such "numbers".
It would be an easy matter to demonstrate that, provided the "single digit operations" (1) are meaningful, one can operate in the same way on objects (subtraction and division being of course barred), such as vectors, matrices, etc., obtaining additivity and commutativity whenever they hold in arithmetics.
These "numbers" or "strings" of digits, with the operations (1), form clearly a commutative semi-ring. As in arithmetics each "digit" plays two different roles: one intrinsic to it ("cardinality"), the other ("position") relative to the string in which it belongs. The next remark is that standard set theory treats only intrinsic properties of sets. If in (1) we interpret + as "union" $\vee$ and x as "intersection" $\wedge$, we can immediately transport all that was said thus far to "strings of sets", or "composite sets", "C-sets" for short.
Operating on C-sets as before, with $\vee$ and $\wedge$ in place of + and x in (1) (a,b denote now the "simple" sets of which C-sets are strings, as the digits in the former example), one has "C-calculus": a commutative semi-ring which permits, from some given C-sets, to generate any number of other C-sets. Inverse operations are neither possible nor required in this context: only direct ones are permissible; one may perhaps see, though, advantages in being able to express in this way long lists of specifications in terms of some few basic ones.
An example of C-operation of special relevance for our present purpose is the following. Consider a segment S partitioned in segments a_1, a_2, a_3, ..., a_k; this partition $A = a_1\, a_2 \,.....a_k$



Consider now the same segment partitioned in a different way $B = b_1\, b_2 \,.....b_L$



Consider now A and B as C-sets: the elements of each partition, or string, are "simple" sets; C-multiplication of A and B gives

$$A \times B = B \times A = a_1\, a_2 \,.....\, a_k \times b_1\, b_2 \,.....b_L = C = c_1\, c_2 \,.....\, c_p$$

and it is immediate to verify that the simple sets of the product are obtained, in order, by joining on the segments the terminal points of both partitions A and B.
The C-product of two partitions gives thus the refinement of one by the other: C-calculus is the natural way of composing partitions, or covarings. In fact, the same property holds true in any number of di-

mensions. (Apostolico et al., 1978). This is the key property of C-calculus as regards its application to Pattern Recognition.

2) Our interest in an approach of this type to Pattern Recognition originates from the instinctive feeling of the physicist when confronted with problems for which a vast number of approaches is proposed, some indeed of remarkable ingenuity and power, but none of general (at least in some sense) applicability: is there some philosophy, or method, that may be applied to all problems of this sort, even if, of course, with less abundance of results than ad hoc techniques will undoubtly provide? Does one need a language for thinning, one for shrinking, one for counting, one for studying textures, one for retrieving objects against a background, and so on? or may we, perhaps, let "patterns" speak for themselves, changing the pattern itself into something algebraic or numeric, out of which, several, if not all, questions may be answered through essentially a single basic algorithm? This is, of course, a "träumerei"; but the search for "laws", rather than "rules", is a professional deformation for which a physicist need not apologize, although he better be - as we certainly are - duly apologetic about results achieved.
Having expressed (not certainly justified) our motivation, we shall substantiate it with a typical instance. Granted a priori that a major crime of Pattern Recognition is the preliminary reduction of a (say) 2-dimensional image into pixels, and that we must so proceed because we are much less bright than a fly or a frog, we find that a rather peculiar situation then arises. Parcelling a picture into pixels (with tones of grey, or colour) is, logically, a parallel process, out of which we can gather the more information, the finer the grid whose windows generate "homogenized pixels" (from each pixel only averages are taken). Suppose now that the same grid is rigidly shifted, over the picture, by a fraction of its window side; we may proceed as before, and obtain some other amount of parallel information. The question arises: can we use both informations, the one from the first and the one from the second grid partitioning, to get a better, more detailed information on the picture? Since we are taking only averages from each pixel, each time, the answer is no (unless, of course, we perform some mathematical acrobaties): one of the two readings has to be thrown away.
It would be nicer, one might feel, if there were a way of performing readings from the grid such as to permit to combine in a natural way the readings of both grids to obtain a more refined information on the picture (as might have been gathered by using a finer grid to begin with). If one can handle this situation, conceivably it may then be possible to use several times (serially) a single coarser grid, read out of it the (parallel) information obtained by shifting the whole grid by one step, and so on. The use we intend to do of C-calculus is aimed at answering just this question. The "reading" from a grid (of a suitable sort) becomes per se a C-set; two C-sets from different positions of the grid can be C-multiplied; this will give finer information, and so on.

