COMPUTER DESCRIPTION i F TEXTURED SURFACES

Ruzena lajcsy

University of Pennsylvania
Philadelphia, Pennsylvania

Abstract

This work deals with computer analysis of textured surfaces. Descriptions of textures are formalized from natural language descriptions. Local texture descriptions are obtained from the directional and non-directional components of the Fourier transform power spectrum. Analytic expressions are derived for orientation, contrast, size, spacing, and in periodic cases, the locations of texture elements. The local descriptions are defined over windows of varying sizes.

Key Words

Computer vision, computer description of texture, texture elements, spatial organization of textured regions.

Introduction

My purpose is to present a new technique for computer description of textured surfaces. Although I use outdoor scenes, involving grass, water, forest, and the like as a starting point of Intuition, textured surfaces generally appear in almost every sort of scene and, therefore, I will also show some examples of isolated and artificial textures.

This study is motivated by a wide range of practical applications. The agricultural survey and analysis of earth resources by means of satellite pictures is one such example. The social benefits of computer-controlled cars using computer vision, as described by John McCarthy, is another. Industrial robots will soon acquire vision, too. Texture synthesis, to which I feel our techniques are also applicable, is useful in computer-aided design and computer-aided art. Interpretation of scanning electron-microscope pictures, e.g., for metallurgy, may also be of Interest.

A primary problem in texture is how we perceive a textured surface as a uniform structure in a non-trivial way. Intuitively speaking, there are many levels on which one can perceive texture. In one situation we may look at the pattern created on a wall and call that a "texture." In another situation we may have a closer look at the same wall from the same distance, but in this case see texture of the individual bricks and ignore the texture given by the overall architectural structure of the bricks. We want to characterize textures in terms of a compact symbolic representation.

We follow a structural description of textured regions in terms of texture elements and their spatial relationships. In turn, textured regions can also be texture elements in a larger structure, etc.

Texture elements cannot be determined in isolation. They are recognized through their similarity relationships, although the measure of a similarity may vary considerably. For example, in a texture of pebbles the size similarity may be important even though the sizes vary significantly. Still, there is uniformity within a factor of 10 or so. Similarities of other properties such as contrast, shape and spatial distributions may also be only approximate.

Texture Descriptions

The problem of texture recognition has been noted already in early computer vision research. ' ' ' '

The best review paper about the current state of texture extraction technology is that of Hawkins. According to him, there are four types of approaches that have been taken toward texture classification:
(1) spatial frequency content; (2) gray level content; (3) local shape content; and (4) higher order measures. All of these approaches are pattern classification techniques. These techniques are not satisfactory for a description of real textures for the following reasons:
(1) pattern classification techniques have concentrated on linear decision procedures and domain independent formulations. In this method the principle of continuity based on local similarity relationships of some features is missing; however, in real scenes this fact is very important. Take, as an example, a grassy field. Usually, grass appears as a lineary horizontal texture, e.g., horizontal parallel lines. However, when the wind blows, some of the grass stalks change direction away from the horizontal lines. Although this change of directionality is continuous and thus detectable, pattern recognition methods would miss it.
(2) in pattern recognition approaches, context appears as a set of numerical coefficients in a linear function and in the choice of features. Better models exist, in terms of context dependent decision trees, which provide a flexible description of the object in the scene. The description is not a simple name of a class to which the object belongs, but rather a description of the object's parts, attributes, and relationships■

In fact, structural description of a texture is the main theme in our work. In other words, our work involves the description of texture elements (their shape, size, orientation, etc.) and relational features (spatial organization, texture gradient, and the like).

In practical implementations we can describe only simple relationships in spatial organization among the elements, e.g., linear, bilinear, periodic, regular but aperiodic, symmetric. Likewise, shape descriptors of the elements must be relatively simple (straight lines, dots, blob-like). One may question the effectiveness of simple relationships and their descriptors; it is reasonable to think that a more complex description of texture elements and their relationships is necessary for adequate description of textures. However, the psychological experiments indicate that human differentiation of textures depends heavily on a few simple descriptors, such as contrast and directionality, and ignores even curvature In making texture groupings. Although we cannot estimate the computational complexity of descriptors, we have an intuitive feeling that in terms of time or In terms of complexity of wiring for parallel systems, simple descriptors, such aa directionality, are clearly preferred.

