M.Tech. (Computer Science) Dissertation Series

Performance Analysis Of Area Morphology Operator In Color Image Processing

A dissertation submitted in partial fulfillment of the requirements for the Master of Technology (Computer Science) degree of the Indian Statistical Institute

By

Debapriyay Mukhopadhyay

Under the supervisions of

Prof. Bhabatosh Chanda

And

Dr. Dipti Prasad Mukherjee

INDIAN STATISTICAL INSTITUTE

203, Barrackpore Trunk Road

Kolkata – 700 108

2002

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled "***Performance Analysis Of Area Morphology Operator In Color Image Processing*** " is an authentic record of the dissertation carried out by **Debapriyay Mukhopadhyay** under my supervision and guidance. The work fulfils the requirements for the award of the M.Tech. degree in Computer Science.

Dated : The 19th of July, 2002.

B. Chanda

Signed :

(Prof. B. Chanda)

Supervisor

P. K. Nandi 19/7/02

Countersigned :

External Examiner

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and indebtedness to my guides **Dr. Dipti Prasad Mukherjee** and **Prof. Bhabatosh Chanda** of Electronics and Communications Science Unit of **Indian Statistical Institute, Calcutta** for their generous and whole hearted support to me in completing this project.

I wish to express my sincere respect and thanks to **Mr. Partha Mohanty** of Electronics and Communications Science Unit for his kind help and cooperation that he extended to me throughout the tenure of my project.

I also wish to place on record my sincere thanks to **Mr. Nabin Jana**, Research Scholar of Statistics and Mathematics Unit for his huge help in making my project thesis colorful and attractive.

Lastly, I would like to express my profound gratitude to all other teachers of Indian Statistical institute, Calcutta and sincere thanks to all of my classmates who were the source of inspiration during the M.Tech. programme.

Debapriyay Mukhopadhyay

Date: 13/07/2002 **Debapriyay Mukhopadhyay**

Contents

Chapter 1 : Overview of the Problem

- Introduction
- Definitions and Concepts
- Objective of the Study

Introduction

The field of mathematical morphology contributes a wide range of operators to image processing, all based around a few simple mathematical concepts from set theory. The operators are particularly useful for the analysis of images and common usages include edge detection, noise removal, image enhancement and image segmentation.

The concept of digital morphology is based on the fact that images consist of set of picture elements called pixels that are collected into groups having a two-dimensional structure called shape. A group of mathematical operators can be applied to the set of pixels to enhance or highlight specific aspects of the shape so that they can be recognized.

Erosion or shrinking of set of pixels having a given pattern (structuring element) and dilation or expanding of a given pattern to, are basic morphology operators. Virtually, all other mathematical morphology operators can be defined [10] in terms of combinations of erosion and dilation along with set operators such as intersection and union. Some of the more important are opening (application of erosion immediately followed by a dilation using the same structuring element) and closing (application of dilation immediately followed by a erosion using the same structuring element).

The idea underneath the mathematical morphology operators is extended in [2] to give rise area morphology operators – viz, area opening and area closing. The composition of area opening with it's dual i.e. area closing, is defined as area open_close. It is a filter by reconstruction which is size dependent in the sense that it removes image components that are smaller in area than a given limit. The study reported in [1] revealed that filters by reconstruction belong to a larger class called connected operators that have the fundamental property that these operators do not remove some frequency components like linear filters or some shapes like median filters or standard morphological opening and closing. Now-a-days, they are becoming very popular because they have been claimed to simplify the image while preserving contours. This rather surprising property makes them very attractive for a large number of applications such as noise cancellation, smoothing and segmentation.

The present study is circled around one such filter by reconstruction, viz area open_close and a systematic (experiment based) approach has been taken to establish the benefits of such a spatial filtering algorithm over others with respect to parameters like noise cancellation, smoothing and segmentation.

Definitions and Concepts

1. Area Operators:- For a set S defined on domain $\Omega \subset Z^2$, we have members of the onset : $(x,y) \in S$ and members of the offset : $(x,y) \in S^c$, where S^c is the complement of S.

Two points (x_1,y_1) and (x_2,y_2) are members of the same connected component S_i of a set S if both are members of the set, and there exists a connected path between the two points that only includes members of the set.

For the onset S, the area open operation is denoted by S o A and removes all connected components S_i with the area (cardinality) less than A . Area close is the complementary operation : S o^c A removes all connected components of area less than A in the offset S^c. So, area A is the parameter of area morphology, similar to the structuring element size in standard morphology.

For images, the area open and close operators are implemented via stacking. In a threshold decomposition of the image **I**, an associated level set L(**I** , t) is a set obtained by thresholding the image intensity: (x , y)$\in$ L(**I** , t) if I(x, y) $\geq$ t. For a discrete domain of k intensities j $\in\{0,1,\ldots,k-1\}$,

$I(x, y) = \Sigma\ 1_{(x, y) \in L(I, j)}$, where $1_{[\,]}$ is the set indicator function and summation is taken over j where j ranges from 0 to k-1. The stacking function allows **I** to be defined by,

$$I(x, y) = \max [\, t : (x, y) \in L(\mathbf{I}, t)\,] \qquad \text{-------------} (1)$$

To implement the area open and area close operators, each level set can be processed independently. Then the resulting image can be reconstructed by stacking using (1).

Within an area open operation on an image **I** o A or area close operation on an image **I** o^c A, the order of level sets processed does not affect the final result. An area open operator on an image will remove all connected components within the level sets L(**I** , t) of **I** that do not have a minimum area of A. Similarly , the area close operator will remove connected components of the complemented (offset) level sets L^c(**I** , t) of I that do not possess the minimum area. In this way, area open " flattens " small bright objects and area close flattens small dark objects in an image.

The concatenation of the are open and close operators leads to area open_close (AOC) I oA o^c A. This operator controls the scale of both positive going bright objects and negative going dark objects. AOC is a connected operator that with either remove or preserves connected components within the level sets L (**I** , t) and the complemented level sets L^c(**I** , t). In this way, two points within the same connected component are treated equally in scale space generation. This property is important in image segmentation.

Connected regions of constant intensity are called flat zones. The AOC operator increases the area of these flat zones in the image, while reducing the total number of flat zones as reported in [2]. Thus, the area operators increase the region homogeneity of the image **I** as the area A is increased.

2. Noise Reduction:- Image enhancement through noise reduction is a fundamental problem in image processing . The goal of "cleaning up" images so that they look better to most people is admittedly subjective. Nevertheless, images that most people regard as " clean " possess two common characteristics,

1) Edges , thin lines , and small features are sharp and clean, and
2) Areas between these features are smoothly varying.

Edges , thin lines , and small features are less affected perceptually by noise , because these features are effectively one dimensional whereas noise is a two dimensional feature as it require area to be perceptually significant.

Many techniques for noise reduction replace each pixel with some function of the pixel's neighbourhood. Linear filters (like mean filters) tend either to amplify the noise along with the 1D features or to smooth out the noise and blur the 1D features.

Morphological filters are, the most well known non-linear filters for image enhancement. These include erosions, dilations, opening, closings and order statistics filters. The action of a morphological filter depends on it's structuring element. The median filters is very good at removing some types of noise(salt and pepper), while preserving some edges.

Other morphological filters include , "alternating sequential filter (ASF)" (application of openings and closings with successively larger structuring elements) by Serra , and "Generalized Morphological Filters"(linear combinations of the results of opening and closing with multiple structuring elements) by Song and Delp. Lastly we have filters by reconstruction such as area open_close which is also very good in removing noise while preserving edges.