3) Under "suitable" circumstances (to be defined explicitly in the sequel) this procedure can be carried through to the extreme limit of perfect reconstruction of the original picture (as digitized at the finest possible level: e.g. with a $2^{10} \times 2^{10}$ grid for the original, it may be reconstructed by covering it stepwise with, say, a $2^3 \times 2^3$ grid). During this process many things which one does with specific tecniques, such as contour extraction, contrast enhancement, feature extraction, etc., can be performed by interpolating in it steps with "answer" questions of this sort and become part of the algorithm.
But the application of C-calculus will often fail; the original image may not be thus reconstructed. There is an element to be considered, which was before ignored through the adjective "suitable": the size of the window. It is a feature of our approach that the critical size, below which total reconstruction of the picture is impossible, is determined by the structure of the picture itself, and is not a matter of guesswork or trial and error.
One can arrange readings, and ways of analyzing them, from grids having sizes appropriates to constitute in fact filters that see some wanted features and are blind to others. Typically, consider a saucer on a chessboard: things can be arranged so as to see only the saucer or only the chessboard (with a hole); or a specific component of a texture, ignoring all others, or suppress some noise, etc. Such filtering does not smear out or enhance; it gives at worst an intended contour to the saucer, as is natural when working with grids.
A study of this filtering process from the point of view of rigorous mathematics has not been undertaken yet, again on the physicist's view that such studies are always possible on no matter what subject, but that it is preferable first to test whether the subject is worth the effort.
At this stage a wealth of tools becomes available, about which it is better to keep a critical than an enthused view. Most times the practical problems at hand require only partial answers for their solution, like distinguishing between given printed characters. Considerations of this nature are somewhat systemized by using one such tool, which, not surprisingly, call C-matrix; it yields useful informations in several situations of interest, we shall exhibit if mostly through examples.

1) Our main tool in the application of C-calculus, to Pattern Recognition will be, as was said at the end of section 1-1, the C-multiplication of two partitions. A and B of an N-dimensional domain

$$A = a_1 \,\ldots\ldots\ldots a_k$$
$$B = b_1 \,\ldots\ldots\ldots b_L$$

which yields $C = A \times B = B \times A = c_1 \,\ldots\ldots\ldots c_p$

i.e. is the refinement of A by B, or viceversa, with elements $c_1\ c_2 \ldots c_p$

The same approach can be used regardless of the number of dimensions of the pattern to be studied. We shall consider, for the sake of simplicity and for specific relevance to P.R., only one- and two-dimensional patterns (e.g. graphs and pictures); we shall also consider only one additional dimension, which may e.g. denote levels of greyness, discretized or not (with colors, one may add as many dimensions as distinct ones are considered, etc.).
We start with one-dimensional patterns, as it is immediate to visualize in this case the procedure. We restrict our attention to "grids", i.e. to partitions of the abscissa x into segments, for simplicity, of equal lenght, the "windows".
Thus, consider the graph of Fig. 1, where the ordinate y denotes greyness (or intensity of sound, or local pitch...).
Discretize now the x coordinate with a grid G, of window w; change the graph into a sequence of rectanles, by substituting the portion of graph corresponding to a given window w_h with the rectangle which projects upon x into w_h and upon y into the segment having as upper and lower extrema the maximum M_h and the minimum m_h reached by the graph within w_h (no matter where, or how many times). We change thus the graph into a string of rectangles

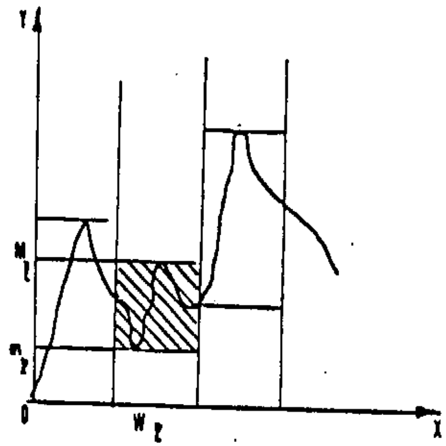


FIG. 1

A C-set: a partition of a graph in a string of quadruples.