Procedures for Texture Descriptors

Basically, there are two domains from which one can derive texture descriptors: the spatial and the Fourier domain. Yet, some features are more visible in one domain than in the other. For example, the local properties, Buch as the shape of a texture element, are grumbled in the Fourier domain while they are preserved in the Bpatial domain; a similar situation exigts in the Bpatial shift among elements. On the other hand, the global organization of texture elements is expressed more succinctly in the Fourier domain than in the spatial domain. We shall describe below the procedures of feature extraction in each respective domain.

Texture descriptors derived in the spatial domain

Since descriptors refer to properties of objects (texture elements) represented in the image space, it is natural to look for operators acting directly in the spatial domain. Several low level operators have to be combined into a procedure to obtain the desired texture description. A skeleton of such a process usually consists of: procedures isolating the image elements, geometric description of image elements, and clustering of elements based on proximity and their spatial organization. A set of simple descriptors has been suggested and implemented by Rosenfeld and Thurston. They use, in parallel, several local averaging operators applied in different directions and on various sizes of windows. Though this method finds some texture boundaries, the operators are too trivial for handling a wide class of real textures. Besides, they do not provide any description of a texture, rather they only detect the texture differences.

Tn the process of isolating the image elements the most important features are the following topological properties: connectivity, continuity, and proximity. These properties, applied to brightness or color, are used in all region finders. Here, the algorithm is based on grouping all adjacent points with similar brightness and/or color. Discontinuity is the basic property to be used in edge and line operators.-[3,6] Current edge and line operators are designed for detecting discontinuities between two large homogeneous regions and they do not operate satisfactorily on small regions. The textured elements that one finds in outdoor scenes are too small in size and too large in number to be processed usefully by any of the above operators.

After completing the isolating of image elements - figures, we shall describe them. We select those descriptors which enable clustering, i.e. based on proximity, those which will find the nearby elements. In passing, I want to emphasize that color and brightness are among the most important descriptors in natural scenes. Image elements cannot be taken separately from their background. In fact, the common background of the elements is a strong clue for their clustering. The relationship between the background and color is expressed in terms of contrast, and, therefore, It can be used as another descriptor.

Since the shape of a two or three-dimensional object in a general situation could be extremely complicated, we cannot hope and, in fact, we do not want to describe it in detail. Instead, complex shapes are decomposed into simpler ones which are hopefully easier to describe; the size of the texture elements must also be simplified to correspond with the shapeB. A typical example is a tree which may be decomposed into its trunk and crown, the trunk being geometrically linear, the crown being blob-like. In shape analysis of outdoor scenes, we find directionality among the most useful features.

Finally, we describe the spatial organization of texture elements. This amounts to the description of a new structure formed by the texture elements. Here, the main problem is to recognize the whole as composed of texture elements. For example, the dots could form a straight line or a random dot pattern; straight lines with the same direction form parallel lines, etc.

Limitatio_of Spatial Domain Procedures Despite the importance of descriptors derived in the spatial domain, we shall not use them in this work for the following reason. Currently available edge finders and region finders are tailored for large homogeneous regions. In natural scenes, textured areas are composed of small texture elements. Even to the extent that the boundaries of small regions are determined, the data structures require unreasonably large memory, since the boundary descriptions are no longer economical. The next steps - description of elements and clustering elements of similar direction, size, color, or brightness - seem prohibitively time consuming and difficult for grass, pebbles, sand, etc.

Texture descriptors derived in the Fourier domain

Why do we Suggest to use the Fourier domain rather than the Bpatial domain? An effective texture operator must have certain virtues. It should describe the spatial distribution of texture elements, and it should characterize the shape and the size of the texture elements.

From the elementary properties of the Fourier operator, it follows that any real periodic function has a symmetric Fourier image with respect to the origin. An equally well-known but somewhat more interesting fact is that the power spectrum is invariant with respect to translation In the spatial domain, but not with respect to rotation. A trivial consiquence of this property is that the directionality of a pattern In the picture is preserved in the power spectrum, but the phase of the transform is not.

If a function is periodic, partially periodic, of almost periodic, then Its Fourier transform provides a more concise representation of the image, and the relational feature derived from the Fourier image forms a good description of periodic or almost periodic patperns. For example, take an image of a texture composed of several parallel lines. Its power spectrum will have only one line. In addition, the power spectrum will contain the information about the width of the line and the number of lines occurring in the image.