3.Image Segmentation:- Image segmentation serves as the key of image analysis and pattern recognition. It's a process of dividing an image domian into different regions, such that each region is homogeneous.

Color of an image carry much more information than gray level. In many pattern recognition and computer vision applications, the additional information provided by color help the image analysis process and yield better result than approaches using only gray scale information. More research has focused on color image segmentation due to it's demanding need. At present , color image segmentation methods are mainly extended from monochrome segmentation approaches by being implemented in different color spaces. Gray level segmentation methods are directly applied to each component of a color space, then the results are combined to obtain the final segmentation result. Color image segmentation approaches that are in existence now can be mainly divided into the following categories :- statistical approaches, edge-detection approaches, region splitting and merging approaches, hierarchical approaches, methods based on human color perception and last of all fuzzy set-theory based approaches.Apart from these there are few morphology based approaches which are now gaining increas-

-ing popularity because of the fact that morphology based operations preserve contours while simplifying the image.

Objective of the Study

Filters by reconstruction are gaining increased attention because of the fact that they simplify image while preserving contours and this fact have made them attractive in many image application areas.

Present study is also meant to make firm foot in the same direction. We are concerned here in this study to measure and compare the performance of such a filter by reconstruction, called area open_close(AOC), with respect to some parameters like noise cancellation, image smoothing and image segmantation. It is note worthy to mention that images we are considering here are all color images. Objectives of the study include the following :-

1) To compare and analyse the performance of AOC over other noise cancellation procedures in different color spaces viz, RGB, HSV, HLS, YCbCr.
2) To analyse the performance of AOC in image smoothing in different color spaces.
3) To prove that AOC followed by Fuzzy C Means algorithm of clustering provides better classification than median filtering followed by FCM when compared with the original ground truth image.
4) We know that, the AOC operator increases the area of the flat zones in the image, while reducing the total number of flat zones. To develop an algorithm of color image segmentation extending this idea of area morphology operator.

At the end we would see that area open_close (AOC) has wide range of applications in different areas of image processing with a better degree of performance than others.

Chapter 2 : Noise Cancellation and Smoothing

- Introduction
- Basis of Measuring Performance
- Methodology
- Results and Discussions

Introduction

Image filtering is one of the central topic in the area of image processing. We are concerned here in this Chapter to measure the performance of AOC , with respect to parameters like noise removal and image smoothing.

We have in existence different color models. By color model we mean specification of a 3D co-ordinate system where each color is represented by a single point. When trying to measure the performance of AOC , we have taken into consideration different color models.

Performance will be measured on the basis of two parameters, viz,

1) noise removal, i.e., how much successful AOC is in removing noise from the noisy image, and

2) smoothing, i.e., how much successful AOC is in smoothing or creating flat zones in the image.

Basis of Measuring Performance

So far there is no objective way to measure the performance of image filtration. In this Chapter two naive, objective way have been taken as the basis of measuring performance and they are defined as follows.

1) **Positive square root of Mean Square Error (RMSE) :-** Positive square root of the mean square error (MSE) between the matrices corresponding to the original input image and the image after removing noise (smoothing) has been taken as one of the basis of measuring performance. Let us denote this positive square root of the mean square error by d.

2) **θ-Measure :-** Let, **I** be the original image and **I'** the image obtained after removing noise from the noisy image $\mathbf{I_n}$ which has been obtained by adding noise to the image **I** .

Each pixel of a colored image is consisted of a triplet , viz, (X ,Y ,Z) , where the 1st, 2nd, and 3rd components denote the values corresponding to the red, green and blue channel respectively. So, each pixel can be conceived as a vector in the RGB plane.

If we consider an image to be a rectangular matrix of size m x n, then (X_{ij}, Y_{ij}, Z_{ij}) denote the pixel at the (i , j) th location. Now, let (X_{ij}, Y_{ij}, Z_{ij}) and $(X_{ij}', Y_{ij}', Z_{ij}')$ be the (i , j) th pixel values of the images **I** and **I'** respectively. Considering both of them as vectors in the RGB plane, we define , θ_{ij} as the angle between these two vectors, i.e.,

$$\theta_{ij} = \frac{X_{ij}.X_{ij}' + Y_{ij}.Y_{ij}' + Z_{ij}.Z_{ij}'}{|(X_{ij}, Y_{ij}, Z_{ij})| \,.\, |(X_{ij}', Y_{ij}', Z_{ij}')|} \qquad \text{----------------------- (1)}$$

Then, the θ-measure between the two images **I** and **I** ' is defined as,

$$\theta = \Sigma \, \Sigma \, \theta_{ij} \qquad \text{--------------} \quad (2)$$

, where summation is taken over i and j where i ranges from 1 to m and j ranges from 1 to n .

The image gets corrupted as we add noise to the image. The performance of a noise filtering procedure can be perceptually conceived by the similarity between original input image and the resulting image after noise removal .

But, the question is how do we define the term 'similarity'. The term similarity suggests here that the deviation of the elements of the matrix corresponding to the resulting image (after noise removal) from that of the original image is significantly small. This deviation has been measured here by positive square root of mean square error (RMSE) and also by θ-value (as defined in (2)) between the matrices representing **I** and **I** ' respectively.

So, lower the value of d and θ, better is the performance of the corresponding filtering procedure in removing noise.

In this Chapter an attempt has also been made to simulate the performance of AOC in image smoothing in different color spaces. The same criterion is also used here , i.e., lower the value of d and θ, better is it's performance in smoothing the image.

Methodology

Our first objective is to compare the performance of AOC in removing noise with other noise cancellation procedures like mean filtering, median filtering and max-min filtering in different color spaces.

In our experiment, noise have been added to the original input image. 'Salt and pepper' noise of 2% density has been added separately on the red, green and blue channels of the input image and the resulting channels are then combined to produce the noisy color image.

Space conversion algorithm from RGB color space to the desired color space then has been applied on the matrix corresponding to the noisy image. Space specific channels (as for example, red, green, blue channels for RGB space; hue, saturation, value channels for HSV space; hue, luminance, saturation channels for HLS space; luminance, crominance (Cb and Cr) channels for YCbCr space) have been separated from the matrix representing the image in the new space. Filtering procedures are then applied on by one on all the channels resulting in new values for the channels.

As 'salt and pepper' noise of 2% density have been added on the image, so an area equals to 12 (the parameter value that is required to run the AOC algorithm) is sufficient to remove noise from the image. The most important to be mentioned here is that for all other filtering procedures a 5 x 5 mask has been used for convolution and the choice of this mask is in accordance with the parameter value (i.e., area = 12) that is required to run the AOC algorithm.

The new values for the channels are then accumulated yielding the matrix pertaining to the filtered image in the space. Appropriate space conversion algorithm is again applied on the above matrix representing the filtered image to bring back the filtered image in the RGB space. The positive square root of MSE and the □-value (as in (2)) between the matrices representing the filtered image and the original input image — both being in the RGB space, then measures the performance of the filtering procedure.

