IMAGE SEGMENTATION FOR OBJECT BASED IMAGE COMPRESSION

By

JUNAID AHMED

Bachelor of Science

Middle East Technical University

Ankara, Turkey

1997

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2000

IMAGE SEGMENTATION FOR OBJECT BASED IMAGE COMPRESSION

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

ACKNOWLEDGMENTS

There are a number of people that I would like to acknowledge. First and foremost my advisor Dr. Scott Acton for his support and encouragement both in my area of study and financially. Dr. Yarlagadda and Dr. Teague for reviewing my thesis and serving on my committee. My colleagues Joseph Bosworth, Badri Raghunathan, Yongjian Yu, Khaled Al-Mughadhawi, Nilanjan Ray, Zhongxiu Hu and David Cary for their friendship and help whenever I needed it. Thanks also go to Mrs Rea Maltsberger for helping me in all the ways that she could. Special thanks go to my Parents, my brothers and sister for always praying for my success and goodwill.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure		Page

Figure		Page

CHAPTER I

INTRODUCTION

This thesis addresses the use of image segmentation for object based coding. Image segmentation and coding are two different and important tasks in image processing. Image segmentation is the first step in many computer vision systems in use today. Object-recognition and target tracking are two salient examples. Whenever images have to be stored or sent over transmission lines they are compressed. Image compression and coding has also been widely studied and used to save disk-space and transmission bandwidth.

There are two categories of image segmentation techniques, edge based segmentation and region based segmentation. The edge based segmentation techniques look for sharp change in intensity values to locate boundaries and the region based approaches look for homogeneity of pixel values to group them together. Different segmentation methods have been designed for different applications. But a dilemma concerning the segmentation researchers has been to find a method to measure the segmentation quality. Object based compression and coding is an important application of image segmentation. It can be used as an effective measure of segmentation quality.

Different image compression methods have been designed and standardized. Block based methods and wavelet based methods are two important types that have emerged in the JPEG and JPEG 2000 standards, respectively. Block based image compression methods are those where image is divided into fixed sized blocks before further processing. In wavelet based methods image is divided into bands of different frequencies before processing each one alone. Object based compression is a relatively new method that is being researched, it divides an image into arbitrarily shaped regions called objects, depending on the contents of an image. It has applications in video editing, motion estimation, content based image retrieval, low bit-rate video coding and biomedical imaging. The performance of an object-based image coding algorithm highly depends on segmentation. An overly segmented image that would result from directly applying edge based segmentation techniques to a texture image would be inefficient. And a semantically meaningless segmentation that would result by segmenting a smooth real image by texture segmentation techniques would be unacceptable. It will not be practical to hand pick a segmentation technique for every image to be later compressed by a general purpose object based coding algorithm. In short there is a need of a segmentation technique that is cost effective and is applicable on a variety of images.

The purpose of this research is two fold, to present a new segmentation method that can be used in a broad range of images and is good in terms of higher PSNR at the same compression ratio. The second purpose is to introduce object based coding as an effective measure to test the performance of a segmentation method. The rest of this thesis is organized as follows. In the next chapter a review of the literature on image compression and segmentation is given. In chapter 3 description and analysis of the segmentation and compression techniques used in the research are presented. Results of the research are presented in chapter 4 and chapter 5 concludes the thesis.

Chapter II

REVIEW OF LITERATURE

2.1 Image Coding Techniques

Several Methods for image coding have been developed, these methods can be divided into three main groups block based coding, wavelet based coding and object based coding. These three methods differ in the way they divide an image before coding starts.

2.1.1 Block Based Coding

The most popular block based coding technique is the standardized JPEG. Other algorithms have also been developed recently that use block based coding technique and give better results depending on the application, these methods are discussed below.

2.1.1.1 JPEG

JPEG is an abbreviation of Joint Photographic Experts Group , this standard was developed to be a compromise between image quality and compression. This algorithm first divides an image into 8 x 8 blocks and processes them separately. A two dimensional discrete cosine transform of these 8 x 8 blocks are taken and quantized using an application dependent quantization matrix. These transform coefficients are then compressed using Huffman compression technique. This algorithm gives a fairly good result but has disadvantages as blocking and blurring of edges which are removed by object based coding, which will be discussed later.

2.1.1.2 Block Classification

This compression technique discussed in [1] and [2] was developed utilizing the properties of human visual system (HVS). HVS recognized images by their shapes and regions and not their intensity values. The algorithm used by this method is explained below:

An image is divided into 8 x 8 blocks as in JPEG and then these blocks are further divided into 3 categories smooth regions, texture regions and edge regions. Smooth regions are coded using the DC value only and that is also coded differentially. Texture regions are coded by taking the discrete cosine transform and coding the coefficients as done in JPEG. Edge regions contain the maximum information, they are also coded using discrete cosine transform but more accurately than the textured regions.

A further improvement in this algorithm can be done by first dividing the image into 32x32 super-blocks and then checked for homogeneity. If a block is homogeneous it is coded as a single 32 x 32 but if the block is not homogeneous it is further divided into 16 x 16 and/or 8 x 8 sub blocks and checked for homogeneity. These homogeneous blocks are then coded using the method described above. The block classification method improves the edge quality over JPEG but is not efficient and does not give the functionality that an object based coding method does.

2.1.1.3 Region of Interest Coding

This is a simple algorithm where blocks of interest are coded more accurately than other blocks and the region of interest is selected by a user. It has wide range of applications including biomedical imaging where the background is of no interest, so it is coarsely coded. Since this method requires user interaction it is not practical for many applications.

2.1.2 Wavelet based Coding

It is a subband image coding technique. Image is passed through a set of bandpass filters and their outputs are quantized and coded separately. These bandpass filters form an orthonormal basis function so that the original image can be reconstructed exactly by passing the filtered signals through a bank of reconstruction filters. A graphical representation of a general wavelet based image coding algorithm is shown below:

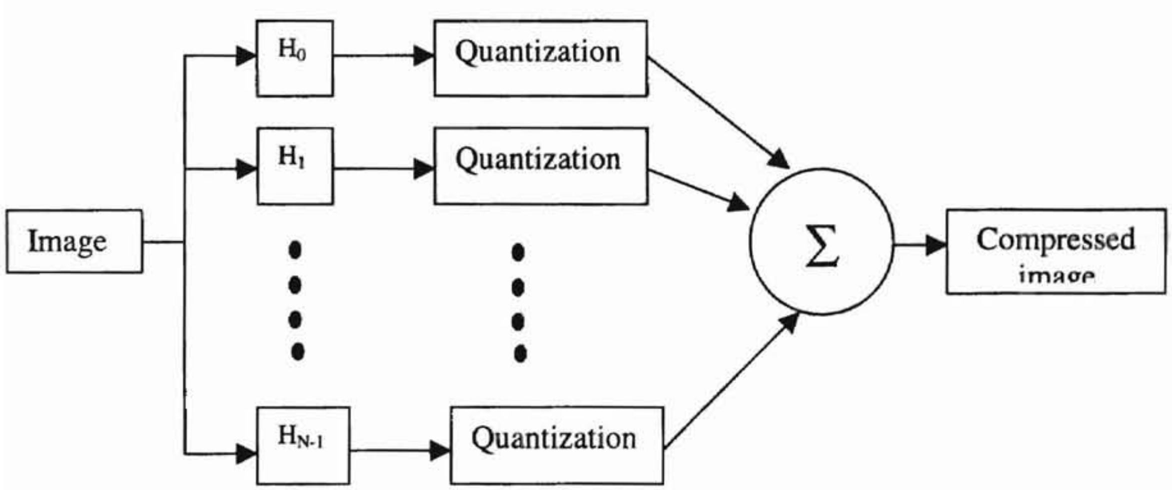


Figure 1: Block diagram of wavelet based coding

One implementation of wavelet-based coding is by using subband filters. These filters decompose an image into high and low-frequency bands. The lowest frequency band is then again passed through these filters to further decompose it into smaller bands.

2.1.3 Object Based Coding

Image can also be divided into objects of arbitrary shape depending on the content instead of a fixed block size. These objects are then coded separately. This division gives more functionality to the compressed images and also improves the sharpness of the edges. Two different categories of object based coding exist depending on whether the shape of the object effects the coding algorithm or not, these are given below.

2.1.3.1 Shape Adaptive Coding

Images are coded according to their shapes, two different ways of coding can be grouped in this category.

