M.Tech. (Computer Science) Dissertation Series

EFFICIENT COMPUTATION OF INVARIANT MOMENTS FOR GRAY- LEVEL IMAGES

A Dissertation Submitted in Partial Fulfilment of the Requirements for the M Tech(Computer Science) Degree of the Indian Statistical Institute

By

DARPAN MAJUMDER

Under the supervision of

Prof. Dr. Bhargab B. Bhattacharya

INDIAN STATISTICAL INSTITUTE
203, Barrackpur Trunk Road.
Calcutta – 700 035.

Computer Science Dissertation Series

EFFICIENT COMPUTATION OF INVARIANT MOMENTS FOR GRAY-LEVEL IMAGES.

Darpan Majumder
M. Tech. (Computer Science)

A Dissertation Thesis
Under the supervision of

Prof. Dr. Bhargab B. Bhattacharya
Advance Computing & Microelectronics Unit

INDIAN STATISTICAL INSTITUTE
203, Barrackpur Trunk Road.
Calcutta – 700 035.
India.

Certificate of Approval

This is to certify that the thesis titled *Efficient Computation Of Invariant Moments For Gray Level Images* submitted by Darpan Majumder, towards fulfillment of the requirement for the Master of Technology in Computer Science at Indian Statistical Institute, Calcutta, embodies the work done under my supervision during the period from *July 1999* to *July 2000.*

Dated :

Prof. Dr. Bhargab. B. Bhattacharya.
(Supervisor)

P. S. Das-gupta

(External Examiner)

Acknowledgements

This gives me immense pleasure to express my indebtedness to my supervisor Professor Dr. Bhargab B. Bhattacharya , who despite his hectic schedule always found time to help me with the difficult problems and knew when to trust me with the simpler ones. I also feel eternal gratitude to him for initiating me into the paradigm of mapping of algorithms to architecture.

I am very grateful to Mr. Arijit Bishnu, who in spite of his own hectic schedule always found time to be with me and guided me in each and every step of my dissertation work. Due to his friendly attitude and sincerity to guide me, I never felt fatigued or distressed. I really don't know how to express my gratitude to him.

I would like to thank Dr. Malay K. Kundu of Machine Intelligence Unit, Indian Statistical Institute, who has also supervised me in solving the critical problem of image feature computing. He also provided me with good references from time to time which were very useful to find out the right direction quickly.

I thank Prof. Dr. Bhabani P. Sinha for providing me with the computing facilities of the unit during last one year. He also gave me his valuable suggestion from time to time.

I thank Jayanta Kumar Dey and Tapas Chakraborty for spending their valuable time to discuss with me about my algorithms and their implementation.

I thank the Professors, scientists and staff members of the unit for providing me all the help during the period I worked in the unit.

I must express my heartiest gratitude to my parents and my brother. Without their constant inspiration and moral support this work might not have been possible.

Last but not the least, I thank all my classmates and friends who helped me in some way or the other and shared the moments of joy and frustration during my stay at ISI.

Darpan Majumder
Indian Statistical Institute
July 21, 2000.

Abstract

The field of digital image processing is continually evolving. It finds massive application in areas like medical imaging, remote sensing, geological surveys, weather forecasting etc. With the rapid growth in Internet and multimedia technology , image processing applications are more in demand. Images not only require a large storage space ,they also demand a lot of real time to get processed.

High performance, special purpose computer systems are typically used to meet these specific application requirements. With advancement in VLSI technology, cost as well as size of chips continue to fall. On the other hand image processing algorithms and their requirements continue to be better understood. The above phenomena resulted in the manufacture of special chips to perform the image processing tasks. This dramatically reduced the processing time.

In this work we have tried to develop methods by which we can assign certain signature values to the images. These signature values are to be used as the parameter for Content Based Image Retrieval to retrieve similar images. Parallel algorithms for the methods used have also been provided. All the images used here are gray level images and the analysis has been done keeping this in mind.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Background

The idea of searching for an image with the help of a signature value assigned to it forms the basis of Content Based Image Retrieval. Image is generally stored in the form of a two dimensional matrix. Our task is essentially to find out a function which maps these two dimensional pixel arrays to a unique value.

The distance between two images will be determined with the help of the difference in the signature values assigned to them. If the distance between these two signature values is within a certain tolerable limit, these two images are considered look-alike. Otherwise the two images are considered to be different. So in this way we are able to separate similar and dissimilar images. Our objective is to determine such a measure which can be used as an image discriminating feature.

1.2 Need For VLSI Implementation

The image is generally stored in the form of two dimensional pixel array. Any sequential algorithm would take at least $O(N^2)$ to compute as it has to visit all the pixels at least once. Hence the need to parallelise, thereby reducing the time complexity of the algorithm.

The VLSI implementation of the algorithms used helps in implementing the whole procedure in the form of a chip, enabling an increase in the speed of computation. However, the advantages of this technology are not fully realized unless simple, regular and modular layouts are used. Fortunately an image processing task can usually be decomposed into a set of sub-tasks distributed over the image. This observation encourage to devise regular and parallel algorithms for image processing tasks which can be easily mapped to regular modules of an architecture.

1.3 Problem Formulation

The basic objective of this project as mentioned earlier is to locate features by which we can assign signature values to the various images taken into consideration. Here we have taken into consideration texture of an image as the feature by which we can discriminate images. The measure used are a set of invariant moments which remain invariant under generalised linear transformations. Parallel algorithms for computation of moments have also been developed. For our analysis purpose we have taken into consideration gray level images.

1.4 Organization Of The Thesis

The thesis consists of four chapters. The first is the introduction and the second is about texture, the feature used for image discrimination. The third chapter is about the measures used and the fourth provides a parallel algorithm for the whole process.

Chapter 2

TEXTURE AS AN IMAGE DISCRIMINATING FEATURE

2.1 Introduction

The recent emergence of multimedia digital libraries makes image retrieval an interesting and challenging problem. The problem involved in this context is the ability to classify images or in other words discriminate between images using some measure which calculates a signature value for each image. In our work we have used texture as one such image discriminating feature.

In search of an image discriminating feature for describing pictorial information, it is only natural to look forward to the types of features which human beings use in interpreting pictorial information. Spectral , textural and contextual features are three fundamental pattern elements used in human interpretation of colour photographs. Spectral features describe the average tonal variations in various bands of visible and/or infrared portion of electromagnetic spectrum, whereas as textural features contain information about the spatial distribution of tonal variations in a band. Contextual features contain information derived from blocks of pictorial data surrounding the area being analyzed. As we are considering only gray level pictures, spectral features are not considered. When small image areas from black and white photographs are independently processed by a machine, texture and tone are most important.

Concept of tone is based on varying shades of gray of resolution cells in a photographic image while texture concerns spatial (statistical) distribution of gray tones. Texture and tone are not independent concepts; rather, they bear an inextricable relationship to one another very much like the relationship between a particle and a wave. Context, texture and feature are always present in an image, although at times one property can dominate the other.

Texture is an innate property of all surfaces. It contains important information about the structural arrangement of surfaces and their relationship to surrounding environment. Although it is quite easy for human observers to recognize and describe texture in empirical terms, it has been extremely refractory to precise definition and to analysis by digital computer.

Although no formal definition of texture exists, intuitively this descriptor provides measures of properties such as smoothness, coarseness and regularity. The notion of texture appears to depend upon three ingredients.

i) Some local 'order' is repeated over a region which is large in comparison to the order's size.
ii) The order consists in the nonrandom arrangement of elementary parts.
iii) The parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.

From the above definitions image texture can be defined as a function of the spatial variation in pixel intensities(gray values). Image texture has a variety of applications and has been a subject of intense study by many researchers. One immediate application of image texture is recognition of images using texture properties. We are using this property as an image discriminating feature which will later form the basis of Content Based Image Retrieval.

2.2 A Taxonomy of Texture Models

Identifying the perceived qualities of texture in an image is an important first step towards building mathematical models for texture. The intensity variations in an image which characterize texture are generally due to some underlying physical variation in the scene. Modeling this physical variation is very difficult , so texture is usually characterized by the two-dimensional variations in the intensities present in the image. As there is no precise definition about texture , there are a number of intuitive properties of texture which are generally assumed to be true.

i. Texture is a property of areas; the texture of a point is undefined. So, texture is a contextual property and its definition must involve gray values in a spatial neighbourhood. The size of this neighbourhood depends upon the texture type, or the type of the primitives defining the texture.

ii. Texture involves the spatial distribution of gray levels. Thus ,two-dimensional histograms or co-occurrence matrices are reasonable texture analysis tools.

iii. Texture in an image can be perceived at different scales or levels of resolution. For example, consider the texture represented in a brick wall. At a course resolution, texture is perceived as formed by individual bricks in the wall; the interior details in the brick are lost. At a higher resolution when only a few bricks are in the field of view, the perceived texture shows the details in the brick.

iv. A region is perceived to have texture when the number of primitives in the region are large. If only a few primitive objects are present, then a group of countable objects is perceived instead of a textured image. In other words, a texture is perceived when significant individual "forms" are not present.

Image texture has a number of perceived qualities which play an important role in describing texture. Some of these properties are uniformity, density, coarseness, roughness, regularity, linearity, directionality, direction, frequency and phase. Some of these perceived qualities are not independent. For example, frequency is not independent of density and the property of direction also applies to directional features. The fact that the perception of texture has so many different dimensions is an important reason why there is no single method of texture representation which is adequate for a variety of textures.

We may classify the methods of texture evaluation in the following way.

2.2.1 *Statistical Methods:*

One of the defining qualities of texture is the spatial distribution of gray level values. The use of statistical features is therefore one of the early methods proposed in machine literature. In the following, we will use $\{I(x,y), 0<=x<=N-1, 0<=y<=N-1\}$ to denote an $N \times N$ image with G gray levels. Some of the proposed methods are

2.2.1.1 *Co-occurrence Matrices*

Spatial gray level co-occurrence estimates image properties related to second order statistics. Haralick suggested the use of gray level co-occurrence matrices which has become one of the most well-known and widely used texture measures. The G X G gray level co-occurrence matrix P_d for a displacement vector d=(dx,dy) is defined as follows. The entry (i,j) of P_d is the number of occurrences of the pair of gray levels i and j which are a distance d apart. Formally, it is given as

$$P_d(i,j) = | \{ ((r,s), (t,v)) : I(r,s) = i, I(t,v) = j \} |$$

where (r,s), (t,v) $\in$ N X N, (t,v) = (r+dx,s+dy), and $|\cdot|$ is the cardinality of the set.

A number of measures can be evaluated from this transformed matrix and These measures are widely used for texture feature extraction.

2.2.1.2 *Autocorrelation Features*

An important property of many textures is the repetitive nature of the placement of texture elements in the image. The autocorrelation function of an image can be used to access the amount of regularity as well as the fineness/coarseness of the texture present in the image. Formally, the autocorrelation function of an image I(x , y) is defined as follows

$$R(x, y) = \frac{\Sigma \Sigma I(u, v) I(u + x, v + y)}{\Sigma \Sigma I^2(u, v)}$$

2.2.2 *Geometrical Methods*

The class of texture analysis that falls under the heading of geometrical methods is characterized by their definition of texture as being composed of "texture elements" or primitives. The method of analysis usually depends on the geometric properties of these texture elements. Once the texture elements are identified in the image , there are two major approaches to analyzing texture. One computes statistical properties from the extracted texture elements and utilizes this as texture features. The other tries to extract the placement rule that describes the texture. The latter approach involves geometrical methods of analyzing texture.

In this method the image is generally tessellated into a number of parts and texture measures are applied on these parts. Generally in geometrical methods the measure used is that of moments. The ($p+q)^{th}$ order moments of area of a closed region **R** with respect to the centroid $(x_0 , y_0)$of the region is given by

$$m_{pq} = \int\int (x - x_0)^p (y - y_0)^q \, dx \, dy$$

where $p+q = 0,1,2,3 \ldots\ldots\ldots$

2.2.3 *Signal Processing Methods*

Psychophysical research has given evidence that the human brain does a frequency analysis of the image. Texture is especially suited for this type of analysis because of its properties. Most techniques based on this method compute certain features from filtered images which are then used in either classification or segmentation tasks. The filtering is done either in the spatial domain or in the frequency domain. Based on the above we have either spatial domain filters or frequency domain filters.

Spatial domain filters are the most direct way to capture texture properties. Earlier attempts on defining such methods concentrated on measuring the edge density per unit area. Fine textures tend to have a higher density of edges per unit area than coarser texture. Measurement of such edges is usually computed by convolving the image matrix with a mask of known size.

Frequency analysis of the textured image is best done in the Fourier domain. As the psychophysical results indicated, the human visual system analyses the images by decomposing the image into its frequency and orientation components. This encouraged the use of filters to obtain image features.