The fact that directionality is preserved in the power spectrum allows us to infer some gross Bhape properties. He are able to distinguiah directional and non-directio-"*1 rormonents of texture. For this reason, it is useful to transform the power spectrum from a cartesian coordinate system <r,<p>. Then in each direction , one can regard P(r,m) as a one-dimensional function $P_{\varphi}(r)$. Similarly, for each frequency r, $P_r(\varphi)$ is a one-dimensional function. In addition, if we Integrate along direction {p, we obtain

$$P(r) = \sum_{\varphi=0}^{\pi} P_{\varphi}(r) ,$$

and similarly, integration along radius produces

$$P(\varphi) = \sum_{r=1}^{W/2} P_r(\varphi) .$$

where W is the window size.

Thus, the description of the texture depends in this method on the form of the pair of functions <P(r), P(o,)>.

The taxonomy of textures that we can describe by using the properties of functions P(r), P(cp) is displayed in Flowchart 1.

Examples of Texture Descriptions Usinp Fourier Techniques

An example of a monodirectional texture is the texture of wood, shown in Fig. 1. In this figure, the upper left picture shows the original texture divided into four windows to show more than one sample of the same texture. The picture in the upper right corner is the resynthesized texture, produced according to the description. As one can see from these pictures, the description is monodirectional texture and, therefore, in the resynthesized version, the non-directional components are filtered out. The picture in the lower left corner shows the power spectrum of the original texture.

An example of bidirectional texture is a piece of canvas displayed In Fig. 2. In thiB figure, just like in the previous one, the upper left picture Bhows the original texture divided into four windows. The picture in the upper right corner is the power spectrum of the original texture. The pictures in the lower part of the figure are separated and resynthesized to two monodirectional textures, produced according to their description. An example of a noisy (pepper and salt) and blob-like texture are shown in Figures 3 (the sand) and 4 (the blob), respectively.

Now we shall study a further possible interpretation of function P(r). Consider a monodirectional pattern that appears as a one-dimensional (in the particular direction) square wave function F(x). Denote the replicative symbol by $\omega(x)$ and the wave form by f(x). We can represent a periodic texture F(x) as a convolution of basic waveform f(x) and a periodic function $\omega(x)$.
Thus, $F(x) = f(x) * \omega(x)$.
The Fourier transform of F(x) is

$$\mathcal{F}\,[F(x)] = \mathcal{F}\,[f(x,v)] \cdot \mathcal{F}\,[\omega(x,l)] = \text{sinc}\,\frac{x}{v} \cdot \omega(\frac{x}{\ell}).$$

Applying the window function of the width w, the operation appears as a convolution in the Fourier domain.

$$\mathcal{F}\,[W(x,w) \cdot F(x)] = \text{sinc}\,\frac{x}{v} * [\text{sinc}\,\frac{x}{v}\,\omega(\frac{x}{\ell})].$$

It is clear that we can measure $\frac{w}{v}$, $\frac{w}{\ell}$ in the power *spectrum from the function P(r), for every direction*ality and window size w. Consequently, we can estimate (how well, depends on the brightness function) the wavelength ℓ, as before and, in addition, the size of the smallest element, v. The quantities v and ℓ will be parameters associated with each description.

Examples of functions $P(\varphi)$ and P(r) of texture samples will be presented next. The size of samples is 32 x 32 points. The points on the y-axis have the corresponding values of the functions $P(\varphi)$ and P(r), respectively. The points (on the x-axis) in the graph for function $P(\varphi)$ represent the value $(x-1)\frac{\pi}{16}$ (for x = 1, 2, ..., 16). The points (on the x-axis) in the graph for function P(r) have just the actual values of frequency f = 1,..., 16.

Each pair of functions $\langle P(\varphi), P(r)\rangle$ will be described by some parameters, listed in a table. Below is the list of the parameters and their description.

NAME: The natural language names of the texture samples.

DESCRIPTOR: A hypothetical description of the sample according some criteria (thresholds) applied on functions $\langle P(\varphi), P(r)\rangle$.

MAX $P(\varphi)$: The maximal value of $P(\varphi)$.

φ_{max}: Is such φ that $P(\varphi_{max}) = \max P(\varphi)$.