2.1.3.1.1 Pixel Shifting

This algorithm was proposed in [3] and is illustrated below in figure 2. In the figure an arbitrarily shaped object (dark) is contained within a rectangular block (light). The size of the block can be set according to the shape of the object or can also be predefined as the maximum size that any object can take. To code the object all the pixels belonging to the object are first moved up and aligned with the upper boundary and then one dimensional discrete cosine transform is taken, the size of the transform is set to be

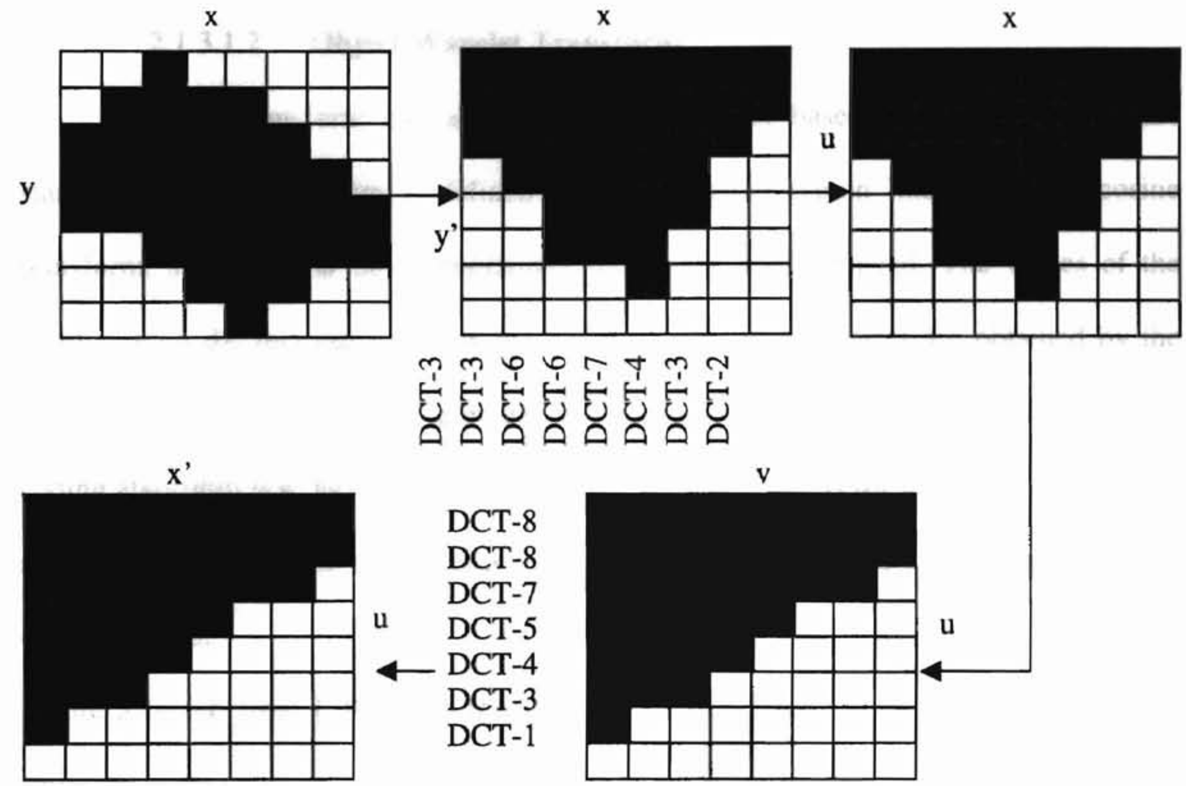


Figure 2. Successive steps involved to perform shape adaptive DCT.

the number of pixels in that column. After the first transform the pixel values are replaced by the transform coefficients. The rows are now shifted towards the left border and a similar one-dimensional DCT along the horizontal axis is calculated.

An improvement to this algorithm was suggested in [4]. The alignment of the vertical DCT coefficients to the left column increases entropy. If we align the DCT coefficients according to a look up table generated for different shapes and sizes, the entropy can be reduced.

2.1.3.1.2 Object Wavelet Transform

Wavelet transform can also be used for object based coding [5]. Since the standard wavelet transform is defined on a rectangular region like a discrete cosine transform, a rectangular block containing the object is constructed. The values of the pixels within the rectangular block but not belonging to the object are obtained by the average value of the neighboring pixels that belong to the object. The redundancy of the coding algorithm is reduced by first using only the object pixels for low-pass filtering and second by replacing the coefficients coming from the pixels that do not belong to the object, to zero.

Figure 3 (taken from [5]) shows the selection of the coding coefficients for an object, shown in dark.

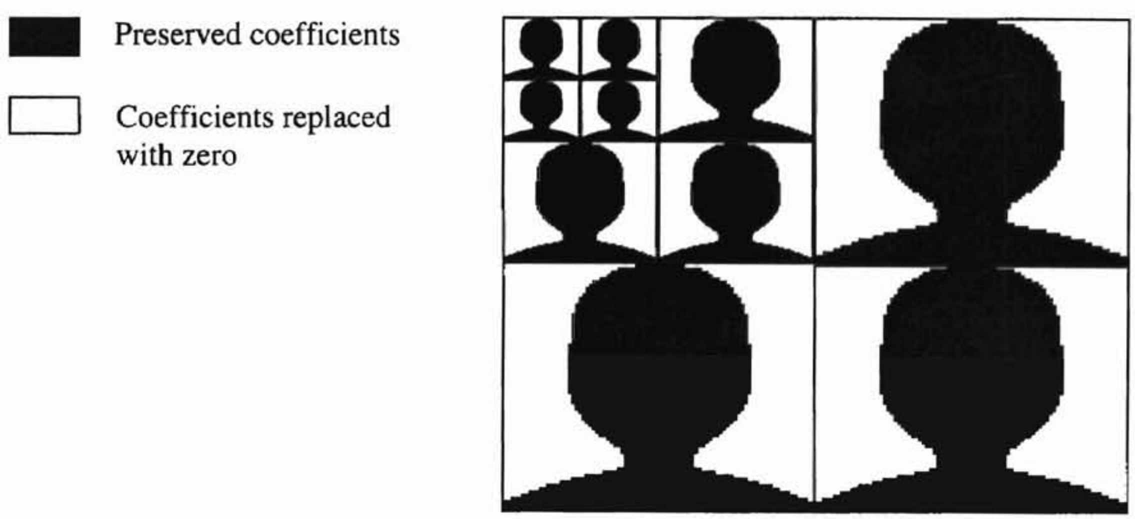


Figure 3. Coding coefficient selection

2.1.3.2 Shape Independent Coding

An image can be approximated by a set of basis functions as follows:

$$f(x,y) = \sum_i \beta_i \psi_i(x,y) \qquad x, y \in S$$

where S is the region occupied by the image and ψ_i are the basis functions. These basis functions have to be orthogonal to each other with respect to the image shape. Since the image shape is arbitrary it is very difficult to always find the mutually orthogonal basis functions for them. Instead we can find the basis functions for a rectangular block and then try to optimize the basis functions on the shape. This method introduces redundancy in the image, in order to overcome this redundancy and be able to best represent the object with least number of coefficients different methods have been suggested and are described below.

2.1.3.2.1 Zero padding

The easiest way to reduce the redundancy is to set all the pixel values outside the object to zero (this method is discussed in [6] and [7]). This seems like a good option at the first look but it introduces new high frequency coefficients in discrete cosine transform. This happens because of the discontinuity at the object boundaries.

2.1.3.2.2 Mirroring

The values of the pixels outside of the object can be set using the pixel values inside the object by mirroring. This can reduce the effects of the boundary. Different ways of mirroring have been suggested *e.g.* edge repetition, mirroring in two directions separately *etc.* These methods give different results depending on the image.

2.1.3.2.3 Exhaustive Searching

Exhaustive searching can be used to achieve this task but it is computationally very expensive. We can exhaustively search at both DCT level and in the spatial domain. Exhaustive searching in the spatial domain is a relatively easier task, since, in general, the number of pixels outside the object will be less than the number of pixels inside it. Still the complexity is high enough to make it unpractical. For example, in an 8x8 block, if there is a single pixel that does not belong to the object itself than it will take 256 iteration to find its optimum value. If there are two such pixels then the number of iterations needed will be 256 x 256 *i.e.* the complexity goes as 256^n.

2.1.3.2.4 Projection onto Convex Sets

Another method proposed by [8] and called *projection onto convex sets* can be used to estimate the outside pixels. The method is described in figure 4 below. First we find the segmentation for the image and then take its discrete cosine transform and zero high frequency or low energy coefficients. An inverse transform is obtained and the value of the pixels that are part of the object is set to the original. We again take the transform and zero some of the coefficients and continue doing process till it converges or a maximum number of iterations have been done. This procedure is successful because by zeroing the low frequency coefficients we slowly remove the high frequency effects of the boundary.