Results and Discussions

The study is solely based on the results obtained through the processing of three standard 'bmp' images. They are as follows.

fig 21

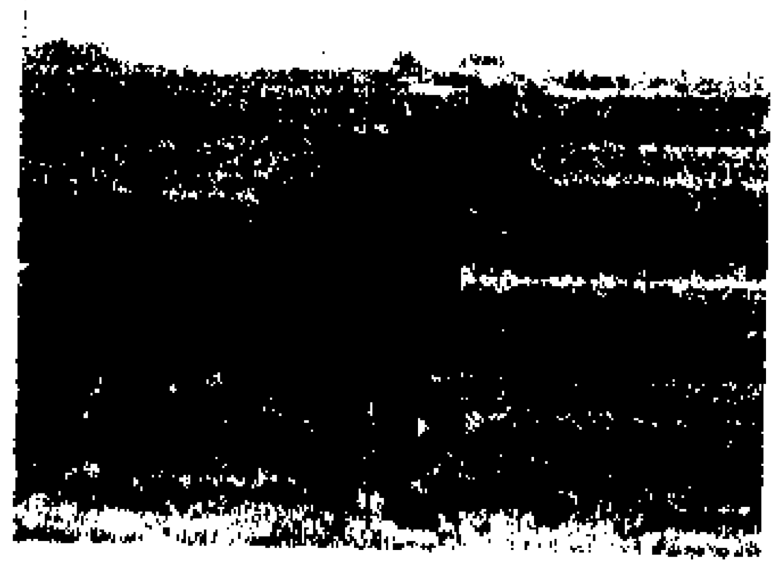


fig 2.2

fig 2.3

Table1 shows the performance of different noise filtering procedures on the given input images in the RGB space. From the table it is evident that the performance of AOC (in removing noise) is better than the other there filtering procedures for all the three images, as d-values and □-values corresponding to AOC are least.

Table1: Performance Table in RGB Space

Images	**Noisy**		**AOC**		**Median**		**Mean**		**Max-Min**	
	d	□	**d**	□	**d**	□	**d**	□	**d**	□
Fig1	36.72	332.7	13.86	144.9	33.56	293.5	38.77	592.3	57.75	715.3
Fig2	35.52	417.0	13.43	235.6	35.23	346.4	41.73	527.1	50.12	942.48
Fig3	40.02	190.6	19.09	181.2	41.77	188.7	59.75	329.2	65.47	608.65

Table 2, 3 and 4 associate themselves with the performance of filtering procedures in HSV, HLS and YCbCr color space respectively and as has been found from these tables performance of AOC is better than the others in all the above mentioned three spaces for all the three images, as d-values and θ-values corresponding to AOC are least. Tables also show that performance of median filtering though not better than AOC, but is better than Median Filtering and Max-Min Filtering in all the four color spaces in consideration.

Table2: Performance Table in HSV Space

Images	Noisy		AOC		Median		Mean		Max-Min	
	d	θ	d	θ	d	θ	d	θ	d	θ
Fig1	36.72	332.7	14.79	163.9	33.43	287.8	40.69	604.3	48.22	618.3
Fig2	35.52	417.0	13.51	146.1	34.55	268.1	42.18	603.1	45.00	599.33
Fig3	40.02	190.6	23.35	181.2	46.65	199.9	61.05	618.8	58.10	650.0

Table3: Performance Table in HLS Space

Images	Noisy		AOC		Median		Mean		Max-Min	
	d	θ	d	θ	d	θ	d	θ	d	θ
Fig1	36.72	332.7	23.78	171.6	33.94	299.0	41.65	787.6	61.14	607.8
Fig2	35.52	417.0	15.41	214.0	35.54	294.9	43.46	846.6	54.44	553.6
Fig3	40.02	190.6	25.48	198.8	49.19	238.2	78.22	853.9	70.40	575.7

Table4: Performance Table in YCbCr Space

Images	Noisy		AOC		Median		Mean		Max-Min	
	d	θ	d	θ	d	θ	d	θ	d	θ
Fig1	36.72	332.7	14.23	171.4	33.72	284.6	47.67	1259	63.02	1055
Fig2	35.52	417.0	14.23	197.1	35.10	286.2	49.17	1097	58.19	972.4
Fig3	40.02	190.6	22.61	191.5	42.10	221.3	78.61	1233	65.63	979.1

So at the end we conclude that with respect to noise removal AOC is superior than other noise removing procedures in all the four color spaces and in the RGB space performance is exceptionally good as can be seen from the Table1.

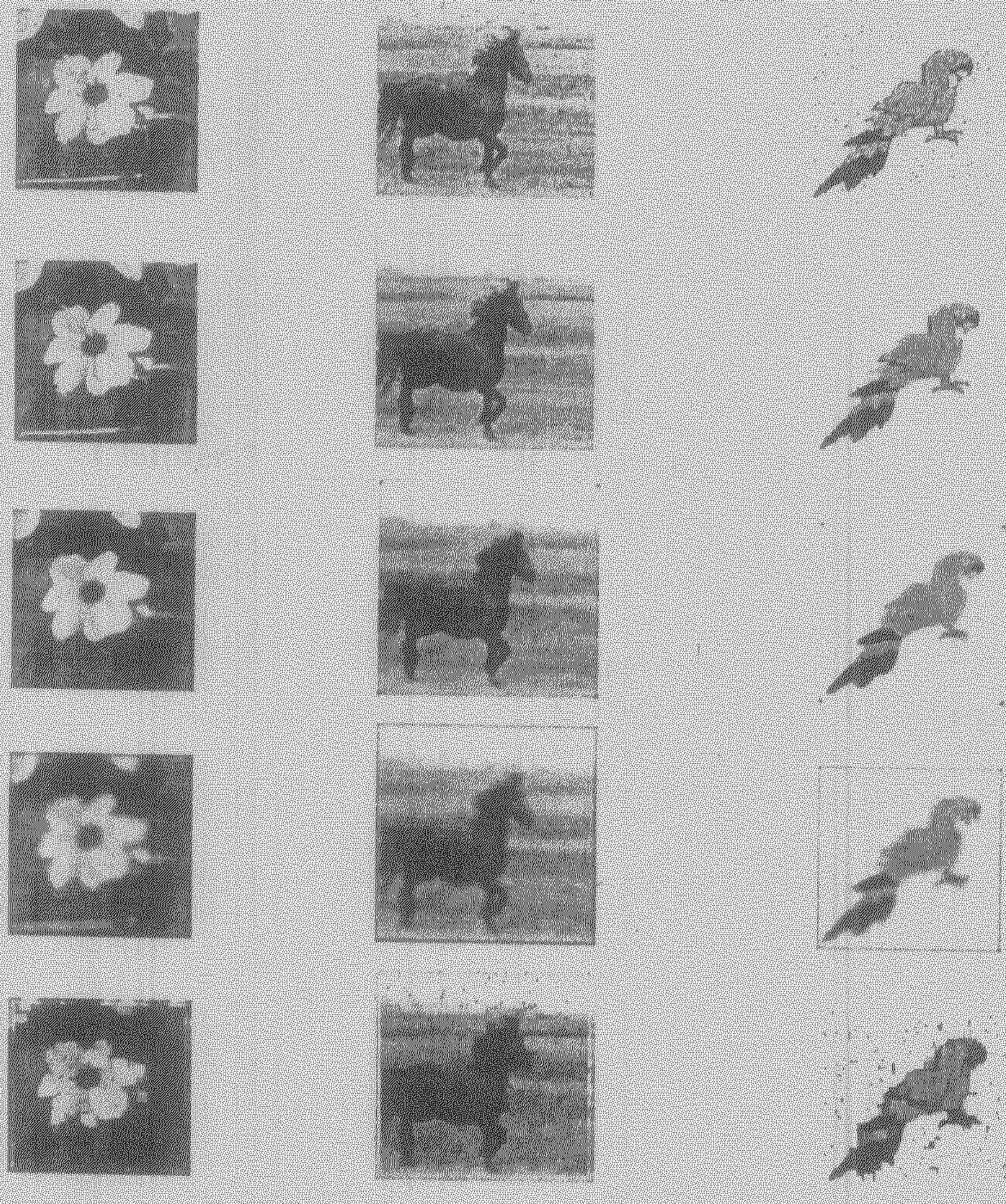
Table 5 shows the performance of AOC in image smoothing in different color spaces. At this point it's of worth much to mention that the minimum area used to run the AOC algorithm has to do a lot in image smoothing, i.e., choice of this area is a critical paramater in measuring performance of AOC in image smoothing. For the present study this area has been chosen to be equal to 100.