WIDTH: The distance between φ_1, φ_2, *where* $\varphi_1 < \varphi_{max} < \varphi_2$ and $P(\varphi_1) = \text{MIN}\ P(\varphi)$, the left side with respect to $P(\varphi_{max})$.
$P(\varphi_2) = \text{MIN}\ P(\varphi)$ (the right side with respect to $P(\varphi_{max})$.

DIR: If the descriptor is a directional, first perform a fan filtering in such a way that the fan filter is centered in φ_{max} and then find MAX $P(n,m) = P(n_{max}, m_{max})$ and thus compute DIR = arctg $\frac{m_{max}}{n_{max}}$.
If the descriptor is non-directional then just find
MAX $P(n,m) = P(n_{max}, m_{max})$
and compute DIR as above.

RO: Is the wavelength computed from the maximal point energy?
RO = window size/ $n^2_{max} + m^2_{max}$.

M : is the mean value of function $P(\varphi)$.

v : is the variance of $P(\varphi)$.

MAX P(r): is the maximal value of P(r).

r_{max}: is such r that $P(r_{max}) = \text{MAX}\ P(r)$.

WIDTH r: is the distance between the center of P(r) and the threshold value of the envelope of P(r).

M_r: is the mean value of P(r).

v_r: is the variance of P(r).

v: is the element size equal to window size/width r of the envelope.

ℓ: is the spacing between elements, equal to window size/frequency of the first peak.

In the case of bidirectional texture, a pair of values is listed for the following parameters:
MAX $P(\varphi)$, φ_{max}, width φ, DIR and RO.

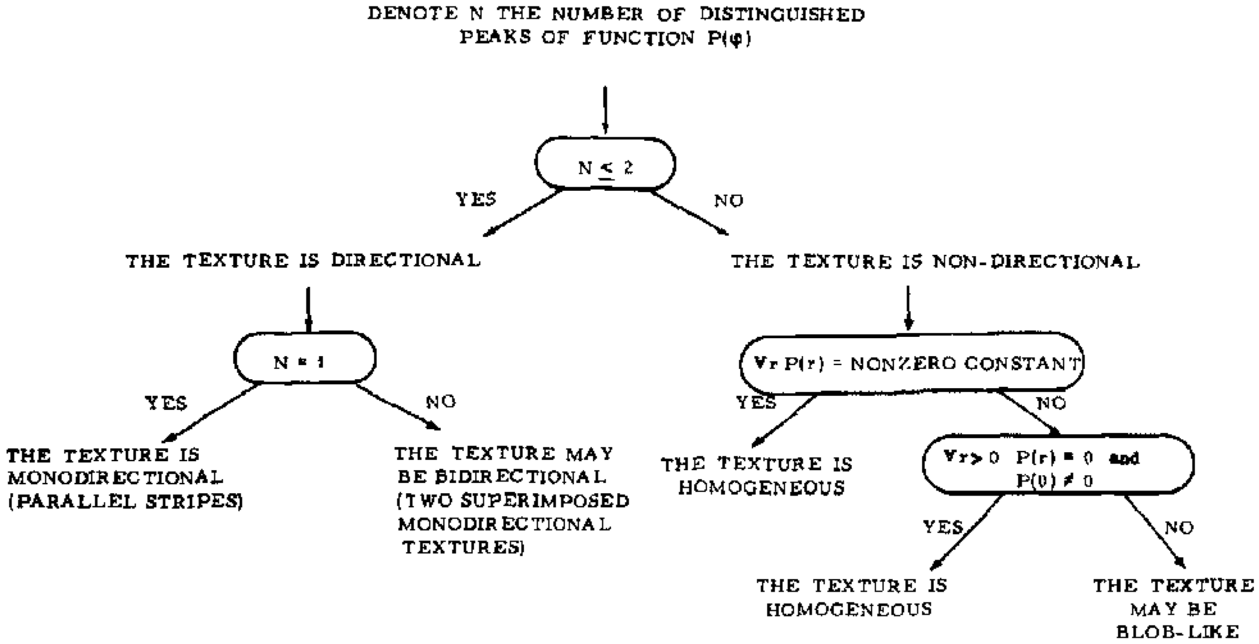
The texture names are on the top of each picture displaying the corresponding function P((p) and P(r). The actual samples of texture - wood, canvas, circle, and Band - are in Figures 1, 2, 3, and 4.