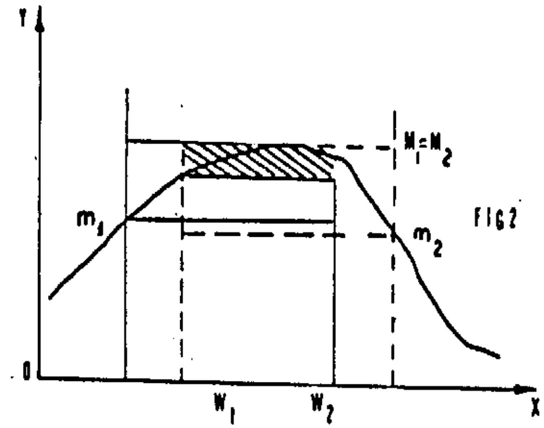


FIG 2

A product of the elements of two C-sets.

The ordered sequel of all these rectangles is the C-set determined from the graph by partitioning the x-axis with the given grid.
We may now proceed as before, after shifting the grid by a step $1 < w$. Denoting with indices 1 and 2 the two C-sets thus obtained, we have, with an obvious notation:

$$C_1 = R_{1,1}, \dots\dots, R_{1,k}$$

$$C_2 = R_{2,1}, \dots\dots, R_{2,k}$$

$$R_{i,h} = \left[w_{i,h} ; m_{i,h} , M_{i,h}\right]$$

We may now define the product of two simple sets $R_{1,h}$ and $R_{2,L}$ as the intersection of the rectangles just defined, i.e.

$$R_{1,h} \times R_{2,L} = \begin{cases} \emptyset \quad \text{iff} \quad W_{1,h} \wedge W_{2,L} = \emptyset \\ \left(W_{1,h} \wedge W_{2,L} ; \max\left[m_{1,h}, m_{2,L}\right], \min\left[M_{1,h}, M_{2,L}\right]\right] \end{cases}$$

In other words: instead of attaching to each window w one value, say the average of y in it, we take two, m_h and M_h; the difference $\Delta_h - m_h = \Delta_h$ is known as the dynamic of the graph in w_h.
This modification is sufficient to carry out our proposed program, because now it is evident that $C_1 \times C_2$ represent a finer partition of some strip within which the graph-line is contained, as is shown in Fig.2.
Consider now any rectangle of the C-set $C_1 \times C_2$; its base is $w_{1,h} \wedge w_{2,h}$, the height is $y_h = \min.(M_{1,h}, M_{2,h}) - \max(m_{1,h}, m_{2,h})$.
Is thus evident that the dynamic of the graph in it is:

$$\Delta (w_{1,h} \wedge w_{2,h}) \leq y_h, \text{ which reduces}$$

$$\Delta (w_{1,h} \wedge w_{2,h}) = y_h \quad \text{if the graph is monotonic in } w_{1,h} \vee w_{2,h}.$$

This remark is essential in order to study under which conditions iterated C-multiplication of C-sets obtained by shifting a given grid will reproduce the given graph to maximum permissible accurancy (that of the original graph, which was supposed digitized at some finer level).
The criterion of convergence to be satisfied for total reconstruction of the original graph, or parts of it, must clearly be the following: convergence is achieved wherever one obtains, at the h^{th} iteration, $m_{i,h} = M_{i,h}$ over the corresponding w_i's. One may substitute to this criterion the weaker one (especially if the ordinate is not discretized) that $M_{i,h} - m_{i,h} < \varepsilon$, prefixed, small as convenient.
The formalization of this procedure is straightforward; the interested reader may reproduce it by himself, or refer to some previous paper.(Apostolico et al.,1977).
For equally spaced grids (it would be only a matter of convenience to relax or change this condition at any wanted step of the procedure), there is a very simple formula which determines whether overall convergence is guaranteed: this will be the case if, and only if:

$$w \leq \frac{D}{2} + 1 \qquad (2)$$

where D denotes the smallest distance between a minimum and a maximum of the graph. The proof is given in(Apostolico et al.,1977). The same formula applies also in two (or more) dimensions if we now read w to mean the side of the square window, D the euclidean distance in the plane between such extrema.

2) We can now change our viewpoint. Instead of shifting the grid, obtaining and multiplying the ensuing C-sets, etc., we ask what this procedure will finally yield at a given, fixed point on x-axis. The answer is that, regardless of the order in which these operations are performed, the final values which are associated with any given point are the highest minimum and the lowest maximum that are seen when the window w moves on an interval of lenght $2w - 1$, centered at that point. The discussion of this point, which is related of course to (2) is in ref.(Apostolico et al., 1977). Of special interest is the case in which (2) is violated. Our procedure will not reconstruct then the original pattern, but produce a new pattern, which suppresses all those details of the original one which could be retrieved only by respecting (2). In other words, the procedure will act now as a filter (C-filter (Caianiello et al., 1978). A trivial, intuitive example will convince us of this fact. Suppose that we wish to study in this way the before mentioned chessboard. Operating with a (square) grid whose window is smaller than the case of the chessboard will readily reconstruct the chessboard. If, however, the window is larger than the case, no matter how we move the grid we shall always find $m = 0$, $M = 1$ (say) in any window: convergence is impossible. It is then a trivial matter to arrange things so that in the first case our procedure reconstructs the chessboard, in the second it yields a total blanc: the chessboard is filtered away\ If now we have a saucer on the chessboard, we shall be able to retrieve only the saucer, obliterating the chessboard background; likewise, one can proceed with textu-

res (Caianiello et al., 1979): it is possible to see an object ignoring a textural background, or viceversa, to extract only some relevant textural elements. Many variations are possible on this theme (Galloway, 1978). One can keep, thus, a window which satisfies (2) but accept only dynamics within given thrsholds; or play both with window size and threshold.
The next part illustrates a computational tool that can be extracted any given pattern, the "C-matrix": it gives automatically optimals criteria for window and threshold size, and can be useful in the study of several classical problem of P.R.

C-MATRIX

1) Consider a pattern F in any number N of dimensions (for the sake of illustration we restrict here N = one or two), and consider only one of some K "attributes" of interest, e.g. the level of greyness (which we may suppose now digitized). An N-dimensional grid of (square) windows generates, as we have seen, a C-set in N + 1 (K = 1) space; C-multiplication of all C-sets obtained by displacing (according to some rule) the grid ever the pattern yields, after a suitable number of iterations, a pattern F, which will coincide with F if condition (2) is respected, differ from F otherwise. This process we have called C-filter; as much, it might deserve per se mathematical investigation. Our interest here is rather with concrete ways of exploiting the typical new feature of C-calculus, that of permitting either "precise" measurements of a whole through serial composition (C-multiplying of "coarser" partial parallel measurements) (each, a C-set from the grid), or "filtering operations". "Precise" and "coarse" are to be understood as in physics. We have found profitable, for this purpose, to introduce a mathematical object which (for any N, K = 1) is always 2-dimensional: the C-matrix.

