M. TECH. (COMPUTER SCIENGE) DISSERTATION SERIES

FRACTAL BASED IMAGE SEGMENTATION

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENTS OF THE REQUIREMENTS FOR THE M.TECH. (COMPUTER SCIENCE) DEGREE OF THE INDIAN STATISTICAL INSTITUTE

BY

ANIMESH ADHIKARI

UNDER THE SUPERVISION OF

DR. C. A. MURTHY

INDIAN STATISTICAL INSTITUTE
203, BARRACKPUR TRUNK ROAD
CALCUTTA - 700 035

-:CONTENTS :-

ABSTRACT

This dissertation addresses to the study of existing algorithms for finding fractal dimension and to find an effective algorithm for estimating fractal dimension and to study the utility of fractal dimension on (satellite) image segmentation problem.

Keywords - Fractals, image segmentation, fractal Brownian surface, Fourier power spectrum, histogram thresholding, smoothing.

1. INTRODUCTION

During the last thirty years, B. B. Mandelbrot [6, 7] has developed and popularized a relatively novel class of mathematical functions known as fractals . Subsequently, it was popularised by Barnsley [4],Pentland [2] etc.. A general description of fractal geometry can be found in [3, 4, 6, 7]. A more mathematical approach for fractal geometry can be found in Edger's book [5]. Fractal geometry has become very popular because of its wide range of applications in numerious fields. It has many applications in image processing and its related topics, e.g., image coding [4, 15] image segmentation [2, 14], image magnification [16] etc.. It also have applications in Physics [3], Chemistry [17]. We shall address here the problem of image segmentation using fractal dimension. Fractal dimension could play an important role in many applications. Pentland [2], Peleg [1] and others have shown many applications of fractal dimension ,e.g.,detection of edge points [2], image segmentation [2], textures comparison [1, 13, 14] etc.. Several techniques are available to segment an image [10, 11]. But, we know that, there does not exist any unified algorithm to solve segmentation problem completely. It can be intuitively stated that different values of fractal dimension imply different regions. Thus, fractal dimension is one such tool for segmenting an image. Here we want to apply this methodology to segment an IRS (Indian Remote-sensing Satellite) image.

There are many ways of defining fractal dimension. One definition was by Hausdorff and Besikovitch [5]. We shall follow here the following definition for fractal dimension [4]. Let $N(M,\epsilon)$ be the minimum number of m-dimensional objects (covering elements are e.g., squares, cubes or hyperspheres) of size ϵ needed to cover M (any fractal set). The number of objects needed to cover M is inversely proportional to ϵ^{D}.

i.e., $N(M,\epsilon) \propto \epsilon^{-D}$

or, $\log(N(M,\epsilon)) \propto (-D) * \log(\epsilon)$

or, $\log(N(M,\epsilon)) \propto D * \log(1/\epsilon)$

$$\text{Thus, } D = \lim_{\epsilon \to 0} \frac{\log(N(M,\epsilon))}{\log(1/\epsilon)}$$

Hence, if M is a curve and if the covering elements are squares of size ϵ , then the length $L(\epsilon) = (N(M,\epsilon) * \text{square}(\epsilon))/ \epsilon$

i.e. $L(\epsilon) = N(M,\epsilon) * \epsilon$.

So, $L(\epsilon) \propto \epsilon^{-D} * \epsilon = \epsilon^{1-D}$.

This finding is the same as that of Mandelbrot. The extension of this finding for surfaces gives rise to $A(\epsilon) \propto \epsilon^{2-D}$, where $A(\epsilon)$ is the area of surface, with diameter of the covering element is ϵ. In otherwords, $A(\epsilon) = F * \epsilon^{2-D}$, where F and D are constants for a particular surface.

2. Peleg's Method

There are several methods of computing fractal dimension for digital igames. We studied first Peleg's method [1], since it is computationally efficient. This method has been described below.

P1. Calculate the surface area $A(\epsilon)$, given ϵ = 1,2,3,.....

P2. Obtain pair of values ($\log(A(\epsilon))$, $\log(\epsilon)$) for every ϵ.

P3. Initialize a variable S by zero.

P4. Fit a straight line passing through three consecutive ordered pairs selected in P2.

P5. Find the slope of the line and add it to S.

P6. Do P4 and P5 until all three cosecutive ordered pairs are considered.

P7. Divide S by number of slopes obtained in step P5 and equate it to 2 - D. Obtain estimate of fractal dimension (D) ■

The step *P1* needs some explanation. All the points in the 3-dimensional space at a distance ϵ from the image surface are considered here resulting in a "blanket" of width 2ϵ around the image surface. Then the surface area of the image surface is the volume occupied by the blanket divided by 2ϵ. The covering blanket is defined by its upper surface u_ϵ and its lower surface b_ϵ. The substeps of step *P1* are defined as follows.

P1.1. Compute $A(\epsilon) = (v(\epsilon) - v(\epsilon - 1))/2$, $\epsilon = \partial, 2\partial, 3\partial, \ldots$; and ∂ = step value = 1; where

P1.2. $v(\epsilon) = \sum_{i,j} [u_\epsilon(i, j) - b_\epsilon(i, j)]$;where

P1.3. $u_\epsilon(i,j) = \max \left[u_{\epsilon-1}(i,j) + 1, \max_{|(m,n) - (i,j)| \le 1} u_{\epsilon-1}(i,j) \right]$

$b_\epsilon(i,j) = \min \left[b_{\epsilon-1}(i,j) - 1, \min_{|(m,n) - (i,j)| \le 1} b_{\epsilon-1}(i,j) \right]$

; where

P1.4. $u_0(i,j) = b_0(i,j) = g(i,j)$; where $g(i,j)$ represents the gray value at the (i,j)-th. pixel position ■

FIGURE SET - I

Notations :

n :Number of iterations (i.e. the number of times value of slope has been calculated)

fd(n) :Estimated fractal dimension using Peleg's algorithm based on n iterations.

The following figures show some output of Peleg's algorithm (based on 8 x 8 artifitially generated images of size 8 x 8).

A.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

n	fd(n)
3	2.00000
5	2.00000
7	2.00000
9	1.99999
11	1.99998
13	1.99997
15	1.99996

Fig. 1.1

B.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	30	30	30	30	30	30	30
0	30	30	30	30	30	30	30
0	30	30	30	30	30	30	30
0	0	0	0	0	0	30	30
0	0	0	0	0	0	30	30
0	0	0	0	0	0	30	30

n	fd(n)
3	2.37783
5	3.02673
7	2.80049
9	2.62259
11	2.50938
13	2.43100
15	2.37353

Fig. 1.2

C.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	30

n	fd(n)	
3	1.97730	
5	1.86486	*
7	1.80608	*
9	1.95723	
11	2.15069	
13	2.32859	
15	2.49574	

Fig. 1.3

D.

28	13	15	17	16	13	15	28
3	4	5	6	28	13	15	16
4	7	8	3	4	5	8	15
13	15	17	15	16	19	13	15
30	13	19	30	32	14	27	18
2	3	6	7	32	14	27	23
4	7	3	4	4	8	3	4
13	28	15	17	34	13	28	15

n	fd(n)
3	2.54260
5	2.83521
7	2.67300
9	2.52343
11	2.42825
13	2.36235
15	2.31404

Fig. 1.4

Note that star (*) show some typical undesirable fractal dimensions.

3. MODIFICATION IN PELEG'S METHOD

There are several problems in the Peleg's algorithm. As a result it has not estimated the desired fractal dimension properly. The problems are as follows.

- The maximum and the minimum gray values slowly spread over the u- and b-matrix respectly. Hence, estimated fractal dimension slowly tends toward the value 2 as the number of iterations increases. Again, it has been found for many cases that the calculated area when step value is 1 is significantly high. As a result average slope for less number of iterations may be even less than -1. Hence, fractal dimension in those cases may be greater than 3. This is because of the fact that the initial value of v is always 0.
- ϵ takes only integer values.
- In course of finding fractal dimension using above definition, the basic idea is to cover the whole set (M) by the covering elements. Thus, to get the fractal dimension of an image surface we require to cover the surface area by covering elements of diameter (or side) ϵ. But the idea of covering surface area in Peleg's method is not much appealing in the sense that it considers only 4-neighbourhood of a pixel (x,y). One may get better results by considering some intermediate points between two consecutive pixels in 8-neighbour of the pixel (x,y).