2.3 Conclusion

In this chapter we have elaborated the concepts the and techniques for processing images using its texture. Texture is a prevalent property of most physical surfaces in the natural world. We have used this property to evaluate signature values for each image and using this we can identify an image from various other images. The measure used by us is the method of moments. It is a statistical as well as a geometric measure. This aspect is dealt with detail in the next chapter.

Chapter 3

MOMENTS AS A MEASURE OF TEXTURE

3.1 Introduction

Moment is considered to be a powerful measure of texture. We will devise a method to use it as a measure of texture. To be more specific we are using a set of four moments, as proposed by Reiss and Hu, as a parameter for identification of images. These measures are considered to be invariant under rotation scaling and under any general linear transformation. The fact that the measure remains unchanged under the above transformations makes it all the more suitable for its consideration as a signature value for an image, which can be used for identification of an image. In the following sections we will describe the idea and the method of computation of invariant moments.

3.2 Mathematical Model

For a 2-D continuous function $f(x, y)$, the moment of order $(p + q)$ is defined as

$$m_{pq} = \int\int x^p y^q f(x, y)\, dx\, dy$$

for $p, q = 0, 1, 2, 3 \ldots\ldots$

A uniqueness theorem (Papoulis[1965]) states that if $f(x, y)$ is piecewise continuous and has nonzero values only in a finite part of the *xy* plane, moments of all order exists and the moment sequence (m_{pq}) is uniquely determined by $f(x, y)$. Conversely, (m_{pq}) uniquely determines $f(x, y)$. The *central moments* can be expressed as

$$\mu_{pq} = \int\int (x - x\_avg)\,(y - y\_avg)\, f(x, y)\, dx\, dy$$

where

$x\_avg = m_{10}/m_{00}$ and $y\_avg = m_{01}/m_{00}$

For digital image the above equation becomes

$$\mu_{pq} = \sum\sum (x - x\_avg)^p (y - y\_avg)^q f(x, y)$$

Then the central order moments up to order 3 can be computed in terms of ordinary moments in the following way

$$\mu_{10} = \sum\sum (x - x\_avg)^1 (y - y\_avg)^0 f(x,y)$$

$$= m_{10} - m_{10} m_{00} / m_{00} = 0$$

$$\mu_{11} = \sum\sum (x - x\_avg)^1 (y - y\_avg)^1 f(x,y)$$

$$= m_{11} - m_{01} m_{10}/m_{00}$$

$$\mu_{20} = \sum\sum (x - x\_avg)^2 (y - y\_avg)^0 f(x,y)$$

$$= m_{20} - 2m_{10}^2/m_{00} + m_{10}^2/m_{00} = m_{20} - m_{10}^2/m_{00}$$

$$\mu_{02} = \sum\sum (x - x\_avg)^0 (y - y\_avg)^2 f(x,y)$$

$$= m_{02} - m_{01}^2/m_{00}$$

$$\mu_{30} = \sum\sum (x - x\_avg)^3 (y - y\_avg)^0 f(x,y)$$

$$= m_{30} - 3x\_avg\, m_{20} + 2x\_avg^2\, m_{10}$$

$$\mu_{03} = \sum\sum (x - x\_avg)^0 (y - y\_avg)^3 f(x,y)$$

$$= m_{03} - 3y\_avg\, m_{02} + 2x\_avg^2\, m_{01}$$

$$\mu_{12} = \sum\sum (x - x\_avg)^1 (y - y\_avg)^2 f(x,y)$$

$$= m_{12} - 2\, y\_avg\, m_{11} - x\_avg\, m_{02} + 2\, y\_avg^2\, m_{10}$$

$$\mu_{21} = \sum\sum (x - x\_avg)^2 (y - y\_avg)^1 f(x,y)$$

$$= m_{21} - 2\, x\_avg\, m_{11} - y\_avg\, m_{20} + 2\, x\_avg^2\, m_{01}$$

The *normalized central moments* [1],denoted t_{pq} , are defined as

$$t_{pq} = \mu_{pq} / \mu_{00}^{c}$$

where $c = (p+q)/2 + 1$

for $p + q = 2,3........$

Reiss derived a set of four invariant moments from the *centralized* second and third order moments. Hu proposed another set of moments using *normalized central order moments*. We have evaluated the moments proposed by Reiss using *centralized* as well as *normalized central order moments*. Results seem to favour the use of *normalized central order moments.*

The moments proposed by Reiss[6] are given below

$$T_1 = \mu_{20}\mu_{02} - \mu_{11}^2$$

$$T_2 = (\mu_{30}\mu_{03} - \mu_{21}\mu_{12})^2 - 4(\mu_{30}\mu_{12} - \mu_{21}^2)(\mu_{21}\mu_{03} - \mu_{12}^2)$$

$$T_3 = \mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)$$

$$T_4 = \mu_{30}^2\mu_{02}^3 - 6\mu_{30}\mu_{21}\mu_{11}\mu_{02}^2 + 6\mu_{30}\mu_{12}\mu_{02}(2\mu_{11}^2 - \mu_{20}\mu_{02}) +$$

$$\mu_{30}\mu_{03}(6\mu_{20}\mu_{11}\mu_{02} - 8\mu_{11}^3) + 9\mu_{21}^2\mu_{20}\mu_{02}^2 - 18\mu_{21}\mu_{12}\mu_{20}\mu_{11}\mu_{02} +$$

$$6\mu_{21}\mu_{03}\mu_{20}(2\mu_{11}^2 - \mu_{20}\mu_{02}) + 9\mu_{12}^2\mu_{20}^2\mu_{02} - 6\mu_{12}\mu_{03}\mu_{11}\mu_{20}^2 + \mu_{03}^2\mu_{20}^3$$

Invariant moments are computed from the above using the following formulae

$$I_1 = T_1/\mu_{00}^4 , \quad I_2 = T_2/\mu_{00}^{10} , \quad I_3 = T_3/\mu_{00}^7 , \quad I_4 = T_4/\mu_{00}^{11}$$

These moments are invariant under translation and general linear transformations.

The above measures have also been computed in the terms of *central order moments.* We have also computed the above mentioned moments in terms of the *normalized central order moments* which gives better results.

3.3 Choice of Function

The measure of moments can be analyzed in the following way. The product terms can be thought of as two parts

i. The spatial distribution part consisting of the product terms comprising x and y.

ii. The image part, which can be taken into consideration by appropriate choice of the function $f(x, y)$.

We choose $f(x,y)$ in such a way that its characteristics are similar to that of the Companting Curve[9](curve depicting the sensitivity of the human eye to the changes in gray levels). Its characteristic is as shown in the following curve

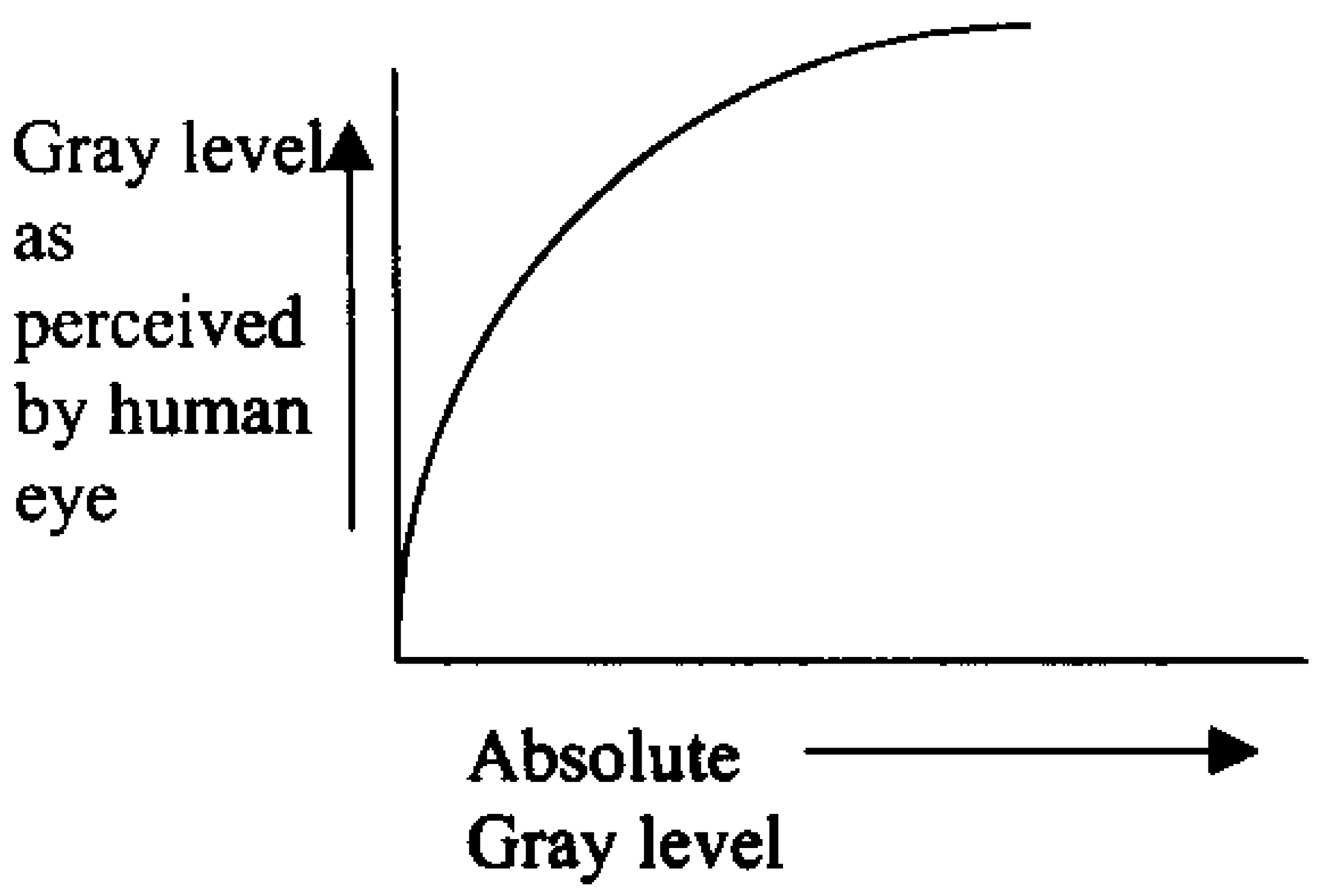


Figure 3.1
Plot of the Companting Curve

As can be seen from the figure, the curve is similar in nature to the curve $(1- e^{-x})$. So, we choose the function as $f(x, y) = 1 - exp(-I[x,y] / I_{average})$

Here $I[x, y]$ = gray value of pixel (x, y)
and $I_{average}$ is the average gray value of the whole image.

The nature of this function is similar to that of the Companting Curve. Actually this function is a non decreasing function of $I[x, y] / I_{average}$. Division by $I_{average}$ is required so that the function becomes independent of gray level scaling.

3.4 Experimental Results

Table1: These are the results obtained by using *central order moments* on the *original* image.

Image	1st moment	2nd moment	3rd moment	4th moment
AFRICA.G	243.53891	-2213.56612	-247.83	4845.63
BLAZE.G	172.884598	9282.257812	15.33	-137.10
CASTLE.G	188.61136	4954.041504	65.68	-599.04
CATHED.G	173.49121	3999.79248	-187.95	2386.73
CATTLE.G	215.6022	-10950.028	-961.85	13594.46
CHIMP.G	202.203	291998.25	190.30	-2368.29
ARMY.G	190.33	5654.43	-56.71	2011.47
CHOPER.G	185.4896	3850.99	57.17	-445.37
COUPLE.G	205.66	-74560.07	-1263.07	18468.88
FISH.G	158.00	9681.68	153.58	-957.02
GOLDFISH.G	175.34	-287959.88	-2648.21	27500.93
HAWK.G	205.00	-25642.10	-1051.82	13793.95
ICE.G	143.9	830.52	-112.42	995.97
INSECT.G	170.97	124.98	20.38	-157.13
KID1.G	178.36	5123.33	33.82	-263.30
KID2.G	191.76	4262.46	8.22	-15.44
KID3.G	178.97	430.4	12.61	-129.04
LEAF.G	175.15	2324.8	-42.14	456.76
NEWENG.G	234.45	15213.28	375.47	-4554.165
PHOTOGRA.G	248.4	34372.32	-74.83	1155.36
RGB3.G	193.66	1429.10	17.88	252.94
ROCK.G	176.26	101134.52	240.27	-2333.20
RODEO.G	218.55	247861.45	-345.68	6294.24
ROSE.G	170.65	0.011844	0.531356	-0.9628
SANTA.G	165.17	11676.23	-53.1356	860.77
SEAFISH.G	175.53	1262.23	-35.9	876.53
SOLDIR.G	200.45	2443.86	235.97	-2582.95
STAR.G	205.28	1.084	4.273	-44.46
SUNSET.G	163.58	78.95	-52.17	637.76
TOWN.G	192.9	524.9	-130.95	1615.79
WAVE.G	140.58	-2136.36	-306.56	3660.56
ZEBRA.G	195.96	1445.36	22.05	232.73

Table2: These are the results obtained by using *central order moments* on the *rotated* image.