2) We define the C-matrix. Its element c_{hk} has as row label h (=2,3, 4,....) the linear size of the window w_h of the scanning grid G_h (C-calculus will be more useful the larger can be kept the minimum h needed for a given analysis); the column label k (= 0, 1, 2, ...) denotes the possible values of the dynamic K = M - m as read through w_h. The grid G_h scans the pattern according to some criterion (in two dimensions, the best has proved to run down the main diagonal) by one step at a time ("1" means the original digitizing window, if any, or simply the "resolving power" of the system); G_h has thus h(= $\sqrt{2}$ h , square grid h, linear) scanning positions; any value k of the dynamic is registered $P_{ik}^{(h)}$ (= 0, 1, 2, ...) times in the windows, for each position i=1, .., h of the grid G_h. In conclusion

$$C_{hK} = \sum_{i=1}^{h} P_{iK}^{(h)} \qquad (3)$$

The building of the C-matrix seems, from the definition just given, a much more imposing task than it actually is. In fact, the operations that lead to c_{hk} are performed in parallel for k = 0, 1, 2, ... so that the C-matrix is built one row at a time, or in one piece with suitable hardwere; also, the procedure has to be started from the bottom row, i.e. largest w_h, and it becomes an obvious matter to implement the algorithm with devices that suppress the need to explore, moving towards smaller h, territories where, e.g.k,=0 (uniform greyness) for a larger h. Nor need one proceed through the full sequence h_{max}, $h_{max} - 1$, $h_{max} - 2$, ..., 2: large jumps may be made.
An element c_{ij} of the C-matrix tells us thus: if it vanishes, that the value j of the dynamic of the pattern can never be seen through windows of size w_1 ; if not, how manytimes that is seen by no matter which window w_1. It is easy then to read from the C-matrix many features of the pattern; e.g. if only $c_{i,o}$ = 0, all i, the pattern is a plauteaux, of constant gryness throughout: the max i for which $c_{i,o} \neq 0$ is the width of the largest plateaux; a highest-slope of a (1-dimensional) signal shows as the sequel $c_{ij} \neq 0$, j max, for each i.

3) CONTOUR EXTRACTION

This is a; relevant step in every problem of P.R., with a wide literature and a large number of techniques available: gradients and threshold, thinning, joining or separating of broken or interpenetrating pieces, template matching, etc.(Roberts, 1965; Hall et al.,1976; German et al., 1977).
We can extract contours with the following role. Consider the first among the columns of the C-matrix, let it be the column h,in which the elements c_{ij} go through a maximum, at some value i. Take the corresponding window w_i as element of the grid of a C-filter, given by C-multiplication of C-sets whose simple sets vanish if smaller than the (hyper) rectangles w_i x h (i.e. h is a threshold on dynamic)