Table5: Performance Table of AOC in smoothing

Images	RGB		HSV		HLS		YCbCr	
	d	θ	d	θ	d	θ	d	θ
Fig1	31.59	237.0	29.40	0.00	31.22	45.59	31.76	86.88
Fig2	17.66	308.0	15.56	0.00	18.26	75.01	17.38	125.8
Fig3	25.14	267.3	25.52	0.00	29.98	45.67	23.01	213.7

If we consider only the d-value as the basis of measuring performance of AOC (in image smoothing), then as has been found from the Table5, it is little better in HSV space while being more or less the same for all the other three spaces. θ measure shows that the performance of AOC (in image smoothing) is much better in the HSV and HLS space as compared to the other two spaces. So, the obvious conclusion is that the performance of AOC (in image smoothing) is far better in the HSV space as compared to the other three spaces.

The experimental results ,i.e., the result of noise cancellation in RGB space for all the three sample images have been given overleaf. The results corresponding to other spaces have not been shown.



First row corresponds to the Noisy Image.
Second row shows the result of noise cancellation through AOC.
Third row shows the result of noise cancellation through Median Filtering.
Fourth row shows the result of noise cancellation through Mean Filtering.
Fifth row shows the result of noise cancellation through Max-Min Filtering.

Chapter 3 : Classification

- Introduction
- Fuzzy C- Means Classification
- Methodology
- Results and Discussions

Introduction

Image smoothing algorithms reduces the local standard deviations from an image and as such the area of the flat zones of he image gets increased. This property can be exploited in image classification.

The objective of image classification is to group image vectors based on a similarity measure. Standard minimum distance based classifiers are used for this purpose. The possibility of a single pixel belonging to different classes, suggests the use of an unsupervised fuzzy c-means classifier. So, smoothing followed by a clustering technique applied on an image may result in providing a meaningful classification.

Both AOC and median filtering can be used to smooth an image. So, any one of them followed by fuzzy c-means (FCM) classifier will provide a classification of the image. In this chapter a comparative study has been made to find out which one of these two (that is, between AOC and median filtering) provides better classification of an image compared to the ground classified image (obtained as a result of perceptual classification) when coupled with fuzzy c-means classifier.

Fuzzy C- Means Classification (FCM)

Within the fuzzy c-means clustering algorithm, the familiar least-squared error criterion is applied :

$$J_m(U , \mu) = \sum_{\Omega} \sum_{i=1}^{c} (u_i(x , y))^m \; \| d_i(x , y) \|^2 \qquad \text{----------------} (1) ,$$

Where Ω is the set which we intend to classify. Here, U is the fuzzy c-class partition of the space. Given a space vector I(x,y) at location (x,y), the measure $\| d_i(x , y) \| = \| I(x,y) - \mu_i \|$ is the distance between the space vector and the i th cluster centre μ_i. The distance is weighted by the fuzzy membership value of each space vector $u_i(x , y)$ corresponding to the i th class. The fuzzy exponent m has the range $m \in [1 , \infty]$.

For every feature vector I(x , y), the error criterion $J_m (U , \mu)$ is minimized subject to the condition ,

$$\sum_{i=1}^{c} u_i(x , y) = 1 \;\; , 0 < \sum_{\Omega} u_i(x , y) < |\Omega| \;\; \text{and} \;\; u_i(x , y) \geq 0.$$

For the space vector at (x ,y), the fuzzy membership is updated according to

$$u_i(x , y) = 1 \,/\, \left[\sum_{j=1}^{c} (d_i(x , y) / d_j(x , y))^{2/(m-1)} \right] \qquad \text{------------}(2)$$

The initial fuzzy membersnip value is generated using uniformly distributed random number generator. For the classified image, the cluster centre is updated for all the classes at each iteration according to

$$\mu_i = \frac{\Sigma (u_i(x , y))^m I(x , y)}{\Sigma (u_i(x , y))^m} \text{ ------------------ (3)},$$

where both the summations has to be taken over Ω. The algorithm is terminated for insignificant (1 – 2 % of the current value) changes in μ_i between consecutive iterations. The final classified image is obtained by classifying the pixels based on highest class membership value for each pixel .

Methodology

Every image can be perceptually classified. Classification arising out of perception is thus implemented on the given input image **I** to produce ground classified image **G** which will be the basis of comparison. Input image **I** is then smoothed with the help of AOC. As we have already classified an image to produce **G** , so we know before hand about the number of clusters present in the image. So, now we can classify the smoothed image with the help of FCM classifier with c, the number of clusters being known previously. The result we get out of FCM classification is an another classified image $\mathbf{G_1}$ with c many clusters present in it.

Input image **I** is then again smoothed with the help of median filtering and the resulting image is then classified with the help of FCM classifier to output an another classified image $\mathbf{G_2}$ with c many clusters present in it.

In all the three images **G**, **G**, and $\mathbf{G_2}$, c distinct colors have been used to identify c different clusters that is present in the image. Colors are not assigned to the clusters arbitrarily , rather perceptual idea and the spatial information have been used to identify a particular cluster with a particular color in all the three images **G**, $\mathbf{G_1}$, and $\mathbf{G_2}$.

Now, we will have to compare the performance of two classifications that is $\mathbf{G_1}$ and $\mathbf{G_2}$. with respect to the ground classified image **G**. As we know that c being the number of clusters present in the image , so we form a c x c matrix for each of the images $\mathbf{G_1}$ and $\mathbf{G_2}$ as follows.

Let , c different clusters be identified by the c distinct colors $k_1, k_2, \ldots ,$ k_c . Let, m x n be the size of the input image **I** and thus all the classified images **G**, $\mathbf{G_1}$ and $\mathbf{G_2}$ will also be of size m x n. Now the following algorithm constructs the matrix , for a classified image **H** compared to the ground classified image **G**. The output of the algorithm is the trace of the matrix thus formed.

```
Algorithm Form_Matrix( H )
  begin
    for i = 1 to c do
      for j = 1 to c do
          matrix( i , j ) = 0
      end for
```

```
end for

for i = 1 to m do
   for j = 1 to n do
        s = Color of the (i , j) th pixel in G
        t = Color of the (i , j) th pixel in H
        [ Clearly , 1 ≤ s , t ≤ c ]
        matrix( s , t ) = matrix (s , t) + 1
   end for
end for

output trace of the matrix i.e. , Σ matrix( i , i ) where the summation being taken
over i , 1 ≤ i ≤ c .
end Form_Matrix
```

Trace of the matrix thus obtained will give the number of pixels of image **I** that are rightly classified when compared with the ground classified image **G**. Thus , with the help of the above algorithm trace of the matrix for each of the images G_1 and G_2 can be calculated and the percentage of the total number of pixels that are rightly classified is then also computed for both the images G_1 and G_2. These percentage values then become the basis for comparison between two classifications G_1 and G_2.