Image	1st moment	2nd moment	3rd moment	4th moment
AFRICA.G	243.08	-2147.03	-247.08	4776.016
BLAZE.G	172.87	9461.56	19.33	-188.99
CASTLE.G	188.67	5245.72	63.96	-574.48
CATHED.G	173.55	3775.91	-189.24	2405.82
CATTLE.G	215.75	-11747.17	-962.74	13615.33
CHIMP.G	202.16	200007.00	212.45	-2644.79
ARMY.G	190.43	4191.64	-65.75	2139.21
CHOPER.G	185.43	3789.21	56.42	-439.70
COUPLE.G	205.77	-74095.75	-1263.03	18441.24
FISH.G	157.86	9833.62	153.70	-954.65
GOLDFISH.G	175.76	-288273.46	-2659.33	27662.94
HAWK.G	204.70	-24982.39	-1053.63	13779.91
ICE.G	144.00	837.05	-111.80	990.88
INSECT.G	170.93	99.18	19.19	-144.19
KID1.G	178.47	5034.01	32.98	-254.50
KID2.G	191.67	4539.8	12.12	-54.10
KID3.G	179.11	452.76	14.34	-146.78
LEAF.G	175.09	1996.9	-43.40	471.45
NEWENG.G	234.51	15217.03	375.67	-4558.12
PHOTOGRA.G	247.93	39841.19	-56.89	871.32
RGB3.G	193.72	1306.55	17.67	260.22
ROCK.G	176.24	99552.32	237.21	-2305.06
RODEO.G	218.61	247194.28	-356.11	6316.27
ROSE.G	170.52	0.019483	-0.42487	8.0233
SANTA.G	165.35	13499.69	-46.10	767.80
SEAFISH.G	175.53	1262.13	-35.94	876.64
SOLDIR.G	200.40	2371.77	233.00	-2551.08
STAR.G	205.34	5.81	5.77	-59.96
SUNSET.G	163.50	57.38	-52.37	636.38
TOWN.G	192.77	579.20	-127.41	1568.44
WAVE.G	140.55	-1954.88	-305.76	3657.10
ZEBRA.G	195.88	1420.53	21.20	238.69

Table3: These are the results obtained by using *normalized central order moments* on the original image.

Image	1^{st} moment	2^{nd} moment	3^{rd} moment	4^{th} moment
AFRICA.G	3.8769	-22.3822	-0.03144	0.9787
BLAZE.G	1.66173	26.5864	0.000804	-0.006918
CASTLE.G	1.6620	11.4198	0.002960	-0.0238
CATHED.G	1.5299	9.2366	-0.008482	0.094978
CATTLE.G	2.5780	-54.1380	-0.074	1.2498
CHIMP.G	1.9772	873.053	0.01029	-0.126195
ARMY.G	2.7152	43.4672	-0.0054	0.3005
CHOPER.G	1.4176	6.2188	0.002008	-0.01196
COUPLE.G	3.0068	-609.33	-0.13806	2.9514
FISH.G	2.0539	58.9863	0.013668	-0.110724
GOLDFISH.G	3.298139	-4419.014	-0.4499	8.78865
HAWK.G	1.82205	-60.39051	-0.04812	0.5609
ICE.G	1.208	1.6959	-0.004655	0.0346515
INSECT.G	1.31825	0.2063	0.000727	-0.004655
KID1.G	1.25825	6.7716	0.001033	-0.005673
KID2.G	1.4279	6.449	0.000276	-0.000387
KID3.G	1.3084	0.6220	0.00041	-0.00307
LEAF.G	1.3208	3.6308	-0.00145	0.01182
NEWENG.G	2.3992	50.9669	0.02199	-0.2198
PHOTOGRA.G	3.1192	192.0647	-0.006268	0.121493
RGB3.G	1.7229	3.374	0.000819	0.010318
ROCK.G	1.5388	227.774	0.010655	-0.010655
RODEO.G	2.416	1007.075	-0.02377	0.4662
ROSE.G	1.1289	0.000013	0.000015	-0.000017
SANTA.G	1.3657	22.9545	-0.002175	0.0287
SEAFISH.G	1.3657	2.110071	-0.001292	0.024448
SOLDIR.G	1.645191	4.7161	0.009391	-0.084371
STAR.G	1.5467	0.001690	0.00014	-0.001148
SUNSET.G	1.18381	0.1112	-0.001666	0.014736
TOWN.G	1.5808	1.009204	-0.005198	0.05256
WAVE.G	1.08936	-3.5706	-0.01103	0.10208
ZEBRA.G	1.7716	3.5524	0.001040	0.009918

Table4: These are the results obtained by using *normalized central order moments* on the *rotated* image.

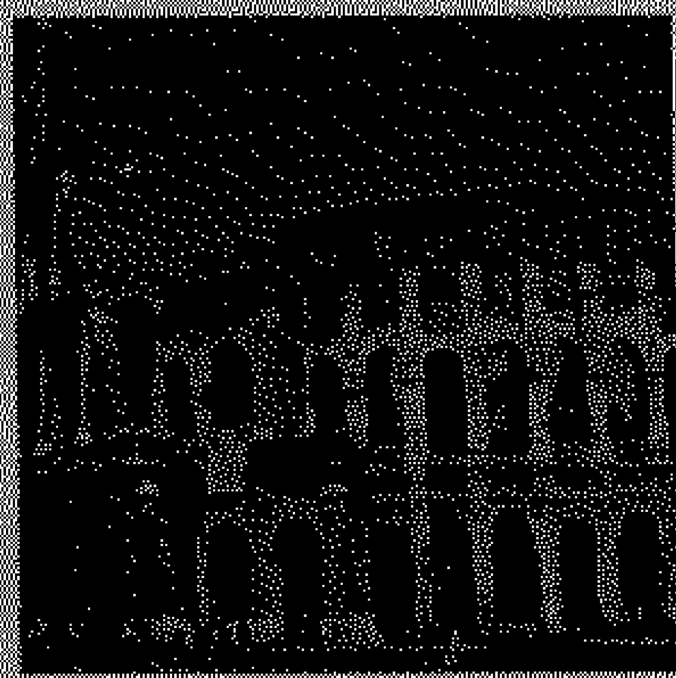
Image	1^{st} moment	2^{nd} moment	3^{rd} moment	4^{th} moment
AFRICA.G	3.861239	-21.5912	-0.03123	0.9588
BLAZE.G	1.66172	27.103	0.001014	-0.009536
CASTLE.G	1.66224	27.1029	0.002882	-0.0228
CATHED.G	1.531095	8.7285	-0.008546	0.095844
CATTLE.G	2.581917	-58.196827	-0.074129	1.2545
CHIMP.G	1.9765	896.671	0.011485	-0.1398
ARMY.G	2.71805	32.2587	-0.006891	0.3199
CHOPER.G	1.4168	6.1147	0.0019	-0.0117
COUPLE.G	3.0100	-606.3880	-0.1381	2.9514
FISH.G	2.0496	59.7247	0.0136	-0.1100
GOLDFISH.G	3.3168	-4459.5590	-0.4543	8.9190
HAWK.G	1.8168	-58.6277	-0.0480	0.5581
ICE.G	1.2098	1.7125	-0.0046	0.0345
INSECT.G	1.3171	0.1635	0.0006	-0.0039
KID1.G	1.2598	6.6645	0.0010	-0.0054
KID2.G	1.4265	6.8608	0.0004	-0.0013
KID3.G	1.3105	0.6556	0.0004	-0.0034
LEAF.G	1.3200	3.1163	-0.0014	0.0121
NEWENG.G	2.4005	51.0179	0.0220	-0.2733
PHOTOGRA.G	3.1078	221.6379	-0.0047	0.0911
RGB3.G	1.7241	3.0879	0.0008	0.0106
ROCK.G	1.5384	224.1005	0.0105	-0.0891
RODEO.G	2.4169	1004.7661	-0.0238	0.4681
ROSE.G	1.1271	0.00000	-0.00000	0.0001
SANTA.G	1.3683	26.5931	-0.0018	0.0256
SEAFISH.G	1.3602	2.1098	-0.0012	0.0244
SOLDIR.G	1.6443	4.5740	0.0092	-0.0832
STAR.G	1.5470	0.0090	0.0001	-0.0015
SUNSET.G	1.1826	0.0807	-0.0016	0.0146
TOWN.G	1.5789	1.1119	-0.0050	0.0509
WAVE.G	1.0890	-3.2661	-0.0110	0.1019
ZEBRA.G	1.7704	3.4888	0.0009	0.0101