In Fig. 5 we display a sample of grass. The upper left window in Fig. 5 is the original sample; the upper right window is its corresponding power spectrum; the lower left window is the power spectrum after a high pass filter; and the lower right window is the resynthesized original picture after the high pass filter.

This example is presented to demonstrate the necessity for separating the slow changes from the real texture pattern. The rationale for this is that most of the objects (texture elements) tend to have the same reflectively and the lighting varies smoothly. Thus shading in the Fourier domain generates a low frequency component.

Functions P(<p) and P(r) of textures grass, wood, and canvas are displayed in Figs. 6a, 7, and Sa, respectively. To consider the main directionality and, thus, to be able to determine *l* and v, we display the filtered alternative in Fig. 6b for grass, Fig. 8b and 8c for canvas (for one directionality). The table of their corresponding parameters is Table 1.

Comments: First of all, notice that grass is described as bidirectional, contrary to what would be expected. The reason is that even after high pass filtering, there is still significant slow change left, (wavelength - 16) which forms the second peak. One needs to know more about the scene (ita illumination, continuity, context) in order to remove this kind of slow change. It Is impossible without further knowledge e-



FLOWCHART 1

TABLE 1

NAME DESCRIPTOR	GRASS BIDIRECTIONAL	WOOD MONODIRECTIONAL	CANVAS BIDIRECTIONAL
MAX $P(\varphi)$	<8.35, 7.5>	64	<108, 80>
φ_{max}	<5, 13>	14	<1, 9>
WIDTH φ	<6, 4>	5	<4, 2>
DIR	<0.463, 2.03>	2.55	<1.57, 0>
RO	<14.31, 16>	8.87	<16, 8>
M_φ	4.76	32.84	49.3
V_φ	0.534	3.76	5.46
MAX $P(r)$	5.62	44.8	120.34
ν	4	3	4
WIDTH r	16	16	9
M_r	4.52	31.46	47.14
V_r	0.324	2.54	7.64
ℓ	<8, 16>	10	<16, 8>
ℓ_{max}	8		8
V for MAX DIR	1	1	1.8

bout the area to handle this situation appropriately, because the same component (wavelength - 16), which in the case of grass is undesirable, is in the case of the canvas texture an essential part of its description.

Function P(r) in the cases of grass and wood shows similarities, which suggests that both of these textures have some noisy irregular backgrounds. On the other hand the canvas texture displays significant peaks in low frequency and decreasing power in higher frequencies.

For more detailed analysis of P(r), one has to separate the different directionalities. This is what we have followed up In Figures 6b, 8b and 8c.

The last two examples of texture of blobs and Band demonstrate the differences between nondirectional texturea. In Fig. 9 and 10 are functions P(tr) and P(r) of samples of texture recorded in Fig. 4 and Fig. 3, respectively. Table 2 contains their corresponding parameters.

The P({p) is a flat function In both textures as is expected. P(r) in the case of blobs has one significant peak; whereas, in the case of sand, P(r) is approximately flat.

TABLE 2

NAME DESCRIPTOR	BLOBS BLOB-LIKE	SAND NOISY
MAX $P(\varphi)$	82.26	73.74
φ_{max}	13	13
WIDTH φ	3	3
DIR	2.35	2.35
RO	11.31	5
M_φ	60.2	52.8
V_φ	2.72	2.48
MAX P(r)	120.46	75.8
V_{max}	3	6
WIDTH r	6	12
Mr	61.70	54.4
Vr	6.52	3.18
φ	10	5
V	2.6	1.3

We must make some comments about the differences between continuous and finite discrete Fourier transforms. The continuous Fourier transform exists for every function with finite energy, while the finite discrete Fourier transform exists for any function. Our interpretations will be based on the continuous transform and the actual computations on the discrete transform (fast Fourier transform). The discrete transform is really a Fourier series. A continuous Fourier transform is rotationally invariant (except for windowing effects), while a descrete transform has distinguished axes along the coordinate axis and the diagonals. Thus, a directional image has a continuous Fourier transform in a very narrow band transform only for directions along the preferred axis. There is a corresponding difficulty in defining fan filters which we have not succeeded in solving.