There is no doubt that the Peleg's method is computationally very efficient. But the above problems motivated us to modify the Peleg's algorithm. Another problem in Peleg's method [2] is that, fractal dimension was not constant over all scales. The modified algorithmn gives good result in most of the practal situations. But, the later problem is yet to be examined. The following modifications have been incorporated in Peleg's method.

M1. Discard v_0.

We have calculated surface areas of some 8 x 8 images to estimate fractal dimension for $\epsilon = 2\partial, 3\partial, 4\partial, \ldots$ and discarding the area for the value of ϵ being ∂, where ∂ = step value. Note that, ∂ may also take fractional value. For the purpose of estimating fractal dimension of larger square (image grid), one may discard $v_0, v_1, v_2, \ldots, v_i$ ($i \geq 1$) during the calculation of areas.

M2. Enriching the covering of surface idea.

Now let us explain this idea in two dimensional case for better understanding. Here we have a set of discrete sample points (integer pairs) for a curve. Let the set be $S = \{ (x,g) : x = x_1, x_2, x_3, \ldots$ and $g = g_1, g_2, g_3, \ldots \}$. The values of x representing some representative sample points along x-axis and values of g representing some pre-defined gray values. For simplicity, let us assume that, we have a straight line between two consecutive sample points. So, we have a set of straight lines in stead of the curve [Fig.4.2]. Let the covering elements are circles of radius ϵ. Here we are applying the Peleg's idea of covering the curve. Each sample point has two neighbours. So each sample point P(x,g) is a end point of two straight lines. We consider a fixed number of intermediate points for each of these two straight lines. Now we draw circles of radius ∂ at each of these intermediate points and check wether it cuts the line segment joined between the points P and (x,0) internally or externally and update b or u accordingly for its next step as done in Peleg's algorithm. Now it is easier to understand in 3-dimensional case. A sample point (x,y), with gray value g can uniquely correspond to a point $P(x,y,g)$ in 3-dimensional space. Let $Q(x_1,y_1,g_1)$ be one of the 8-neighbours of P. We have considered a fixed number of intermediate point between the line segment PQ. For each intermediate point we draw a circle of radius δ and check whether it cuts the line(L) formed by joining the two points $(x_1,y_1,0)$ and (x_1,y_1,g_1) or not. Let T and B be two local variables, initialised by $u_{\epsilon-1}(x_1,y_1) + \epsilon$ and $b_{\epsilon-1}(x_1,y_1) - \epsilon$ respectly. If the drawn circle cuts the line L above (or, below) the point (x_1,y_1,g_1).Let these cuts are <u>ucut</u> and <u>lcut</u> respectly. Update T (or, B) by max { T, <u>ucut</u> } (or, min { B, <u>lcut</u> }). Do the above procedure for all points in the 8-neighbours of Q. Based on the above idea we have shown some outputs in Figure Set - II. But the optimum number of iterations, optimum choices of δ and the optimum number of intermediate points are still open questions. Based on the outputs in Figure Set - II, we have drawn some figures showing the relationships between fractal dimension and above parameters. These figures are shown in Figure Set - III.

FIGURE SET - II

Notations :

M : The image under consideration

n : The size of the square image

Ni : The number of iterations (i.e. the number of times the area is calculated)

Np : The number of intermediate points

ϵ : Epsilon under consideration

Fd : Estimated fractal dimension

1. M = image A in Figure Set - I and n = 8

a. Np = 25 & ∈ = 0.5

Ni	Fd
3	2.00000
5	2.00000
7	2.00000
9	2.00001
11	2.00001
13	2.00002
15	2.00001
20	2.00008

b. Ni = 7 & Np = 25

∈	Fd
0.01	2.00000
0.05	2.00000
0.10	2.00000
0.30	2.00001
0.50	2.00000
0.70	2.00000
1.00	2.00000
1.50	2.00001
2.00	2.00000
3.00	1.99999

c. Ni = 7 & ∈ = 0.5

Np	Fd
5	2.00000
10	2.00000
15	2.00000
20	2.00000
25	2.00000
30	2.00000
40	2.00000
50	2.00000
100	2.00000
200	2.00000

Fig. 2.1

2 .M = image B in Figure Set - I and n = 8

a. Np = 25 & ∈ = 0.5

Ni	Fd
3	2.11152
5	2.40876
7	2.68455
9	2.84427
11	2.88054
13	2.84282
15	2.77910
20	2.62617

b. Ni = 7 & Np = 25

∈	Fd
0.01	2.00000
0.05	2.03065
0.10	2.07320
0.30	2.34733
0.50	2.68455
0.70	2.91027
1.00	2.79305
1.50	2.64620
2.00	2.55443
3.00	2.43715

c. Ni = 7 & ∈ = 0.5

Np	Fd
5	2.44110
10	2.65782
15	2.63472
20	2.70671
25	2.68455
30	2.71614
40	2.72527
50	2.72988
100	2.73440
200	2.73649

Fig. 2.2

3 .M = image C in Figure Set - I and n = 8

a. Np = 25 & ∈ = 0.5

Ni	Fd
3	1.98872
5	1.94340
7	1.89191
9	1.84910
11	1.81932
13	1.83122
15	1.89744
20	2.17271

b. Ni = 7 & Np = 25

∈	Fd
0.01	2.00000
0.05	2.00390
0.10	2.00090
0.30	1.95000
0.50	1.98091
0.70	1.83866
1.00	1.75627
1.50	1.74956
2.00	1.74303
3.00	1.73980

c. Ni = 7 & ∈ = 0.5

Np	Fd
5	1.90682
10	1.89901
15	1.89154
20	1.89166
25	1.89191
30	1.89299
40	1.89029
50	1.89116
100	1.89016
200	1.89063

Fig. 2.3

4 .M = image D in Figure Set - I and n = 8

a. Np =25 & ∈ = 0.5

Ni	Fd
3	2.23998
5	2.55977
7	2.70919
9	2.73583
11	2.70759
13	2.66418
15	2.62012
20	2.52680

b. Ni = 7 & Np =25

∈	Fd
0.01	2.00000
0.05	2.06975
0.10	2.14749
0.30	2.50534
0.50	2.70919
0.70	2.64095
1.00	2.60252
1.50	2.49407
2.00	2.43660
3.00	2.35131

c. Ni = 7 & ∈ = 0.5

Np	Fd
5	2.61120
10	2.67931
15	2.68203
20	2.71212
25	2.70919
30	2.71029
40	2.71698
50	2.71749
100	2.71825
200	2.71860

Fig. 2.4

FIGURE SET - III

Notations :

Fd : Fractal dimension

Ni : Number of iterations (i.e. # of values of ϵ)

Np : Number of intermediate points.

ϵ : Distance from the surface (toward top and bottom) to form a blanket of width 2ϵ.

The following figures show the emperical relation between the fractal dimension and the parameters of the modified algorithm.