3.5 Image Set

SET 1 : Original Images


ARMY1.JPG

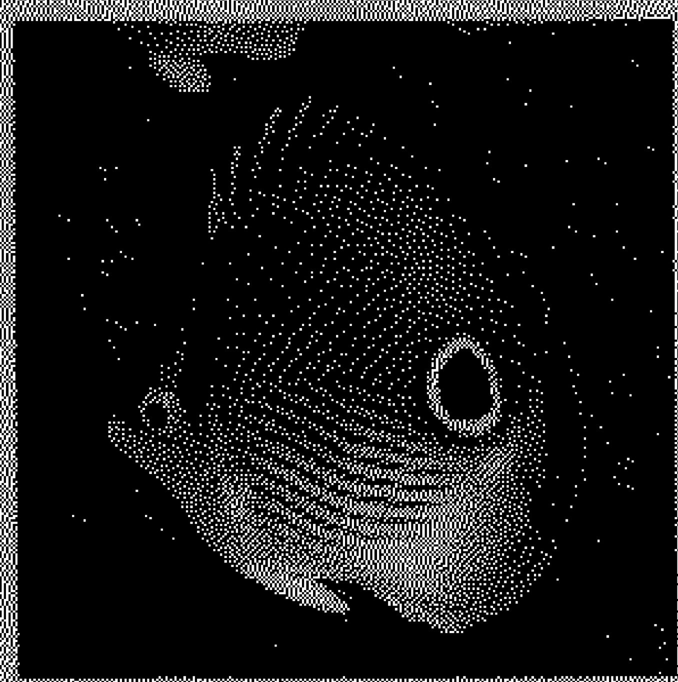

BLAGE1.JPG


CASTLE1.JPG


CATHED1.JPG


CHIMP1.JPG

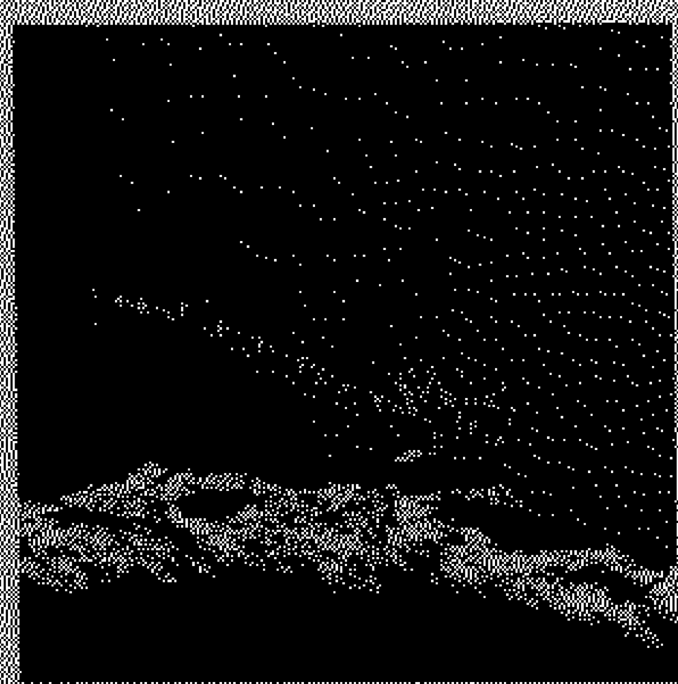

CHOPER1.JPG


FISH1.JPG


GOLDFISH1.JPG


HAWK1.JPG

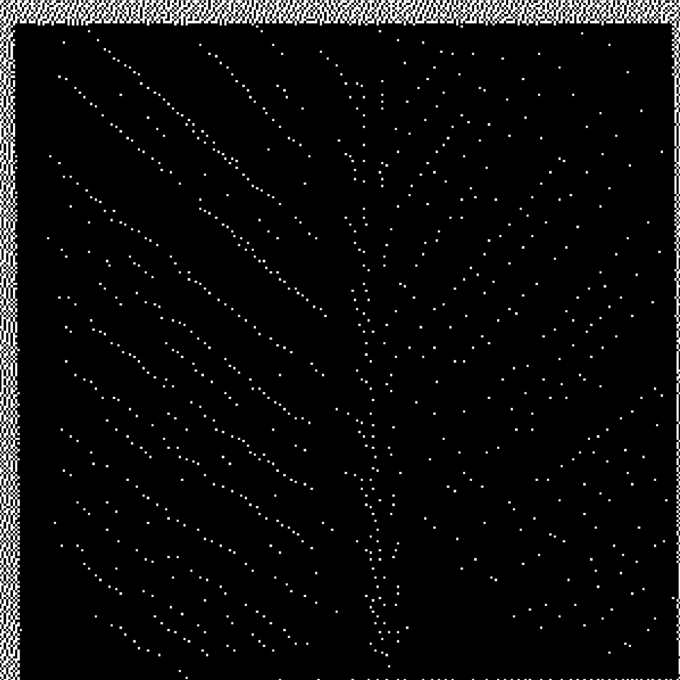

ICE1.JPG


INSECT1.JPG


KID11.JPG


KID21.JPG


KID31.JPG


LEAF1.JPG

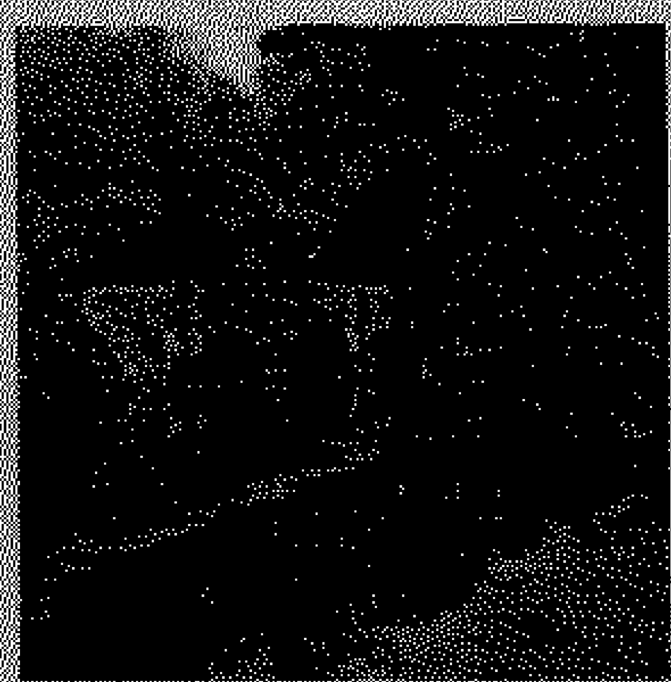
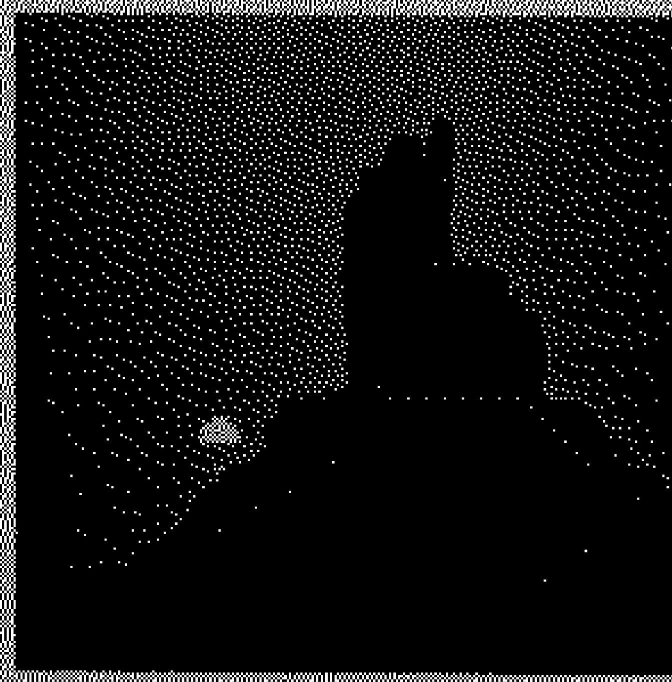