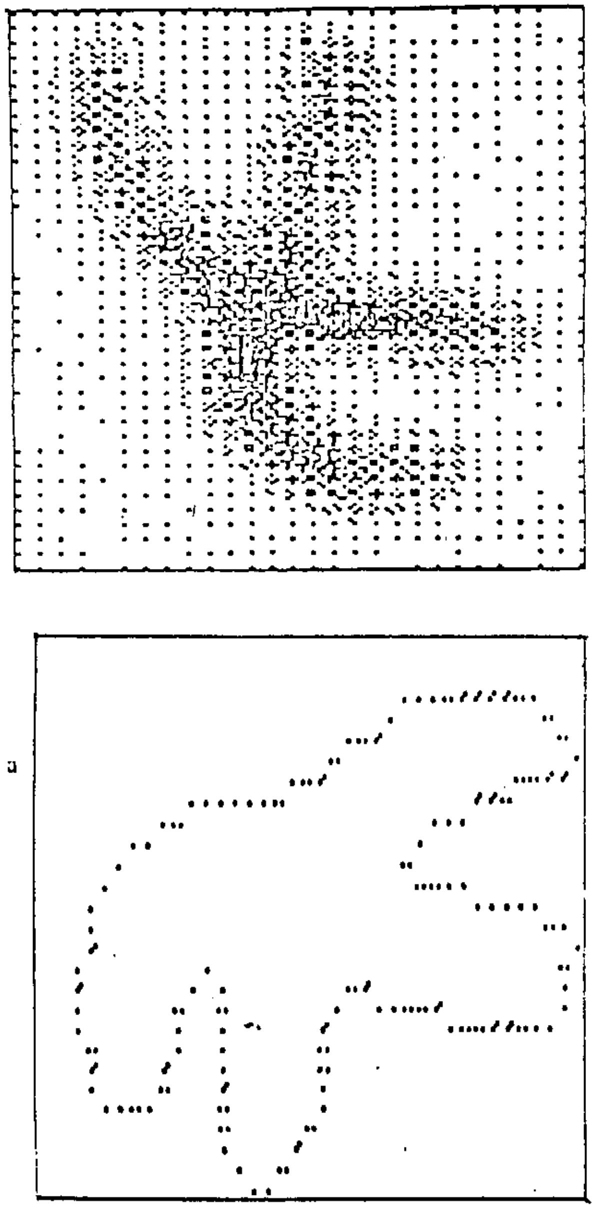
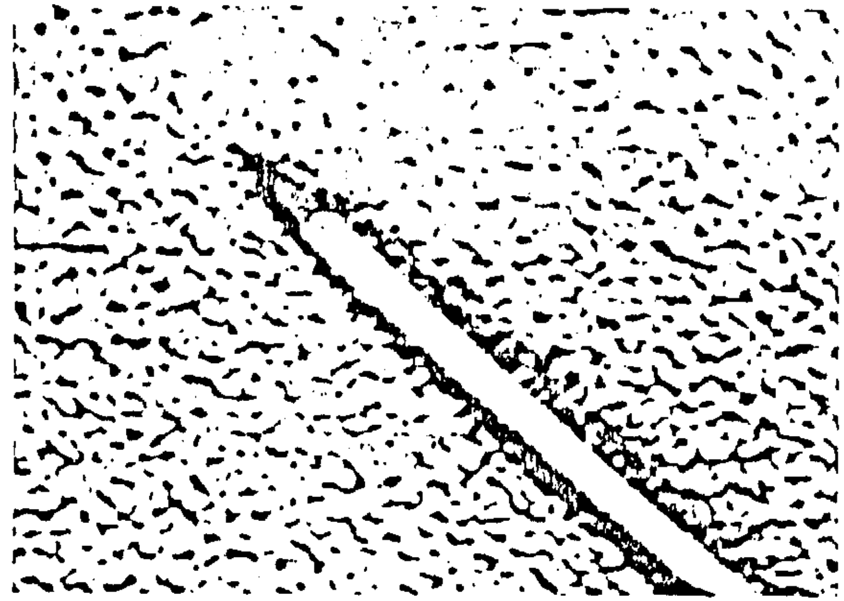
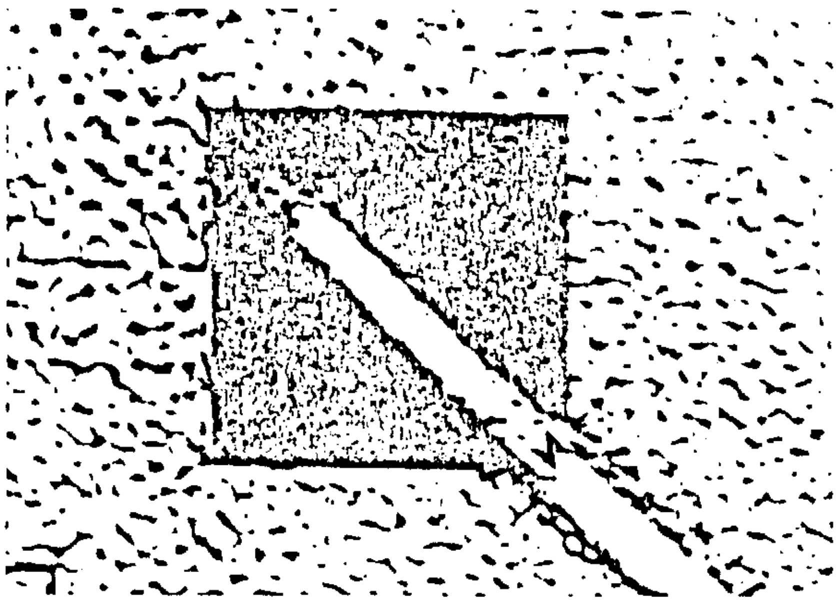


Fig.4
contour extraction of a human chromosome.