Fig.3.1 : Relationship between Fd and Ni (when ϵ and Np are fixed)

Fig.3.2 : Relationship between Fd and ϵ (when Ni and Np are fixed)

Fig.3.3 : Relationship between Fd and Np (when ϵ and Ni are fixed)

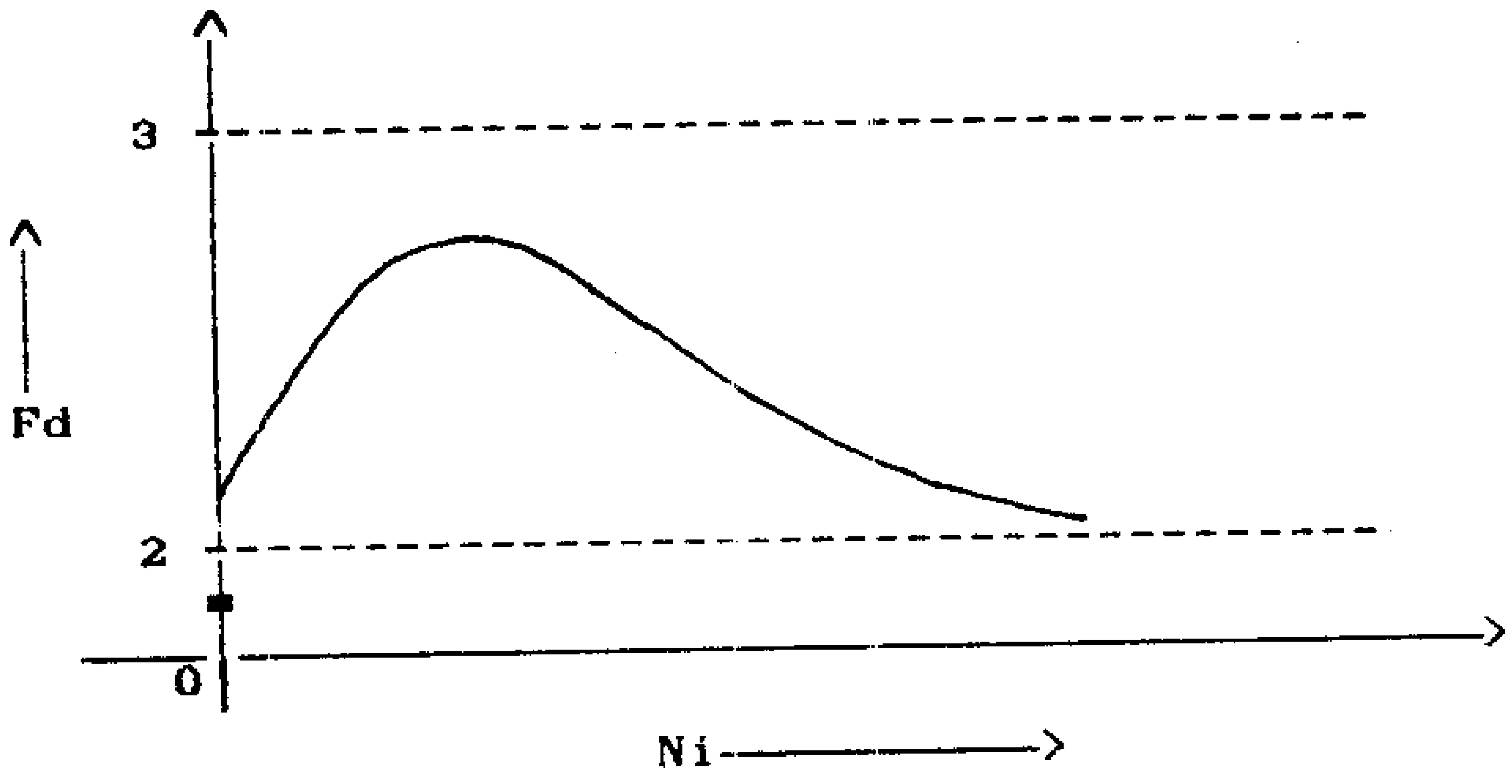

Fig.3.1

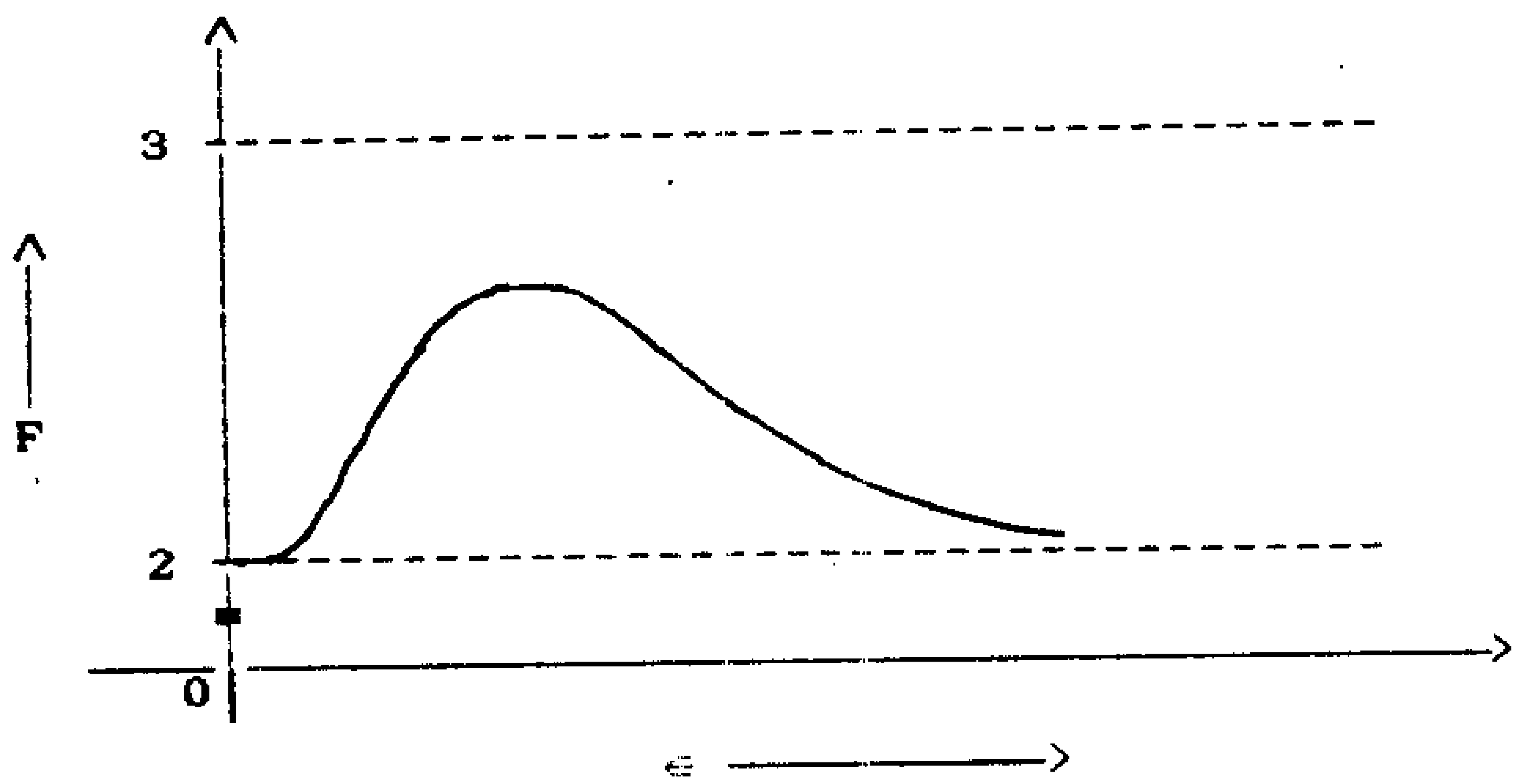

Fig.3.2

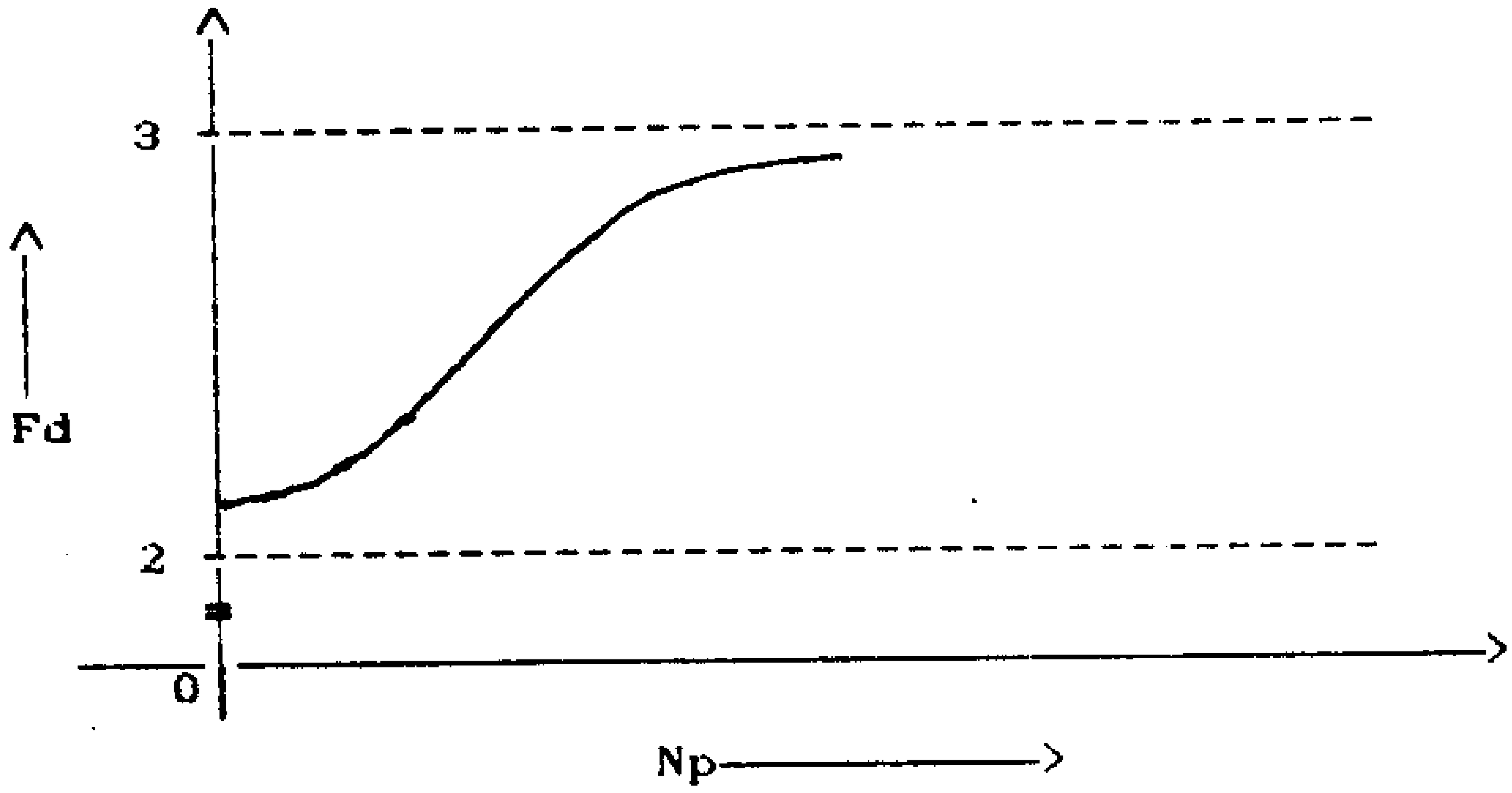