NEWENG1.JPG

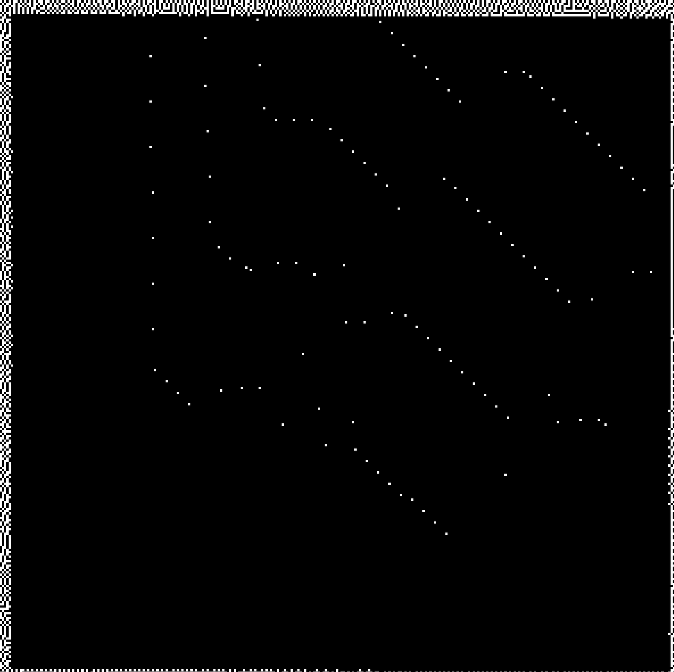
PHOTO1.JPG

RGB31.JPG

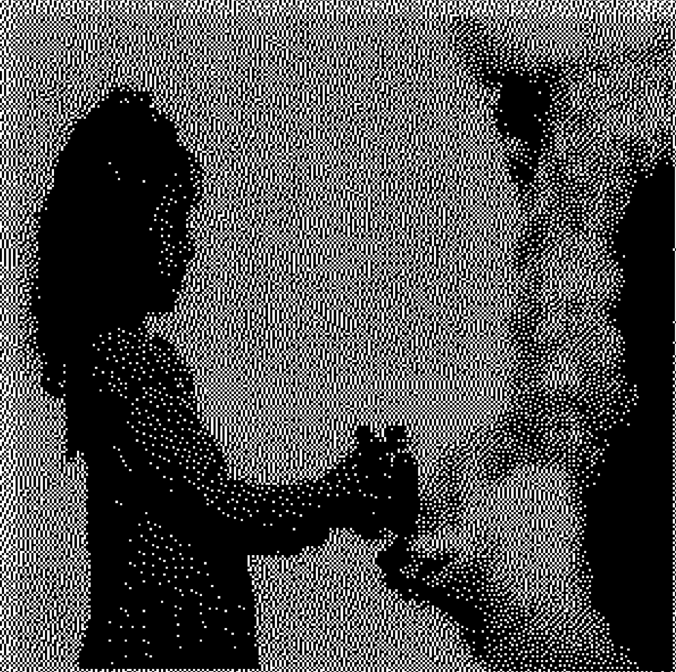


ROCK1.JPG

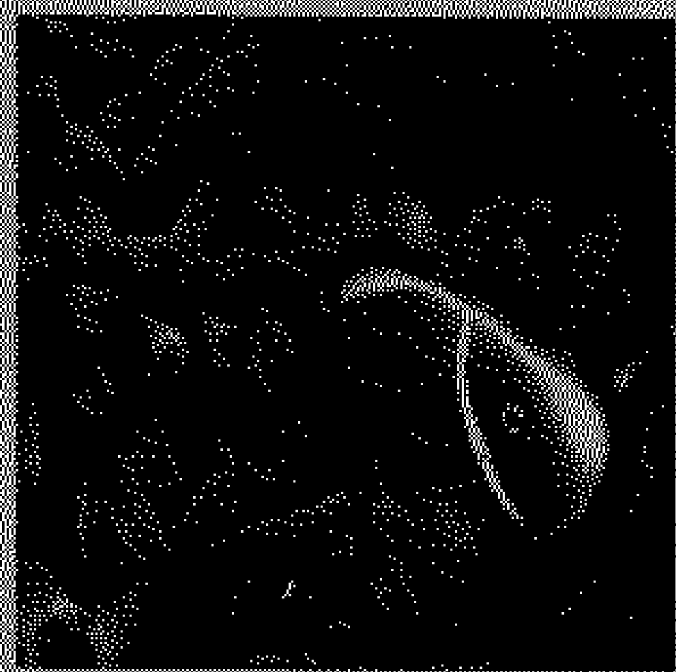
RODEO1.JPG



ROSE1.JPG

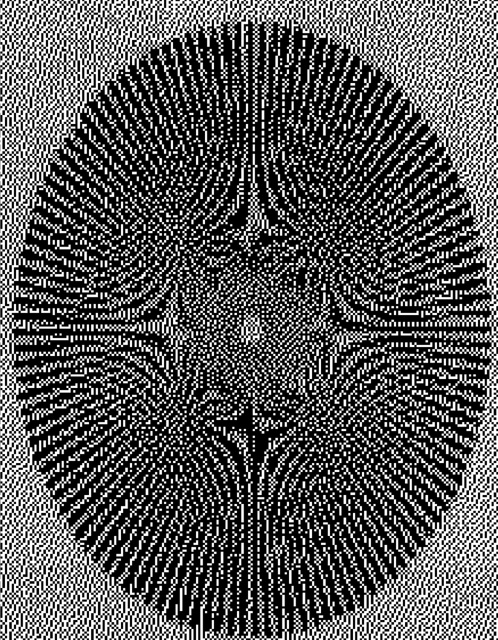


SANTA1.JPG



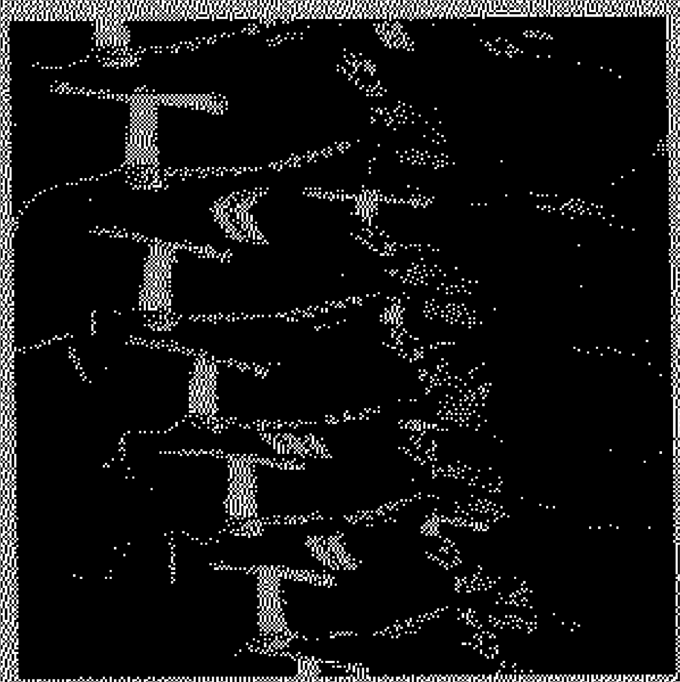
SEAFISH1.JPG

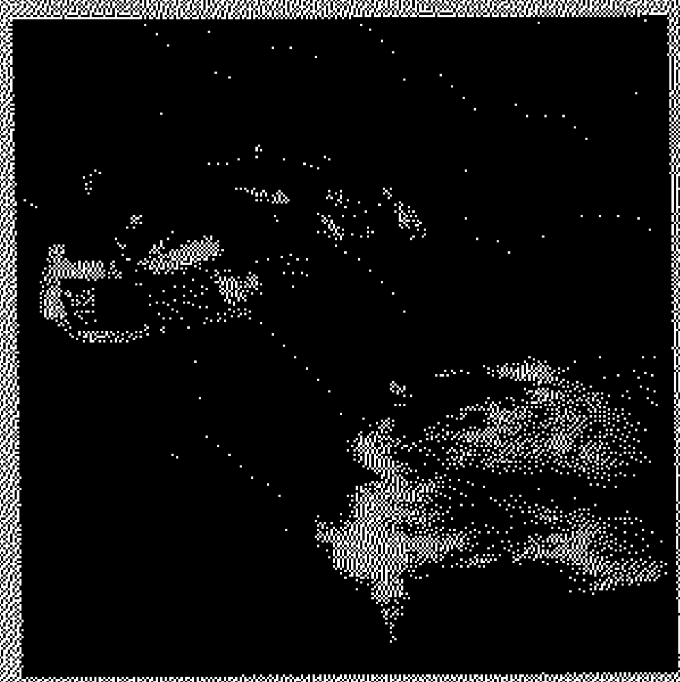
SOLDER1.JPG

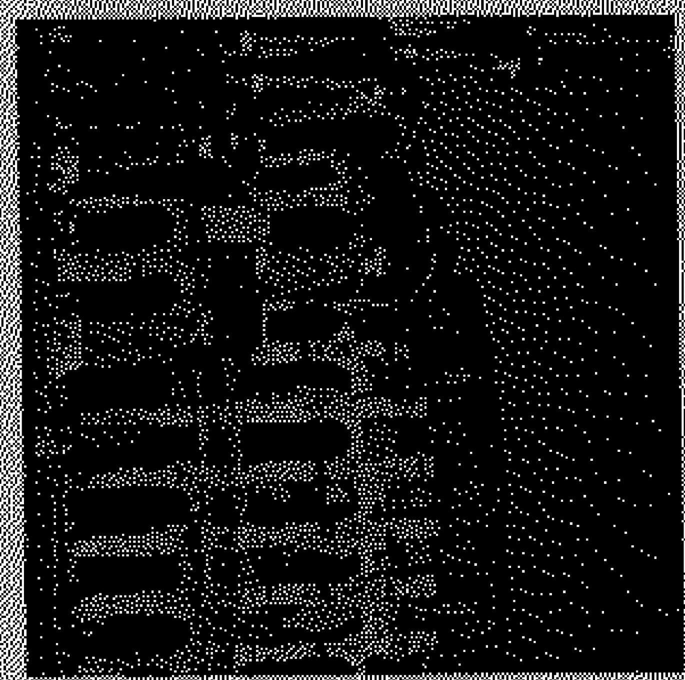


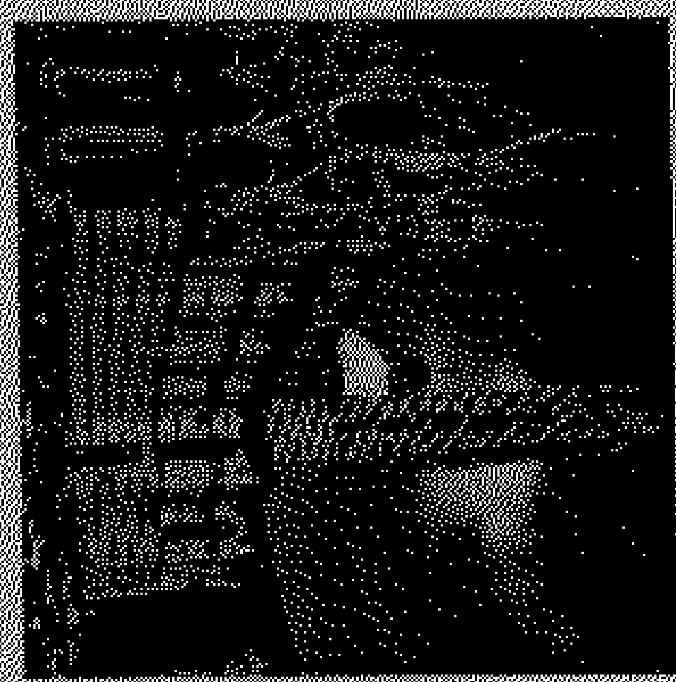
STAR1.JPG

SET 2 : Images with 90° rotation (clockwise) of actual images

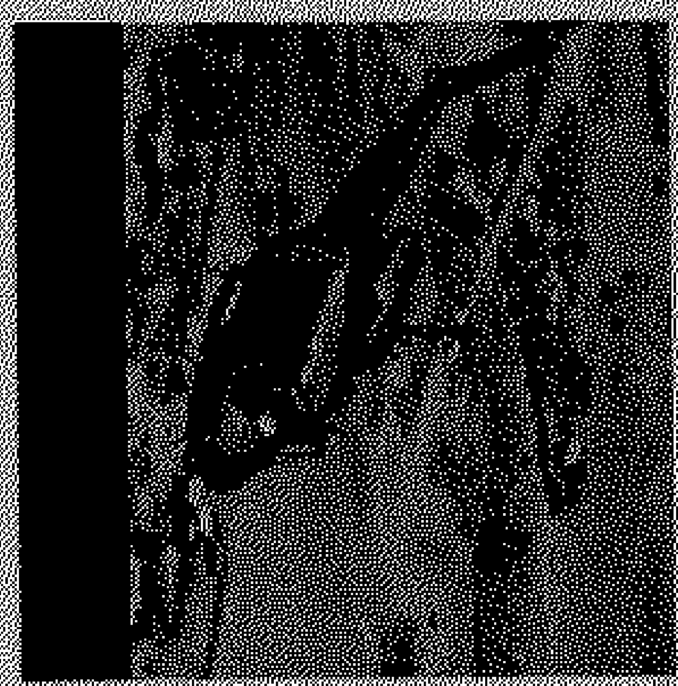

ARMY2.JPG

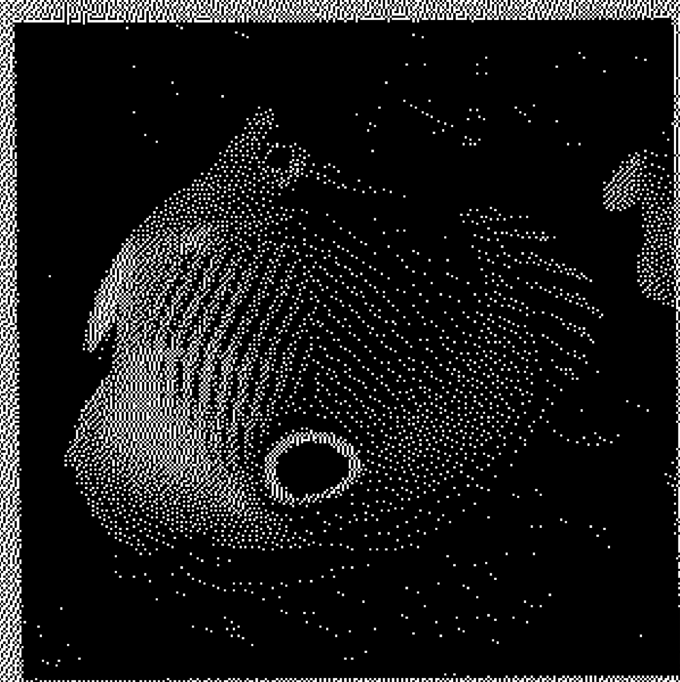

BLAGE2.JPG


CASTLE2.JPG


CATHED2.JPG


CHIMP2.JPG

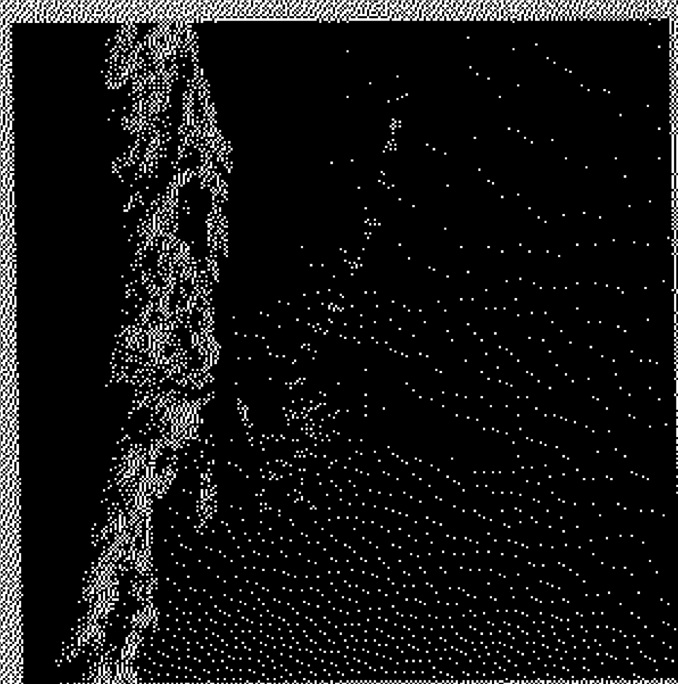

CHOPER2.JPG

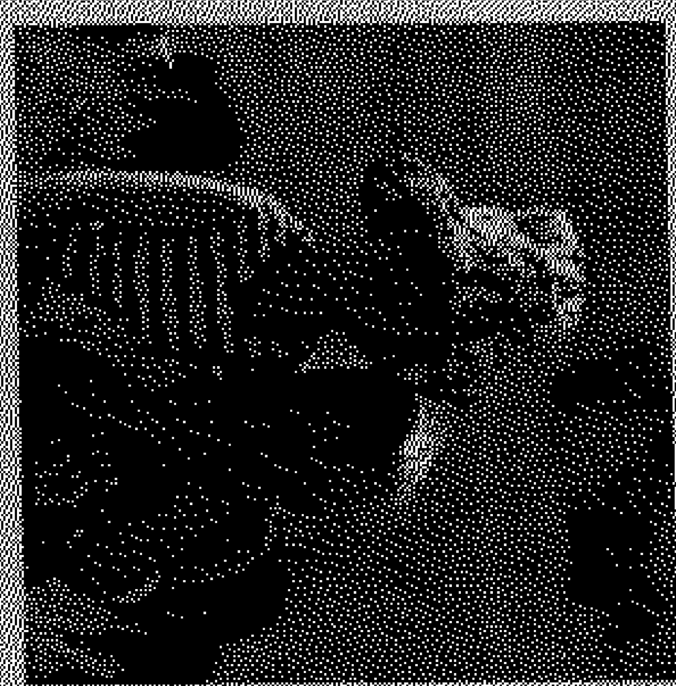

FISH2.JPG


GOLDFISH2.JPG


HAWK2.JPG

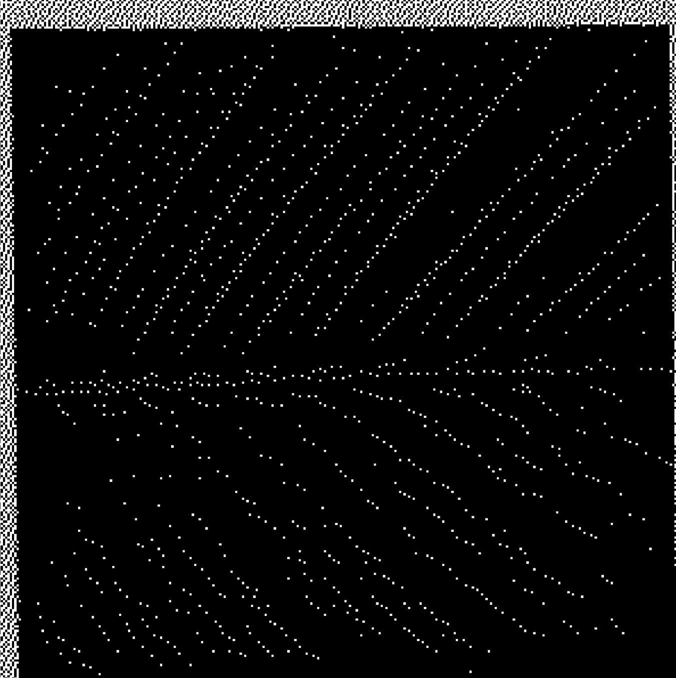

ICE2.JPG


INSECT2.JPG


KID12.JPG


KID22.JPG


KID32.JPG


LEAF2.JPG


NEWENG2.JPG

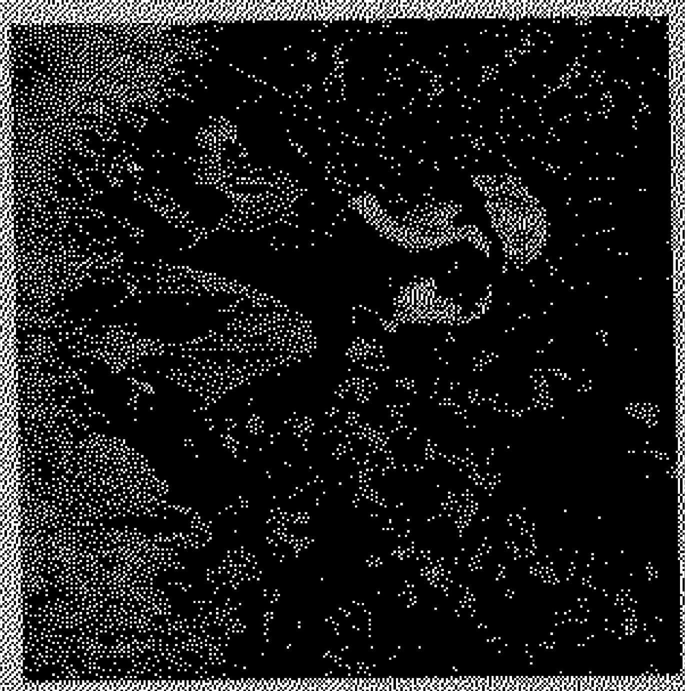
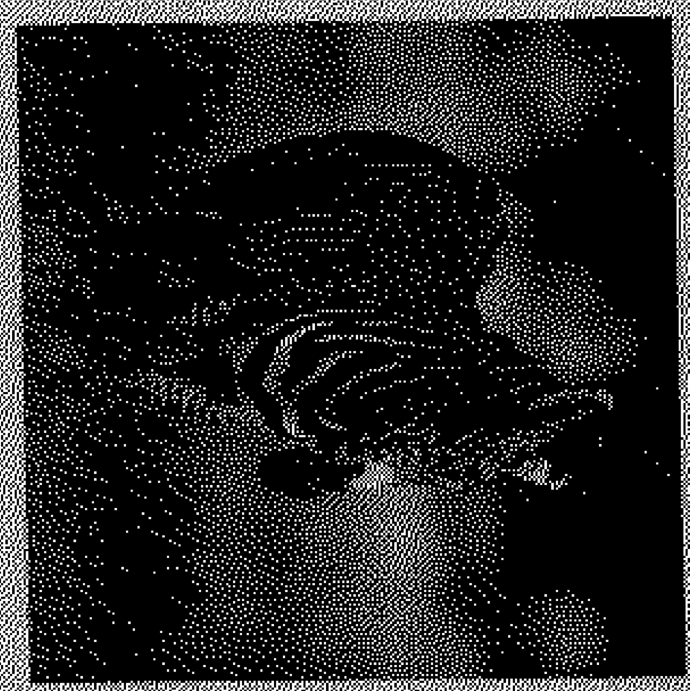
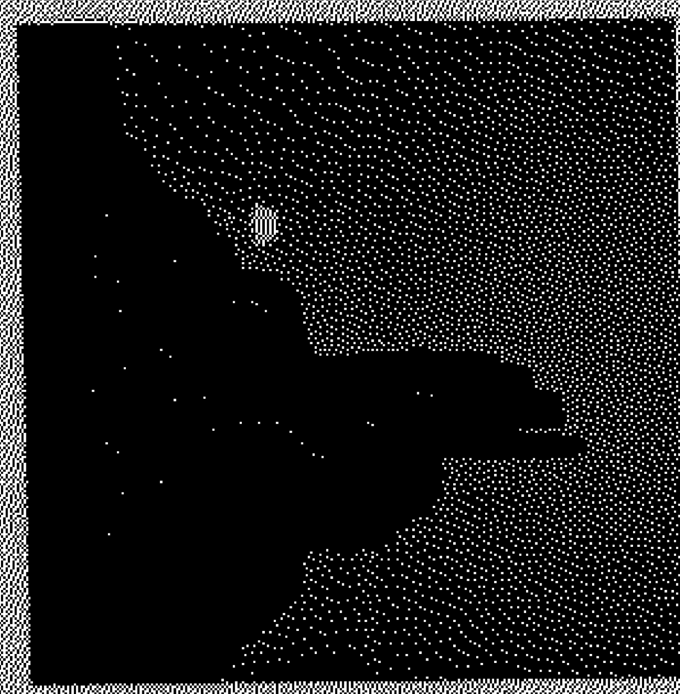


PHOTO2.JPG

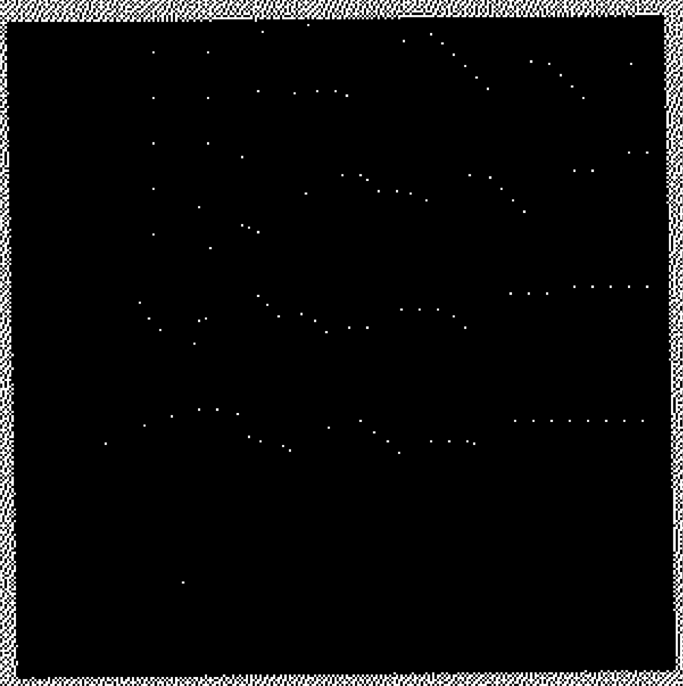


RGB32.JPG



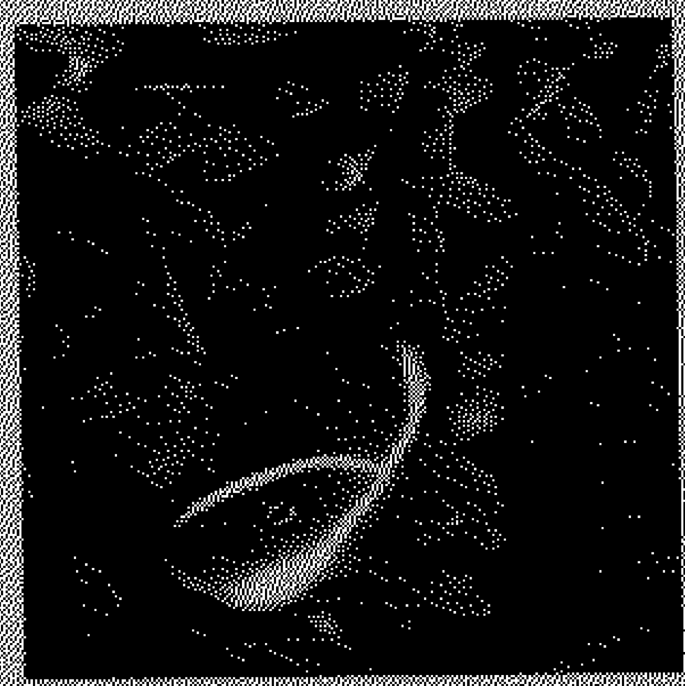
ROCK2.JPG

RODEO2.JPG



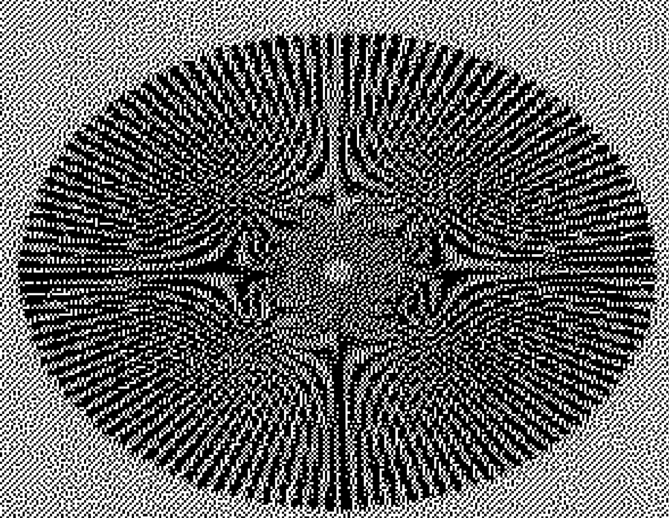
ROSE2.JPG

SANTA2.JPG



SEAFISH2.JPG

SOLDER2.JPG



STAR2.JPG

3.6 Conclusion

From the results we can say that the method of moments provides us with an important tool by which we can recognize images by assigning each image a corresponding signature value. The results obtained may be improved by using methods to extract the Region of Interest, and computing moments only on the region of interest. This would remove unnecessary background information and would give better results.

After assigning each image a corresponding signature value we can use this signature as a key for Content Based Image Retrieval. The method of moments described above may appear to be computationally intensive but it can be implemented parallely. Methods of parallel implementation of moments are dealt with in the next chapter.

Chapter 4

MOMENT COMPUTATION

4.1 Introduction

The measure used in computation of texture features of an image includes moments. In this section, a method of evaluation of moments, keeping parallel implementation in mind is suggested. The application of parallel implementation would reduce the time complexity and hence result in faster evaluation.