One should make a note of a fairly important though elementary mathematical fact: namely, that the Fourier transform does not preserve functional restriction. More specifically, if g(x,y)/W denotes the restriciton of the image function g(x,y) to *a* window W (so that g(x,y) is truncated outside W), then

F[g(x,y)|W] - F[g(x,y)]|w

is true for every W only when g(x,y) Is periodic with period equal to the si2e of W. Thus a Fourier image of a truncated function, truncated outside a window, will in general depend also on the part of the function g(x,y) whose domain is outside W. What this means practically is that certain texture elements could be split in half by windowing, and, as a consequence, an improper interpretation would be derived. This problem can be partly compensated for by overlapping windowing.

Conclusion

In this paper we have presented procedures for describing textured surfaces by operating in the Fourier domain of the image. Although the directionality in the Fourier domain has already been recognized , it has not been used in texture descriptions. The novelty of this method is that it recognizes some gross shape features (raonodirectional, bidirectional, nondirectional, etc.) of textures in the Fourier domain. In addition to the frequency properties, we are also able to make some estimates about the size of texture elements. The descriptions are symbolic. They are associated with a list of parameters (with corresponding numerical values) that are used at a higher level in the hypothesis-verification process. The data structure of texture descriptors is flexible and is expected to change during the hypothesis-verification activity. We are aware of several weaknesses inherent in this method. For instance, human perception tends to discount smooth changes In shading; yet the Fourier transform reflects not only the edges but also the slow changes. We are accustomed to regarding images in terms of homogeneous regions with sharp boundaries and to describing elements by brightness and color contrast and outline shape. In the Fourier domain, these become jumbled In a way that is only approximately resolved by our heuristics; thus, they are not always usefully described.

In addition, the texture elements (their shape) and their organization are also jumbled together in the Fourier domain. For instance, dots and small segments of lines organized in parallel-lined fashion will be described equally as monodirectlonal texture and, therefore, not in full detail. For more detail, one has to apply the spatial, local operators.

To sum it up, there are several possible texture operators, such as statistical and Fourier operators, interval analysis, and others. Some operators are better than others. Thus, the Fourier operators are very efficient for linear perodic textures as well as for linear regular but not periodic textures. Take grassy fields, waves in the water, etc. They are also useful for random dot patterns as long as the question is only the recognition of random organization versus regular organization (Example the sand texture). However, Fourier operator will fail to recognize the differences in more detailed shapes of texture elements. For example, it will not distinguish texture made of "capital A" and triangles (A) of the same size and organization. Here more local operators are necessary. So the spatial technique, being more local and therefore more accurate in some sense, can complement the Fourier technique. Which operator is used where should be determined by higher level program.

Acknowledgements

Tom Binford and John McCarthy have commented critically and helpfully on an earlier draft of the material. I wish to express my thanks also to the reviewers of this paper for their constructive criticism. Most of the work on this paper was carried out when the author was at the Artificial Intelligence project, Stan-

ford University, Stanford, California. Final preparation of the paper was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense under Contract No. SD-183.

FIGURE 1

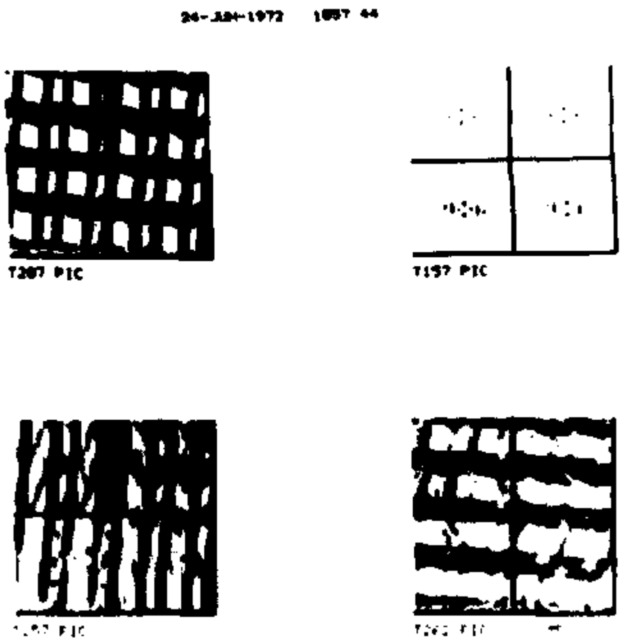


FIGURE 2

References

1. Bajcsy, R., "Computer Identification of Textured Visual Scene," AIM-173, Artificial Intelligence Project, Stanford University, Stanford, California, 1972,

2. *Beck, J., "Perceptual* Grouping Produced *by* Line-figures," Perception and Psychophysics 2. 1967, 491-495.