Fig.3.3

FIGURE SET - IV

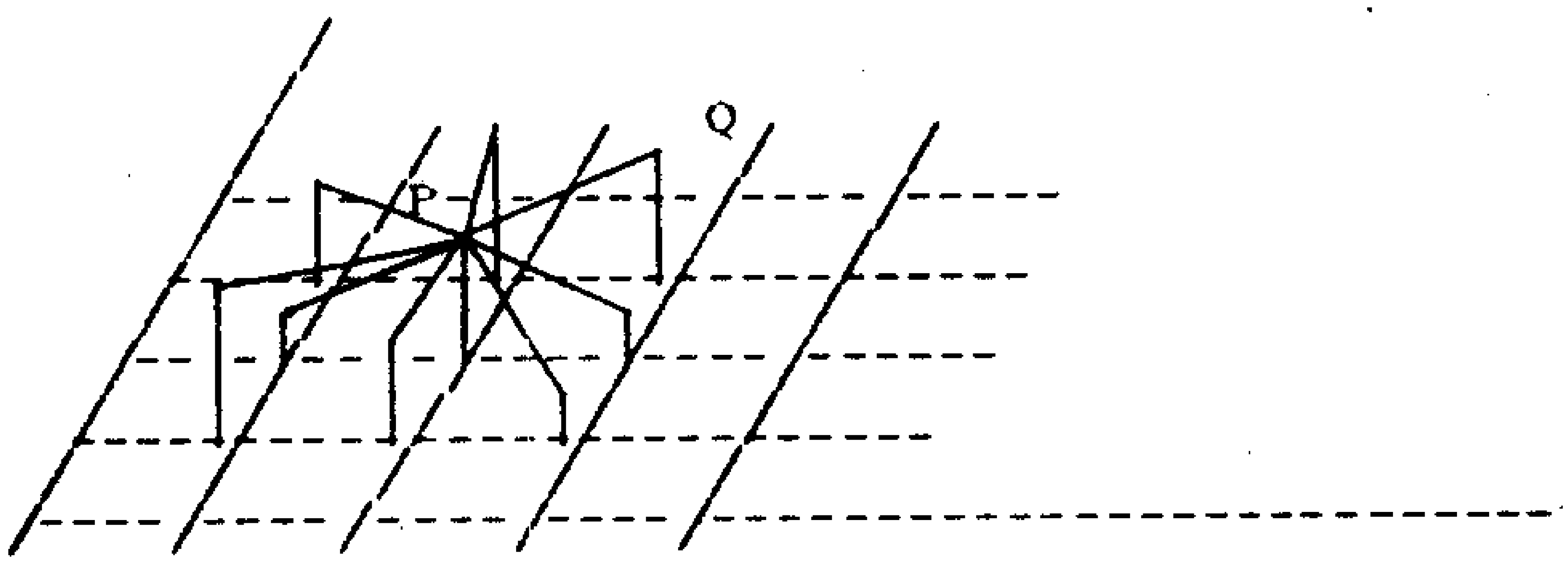

Fig.4.1 :A gray value (g) representing a 3-dimensional point P (x,y,g) when sampling done in X- and Y-direction.

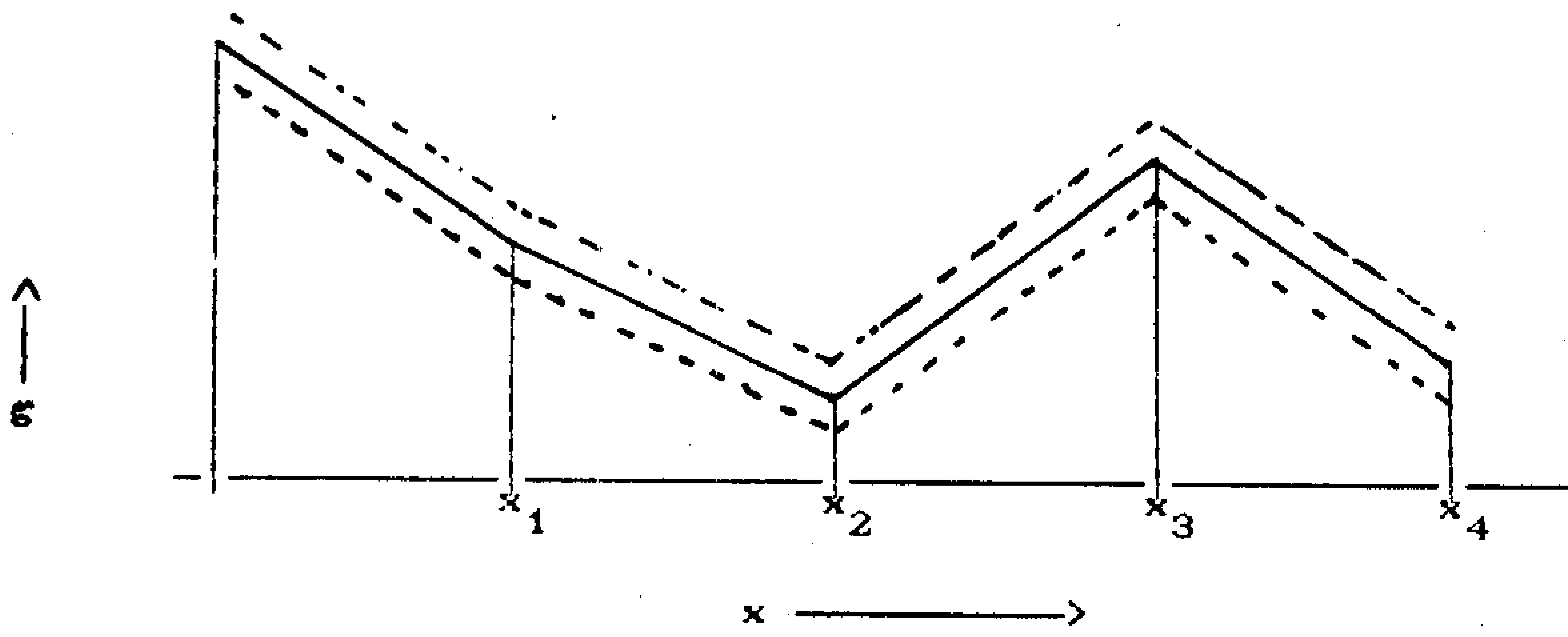

Fig.4.2 :Shows an iteration step in Peleg's method when ∂ is very small (in 2-dimension).