For parallel implementation we take into consideration Method of Projections. By this method we compute up to third order moments.

4.2 Mathematical Foundation

We describe a method of finding the ordinary moments, and then derive central order moments from them. Parallel algorithms have been developed for computing ordinary moments from which the central order moments can be calculated.

Ordinary moments are given by :

$$m_{pq} = \sum\sum x^p y^q f(x,y)$$

Here x and y denote the x and y co-ordinates of the image pixel and $f(x, y)$ is a function such that it is piecewise continuous and has nonzero values only in a finite part of the x y plane.

Central order moments are given by

$$\mu_{pq} = \sum\sum (x - x\_avg)^p (y - y\_avg)^q f(x,y)$$

where $x\_avg = m_{10}/m_{00}$

$y\_avg = m_{01}/m_{00}$

We compute moments up to third order using the proposed technique .The technique adopted here is based on the method of projections. Consider a line L through the origin which has an inclination of θ^{o} with the horizontal X axis . Consider another line M perpendicular to L at distance 'a' from the origin. Let the coordinate point be at a distance 'b' along the perpendicular line. Then x and y co-ordinates in terms of the new L and M coordinates can be represented as follows.

$x = a\cos\theta - b\sin\theta$

$y = a\sin\theta + b\cos\theta$

In this model we take projections along x axis (vertical projection),y axis (horizontal projection), and along the axis L (projection on the inclined line passing through the origin).

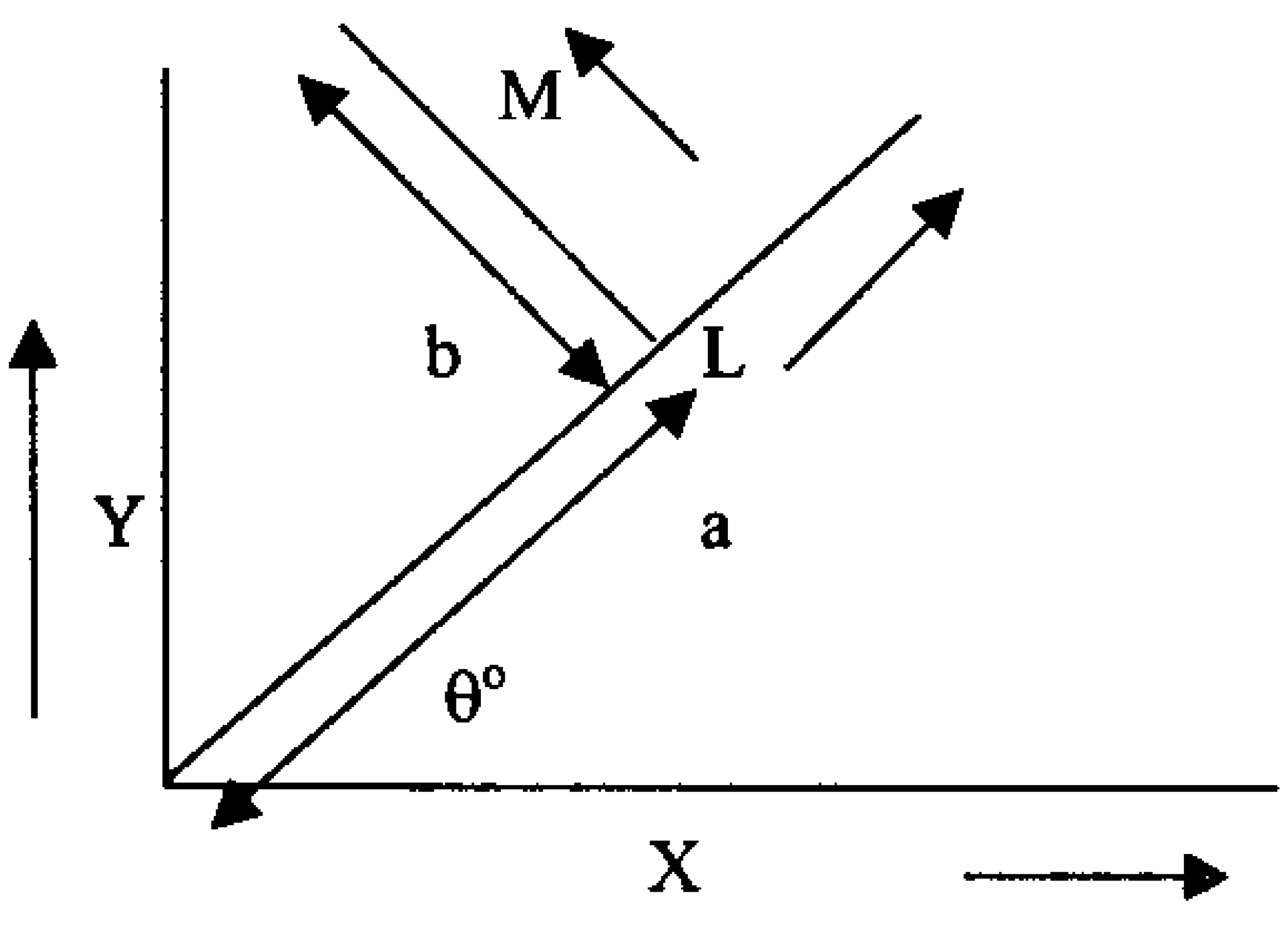


Figure 4.1

Method of Projection

Now the integral of $f(x,y)$ along the new line L gives the value of the inclined projection .Here integral is considered over the part of the line that includes the projection of the image.

$$P(t) = \int_L f(a\cos\theta - b\sin\theta, a\sin\theta + b\cos\theta)$$

Similarly the integral over the x and y axes will yield horizontal and vertical projections. They are given by

$$v(x) = \int_X f(x,y)dy$$

$$h(y) = \int_Y f(x,y)dx$$

Now we have ,

$$A = \int f(x,y)dx\,dy$$

$$= \int v(x)dx = \int h(y)\,dy$$

From the above relation we can obtain zero[th] order moment m_{00}.