3. Blnford, *T.*, "A Visual Processor," Internal Report, Cambridge, Mass., MAC, MIT, 1970.

4. Brlce, C. R. and Pennema, C. L., "Scene Analysis Using Regions," Artificial Intelligence, 1, (1970), No. 3.

5. Hawkins, J. K., "Textural Properties for Pattern Recognition" In B. S. Lipkin (Ed.) Picture Processing and Psychopictorics, Nev. York: Acad. Press, 1970.

6. Husckel, M. H., "An Operator Which Locates Edges in Digitized Pictures," Journal of ACM, 18, (1971), 113-125.

7. Julesz, B. "Visual Pattern Discrimination," IRE TRANS. IT-8, (Feb., 1962), 84-92.

S. Rosenfeld, A. and Thurston, M., "Edge and Curve Detection for Visual Scene Analysis," IEEE Trans, on Computers, C-20 (1971), No. 5.

9. Rosenfeld, A., "Automatic Recognition of Basic Terrain Types from Aerial Photographs," Photogram. Enp. 28. (March, 1962), 639-646.

10. Rosenfeld, A., "Picture Processing by Computer," Computing Surveys, Vol. 1, No. 3, (September, 1969).

11. Rosenfeld, A., "Progress in Picture Processing: 1969-1971," Technical Report TR-176, University of Maryland, (January, 1972).

12. Rosenfeld, A., "Picture Processing: 1972," Technical Report TR-217, University of Maryland, (January, 1973).

13. Lendaris, G. and Stanley, G., "Diffraction-Pattern Sampling for Automatic Pattern Recognition," Proceedings of the IEEE. Vol. 58, No. 2, (February, 1970).