Results and Discussions

The whole procedure is executed on three standard 'bmp' images fig2.1, fig 2.2 , and fig 2.3 , that have been shown in Chapter2. Here we will show the results of the aforementioned procedure on these three images.

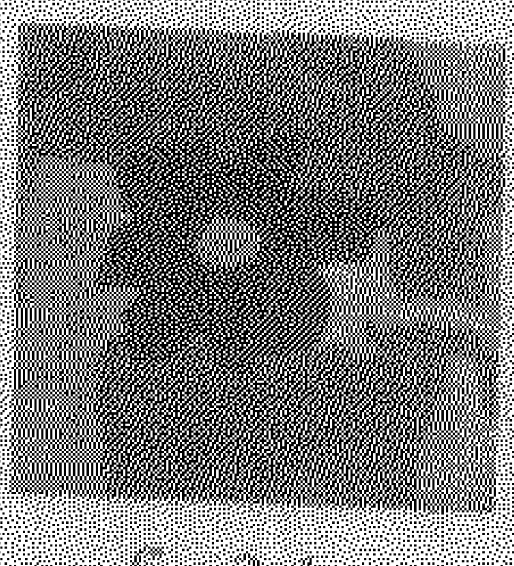


fig 3.1

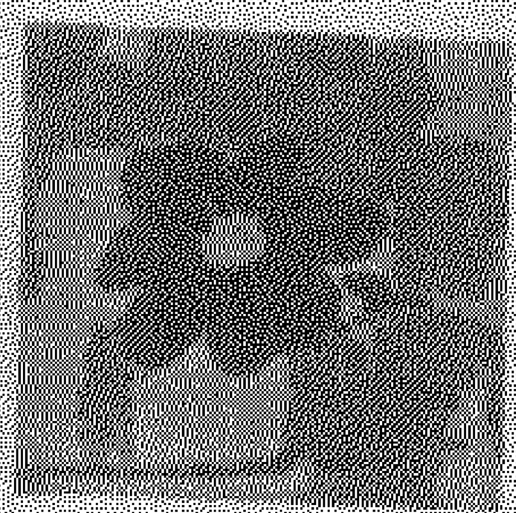


fig 3.2

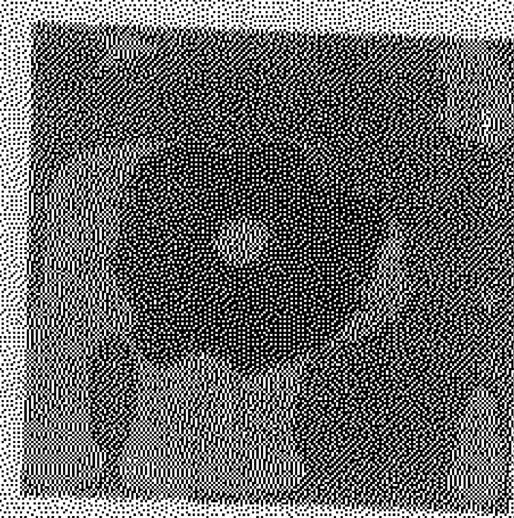


fig 3.3

Figure 3.1 , 3.2 and 3.3 shows the classified images **G** , G_1 and G_2 of figure 2.1. As can be seen from the classified images that the number of clusters present here are 3. So , we form two 3 x 3 matrices — one for each of G_1 and G_2 and the matrices *mat1* and *mat2* that have been shown below are those that we get out of G_1 and G_2 respectively.

	Cl1	Cl2	Cl3
Cl1	2061	43	29
Cl2	204	3716	1242
Cl3	228	394	2083

Mat1

	Cl1	Cl2	Cl3
Cl1	1952	107	74
Cl2	200	3723	1239
Cl3	267	554	1884

Mat2

Figures 3.4 , 3.5 and 3.6 are the classified images **G** , $\mathbf{G_1}$ and $\mathbf{G_2}$ of the input image fig2.2 and matrices *mat3* and *mat4* are performance matrices that we get out of $\mathbf{G_1}$ and $\mathbf{G_2}$ respectively.