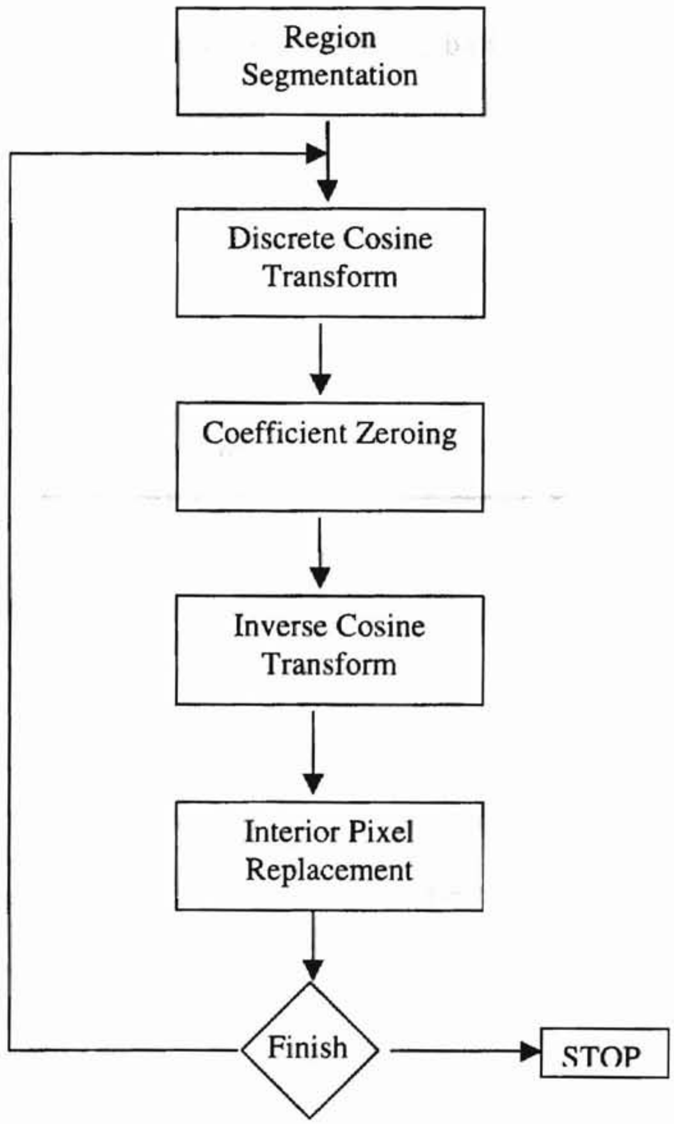


Figure 4: Block diagram of projection onto convex sets algorithm

2.1.3.2.5 Shape-Independent Basis Functions

An arbitrarily shaped image segment can be approximated by a set of orthonormal basis functions defined on a rectangle around the object. These basis functions can be selected one by one depending on the energy associated with each one within the object. The algorithm developed by [9] is applied on each object separately. First we define the object within a rectangle and estimate it using one basis function. This estimate is then subtracted from the original to produce a residue. This residue is used to select more basis functions. We select the basis functions by looking at its energy within the object defined by the following equation:

$$\Delta E_A = \frac{\left[\sum_{(m,n)\in A} r(m,n)\varphi_{uv}(m,n) \right]^2}{\sum_{(m,n)\in A} \varphi^2_{uv}(m,n)}$$

Where ΔE_A is the energy of the basis function φ_{uv} within the residue $r(m,n)$ of the object represented by the area A.

A graphical representation of this algorithm is shown in figure 5. In the figure $f(m,n)$ is the input image $g(m,n)$ is approximation using the selected basis functions. ΔE_{uv} is the energy associated with basis function φ_{uv} where u and v represent the frequencies of the basis function.

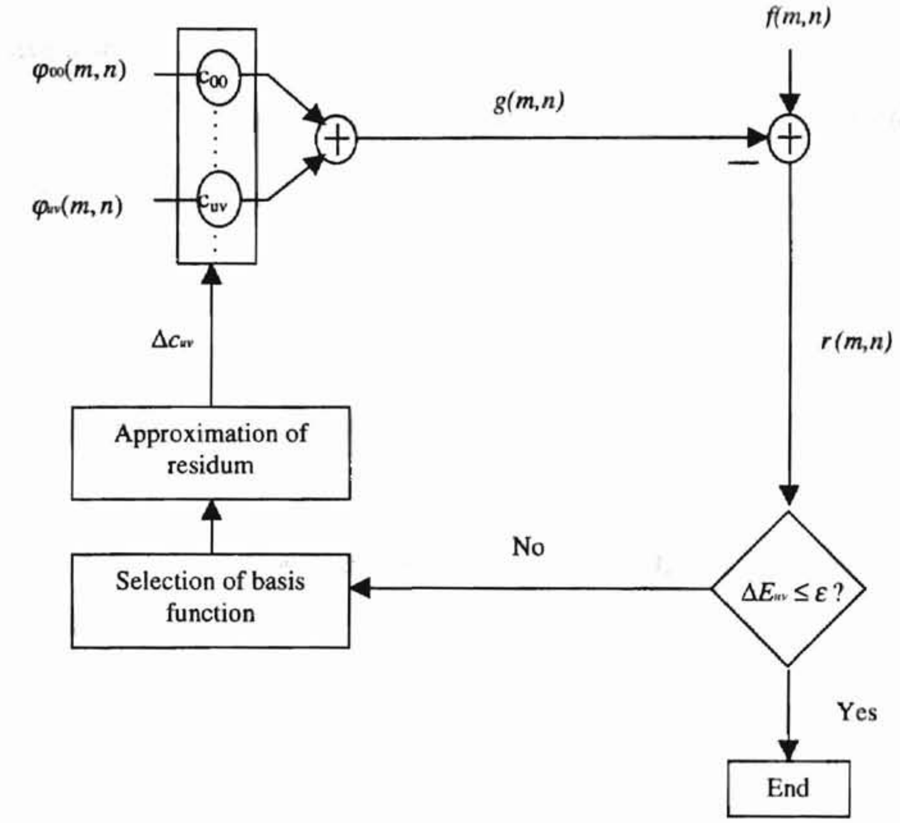


Figure 5: Block diagram of shape-independent basis function

2.2 Segmentation Techniques

Image segmentation is highly dependent on the application, for example in a tracking application only the target is important, so if we segment the target from the background, we have a small portion of the image to work on. Several methods have been developed for segmentation utilizing different aspects of an image. Some methods look at the edges in an image others try to concentrate on neighboring pixel values and some methods consider both of them. This brings us to the question, which segmentation technique is better than the other. This question can be answered according to the application that uses the segmentation. We shall try to answer this question for object based coding in this thesis.

Some of the segmentation techniques developed in the last few decades are discussed below.

2.2.1 Edge Based Segmentation

An edge in an image is the region where there exists a sudden, sustained jump in the gray level. Therefore the first step in each edge based segmentation technique is to first detect an edge using a local edge detector. Taking derivative across an edge produces a peak value at the edge and thresholding will give the edge locations. If we take a second derivative this peak becomes a zero crossing, which is easier to detect. The figure 6 [12] explains this method of edge detection:

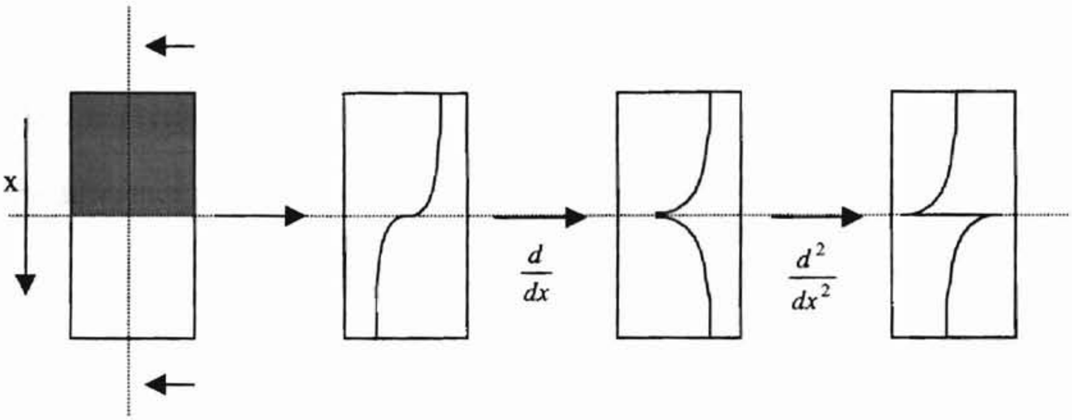


Figure 6: First and second derivatives across a local edge

Since an image is a two dimensional signal we need to calculate its two dimensional derivatives to find edges. Gradient and Laplacian based methods are common.

2.2.1.1 Gradient

The gradient of an image *f(x,y)* at a location *(x,y)* is defined as a vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is generally used in edge detection it is simply called the gradient

$$\nabla f = [G_x^2 + G_y^2]^{1/2}.$$

It is also a common practice to approximate the gradient with the magnitude value of the horizontal and vertical gradient vectors as with

$$\nabla f \approx |G_x| + |G_y|.$$

There can be different implementations of the gradient for image processing, however two implementations are famous these are Sobel and Prewitt edge detectors.

The computation of the partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial x}$ in an image is similar to calculating the difference of the neighboring pixels *e.g.* $\frac{\partial f}{\partial x}$ at a location (x,y) can be calculated by subtracting the pixel values at (x+1,y) from the one at (x,y). Sobel oprators [10] use nine pixel values to calculate gradient in one direction at one pixel. The masks in figure 7 are used to find the partial derivatives G_x and G_y.

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

Figure 7: (a) mask to calculate G_x (b) mask to calculate G_y

The benefit of using 3x3 mask is it gives a smoothing effect. Since differentiation is very sensitive to noise smoothing the image gives better results. An edge map of an image can be obtained from the gradient of the image by thresholding. There is no automatic way to select the threshold it is selected by the user according to the application.

The Prewit algorithm [11] is similar to the Sobel algorithm, the only difference is the mask. The masks of the Sobel algorithm are given in the figure 8.