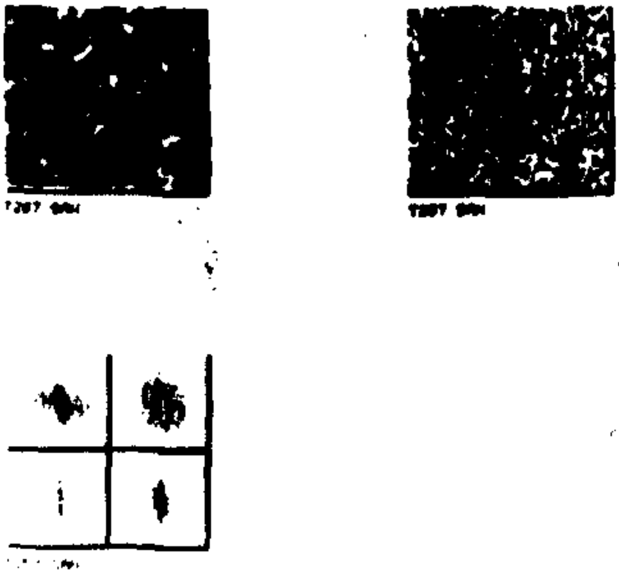


FIGURE 3

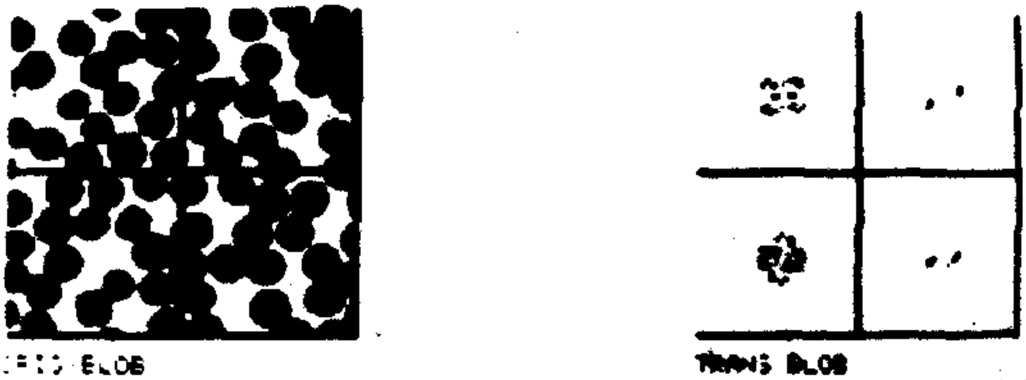


FIGURE 4

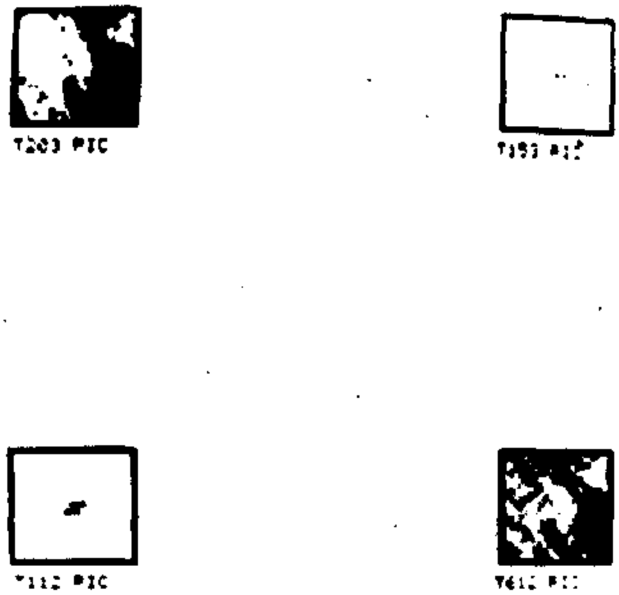


FIGURE 5

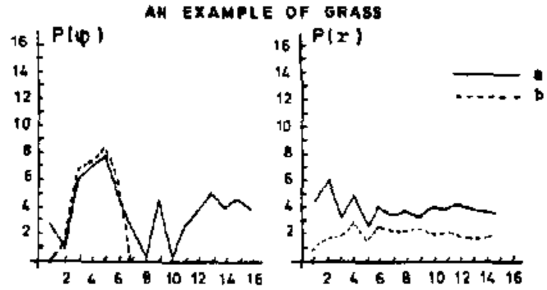


Figure 6

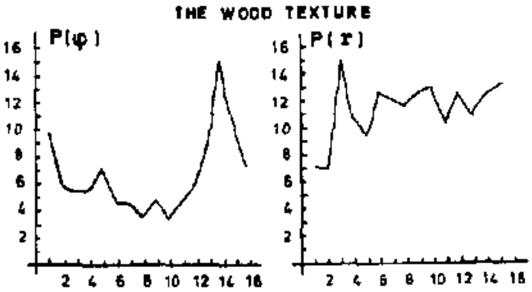


Figure 7

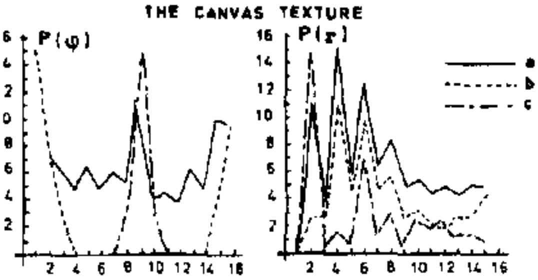


Figure 8

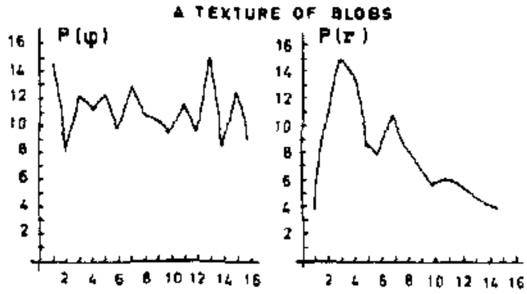


Figure 9

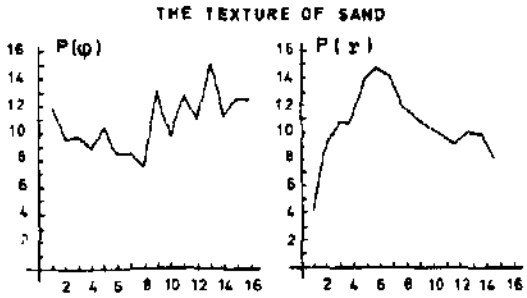


Figure 10