M3. Restricting number of intermediate points.

Intermediate points are obtained between two points of which one belongs to the 8-neighbour of other and the estimated fractal dimension are shown in Figure Set - II. From the outputs in Figure Set - II, the emperical relationship between the number of intermediate points and fractal dimension is shown in Fig. 3.3. The more the number of intermediate points select the better the accuracy of the estimates. From the experimental results it has been found that, if the number of intermediate points are within 50 to 70, fractal dimension converges to some fixed value, provided the image size is 8 x 8. This observation is dependent on the nature of the gray values. If the image surface is smooth enough convergence requires less number of intermediate points (viz. less than 20). So, we feel that 60 will be good enough for our case. But due to the constraint on the processing time, we have taken number of intermediate points to be equal to 5.

M4. Restricting the choice of step value.

From the outputs in Figure Set - II, the emperical relationship between ϵ and the fractal dimension is shown in Fig. 3.2. Our experimental result shows that in most of the cases, estimated value of fractal dimension takes maximum value when $0.7 \leq \partial \leq 0.9$. Since input to the modified algorithmn is a random image, we suggest to draw a random number between 0.7 and 0.9, and take that value as the value of ∂. For our case, we took $\partial = 0.85$.

Remark : We can not make ∂ as small as possible. Let us explain the reason in 2-dimentional case with reference to Fig.4.2. Let x_1 and x_2 be two sample points. We are considering a fixed number of sample points in the line segment formed by the two points $P_1(x_1,g(x_1))$ and $P_2(x_2,g(x_2))$. At each intermediate point we are drawing a circle of radius ϵ and checking whether it cuts the line segment between the points P_1 and $(x_1,0)$ internally or externally. But if ϵ is very small the drawn circle can never cut it. Hence, at the next iteration, we shall get new u snd b -surface given by the upper dotted and lower dotted line respectly. This in turn implies that $(v_{\epsilon} - v_{\epsilon-\partial})$ is constant, where ∂ = step value; i.e., $A(\epsilon)$ becomes constant i.e., estimated slope becomes zero. Similarly, it is not appealing to set ∂ to a high value.

M5. Restricting the number of iterations.

With each iteration, the maximum and minimum value of the image spread over the matrices u and b respectly. Performance deteriorates when maxium or minimum elements cover the entire u or b matrices respectively. So, our objective is to restrict the number of iterations. To calculate the desired number of iterations the following procedure has been suggested.

Step1. Identify all the maximum (G)and minimum (g)gray values and their corresponding positions. Let m_1 be the number of times the maximum gray value has occurred and m_2 be the number of times the minimum gray value has occurred. If G = g then set $d^*= 3$ and stop .

Step2. Let the coordinates of maximum gray value be $(x_i, y_i), i = 1,2,3....,m_1$.

for i = 1 to m_1 do
  for j = (i+1) to m_1 do
    Calculate $d_{ij} = \lceil (|x_i-x_j| + |y_i- y_j|) / 2 \rceil + 1$.

Step3. For each maximum gray value with coordinates (x,y) do
  Calculate $|x-0| + |y-0|$, $|x-n+1| + |y-0|$, $|x-0| + |y-n+1|$, and $|x-n+1| + |y-n+1|$, where e = (x,y).

Step4. Calculate the following
  Let $d_1 = \min_i |x_i-0| + |y_i-0|$, $d_2 = \min_i |x_i-0| + |y_i-n+1|$,
  $d_3 = \min_i |x_i-n+1| + |y_i-0|$, $d_4 = \min_i |x_i-n+1| + |y_i-n+1|$,

Step5. Calculate $\bar{d} = \max \{ d_1, d_2, d_3, d_4, d_{ij}$ for all i,j }
  where d_{ij}'s are obtained in *Step2*.

Step6. In case of minimum elements, do the *Steps 2* to *5*, and get $\underline{d}$.

Step7. Let d = k * min { $\underline{d}$, $\bar{d}$ }, where k ∈ (0,1). As explained earlier k is very close to 0.5. In our case we took k = 0.45.

* d stands for desired number of iterations.

Remarks :

1. Keeping other parameters fixed, it can be intuitively seen that as the number of iterations increases, estimated fractal dimension increases and ultimately tends to a constant value.

2. The algorithm for computing the number of iterations is not optimal. But it gives a reasonable estimate.

4. Fourier Power Spectrum Method

In Pentland's paper [2], fractal dimension has been calculated based on the fact that power spectrum P(g) is proportional to g^{-2H-1}. One may fit linear regression using the pair of observations (log(P(g)), log(g)), for various values of gray value g to determine H (i.e. 2-D). Hence we get the required estimate of fractal dimension D. The frequency distribution of estimated fractal dimensions of IRS Bombay image are shown in Table - III of Table Set - I, and the estimated values are not at all satisfactory.

5. Fractal Surface Generation

We have used Feder's [3] algorithm to generate fractal surface. This algorithmn is based on fractal Brownian motion. We have generated a few 16 x 16 images with known fractal dimension and compared the fractal dimension with the modified Peleg's method and the existing methods.

FIGURE SET - 5

Notations :

n	: Number of iterations (i.e. number of distinct values of ϵ taken)
$fd_P(n)$	: Estimated fractal dimension using Peleg's algorithm, with number of iterations is equal to n
fd_F	: Value of fractal dimension of a matrix generated by Feder's surface generation algorithm
fd_f	: Fractal dimension using Fourier power spectrum
fd_m	: Fractal dimension using modified Peleg's algorithm

The following 16 x 16 images (shown in terms of gray values) has been generated by Feder's surface generation algorithm , with fractal dimension fd_F.

1. Image with $fd_F = 2.5$

0	32	71	69	75	64	78	52	28	50	75	99	106	104	129	153
18	18	50	49	47	43	58	56	57	54	87	126	119	129	129	159
23	44	62	61	19	51	56	75	91	82	94	112	114	110	99	104
45	72	77	113	82	85	71	95	103	90	110	115	115	99	91	48
26	53	61	87	111	74	45	89	135	128	142	142	132	134	119	76
48	54	62	65	80	54	60	80	111	145	142	176	152	187	136	103
36	51	69	80	89	78	61	74	84	121	104	125	111	153	134	123
36	45	78	97	95	110	77	72	108	150	134	106	140	180	158	143
61	93	102	98	85	79	67	73	126	140	173	147	164	171	173	192
110	163	125	110	93	53	53	27	107	111	148	146	174	165	196	242
143	128	125	101	123	86	66	90	163	137	162	179	220	188	202	232
111	80	84	47	96	101	105	104	129	114	161	186	186	164	195	222
111	112	86	88	113	106	149	138	136	172	182	192	174	165	193	183
150	170	125	105	86	60	108	162	151	255	197	227	170	130	153	144
172	159	140	102	67	69	63	83	49	133	126	162	150	135	142	144
161	153	124	96	91	87	73	59	81	103	125	147	133	120	132	144