-1	-1	-1
0	0	0
1	1	1

(a)

-1	0	1
-1	0	1
-1	0	1

(b)

Figure 8: (a) mask to calculate G_x (b) mask to calculate G_y

The Sobel algorithm gives higher weights to the horizontal and vertical differences than others while the Prewitt's algorithm gives the same weight to all the differences.

Figure 9 gives an example of edge detection using Sobel's algorithm.

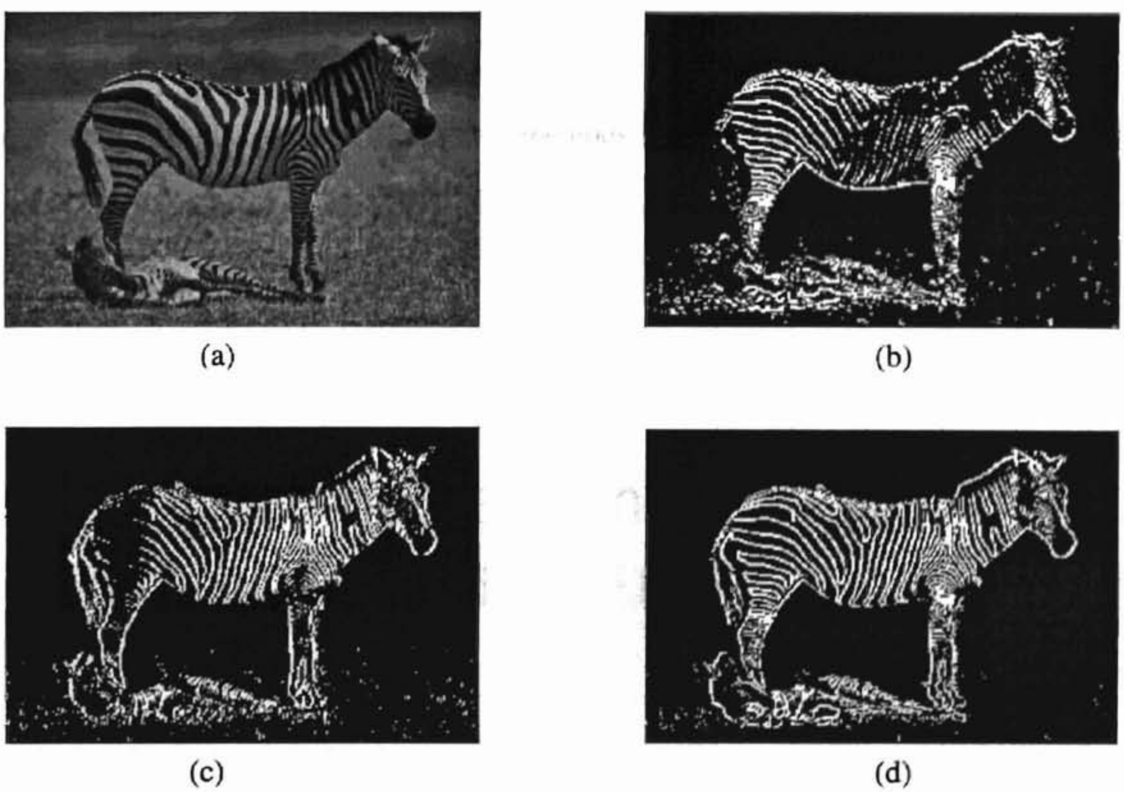


Figure 9: (a) Original image (b) Result of finding edges in vertical direction (c) result of finding edges in the horizontal direction and (d) absolute sum of (b) and (c)

2.2.1.2 Laplacian based Detectors

The Laplacian of an image f(x,y) at a location (x,y) is defined as

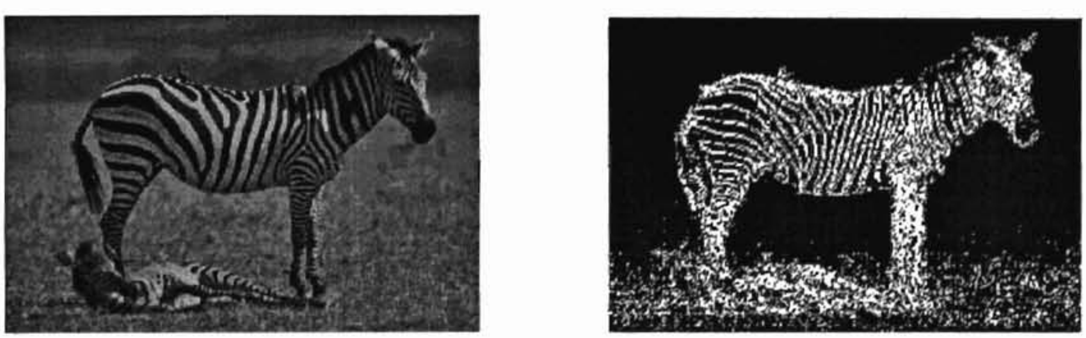
$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} .$$

It can be computed digitally via convolution with the following mask:

0	-1	0
-1	4	-1
0	-1	0

Figure 10: Mask to calculate Laplacian of an image

Since the Laplacian uses the second derivative it is very sensitive to noise and it does not smooth the image like the Sobel or Prewit methods. In the example in figure 11 the same image of zebra that was used before for gradient operators is used to show the edges detected from zero crossings of the Laplacian, the difference is evident. Laplacian does not give better result than the gradient operator.



(a) (b)

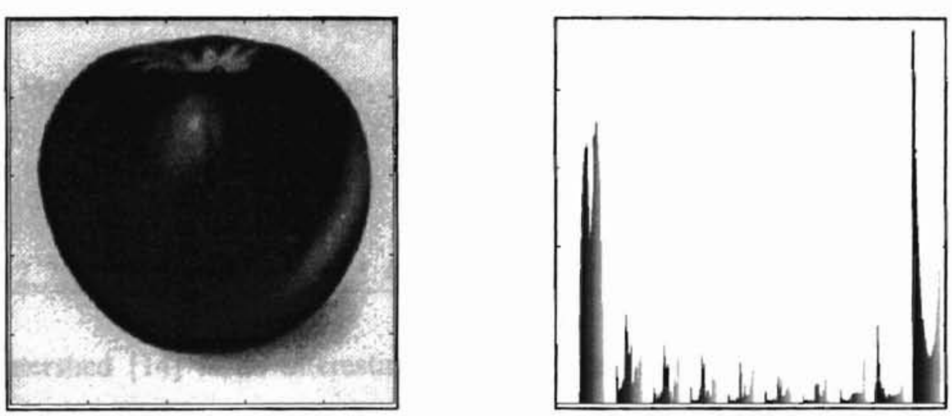
Figure 11. (a) Original image (b) Edge detection using Laplacian

2.2.2 Region Based Segmentation

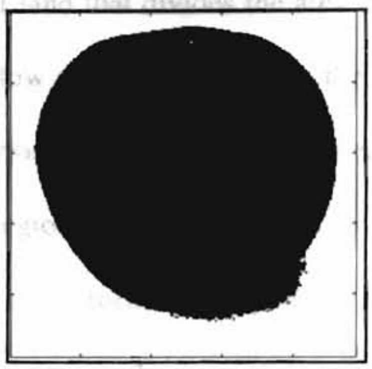
The Region based segmentation techniques look for homogeneity in pixel values to group them together.

2.2.2.1 Thresholding

If an image has only two objects with a marked difference in the gray scale, thresholding can be applied. An example can be a bright object with a dark background or a dark object with a bright background. In this simple case of a dark object on a bright background, thresholding is applied by looking at its histogram and setting the threshold at the valley of the two peaks. These peaks will always exist due to the difference in the gray scale level of the two distinct objects. Figure 12 shows an example of thresholding. Histogram of the image of an apple in figure 12a, is shown in figure 12b, here two peaks are clearly visible. If we apply a threshold of 170 we can divide the image into two parts, apple and background. The result of this thresholding is shown in the figure 12c.



(a) (b)



(c)

Figure 12 (a) Original image of an apple (b) Histogram of the image (c) Segmentation of the image when a threshold of 170 is applied

2.2.2.2 Split and Merge Algorithm

In this algorithm [13] initially whole image is considered one object and then divided into smaller pieces. These pieces are then further divided depending on a given criterion. If a segment meets the criterion it is not divided, else it is divided into two or four segments. This criterion is selected by the user according to the application. It can be the difference between the largest and the smallest gray level intensity, or the variance *etc.* Since the image is divided into square blocks the segmentation that it produces looks blocky and unnatural. Furthermore this algorithm should be fine-tuned according to the application and has high computational complexity.

2.2.2.3 Watershed Algorithm

The watershed [14] is an interesting and effective region based image segmentation algorithm. It considers the gradient of an image as a topographical surface. A topographical surface can be divided into watersheds and catchment basins. A watershed boundary is a high point of land that divides the area drained by one stream from that of another stream. The local low areas of land where these streams collect water are called basins. Similarly if we consider high values of gradient magnitude of an image as high points in a topographical region and dark points as low values, we can cluster all pixels that drain in the same basin to form one segment. This has two main advantages for object based coding. First it gives the result as regions so no extra time is wasted in closing the contours to form regions. Secondly the boundaries are of zero thickness since this algorithm divides the image into objects. In figure 13, catchment basins and watershed lines of the gradient of an image are shown in 3 dimensions as a topographical surface.