Further we have

$$x\_avg\ A = \int\int x\, f(x,y)\,dx\,dy = \int x\, v(x)\,dx$$

$$y\_avg\ A = \int\int y\, f(x,y)\,dx\,dy = \int y\, h(y)\,dy$$

Here we obtain first order moments m_{10} and m_{01} .

For computing second and third order moments, we use the same technique.

$$\int\int x^2\ f(x,y)\,dx\,dy = \int x^2\ v(x)\,dx$$

$$\int\int x^3\ f(x,y)\,dx\,dy = \int x^3\ v(x)\,dx$$

$\int\int y^2 \; f(x,y)\,dx\,dy \; = \; \int y^2 \; h(y)\,dy$

$\int\int y^3 \, f(x,y)\,dx\,dy \; = \; \int y^3 \; h(y)\,dy$

From the above relations , we can obtain m_{20}, m_{02} , m_{30} and m_{03}.

For evaluating the integrals of x y , x^2 y and x y^2 we have to take two additional projections along with the two projections introduced so far(vertical and horizontal projections).These new projections are taken along two diagonals M and L which have a slope of 1 and –1 respectively. Line M passes through the origin and line L through the point (N,0)(here we are considering the values of the function f(x,y) to be the entries of a N X N matrix.)

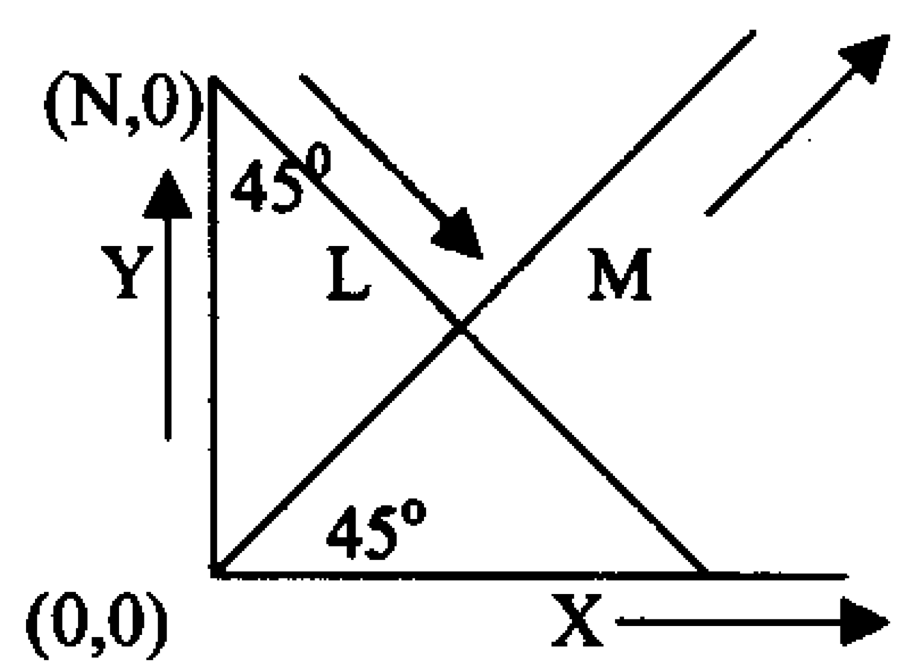


Figure 4.2
Projection along the diagonals

Considering projections along the lines M and L we find that the transformed co-ordinates (a,b) in terms of the original co-ordinates as in figure 1. Here a is the distance along the line of projection and b is the perpendicular distance from the line of projection.

For '+1' slope

x = (a – b)/sqrt(2)
y = (a+b)/sqrt(2)

Then
$a = (x+y)/sqrt(2)$

For a '-1' slope

$x = (a-b)/sqrt(2)$
$y = N- (a+b)/sqrt(2)$

Then
$(x-y) = sqrt(2)\ a - N$

Considering projection along the line M we have

$f(x,y) = f\{(a-b)/sqrt(2),(a+b)/sqrt(2)\}$

and $d_1(a) = \int f\{(a-b)/sqrt(2),(a+b)/sqrt(2)\}\ db$

$\int\int (x + y)^2 f(x , y)\ dx\ dy = \int\int 2\, a^2\ f(x,y)\ da\ db = \int 2\ a^2\ d_1(a)\ da$

Hence

$\int\int x\ y\ f(x,y)\ dx\ dy = \int a^2\ d_1(a)\ da - ½ \int x^2\ v(x)\ dx - ½ \int y^2\ h(y)\ dy$

From here we can obtain the value of m_{11}.

$\int\int (x + y)^3 f(x, y)\ dx\ dy = \int\int 2\ sqrt(2)\ a^3\ f(x,y)\ da\ db$

$= \int 2\ sqrt(2)\ a^3\ d_1(a)\ da$

Then

$\int\int 3(x\ y^2 + x^2\ y)\ f(x,y)\ dx\ dy = \int 2\ sqrt(2) a^3\ d_1(a)\ da - \int x^3\ v(x)\ dx - \int y^3\ h(y) dy$
$= first\_value(say)$

Taking projection along L we find

$f(x,y) = f\{ (a-b)/sqrt(2) , (N-(a+b)) / sqrt(2) \}$

and $d_2(a) = \int f\{ (a-b)/sqrt(2) , (N-(a+b)) / sqrt(2) \} \; db$

$$\iint (x - y)^3 f(x , y) dx\, dy = \iint \{2\, sqrt(2)a^3 - 6a^2N + 3\, sqrt(2)aN^2 - N^3\} f(x,y)\, da\, db$$
$$= \int \{2sqrt(2)a^3 - 6a^2N + 3\, sqrt(2)aN^2 - N^3\}\, d_2(a)\, da$$
$$= \text{second\_value (say)}$$

Then

$$\iint 3(x y^2 - x^2 y) f(x,y)\, dx\, dy = \text{second\_value} - \int x^3 v(x)\, dx + \int y^3 h(y)dy$$

$$= \text{third\_value (say)}$$

Therefore

$$\iint 6 x^2 y f(x,y)\, dx\, dy = \text{first\_value} - \text{third\_value}$$

and

$$\iint 6 x y^2 f(x,y)\, dx\, dy = \text{first\_value} + \text{third\_value}$$

From here we can obtain the moments m_{11} ,m_{12} and m_{21}.

Since we are dealing with digital images, we replace the integral signs by summations. On proper replacements the equations for moment computation evaluate to

$v(x) = \sum f(x,y)$
along y axis

$h(y) = \sum f(x,y)$
along x axis

$d_1(a) = \sum f((a-b)/sqrt(2) , (a+b)/sqrt(2))$ and
$d_2(a) = \sum f\{(a-b)/sqrt(2) , (N-(a+b))/sqrt(2)\}$

$m_{00} = \Sigma\, v(x) = \Sigma\, h(y)$

$m_{10} = \Sigma\; x\, v(x)$

$m_{01} = \Sigma\; y\, h(y)$

$m_{20} = \Sigma\; x^2\; v(x)$

$m_{02} = \Sigma\; y^2\; h(y)$

$m_{30} = \Sigma\; x^3\, v(x)$

$m_{03} = \Sigma\; y^3\, h(y)$

$m_{11} = \Sigma\; a^2\, d_1(a) - ½\, m_{20} - ½\, m_{02}$

$$m_{12} = {}^1/_3\ \{sqrt(2)\ \Sigma\, a^3\, d_1(a)\ + \Sigma\,(\, sqrt(2)a^3 - 3a^2N + 3aN^2 / sqrt(2) - N^3/2\,)\, d_2(a)\ - m_{30}\,\}$$

$$m_{21} = {}^1/_3\ \{sqrt(2)\ \Sigma\, a^3\, d_1(a)\ - \Sigma\,(\, sqrt(2)a^3 - 3a^2N + 3aN^2 / sqrt(2) - N^3/2\,)\, d_2(a)\ - m_{03}\,\}$$

We now describe a parallel algorithm to compute the above moments using the method of projections. For computing the horizontal and vertical moments, summation along the x and y axes would be required. For storing the summations, the corner pixels are used. For computing the projection on the diagonal , the pixels along the diagonal cannot hold all the data. So we take an approximation and consider a staircase diagonal to get rid of this problem.