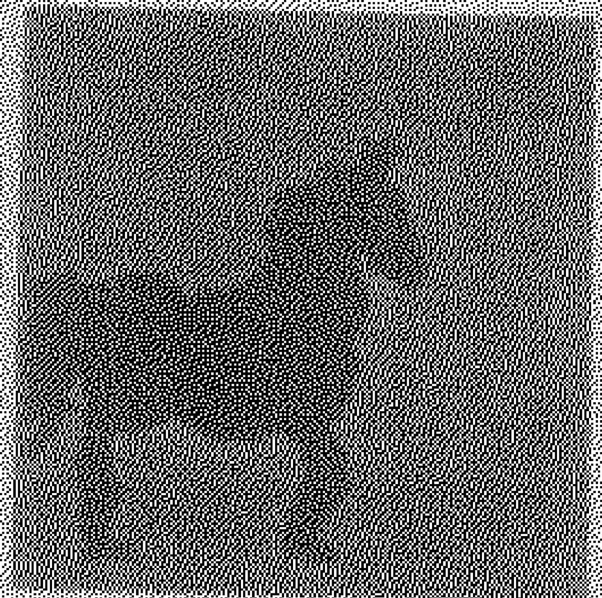


fig 3.4

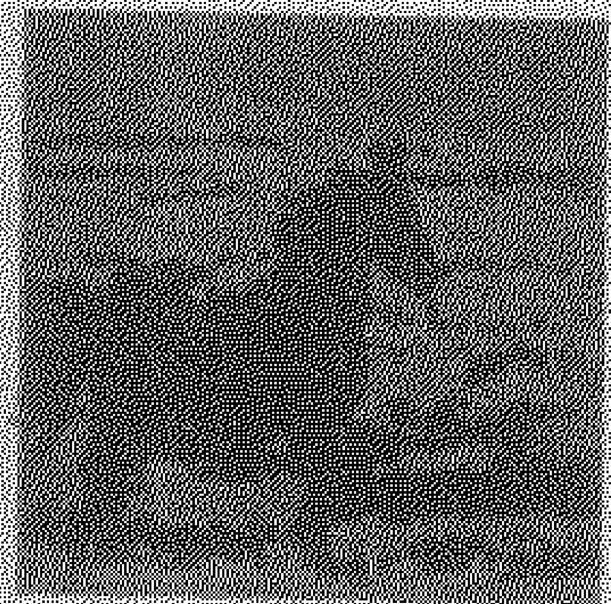


fig 3.5

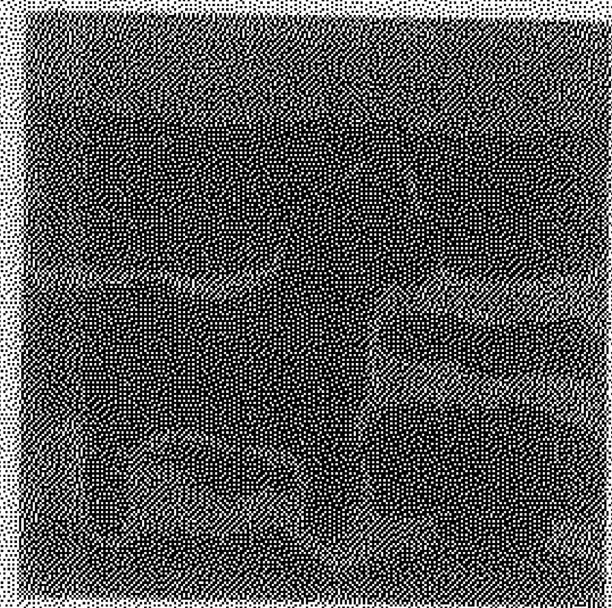


fig 3.6

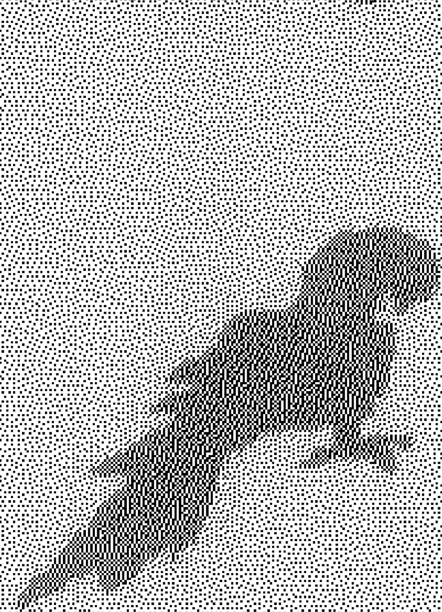
	Cl1	Cl2	Cl3
Cl1	2247	1	0
Cl2	102	6652	2250
Cl3	0	34	3114

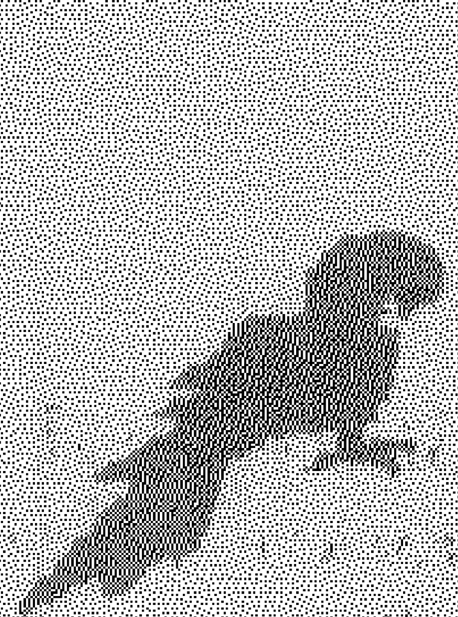
Mat3

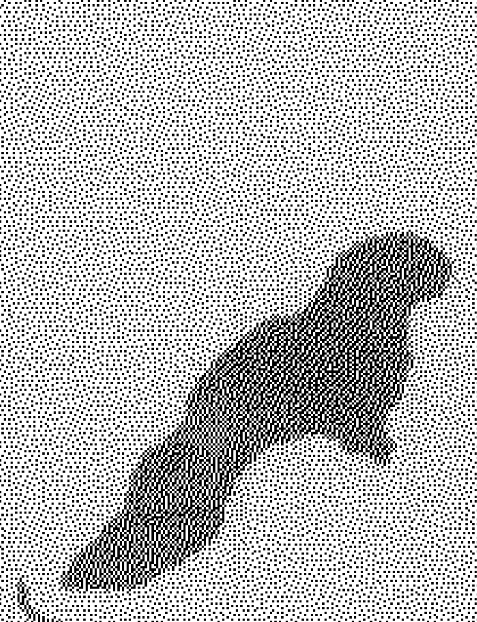
	Cl1	Cl2	Cl3
Cl1	2020	35	193
Cl2	546	2349	6109
Cl3	0	210	2938

Mat4

Figures 3.7 , 3.8 and 3.9 are the classified images **G** , $\mathbf{G_1}$ and $\mathbf{G_2}$ of the input image fig2.3 and matrices *mat5* and *mat6* are performance matrices that we get out of $\mathbf{G_1}$ and $\mathbf{G_2}$ respectively.







	Cl1	Cl2	Cl3	Cl4
Cl1	121	331	3	8
Cl2	390	96	78	0
Cl3	64	0	75	0
Cl4	0	0	0	7737

Mat5

	Cl1	Cl2	Cl3	Cl4
Cl1	1171	300	12	71
Cl2	286	88	113	77
Cl3	25	9	24	81
Cl4	44	51	33	7609

Mat6

Table 3.1 shows the percentage of the total number of pixels that are rightly classified when AOC is followed by FCM and also for median filtering followed by FCM for each of the input images when compared with the ground classified image. As can be seen from the table that AOC followed by FCM provides better classification than the other one for all the three images.

Table 3.1

Images	AOC + FCM	Median + FCM
Fig 2.1	78.60 %	75.59 %
Fig 2.2	83.42 %	50.74 %
Fig 2.3	91.26 %	88.98 %

Chapter 4 : Segmentation

- Introduction
- Segmentation Process
- Segmentation Algorithm
- Results and Discussions

Introduction

The beauty of the AOC operator is that it smooths an image by increasing the area of flat zones and with the increase in area (i.e., the only parameter value that is required to run the Aoc algorithm) the region homogeneity of the image also increases. It also has the interesting property that it preserves contours while smoothing an image. These two properties are very important in image segmentation and that is why in this chapter we have made an attempt, utilizing these two properties of AOC operator coupled with the idea that very close intensity levels in a gray-level image are perceptually indistinguishable, to develop an algorithm for gray level image segmentation. As AOC effectively utilizes the information supplied by the gray-level values of a gray-level image, gray-level segmentation methods have been applied to each component of the RGB space, to obtain the final segmentation result of the color image.

Segmentation Process

The process of segmentation can be divided into three main stages of processing :- increasing the region homogeneity through AOC, merging the homogeneous regions to yield gray-level segmentation and lastly combining the gray-level segmentation result of the three color components.

The first stage of the processing has been described in detail in the first chapter . This stage results in yielding some homogeneous regions. These homogeneous regions are treated as the initial segments which are then subject to merging in the second stage.

The merging process of the homogeneous regions have been done in the frequency domain. In the frequency domain each homogeneous region is represented by it's representative gray level value. A homogeneous region may be divided into several disjoint smaller (in size) homogeneous regions in the spatial domain. Had the merging process be done in the spatial domain, the spatial proximity of the regions have to be taken into account. But, as we are doing the merging process in the frequency domain — so merging the two homogeneous regions means only to replace the gray level representative of one region by the gray level representative of other region. This merging process is implemented by replacing the old value by the new value in the corresponding positions of the image without affecting it's position. So, the benefit of merging in the frequency domain is that the spatial proximity of the homogeneous region is never lost due to merging.

The first stage yields some homogeneous regions which are considered as initial segments in the second stage. The frequency domain representation of these homogeneous regions is a set of gray level values with non-zero frequencies. By frequency we mean here the cardinality of the homogeneous region, i.e., the number of pixels comprising the image.

This initial segments are then merged to give rise a new set of segments. These new set of segments are again subject to merging and this iterative merging procedure continues until the segments are sufficiently separated. At the end of the first two stages , we have the gray level segmentation of a gray-level image. The last step is to combine the gray level segmentation result of three color components, viz, of R, G and B channels, to obtain the desired segmentation of the colored image.

So far we haven't described in detail the merging process, i.e., how the segments in different iterations are merged. The merging process is centered around the answers of two questions and they are,

1) What will be the criteria of merging, that is to say, a deterministic criteria which when will be true merging will be continued. When the criteria will no longer be valid the process of merging will be stopped.
2) Which region is going to merge with which segment.

This questions will be answered as and when the algorithm will be described.