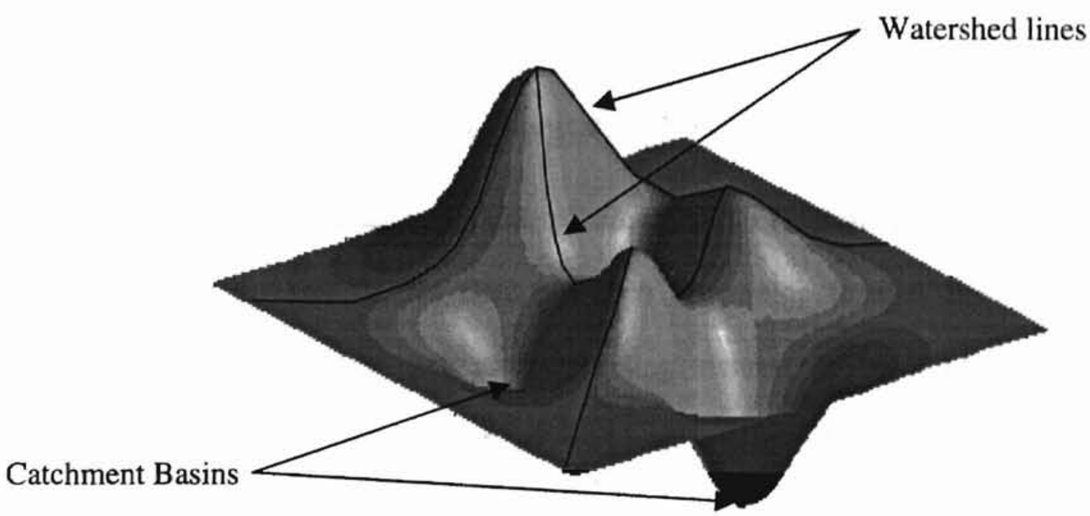


Figure 13: Catchment basins and watershed lines on a segment of a gradient image

The watershed algorithm has a disadvantage of over-segmenting an image. In order to overcome this problem, region merging and multi-resolution watershed pyramids are used. In watershed pyramid, an image pyramid is first obtained by downsampling the image to a predefined coarse level and then applying watershed to the coarsest level. Traversing down the pyramid to the finer level then refines the segmentation obtained while keeping the number of segments the same as before. It is an important branch of watershed and its results will be compared to those of watershed for object-based coding (WOC) in chapter 4.

Watershed algorithm only considers the pixel gray scale values for that reason it gives false edges on textured images. For objects based coding we want to segment an image according to its smoothness and texture. For that reason watershed for object based coding was developed, which tries to eliminate this effect, it shall be discuss in the next chapter.

2.2.2.4 Texture Segmentation by Gabor Filters

Texture segmentation is an important technique in the field of image segmentation. A major goal of research in this field has been to develop features that can represent textures. Different models such as representing textures as amplitude and frequency modulating functions [15] have been proposed. Most of the texture segmentation techniques use two-dimensional Gabor filters to separate areas of a specific texture before further analysis.

Gabor filters are sinusoidally modulated gaussian filters. They have tunable radial frequency and phase and achieve optimal frequency and space resolution. A general 2-D Gabor filter has an impulse function:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\{ -\frac{1}{2}[(\frac{x}{\sigma_x})^2 + (\frac{y}{\sigma_y})^2]\},$$

where $$h(x,y) = g(x,y)\exp[j2\pi(Ux+Vy)].$$

and U and V represent the center frequencies of Gabor filter and σ_x and σ_y are standard deviation in x and y directions.

Figure 14 shows an example of a cosine Gabor filter in the spatial domain.

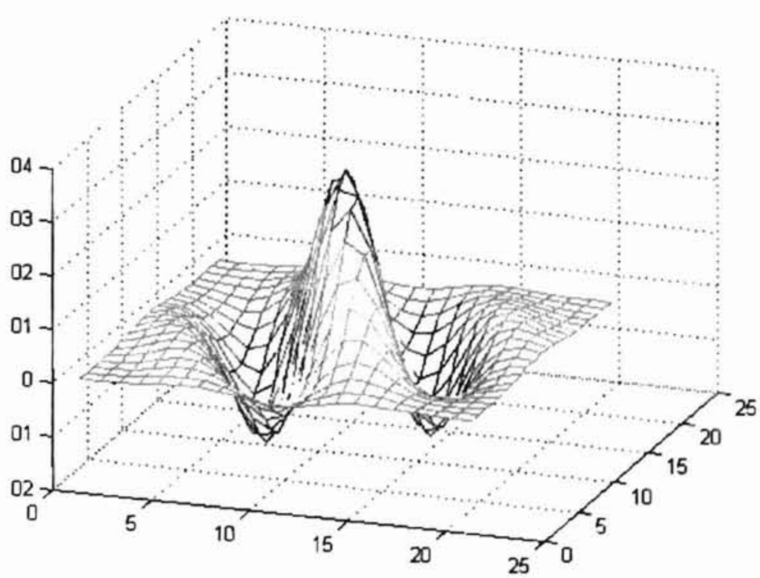


Figure 14: A 2-D Gaussian filter in spatial domain

In a typical texture segmentation technique using Gabor filtering, a bank of Gabor filters with different frequencies and orientations is created. The image is passed through this filter-bank and the outputs are used to segment the image using clustering such as fuzzy c-means clustering [16] or a winner-take-all approach [23].

In this chapter different techniques of image segmentation and compression have been discussed. In the next chapter the segmentation and compression techniques used in the research are explain and analyzed.

CHAPTER III

SEGMENTATION AND COMPRESSION METHODS

3.1 Watershed for Object-Based Coding (WOC) Segmentation

Watershed segmentation only considers the rate of change of the pixel values in an image at a very localized level. In many applications we need to segment the image according to its frequency contents. WOC segmentation does this by first transforming the image into another domain which we shall call difference of discrete cosine transform (DDCT) domain. DDCT for a particular pixel *(i,j)* in an image *I* is found by first taking an 8x8 block around that pixel and finding its DCT coefficients using the following formula:

$$DCT_{i,j}(u,v) = \alpha(u)\alpha(v)\sum_{x=i-3}^{i+4}\sum_{y=j-3}^{j+4} I(x,y)cos[\frac{(2x+1)u\pi}{16}]\,cos[\frac{(2y+1)v\pi}{16}],$$

for u, v = 0, 1, 2, 3, 4, 5, 6, 7.

$$\text{where} \qquad \alpha(u) = \begin{cases} \sqrt{\frac{1}{8}} & \text{for } u=0 \\ \sqrt{\frac{2}{8}} & \text{for } u=1,2,3,4,5,6,7 \end{cases}$$

In the equation u and v are frequency coefficients and $DCT_{i,j}(u,v)$ is the Discrete Cosine Transform of the block around *(i,j)*. These DCT coefficients are then subtracted from the DCT coefficients of the blocks around the neighboring pixels. These DCT differences are weighted and added together. Square root of the result gives us the DDCT.

$$DDCT(i,j)=\sqrt{\begin{array}{l}\sum_{u=0}^{7}\sum_{v=0}^{7}[DCT_{i,j}(u,v)-DCT_{i+1,j}(u,v)]^2\,/W(u,v)+\\ \sum_{u=0}^{7}\sum_{v=0}^{7}[DCT_{i,j}(u,v)-DCT_{i,j+1}(u,v)]^2\,/W(u,v)\end{array}}$$

In the equation *DDCT(i,j)* is the difference of DCT for the pixel *(i,j)* and *W(u,v)* are the weights given to the DCT values. These weights are selected such that high frequency coefficients get lower weights than the low frequency ones so that the resulting sum is more semantically meaningful. The weights used in this thesis are given below:

$$W = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Figure 15 shows these steps for a pixel in a block of a typical grayscale image. Each pixel value in the difference of DCT domain represents a change in both DC and AC in its neighborhood. After we have found the sum of the DCT differences for all the pixels we segment it using the watershed segmentation method.