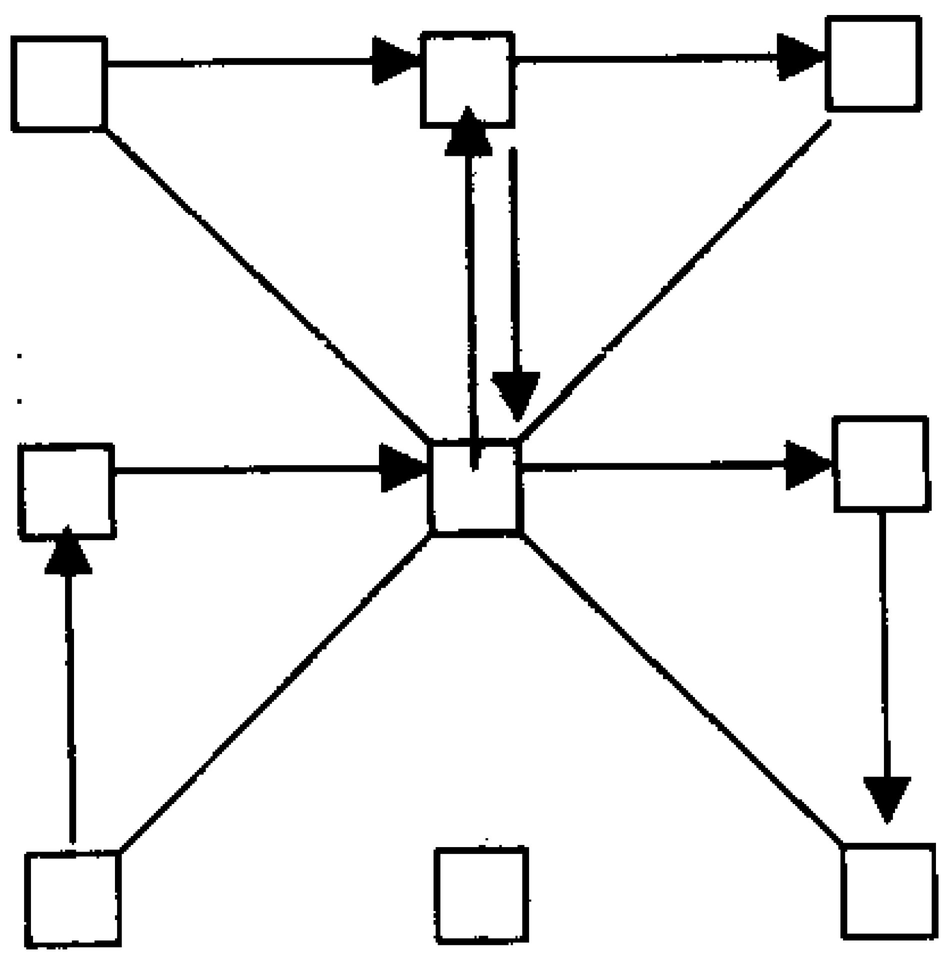


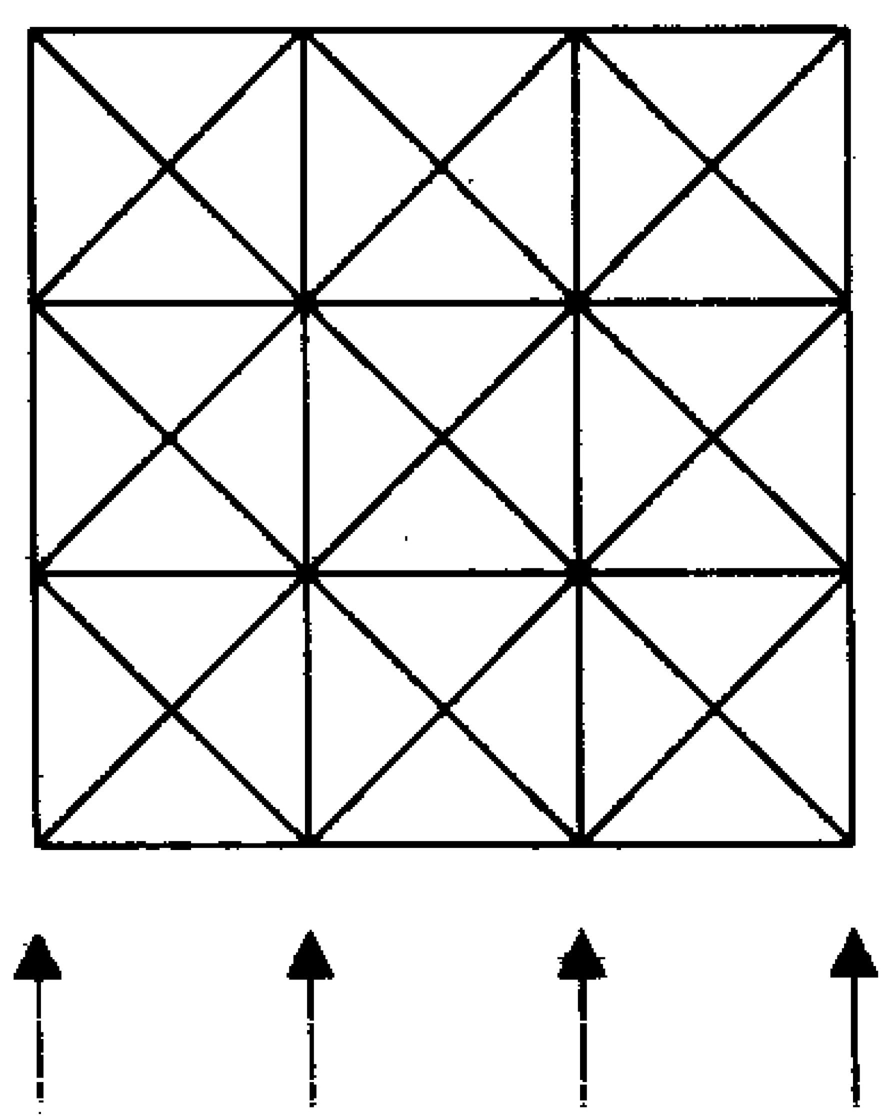
Figure 4.3 Staircase approximation of the diagonal.

4.3 Architecture

In this section we describe an architecture for computation of moments. Here we are considering a N X N matrix to represent the image . An element of the matrix is a function of the gray level intensity of the image at the corresponding point (pixel).

Each cell in this proposed architecture has a local memory and can perform simple arithmetic operations and data movement. The algorithm is capable of evaluating moments up to third order. The algorithm consists of three phases. In the first phase, it initialises the matrix elements and computes the vertical projection , next it computes the horizontal and diagonal projections, and finally it computes the various other moments .

For performing computation, each cell consists of five registers p v h d_1 and d_2. The cell stores its own data corresponding to a function of the gray level value at that pixel in p. For vertical data movement v is used, for horizontal h and for the diagonals, d_1 and d_2. Vertical projections are taken in the cells in the bottom boundary. Horizontal projections are taken in the left hand boundary . Diagonal projections are taken in the cells which are on the staircase diagonal. The algorithm required for the above computation is described in the next section. Figure corresponding to this architecture is given below.



Pixel data are fed here

Figure 4.4

Architecture and cell organisation for computing moments by projection method.

4.4 Algorithm

```
/*

variable usage:

        k is used to denote the number of iterations
        i is used to denote the Y co-ordinate
        j is used to denote the X co-ordinate

*/
```

Phase 1 : **/* Initialisation and vertical projection computation*/**

```
/*
        Pixel data is fed from the bottom of the cell boundary to provide PEij
        with pij . After the end of this phase initialisation and computation of
        vertical projection is complete.

*/

begin
   for k=1 to N
      for i = 1 to N do in parallel
         for j = 1 to N do in parallel
              PEij receives pixel data from PEi,j+1 in p;
              if(data_received and i>1)
                  send received_data to PEi-1,j;
              if( i = N)
                  v = v + received_data;
         end for
      end for
   end for
end
```

horizontal and diagonal projection computation is computed in this phase.
*/

```
Phase 2:
    begin
        for k = 1 to N do
            for i = 1 to N do in parallel
                for j = 1 to N do in parallel
                    PE_i,j recieves data from PE_i,j+1  in h ;
                    if ( i + j <= N+1)
                       PE_i,j receives data from PE_i-1,j-1 in d_1;
                    else if ( i + j >=  N)
                       PE_i,j receives data from PE_i+1,j+1 in d_1;
                    if ( no data received from any neighbour )
                       PE_i,j stop operation;

                    if ( i - j >=0)
                       PE_i,j receives data from PE_i-1,j+1 in d_2;
                    else if ( i - j <  0)
                       PE_i,j receives data from PE_i + 1, j -1 in d_2;
                    if ( no data received from any neighbour )
                       PE_i,j stop operation;


                    if ( j != 1)
                       PE_i,j sends h to PE_i,j-1;
                    else
                        h = h + received_data from PE_i,2   ;
                    if ( (N – ( i + j)) < 0)
                        PE_i,j  sends d to PE_i+1,j+1;
                    else if ( (N – ( i + j)) > 1)
                        PE_i,j  sends d to PE_i-1,j-1;
                    else  /*  for cells on the first staircase diagonal */
                        d_1 = d_1 + received_data from PE_i-1,j-1 ;

                    if ( ( i - j)) < 0)
                        PE_i,j  sends d to PE_i+1,j+1;
                    else if (  i - j)) > 1)
                        PE_i,j  sends d to PE_i-1,j-1;
                    else  /*  for cells on  the second staircase diagonal */
```

```
                d2 = d2+ received_data from PE(i-1,j-1) ;

            endfor
        endfor
    endfor
end
```

Phase 3:

Step 1 through 3 do in parallel

Step 1: /* Compute m_{30}, m_{20}, m_{10} and m_{00} */

```
/*
        Data received from right neighbour:
                v3 : for computing m30
                v2 : for computing m20
                v1 : for computing m10
                v0 : for computing m00
*/

begin
        for k = N down to 1 do
          if ( k = =N)
                PE(N,k) sends v.N^3 to PE(N,k-1) ( v3 to PE(N,k-1) ) ;
                PE(N,k) sends v.N^2 to PE(N,k-1) ( v2 to PE(N,k-1) ) ;
                PE(N,k) sends v.N   to PE(N,k-1) ( v1 to PE(N,k-1) ) ;
                PE(N,k) sends v     to PE(N,k-1) ( v0 to PE(N,k-1) ) ;

          else if ( k = = 1 )
                PE(N,k) computes  m30 = v + v3 ;
                PE(N,k) computes  m20 = v + v2 ;
                PE(N,k) computes  m10 = v + v1 ;
                PE(N,k) computes  m00 = v + v0 ;
```

else

$PE_{N,k}$ sends $v.k^3 + v_3$ to $PE_{N,k-1}$;
$PE_{N,k}$ sends $v.k^2 + v_2$ to $PE_{N,k-1}$;
$PE_{N,k}$ sends $v.k + v_1$ to $PE_{N,k-1}$;
$PE_{N,k}$ sends $v + v_0$ to $PE_{N,k-1}$;

endfor

end

Step 2: /* compute m_{03}, m_{02} and m_{01} */

/*

Data received from top neighbour;

h_3 : for computing m_{03}
h_2 : for computing m_{02}
h_1 : for computing m_{01}

*/

begin

for k = 1 to N **do**

if (k = = 1)

$PE_{k,1}$ sends $h.N^3$ to $PE_{k+1,1}$ (h_3 to $PE_{k+1,1}$);
$PE_{k,1}$ sends $h.N^2$ to $PE_{k+1,1}$ (h_2 to $PE_{k+1,1}$);
$PE_{k,1}$ sends $h.N$ to $PE_{k+1,1}$ (h_1 to $PE_{k+1,1}$);

else if (N = = 1)

$PE_{k,1}$ computes $m_{03} = h + h_3$;
$PE_{k,1}$ computes $m_{02} = h + h_2$;
$PE_{k,1}$ computes $m_{01} = h + h_1$;

else

$PE_{k,1}$ sends $h.(N - k + 1)^3 + h_3$;
$PE_{k,1}$ sends $h.(N - k + 1)^2 + h_2$;
$PE_{k,1}$ sends $h.(N - k + 1) + h_1$;

endfor

end

```
Step 3:     /* compute m11, m21 and m12 */
/*
        ti,j is the distance of the cell PEi,j from PEN,1.=(N-i+j-1)/sqrt(2)

        u-q,r is the distance of the cell PE q,r from PE1,1. = (i +j-2)/sqrt(2)

        c1 and c2 are the data received by PEi,j from either PEi-1,j or PEi,j+1.

        c3 is the data received by PEq,r from either PEq,r-1 or PEq-1,r.

*/


begin
        i = 1;
        j = N;
        q =1;
        r = 1;
        count = 0;
        while ( count < 2N - 1) do
                if ( count != 2N - 1)
                        if ( (i + j -N)= =0)
                           PEi,j sends d1 t^2 +c1 to PE i,j-1 ;
                           PEi,j sends 2 sqrt(2) d1 t^3 + c2 to PE i,j-1 ;

                           j = j - 1 ;

                        else
                           PEi,j sends d1 t^2 + c1 to PE i+1,j ;
                           PEi,j sends (d1 t^3 )/2 sqrt(2) + c2 to PE i+1,j ;
                           PEi,j sends d u^3 + c3 to PE i,j-1 ;

                           i = i + 1 ;

                        fi
                        if( q= =r)
                          PEq,r sends d2 (2 sqrt(2)u^3 -6u^2 N +
                          3sqrt(2) u N^2 - N^3 ) + c3 to PE q,r+1 ;

                        r = r + 1;
```

else

$PE_{q,r}$ **sends** d_2 (2 sqrt(2) u^3 –6 u^2 N + 3sqrt(2) u N^2 – N^3) + c_3 to PE $_{q+1,r}$;

q = q + 1;

fi

else

$PE_{i,j}$ **computes** $m_{11} = d + c_1 - ½\, m_{20} - ½\, m_{02}$

$PE_{i,j}$ **computes** $m_1 = d_1 + c_2$

$PE_{q,r}$ **computes** $m_2 = d_2\ (\text{sqrt}(2)\ N - N)^3 + c_3$

Send data from $PE_{q,r}$ **to** $PE_{i,j}$

PE computes $m_{21} = m_1 - m_2 - 2\, m_{03}$

PE computes $m_{12} = m_1 + m_2 - 2\, m_{30}$

fi

count = count + 1

endwhile

end

4.5 Complexity Analysis

The proposed algorithm consists of three sequential phases. The first, which involves loading of data and computation of vertical projection requires *O(N)* time. The second phase which computes horizontal and diagonal projections also requires *O(N)* time. This is due to the fact that in both these phases iteration is carried out up to N steps. The third phase also requires *O(N)* time (2N – 1 to be precise). So time complexity of the above algorithm is *O(N)*. The number of processors used here is $O(N^2)$. So the AT measure turns out to be $O(N^3)$.

4.6 Conclusion

Here we have described and implemented an algorithm capable of finding moments up to third order for gray level images. The only assumption made here is that the function *f*(x,y) is computed before hand and fed to the different pixels. It seems that moments of higher order can be obtained by this method by changing the architecture used.

Bibliography

[1] Gonzalez.R.C , Woods.R.E, " Digital Image Processing" , *"Addision-Wessely"*.

[2] Haralick .R.M , Shanmugam.K , Dinstein.I , "Texture Features for Image Classification" , *" IEEE Transactions on Man And Cybernatics"*. Vol 6. November 1973.pp. 610-621.

[3] Haralick .R.M " Statistical Image Texture Analysis" , *" Handbook of Pattern Recognition and Image Processing"*.

[4] Horn .B.K.P " Robot Vision" , *"MIT Electrical Engineering and Computer Science Series"*.

[5] Pala.P , Santini.S , " Image Retrieval by Shape and Texture" , *" Pattern Recognition 32(1999) 517 – 527."*

[6] Trier.O.D , Jain.A.K , Taxt.T , " Feature Extraction Methods For Character Recognition – A Survey, *" Pattern Recognition Vol 29 pp 641 – 662, 1996"*.

[7] Tuceryan.M , Jain.A.K , " Texture Analysis" , *" Handbook of Computer Vision and Pattern Recognition."*

[8] Tuceryan.M , Jain.A.K , " Texture Segmentation using Voronoi Polygon" , *" IEEE Transactions on Pattern Recognition and Machine Intelligence"*.

[9] Johnson , " Human Visual System", *" MIT Press"*.