(a) Using Peleg's algorithmn

n	$fd_P(n)$
5	2.44458
10	2.66701
15	2.89690
20	3.08609
25	3.22034

(b) Using Fourier power spectrum

$fd_f = 1.96699$

(c) Using modified Peleg's algo-rithmn

$fd_m = 2.60167$

Fig. 5.1

2. Image with $fd_F = 2.7$

112	91	111	126	136	121	126	108	48	108	135	160	190	173	154	186
77	41	107	162	113	59	73	88	66	55	103	107	137	122	136	156
79	74	115	114	97	78	19	68	74	111	115	134	134	115	110	146
87	62	65	83	103	138	82	93	111	199	170	186	160	101	106	137
111	46	0	65	89	106	77	101	80	146	179	185	225	158	76	131
50	9	49	87	95	119	118	156	105	125	152	148	187	228	155	126
25	44	101	84	85	119	122	124	62	105	156	138	146	183	189	166
49	40	63	62	104	151	159	157	114	76	121	102	157	169	182	206
96	80	49	90	120	157	204	170	161	123	148	170	210	189	166	187
104	136	109	130	123	152	164	157	149	107	175	219	217	213	194	169
70	115	120	106	91	143	143	160	170	181	227	225	226	229	230	183
117	132	100	85	119	188	156	170	200	221	233	230	227	245	214	198
115	106	64	107	111	152	125	189	240	231	254	236	207	202	183	191
129	114	123	170	160	185	168	223	213	210	137	252	218	174	188	184
121	140	143	172	173	160	138	168	181	207	231	238	240	212	211	205
156	183	193	202	173	144	137	129	168	207	218	230	226	221	224	226

(a) Using Peleg's algorithmn

n	$fd_p(n)$
5	2.51689
10	2.67654
15	2.98456
20	3.18860
25	3.29219

(b) Using Fourier power spectrum

$fd_f = 2.36041$

(c) Using modified Peleg's algo -rithmn

$fd_m = 2.73409$

Fig. 5.2

3. Image with $fd_F = 2.3$

0	23	31	45	82	883	104	157	173	184	168	150	153	141	168	138
17	21	26	0	67	68	120	168	184	208	182	153	140	106	133	142
47	40	53	59	118	109	141	174	186	197	198	176	146	131	116	127
28	40	61	65	88	111	158	200	192	194	185	207	166	156	138	111
30	69	88	77	60	117	180	184	188	171	140	174	153	157	168	149
83	119	100	94	106	119	170	169	171	164	171	198	164	150	162	187
86	97	98	124	153	175	214	200	164	182	183	189	157	165	144	202
97	87	120	152	162	213	220	254	221	216	210	221	191	210	186	217
84	94	146	139	129	177	200	220	250	217	221	195	178	185	173	217
81	58	111	131	137	168	179	177	210	180	192	159	174	181	187	218
111	102	110	113	119	144	170	177	195	196	207	199	180	184	176	198
106	128	118	91	124	118	143	164	175	203	199	250	194	198	184	178
103	130	142	134	167	145	118	144	139	158	136	180	148	171	185	182
105	146	137	136	150	176	145	153	152	154	163	186	168	152	187	186
106	128	125	138	122	147	133	148	162	180	178	196	186	183	223	196
98	136	152	168	162	156	150	143	184	225	230	235	202	169	188	206

(a) Using Peleg's algorithmn

n	$fd_P(n)$
5	2.36565
10	2.53090
15	2.84284
20	3.08321
25	3.20666

(b) Using Fourier power spectrum

$fd_f = 1.65563$

(c) Using modified Peleg's algo -rithmn

$fd_m = 2.37682$

Fig. 5.3

6. SEGMENTATION

We have used Pentland's segmentation algorithm to segment an image. For that purpose we require to calculate the fractal dimension of several m x m sub-squares of the given image. The algorithm for segmenting the image is as follows.

Step1. For each m x m subimage estimate the fractal dimension.
Step2. Find the histogram of the estimated dimensions.
Step3. Find the valleys of the histogram.
Step4. Take the valley points as the thresholds and classify the image.

We have applied the above algorithm on an IRS Bombay image of size 128 x 128 (Fig.6). Each pixel in the image occupies 36.25 x 36.25 sq. meters area on earth and the image corresponds to infrared band [12]. *Step1* in the above algorithm has been calculated using Peleg's method, modified Peleg's method and Pentland's method. We have taken the value for m as 9. The values of the fractal dimensions are calculated up to two decimal places correctly. The histograms ,in the form of frequency tables are shown in Table Set - I. After obtaining the histogram, we have applied a smoothing technique [8, 9] on it to detect valleys. The results of the segmentation of the image are shown in Fig.7, Fig.8, Fig.9. The modified Peleg's method shows better segmentation, since they match better with the ground truth. We have obtained three classes after segmentation in this method. These classes are named as class-I, class-II and class-III. Class-I (having very low fractal dimension) consists of water. Class-II (having high fractal dimension) consists of land. Class-III (having moderate fractal dimension) consists of land

and water. But it is predominantly land. The actual water pixels those are covered by Class - III, fall in two categories. The first category consists of all those pixels which fall in the boundary region of water and land. The second category consists of all those water pixels where the variations in the gray levels of the surrounding pixels are comparatively high. Similarly, the Class - I also covers some land areas, where the variations are very low. The modified Peleg's method shows better segmentation. But the last method based on Fourier power spectrum segments image very poorly. The results of the Peleg's method in segmentation is also given here.

There are several reasons for misclassification of pixels using a fractal dimension based criterion.

- Different textures may possess the same fractal dimension.
- Inefficiency of the algorithm for estimating fractal dimension properly.

But, a good fractal dimension estimating method should provide informations to segment a given image (at least) partially. Fractal dimension based segmentation works well when the image is very textured on smooth background.In the satellite image segmentation problem some land areas may appear very smooth so that their estimated dimensions are very close to 2. Hence after segmentation they appears to be water. Again some water areaswhose variations are high due to the presence of inside land, have high fractal dimension. Hence, after segmentation some water may appear to be land.

In general, fractal dimension is just one criterion for segmentation. For a proper segmentation of real life image, some more criteria are to be considered to make segmentation closer to ground truth.

TABLE SET - 1

Notations:

Q : Class number F : Frequency

The following tables represent the frequency distributions of the estimated fractal dimensions. Q = 1 represents the class with estimated value of fractal dimension (D) < 0. Q = 2 represents the class with $0 \leq D < 1$. Q = 3 represents the class with $1 \leq D < 2$. Q = 104 represents the class with $3 \leq D < 4$. Q = 105 represents the class with $4 \leq D < 5$. Q = 106 represents the class with ≥ 5. For $4 \leq Q \leq 103$, the corresponding class is represented by the value for $D = 2 + (Q - 4) / 100$. Table - I corresponds to the Peleg's method with anticipated number of iterations = 5. Table- II corresponds to modified Peleg's method with $\epsilon = 0.85$ and number of intermediate points = 5. Table - III corresponds to Fourier power spectrum method. The results are based on an IRS Bombay image of size 128 x 128.

Table-II

Q	F	Q	F	Q	F	Q	F
1	0	20	458	39	128	58	3
2	0	21	434	40	0 *	59	3
3	168	22	431	41	115	60	5
4	262	23	355	42	118	61	2
5	414	24	415	43	83	62	2
6	610	25	355	44	84	63	1
7	748	26	378	45	81	64	1
8	621	27	356	46	62		
9	579	28	335	47	51		
10	503	29	712	48	30		
11	469	30	319	49	42		
12	444	31	296	50	34		
13	355	32	302	51	24		
14	383	33	235	52	21		
15	0 *	34	202	53	23		
16	388	35	229	54	7		
17	362	36	167	55	20		
18	409	37	174	56	6		
19	435	38	138	57	5		

■ other classes, those are not mentioned have freqency zero.

* corresponding value of x represents valley after smoothing histogram

Table-I

Q	F	Q	F	Q	F	Q	F
1	0	20	319	39	53	58	3
2	0	21	326	40	0 *	59	3
3	5255	22	253	41	29	60	2
4	703	23	305	42	32	61	1
5	304	24	259	43	24	62	0
6	295	25	216	44	14	63	2
7	296	26	259	45	21		
8	264	27	229	46	18		
9	257	28	209	47	17		
10	231	29	395	48	10		
11	250	30	173	49	9		
12	297	31	154	50	10		
13	331	32	121	51	7		
14	396	33	139	52	10		
15	0 *	34	120	53	3		
16	412	35	96	54	6		
17	381	36	78	55	9		
18	346	37	65	56	3		
19	330	38	60	57	2		

■ other classes, those are not mentioned have frequency zero.