A pencil on a natural texture



He application of our filtering method.

SEGMENTATION

This subject, which includes discrimination aginst a background, is the object of much literature, and can be carried to deep contextual and grammatical analysis.
A simple approach to it through C-calculus is obtained by fixing an iterative procedure which again uses data read from the C-matrix. Details can be found in ref.(Feng et al., 1976; Galloway, 1978; Weszka et al., 1979).
Comparison with other algorithms (Feng et al., 1976; Galloway, 1978; Weszka et al., 1979) was made; it appears that thresholds come out of the C-matrix and do not have to be guessed, and that computation time is here less than with RAG, LAG, PT.

FEATURE EXTRACTION

We refer for details to ref 's (Deutsch, 1955). Neurophysiological modeling, Gestalt and invariance requirement: (Deutsch 12) have suggested an algorithm with concentric window so that the inner one acts as "excitatory", the annulus around it as "inhibitory". As before, all relevant parameters are read from the C-matrix. Our experimental results show that C-calculus, as it is, can be adjusted to higher degrees of definition if desidered, because it mimics basic neuronal mechanism.

TEXTURES.

The analysis of textures, both per se and as backgrounds, is almost a science within a science; see, e.g. Gibson 195o, Koehler 1975, Haralick 1973). Our views on this subject do not belong to this short summary of our work; more than in other tonics of P.R. we must remind that "percepta" are "phenomena", in the sense of Kant, quantum mechanics or the Vedas, to which the noumenon" and the "observer" equally concur. We wish only to report here, in conclusion, that C-calculus has proved especially "natural" in this context, leading to ready and elementary classification, analysis and discrimination.
We refer the reader to previous works Gisolfi et al., 1980); some examples in this are reported. It is the "chessboard and saucer" game we were mentioning earlier, which is easily implemented through C-calculus into algorithmic simulations which may be rather close in principle to the actual operation of neural tissues.

1) Apostolico A.,Caianiello E.R.,Fischetti E.,Vitulano S. C-calculus: an elementary approach to some problems in Pattern Recognition. Pattern Recognition n.5 1978.
2) Apostolico A.,Vitulano S. A image transform, in Informatik-Fach berichte n.8 Springer Verlag 1977.
3) Brodatz P. Textures, Dever, 1966
4) Caianiello E.R. A calculus for hierarchical system, Proc. First Int. Joint Conf. on Pattern Recognition, Washington, 1973.
5) Caianiello E.R.,Gisolfi A.,Vitulano S. A technique for texture analysis using C-calculus. Signal Processing n. 1,1979.
6) Deutsch J.A. A theory of shape recognition, Brit. J. Psych. 46,
7) Caianiello E.R.Cisolfi A.,Vitulano S. A method of filtering biomedical specimens, Proc. The Int. Conf. of Cybernetics on Society. Tokyo, 1978.
8) Dinstein I. Haralick R.M., Textural features for image classification. Proc. IEEE trans.of System, man and Cybernetics, n.6, 73
9) Feng U.Y. Pavlidis T., Feature generation for syntactic Pattern Recognition, IEEE trans. Comput. n. 12.
10) Galloway M., Texture analysis using gray level run lenghts. Comput. Graphics, Image Process n.2,1978.
11) Cerman F.V., Clowes M.B., Finding picture edges through collinearity of feature point. IEEE trans. Comput. n. 24.
12) Gibson J.J., The perception of the visual world. Houghton 1950.
13) Gisolfi A. Mlodkowsli M. Vitulano S. A method for classifying and filtering textures. Progress in Cybernetics and System Research, Hemisphere Pub. Washington, 1980.
14) Hall E. Frei W. Invariant features for quantitative scene analysis Univ. Southern California. 1976.
15) Kohler W. La psicologia della forma. Feltrinelli 1975.
16) Roberts L.G. Machine Perception of three dimensional solids. M.I.T. Press, 1965.
17) Weszka J.S. Nagel R.N. A threshold selection technique. Trans. Comput. 23; n. 12.