Segmentation Algorithm

We start with describing some terminologies that will be used in the algorithm of color image segmentation. AOC is the function implementing the area open_close operator which takes as input the gray-level image I and a parameter value, called area and results in producing an another gray-level image SI . HISTOGRAM is another function which takes as input a gray-level image and outputs an array FREQ of size 256 , where FREQ(i) gives the frequency of the intensity-level i in the input image.

We define INDEX = $\{ i : i \in \{0 , 1 , 2 , \dots , 255\}$ and such that FREQ(i) $\neq 0\}$. The values of i in INDEX are arranged in ascending order and let them be $X_1, X_2, \dots , X_N$ such that $X_1 < X_2 < \cdots < X_N$. The cardinality of the set INDEX is N which signifies that the initial number of segments at the end of the first stage is N.

Let , PASS = min $\{ \mid X_{(I+1)} - X_I \mid \}$ where minimum is to be taken over I where I ranges from 1 to N-1. This the initial value of PASS which gives the minimum distance between two initial segments. We then define another array SLOPE of size N-1 as follows ,

SLOPE(I) = { FREQ(INDEX(I+1)) – FREQ(INDEX(I)) } / PASS, where INDEX(J) denotes the J th element of the set INDEX and such that INDEX(I+1) – Index(I) is equal to PASS. SLOPE(I) = –1 ; if INDEX(I+1) – INDEX(I) $\neq$ PASS. We have an another function called SORT which takes as input the array SLOPE and outputs it's elements in descending order via an array SORTED_SLOPE. While sorting care has to be taken to ensure that the position of the elements in SLOPE are not changed. This is required , because by looking at the index I of this array we can say the pair of gray levels (X_{I+1} = INDEX(I+1) and X_I = INDEX(I)) for which this value of SLOPE (that is SLOPE(I)) has occurred. Now, we will start describing the algorithm of gray-level image segmentation.

Algorithm Gray_Segmentation(Gray-level Image I , Area)
begin
1) SI = **AOC**(I , Area).
2) FREQ = **HISTOGRAM**(SI).
3) Form the set INDEX as defined above.
4) Assign Initial Value to PASS as defined above.
5) Construct the array SLOPE as defined above.
6) SORTED_SLOPE = **SORT**(SLOPE).
7) t ← 1 ; cnt ← 0
while (SORTED_SLOPE (t) ≠ -1)
Form INDEX1 = { I : SLOPE (I) = SORTED_SLOPE(t) }.
for each element I in INDEX1 **do**
if FREQ(INDEX(I)) ≠ 0 AND FREQ(INDEX(I) + PASS) ≠ 0 **then do**
cnt = cnt + 1
STORE(cnt) = INDEX(I)
Merge the Regions corresponding to the gray levels INDEX(I) and INDEX(I) + PASS.
Update the array FREQ appropriately.
Update the image by replacing the gray level of one region with the gray level of the other region.
end if
end for
t = t + 1.
end while
8)a) STORE = **SORT_ASCEND**(STORE).
b) Calculate the Maximum Length of Consecutive Merges between the gray levels that has taken place in step (7) where the merged gray levels are all PASS distance apart and store the value in the variable Max_Count.
c) Deallocate STORE.

[**Illustration:** STORE = { J : Merging of the gray levels J and J + PASS has taken place in step (7). The function SORT_ASCEND then sorts the elements in STORE in ascending order. The elements of the set STORE are then scanned to find out the maximum length of the consecutive merging that has taken place in step (7). The variable Max_Count stores the maximum length of this consecutive merging in step (7) , that is , there exists a J belongs to STORE such that J , J + PASS , J + (2 * PASS) , ..., J + (Max_Count * PASS) is the maximum length such that there always exists a merging between two consecutive gray levels J + (I – 1) * PASS and J + I * PASS ,where I = 1 , 2 , ... , Max_Count. It is interesting to note here that in the worst case (i.e. , when all the mergings have occurred in on fixed direction) , the two consecutive gray levels for which frequency is still not equal to zero will be at the most (Max_count + 1) * PASS distance apart. This is the first phase of merging.]

```
9) Flag = 1 ; Max_Dist = PASS.
   Deallocate SLOPE , SORTED_SLOPE , INDEX , INDEX1
   while Flag = = 1
     Flag = 0
     Max_Dist_Merge = Max_Dist
     Range = Max_Count * Max_Dist
     for PASS = (Max_Dist_Merge + 1) to (Max_Dist_Merge + Range) then do
        Form INDEX again as in step ( 3) with the help of updated FREQ.
        /* FREQ gets updated always whenever there is a merging */
        Construct SLOPE again with the help of new INDEX as in step (5) using the
        current value of PASS.
        if IS_NON_NEGATIVE( SLOPE ) then do
           /* The function IS_NON_NEGATIVE scans through the elements of the
            array SLOPE and returns TRUE if there exists at least one non-negative
            element in the array SLOPE , otherwise returns FALSE */
           Flag = 1.
           Max_Count = 0.
           Max_Dist = PASS.
           cnt = 0.
        end if
        Repeat step (6).
        Repeat step (7).
        Deallocate SLOPE , SORTED_SLOPE , INDEX , INDEX1.
     end for
```

[**Note :** PASS will vary within a range in the for loop and for each value of PASS = X within that range there will be merging if there are gray levels X distance apart. The value of PASS within the range for which the merging has last taken place i.e. , the maximum value of PASS within the range for which there is a merging helps in finding the range in the next level of iteration. If there is no merging in the current iteration then the algorithm will stop. If the maximum value of PASS within the range for which there is a merging is D , then PASS will range from (D + 1) to (D + D * Max_Count) in the next level of iteration.]

```
     if Flag = = 1 then do
          Repeat step (8).
     end if
  end while
```

[**Illustration:** The number of gray levels with non-zero frequency will reduce in each iteration and also the distance between the consecutive gray levels (gray levels with non-zero frequency are the representatives of the current segments in frequency domain. After some iterations the gray levels with non-zero frequency will be sufficiently separated such that there will be no chance for Flag to set again and hence the algorithm converges.]

10) Output the Updated Image we have in hand at the end. This image corresponds to the gray level segmentation of the gray level image given to the algorithm as input.