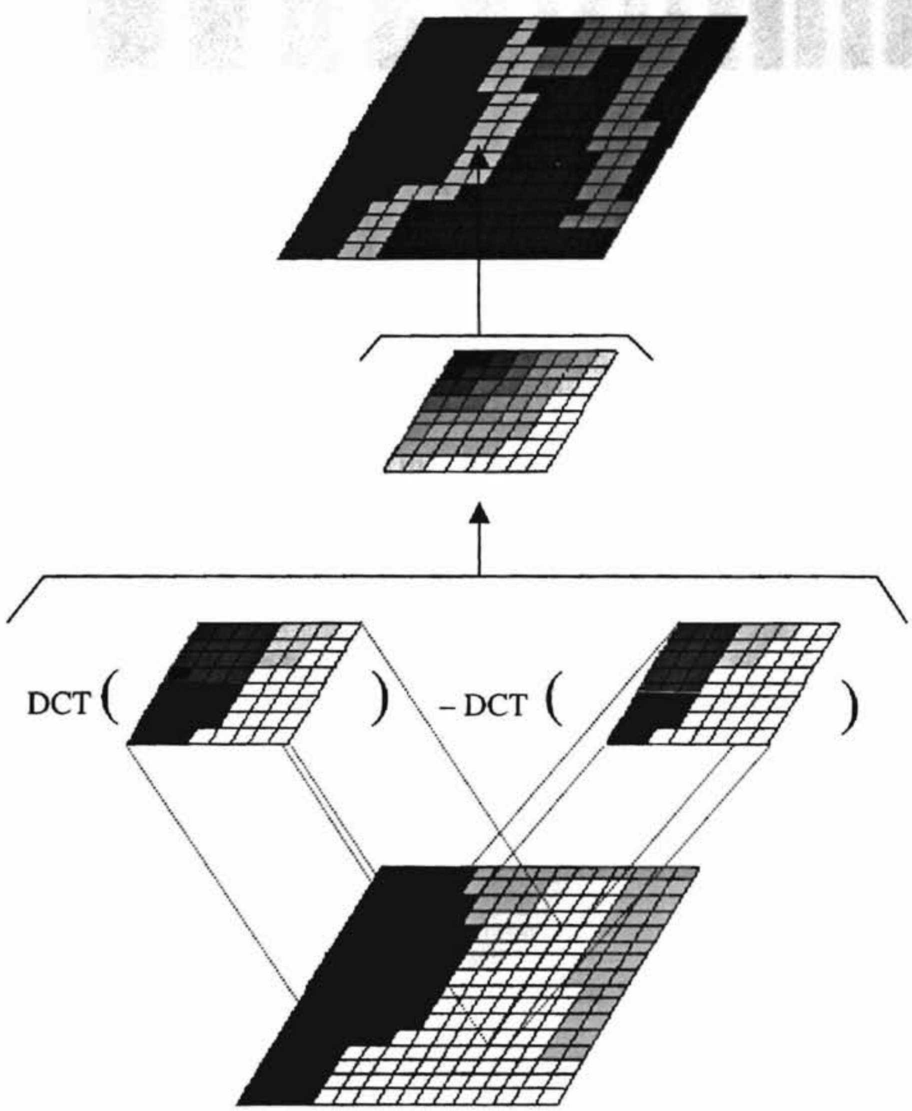


Figure 15: WOC segmentation steps

This algorithm is illustrated below in a simple example of two cosine waves in one dimension.

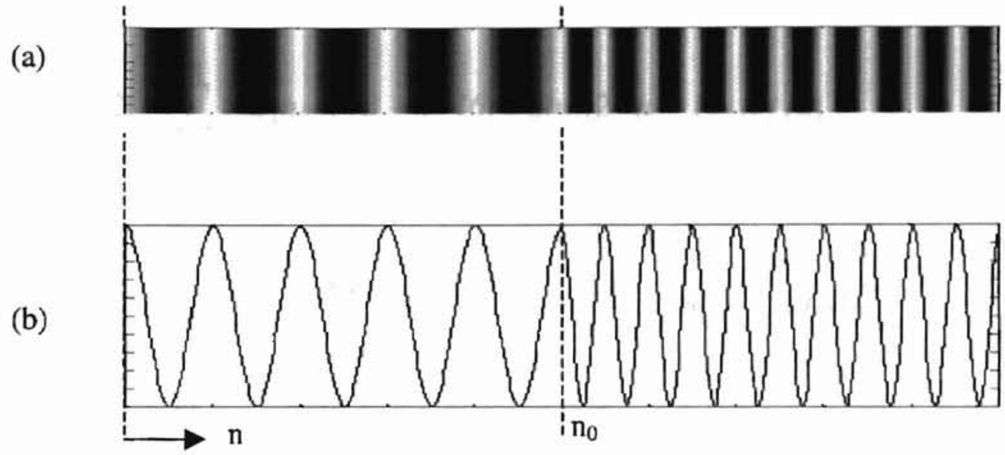


Figure 16: (a) Slice of an image containing only 2 cosine waves in one dimension (b) One row of the image in (a)

We consider a one-dimensional signal composed of two cosine waves of different frequencies. This signal can be represented as

$$cos\left[-\frac{3\pi}{16}+\frac{3n\pi}{8}\right]u[n-n_0]+cos\left[-\frac{\pi}{16}+\frac{n\pi}{8}\right]u[-n+n_0] \qquad \text{where } n > 0$$

Where $u[n]$ is the unit step function and n_0 is the boundary of the two cosines. For simplicity I shall apply a window of length 8.The discrete cosine transform of the window, when it is completely in the low frequency cosine wave region is given by:

$$y[k]=\sum_{n=1}^{8} cos[n\pi/8-\pi/16]*cos[(2n-1)(k-1)\pi/16]$$

This is maximum when k=2 and reduces at other values of k. Lets consider this value of k and see what happens when we slide the window towards right.

$$y[2] = \sum_{n=1}^{8} Cos[n*\pi/8 - \pi/16] * Cos[(2*n-1)*\pi/16]$$

$$y[2] = Cos\left[\frac{\pi}{16}\right]^2 + Cos\left[\frac{3\pi}{16}\right]^2 + Cos\left[\frac{5\pi}{16}\right]^2 + Cos\left[\frac{7\pi}{16}\right]^2 + Cos\left[\frac{9\pi}{16}\right]^2 + Cos\left[\frac{11\pi}{16}\right]^2 + Cos\left[\frac{13\pi}{16}\right]^2 + Cos\left[\frac{15\pi}{16}\right]^2 = 4$$

The value of $y[2]$ reduces as the window enters the high frequency area. The amount reduced is the factor on the right i.e. when it has seven pixels in the low frequency part and one pixel in the high frequency part its value is:

$$y[2] = \sum_{n=1}^{7} (Cos[n*\pi/8 - \pi/16] * Cos[(2n-1)\pi/16]) + \sum_{n=8}^{8} (Cos[3*n*\pi/8 - 3*\pi/16] * Cos[(2n-1)\pi/16]) = 3.853$$

At half way through $y[2]$ reduces to

$$y[2] = \sum_{n=1}^{4} (Cos[n*\pi/8 - \pi/16] * Cos[(2n-1)\pi/16]) + \sum_{n=5}^{8} (Cos[3*n*\pi/8 - 3*\pi/16] * Cos[(2n-1)\pi/16]) = 2.$$

Since two consecutive values of $y[2]$ will be subtracted we only consider the absolute difference. Figure 17 shows a plot of the difference as we slide the window towards right.

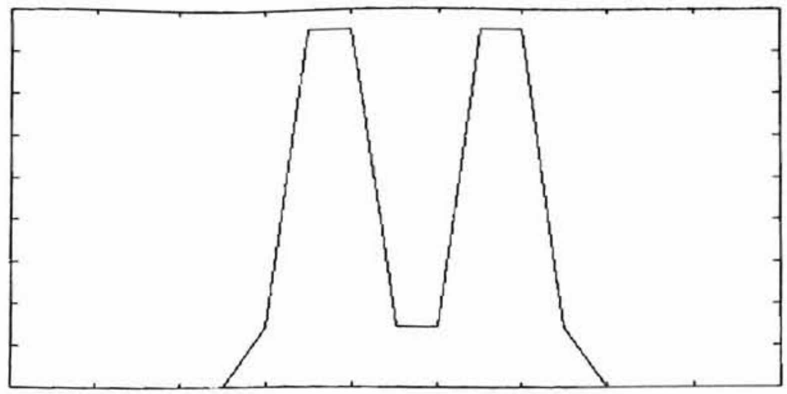


Figure 17: Plot of the difference between two consecutive sum of DCT

If we smooth this difference with a gaussian filter of the same length as the length of the window shown below we get a peak in the center.

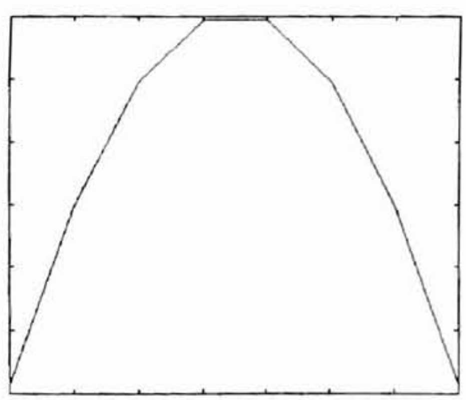


Figure 18: Gaussian filter

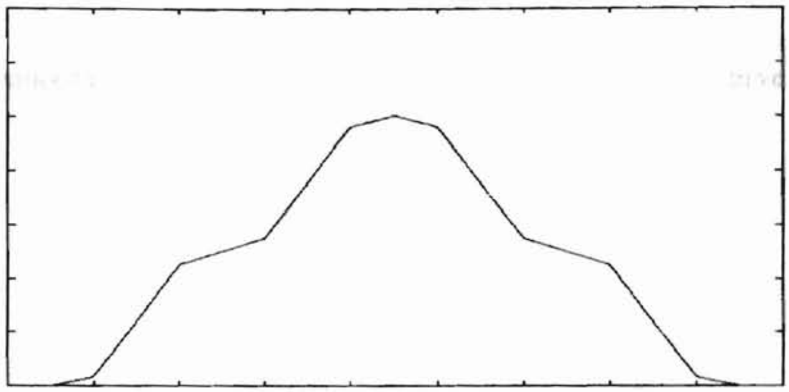


Figure 19: Result of applying Gaussian filter of figure 18 on the difference in figure 17.