* corresponding value of x represents valley after smoothing histogram.

Table-III

Q	F	Q	F	Q	F	Q	F	Q	F
1	789 +	20	55	39	61	58	56	:	
2	859 +	21	51	40	60	59	56	.	
3	2491 +	22	50	41	0 *	60	60	90	58
4	148	23	52	42	55	61	51	91	0 *
5	40	24	71	43	54	62	54	92	49
6	44	25	61	44	56	63	51	93	44
7	44	26	51	45	62	64	65	94	50
8	41	27	57	46	54	65	56	95	47
9	49	28	55	47	73	66	0 *	96	44
10	42	29	66	48	62	67	66	97	29
11	56	30	115	49	68	68	50	98	46
12	62	31	62	50	53			99	30
13	50	32	55	51	77			00	39
14	48	33	60	52	57			01	51
15	52	34	76	53	56			02	43
16	0 *	35	49	54	56			03	31
17	57	36	57	55	47			04	2205 +
18	55	37	65	56	139			05	0
19	52	38	75	57	53			06	2

□ The values which are not mentioned above are more or less homogenious, and are around 45.

* Corresponding value of x represents valley after smoothing histogram.
+ Corresponding value of frequency represents an undesirsble value.

7. REFERENCES

1. S. Peleg ,J. Naor , R. Hartley and D. Avnir , "Multiple Resolution Texture Analysis and Classification", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 518-523, July 1984.

2. A. P. Pentland, "Fractal-based description of natural scenes", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 518-523, July 1984.

3. J. Feder, *Fractals*, Plenum Publishing Corporation, New York, NY, 1988.

4. M. T. Barnsley, *Fractals Everywhere*, Academic Press, New York, NY, 1983.

5. Gerald A. Edger, *Measure, Topology and Fractal Geometry*, Springer Verlag, New York, 1990.

6. B. B. Mandelbrot, *Fractals :Form, Chance and Dimension*, San Francisco, CA : Freeman, 1977.

7. B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman, New York, NY, 1983.

8. C. A. Murthy and S. K. Pal, "Fuzzy thresholding :Mathematical framework, bound functions and weighted moving average technique", *Pattern Recognition Letters*, vol. 11, pp. 197-206, 1990.

9 C. A. Murthy and S.K. Pal, "Histogram thresholding by minimizing gray level fuzzyness", *Information Sciences*, vol. 60, no. 1/2, pp. 107-135, 1992.

10 R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Reading, MA :Addision Wesley, 1987.

11. A.Rosenfeld an A. C. Kak, *Digital Picture Processing*, New York :Academic Press, 1982.

12. "Indian Remote Sensing SatelliteData Users Handbook",

Document No. - IRS/NRSA/NDC/HB-02/89, NRSA, Hyderabad (India), Dec., 1989.

13. J. M. Keller, R. M. Crownover and R. Y. Chen, "Characteristics of natural scenes related the fractal dimension", *IEEE Trans. on Pattern Anal. and Machine Intell.*, 9:621-627, 1987.

14. J. M. Keller, S. Chen and R. H. Crownover, "Texture description and segmentation through fractal geometry", *Computer Vision, Graphics, and Image Processing*, 45:150-166, 1989.

15. A. E. Jacquin, "A novel fractal block coding technique for digital images", *Proceedings of IEEE International conference on Accoutics, Speech, and Signal processing*, pp. 2225-2228, 1990.

16. L. Lalitha and D. D. Majumder, "Fractal based criteria to evaluate the performance of digital image magnification techniques", *Pattern Recognition Letters*, 9:67-75, 1989.

17. D. Avnir and P. Pfeifer, "Fractal Dimension in Chemistry :An intensive characteristic of surface irregularity", *Nouv. J. Chim.*, vol. 7, pp. 71-72, 1983.

8 IMAGES

Fig.6 :**IRS (infrared** band) Bombay image of **size 128 x 128**.

Fig.7 :**Segmentation** of IRS Bombay **image using Pentland's algorithm** (**fractal** dimensions are **estimated by Peleg's method with anticipated** number of iterations is equal to 5)

Fig.8 :**Segmentation** of IRS Bombay image using Pentland's **algorithm (fractal dimensions** are **estimated by modified Peleg's method with** ∈ = 0.85 and number of **intermediate points** = 5)

Fig.9 :**Segmentation** of IRS Bombay **image using** Pentland's **algorithm** (fractal dimensions are estimated by the method **described in** [2]).