end Gray_Segmentation

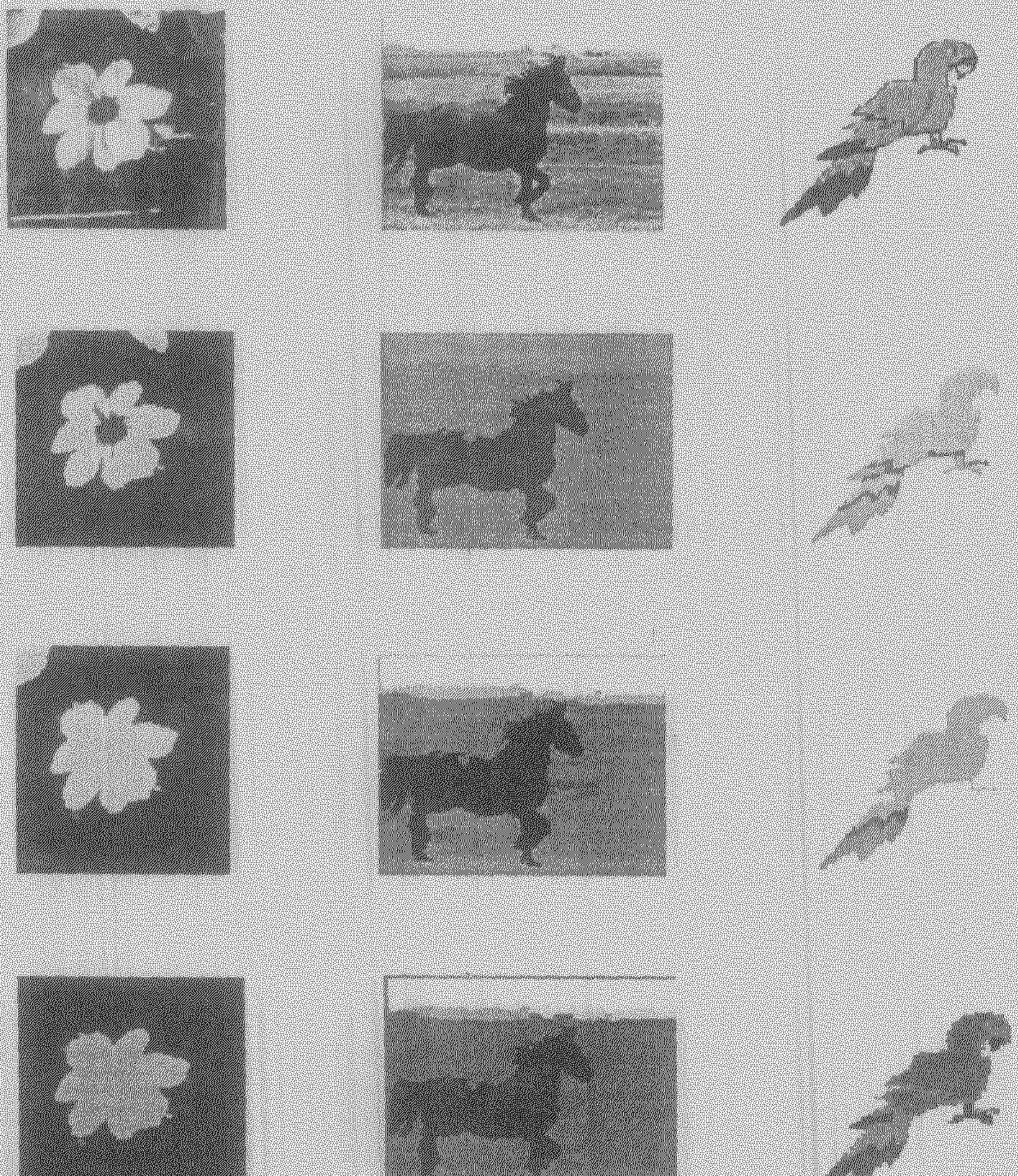
Algorithm Color_Image_segmentation(Color_Image I)
begin

1) Separate the Color Components R , G and B from the given image.
2) Area = Input from the user.
3) Segmented_R = Gray_Segmentation(R , Area).
4) Segmented_G = Gray_Segmentation(G , Area).
5) Segmented_B = Gray_Segmentation(B , Area).
6) Combine the Color Components Segmented_R , Segmented_G and Segmented_B to produce Segmented_Color.
7) Output Segmented_Color

end Color_Image_segmentation

Results and Discussions

The algorithm does not give any idea about what could be the effective value of area (the only parameter to run the AOC algorithm) which will yield effective segmentation of the input image. The above algorithm has been executed on the sample images shown in Chapter 2 for different values of area (100, 200 and 500) and the result of segmentation have been shown in overleaf.



First row shows the original input images.
Second row shows the result of segmentation for Area is equal to 100.
Third row shows the result of segmentation for Area is equal to 200.
Fourth row shows the result of segmentation for Area is equal to 500.

Chapter 5 : Conclusion and Reference

- Conclusion
- References

Conclusion

At the end it would be better to look back and to provide a critical review of what we have done so far. Throughout the study we have tried slowly and logically to gather points in favour of area open_close (AOC) operator through exploration of it's application in different image application areas. But still the present study is not devoid of limitations and subjectivities. It's now the time to note them down too.

In Chapter 2, RMSE and θ- measure have been taken as the basis of comparing performance of AOC over other noise cancellation procedures in different color spaces. It is true that if two images are exactly similar, then RMSE and θ- measure between them will be both zero. On the other hand, if the RMSE and θ- measure of two images I_1 and I_2 with respect to a third image I are same, then that doesn't imply that I_1 and I_2 are similar images. That means one of I_1 and I_2 may be perceptually similar to the image I, while the other one is an altogether different image, but still their RMSE and θ- measure value w.r.t the image I may be the same. This conclusion somewhat weakens our motivation of measuring and comparing performance with the help of these two measures. Further research is needed in this direction to come up with a novel solution of measuring performance of filtering procedures.

In Chapter 3, we have tried to establish experimentally that AOC followed by FCM provides better classification than median filtering followed by FCM when compared with the ground classical image (perceptually classified). Human perception is not devoid of subjectivity and as because comparison has been done with respect to the perceptually classified image, so certain amount of subjectivity will always be there in the results that we get out of this type of experiments. Though we prove our claim in favour of AOC, but still it can't be proved objectively.

In Chapter 4, we have made an attempt to develop a gray level image segmentation algorithm using the flat zones that come out of as a result of AOC. The result of this gray level segmentation applied on three channels Red, Green and Blue, combined together yields the desired segmentation of the colored image. For all the three channels the same area has been used as threshold. Different values of area threshold for different channels may even yield much better result. Actually, we have failed to find out an objective method to predict about what could be the effective value of area threshold which will yield the most effective result of segmentation. Further research is also needed in this direction. Instead of all the limitations, however we have been successful in obtaining a lot many points in favour of area open_close operator, and these points again stand behind the increasing popularity of filters by reconstruction like AOC in many image application areas.

References

[1] J. Serra and P. Salembier, " Connected Operators And Pyramids", in *Proc. Of SPIE Image Algebra and Mathematical Morphology*, San Diego, California, USA, SPIE Vol. 2030, pp. 65 - 76, 1993.

[2] S. T. Acton and D. P. Mukherjee, "Scale Space Classification Using Area Morphology", *IEEE Transactions on Image Processing*, Vol. 9, No. 4, April 2000.

[3] P. Salembier and J. Serra, "Flat Zones Filtering, Connected Operators, And Filters By Reconstruction", *IEEE Transactions on Image Processing*, Vol. 4, pp. 1153 - 1160, Aug. 1995.

[4] Heng-Da Cheng and Ying Sun, " A Hierarchical Approach To Color Image Segmentation Using Homogeniety", *IEEE Transactions on Image Processing*, Vol. 9, No. 12, December 2000.

[5] R. Schettini, " A Segmentation Algorithm For Color Images", *Pattern Recognition Letters*, Vol. 14, pp. 499 – 506, 1993.

[6] P. Salembier and J. Serra, "Morphological Multiscale Image Segmentation", in *Proceedings, Visual Communication and Image Processing*, 1992, Boston, MA, pp. 620 – 631.

[7] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI – 9, pp. 532 – 550, July 1987.

[8] M. Celenk, " A Color Clustering Technique For Image Segmentation", *Computer Vision, Graphics, and Image Processing* 52, 145 – 170, 1990.

[9] F. Meyer and S. Beucher, " Morphological Segmentation", *Journal of Visual Communication and Image Representation*, Vol. 1, No. 1, pp. 21 – 46, 1990.

[10] A. Rosenfield and A. C. Kak, Digital Picture Processing, 2nd ed., Vol 1 & 2.