Similarly if we consider the case when $k = 4$ and do the same analysis above we start with a very small $y[4]$.

$$y[4] = \sum_{n=1}^{8} (\mathrm{Cos}[n\pi / 8 - \pi / 16].\mathrm{Cos}[3(2n - 1)\pi / 16]) = -2.22045 \times 10^{-16}.$$

This value increases as the window enters the high frequency region. A plot of the difference between two consecutive values at this frequency is given below

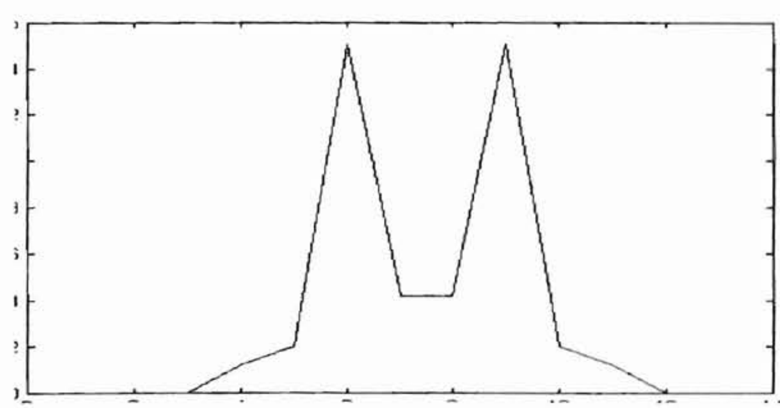


Figure 20: Plot of the difference between two consecutive sum of DCT values

When we smooth the difference with a Gaussian filter we get a curve similar to the one for k=2.

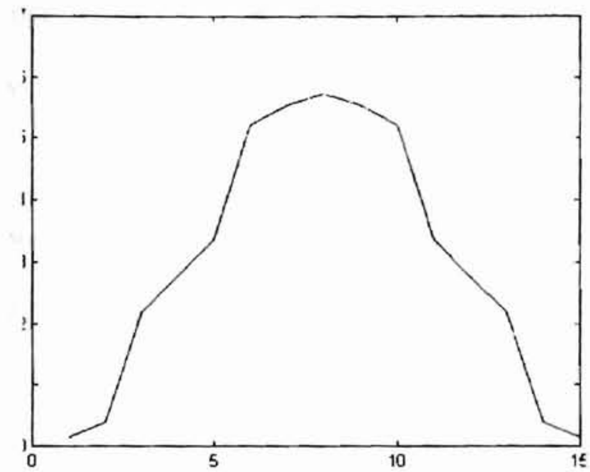


Figure 21: Result of applying Gaussian filter of figure 18 on the difference in figure 20.

One drawback in watershed segmentation is that it tends to over segment an image this drawback can be removed by region merging after watershed.

3.2 Region Merging

The value of the difference of DCT is always high near the edges and low within an object. This property can be used to design a region merging algorithm. Watershed algorithm always computes a basin for each region, this basin is the point in the object where the DCT difference is the minimum *i.e.* the region around it, is mostly uniform. Straight lines can be drawn from the basin of one region to that of its neighboring regions and checked for the height that it reaches. Figure 22 shows how straight lines are drawn between basins of different regions

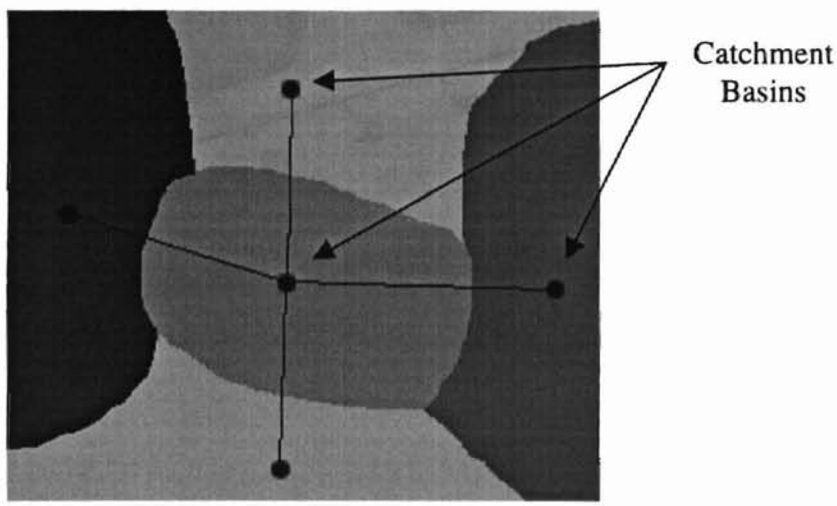


Figure 22: Straight lines connecting basins of adjacent regions

Figure 23 shows a typical boundary between two adjacent regions and the path that is drawn in between them. The path on which the DCT difference does not reach a predefined height is noted down and later these two regions are merged together.

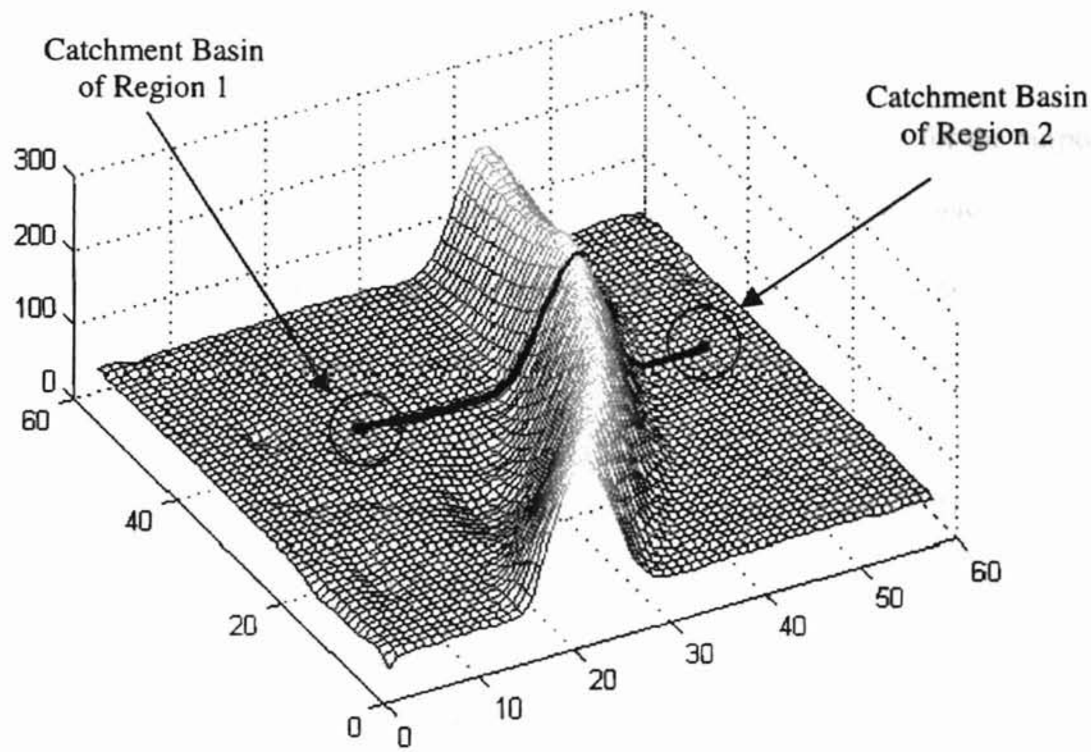


Figure 23: Straight line connecting basins of two adjacent regions

3.3 Object Based Coding

Different object-based coding methods are discussed in the chapter 3, for the purposes of this thesis an object based compression method has to be chosen that is easy to use *i.e.* its ability to use the preexisting algorithms and gives higher compression ratio for the same segmentation. Object based coding method used was shape independent coding with projection onto convex sets. Every object in an image is compressed separately. A rectangular block is used to provide boundary to the segmented object. This block is now further divided into 8 x 8 sub-blocks. Three types of sub-blocks are possible. First, sub-blocks that are not part of the object are ignored (not coded). Second, sub-blocks where all the constituent pixels are part of the object, these blocks are coded in a manner similar to the standard JPEG using the discrete cosine transform (DCT) and quantization. The third type of sub-blocks contain segment boundary pixels requiring extra processing to be able to improve compression quality. These sub-blocks include pixels that are not part of the object, giving us extra information to compress. Since these pixel values will not be used after decoding the sub-block, it gives us the freedom to set their values in such a way that would reduce the number of DCT coefficients required to code them. For this purpose, successive projection onto convex sets (POCS) is applied on these sub-blocks. A simple example below shows how POCS reduces the number of coefficients in a DCT than zero-padding.