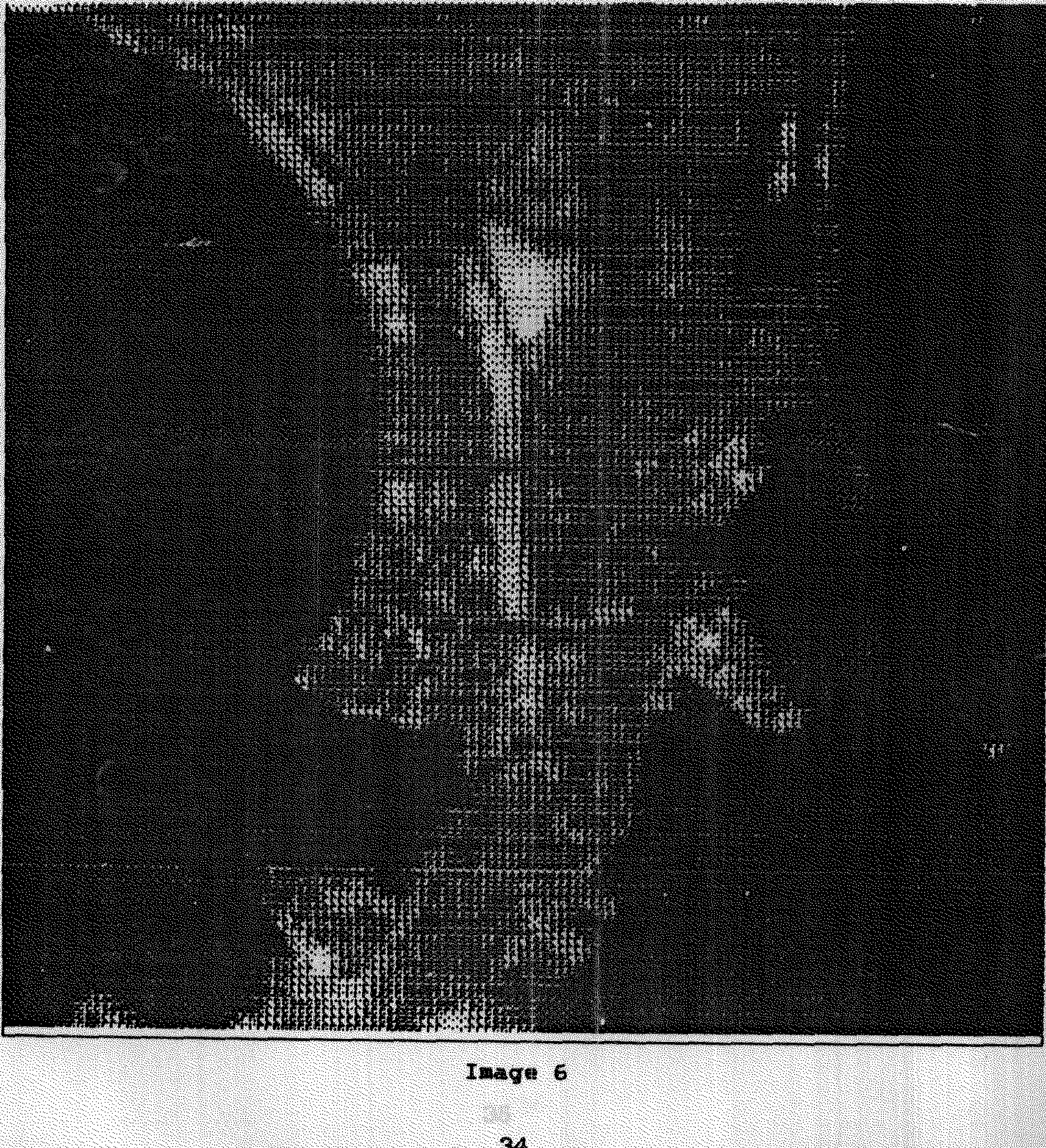

Image 6

Image 7

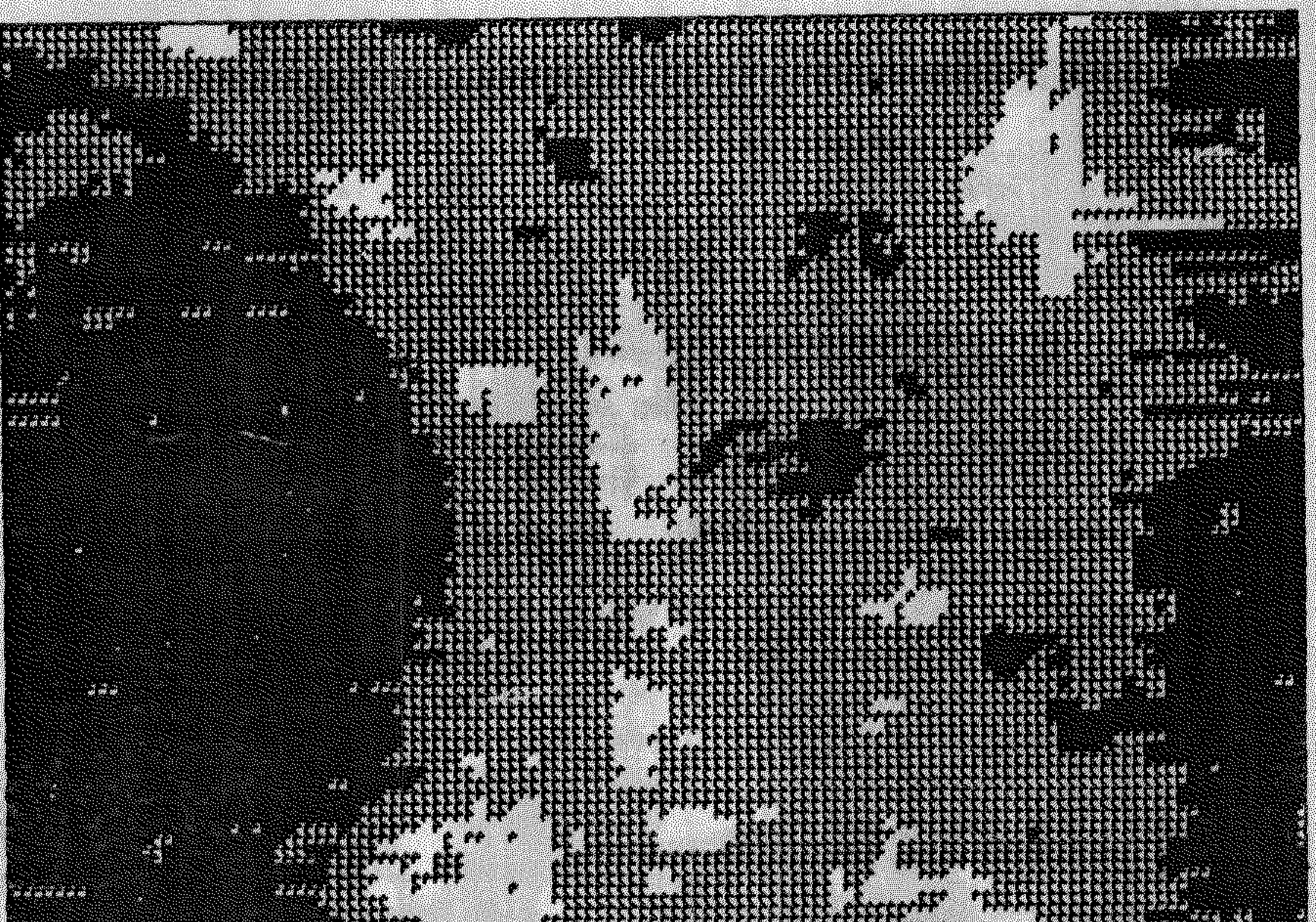

Image 8

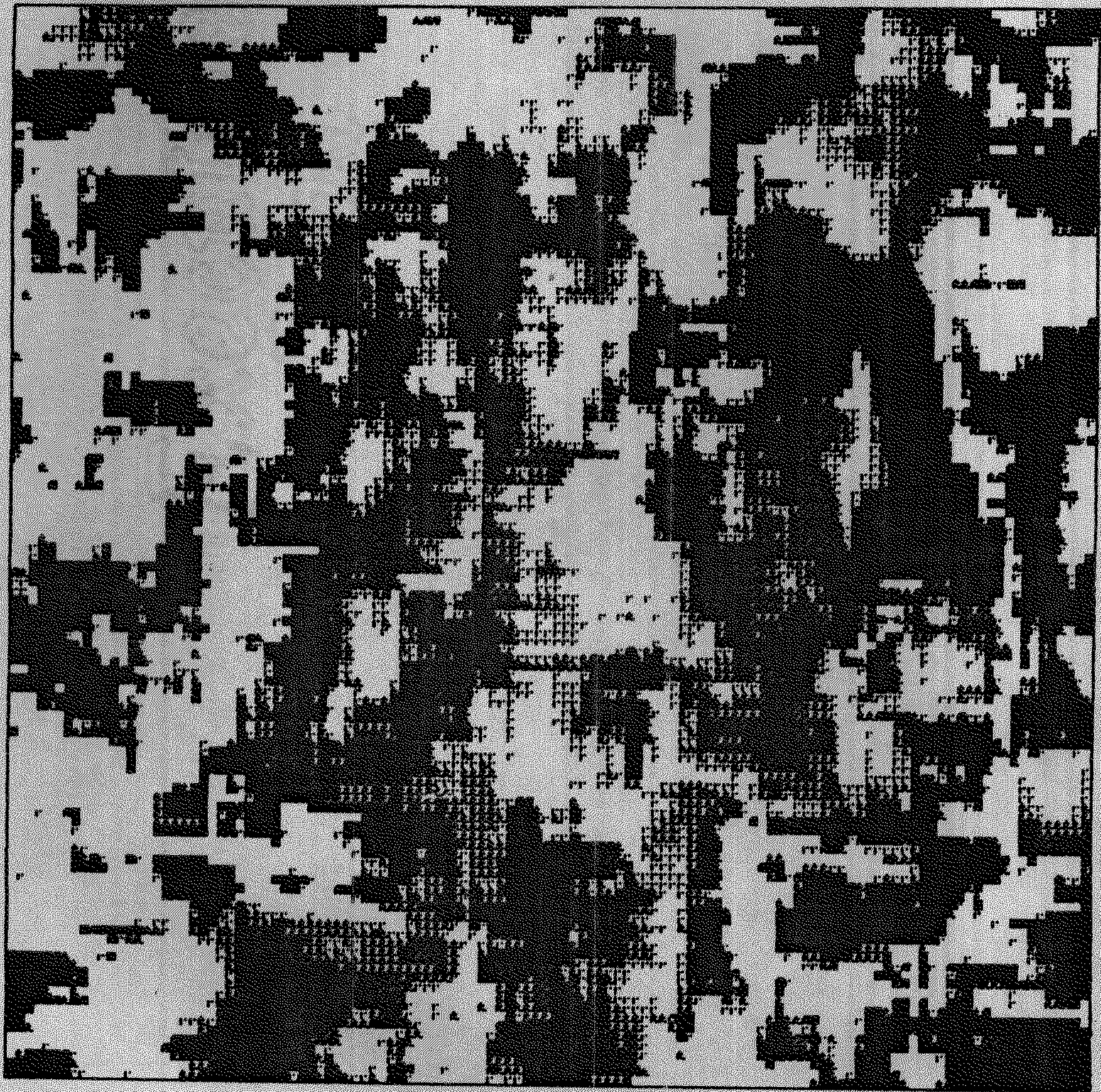

Image 9