We take a 2x2 image with two pixels belonging to the object and two are outside pixels. We shall use this 2x2 image to show the objective of preprocessing in object based coding. Let the image matrix be $\begin{bmatrix} 5 & 0 \\ 10 & 0 \end{bmatrix}$ where the left two pixels are part of the object and the right two pixels are not, the DCT of this matrix is $\begin{bmatrix} 7.5 & 7.5 \\ -2.5 & -2.5 \end{bmatrix}$ The DCT contains 4 nonzero coefficients while matrix itself contains 2 valid pixels. Now if we apply projection onto convex sets on the matrix $\begin{bmatrix} 5 & 5 \\ 10 & 10 \end{bmatrix}$ we get $\begin{bmatrix} 5 & 2.5 \\ 10 & -2.5 \end{bmatrix}$ if we take its discrete cosine transform we get $\begin{bmatrix} 7.5 & 7.5 \\ 0 & -5 \end{bmatrix}$ here we have three non-zero coefficients reducing the number of coefficients needed to send.

After applying POCS on the third type of sub-blocks they are also compressed in a way similar to JPEG *i.e.* by quantizing the DCT coefficients.

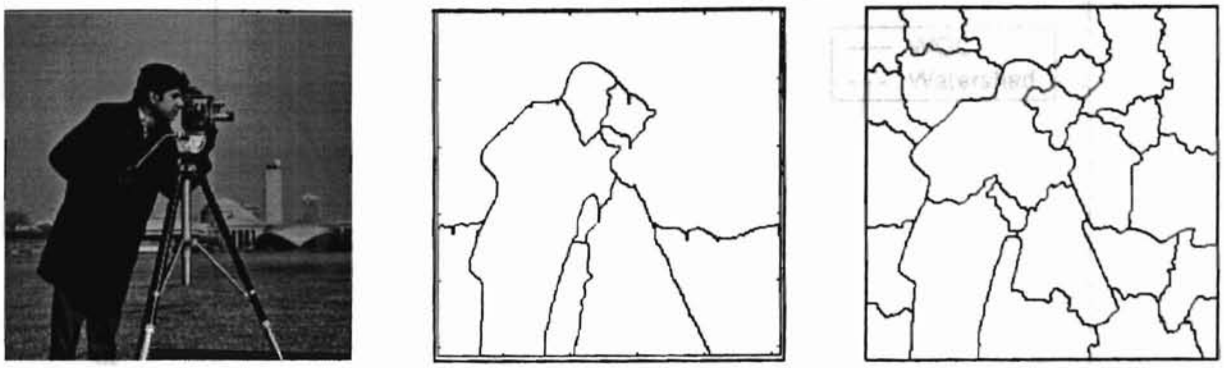
In this chapter the segmentation and compression method that was used in the research were discussed the next chapter gives the results of applying these methods on real and synthetic images.

CHAPTER IV

RESULTS

In this chapter results of WOC segmentation technique when applied on real and synthetic images will be presented. These results will be compared with other segmentation techniques namely watershed segmentation and texture segmentation using Gabor filters. Watershed segmentation is used for images that do not contain textures, texture segmentation is achieved using Gabor filters. Texture segmentation techniques fail to segment smooth images with little or no texture. WOC segmentation can be used for both kinds of images, it successfully segments both smooth, real images and textures. This segmentation ability of WOC segmentation is explained below with detailed comparisons with watershed and Gabor segmentation

WOC segmentation and watershed segmentation methods were used to segment 3 images namely 'cameraman', 'swan' and 'peppers'. Both of these methods suffer from over segmentation, which is removed by region merging. The results of these segmentation methods on the 'cameraman' image are shown in figure 24.



(a) (b) (c)

Figure 24: (a) cameraman (b) WOC segmentation (c) watershed segmentation

When cameraman image was object based coded using the two segmentation methods the resultant compression ratio, mean square error and CPU time taken to compress it are noted below in the table 1.

	Compression Ratio	PSNR	CPU time (sec)
WOC	5.04	31.76	183
Watershed	4.73	31.80	251

Table 1: Example results for 'cameraman'

Figure 25 shows rate-distortion curve of the two segmentation methods. As you can see WOC gave higher PSNR value for a fixed compression ratio compared to watershed. This difference is around 1DB at high compression ratios, which is markedly better.

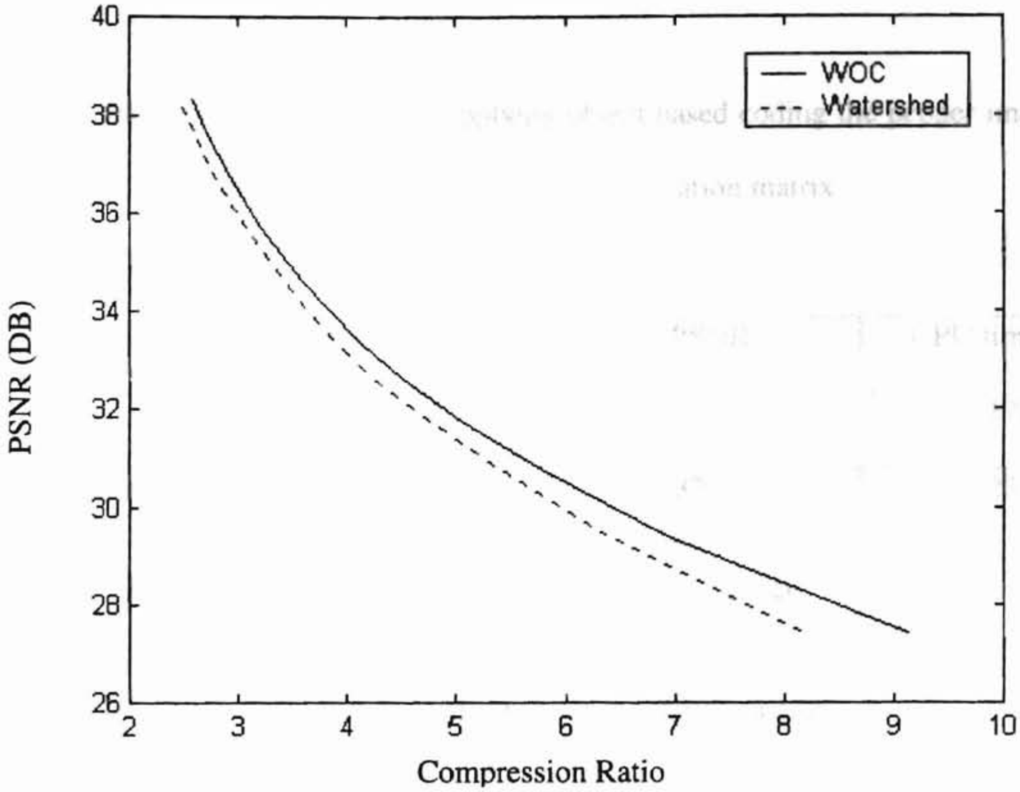


Figure 25: Rate-distortion curve for the image 'cameraman'

Figure 26 shows the results of WOCsegmentation and watershed segmentation on the image 'peppers'.

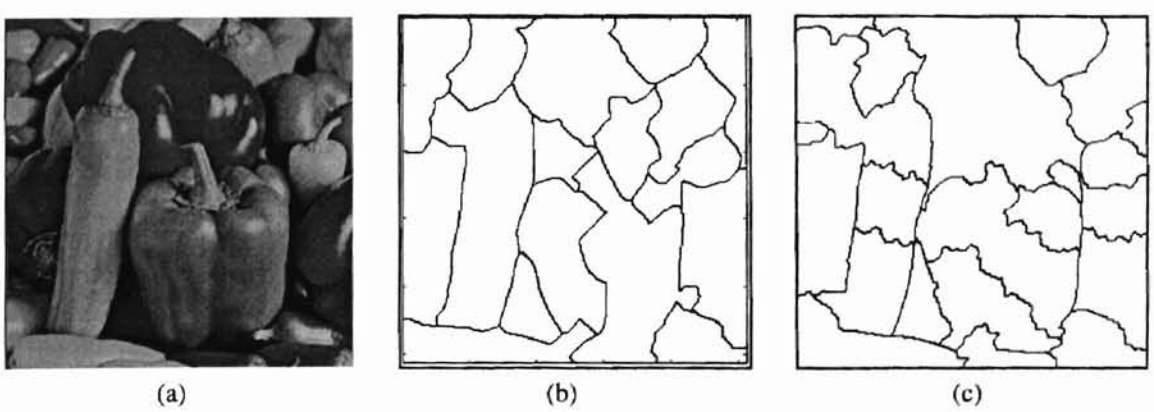


Figure 26: (a) Peppers (b) WOC segmentation (c) watershed

Table 2 gives an example result of applying object based coding the pepper image using the two segmentation methods using the same quantization matrix.

	Compression Ratio	PSNR	CPU time (sec)
WOC	4.53	32.42	172
Watershed	4.37	32.58	230

Table 2: Example results for 'peppers'

Figure 27 shows the rate-distortion curve for the image 'peppers'.

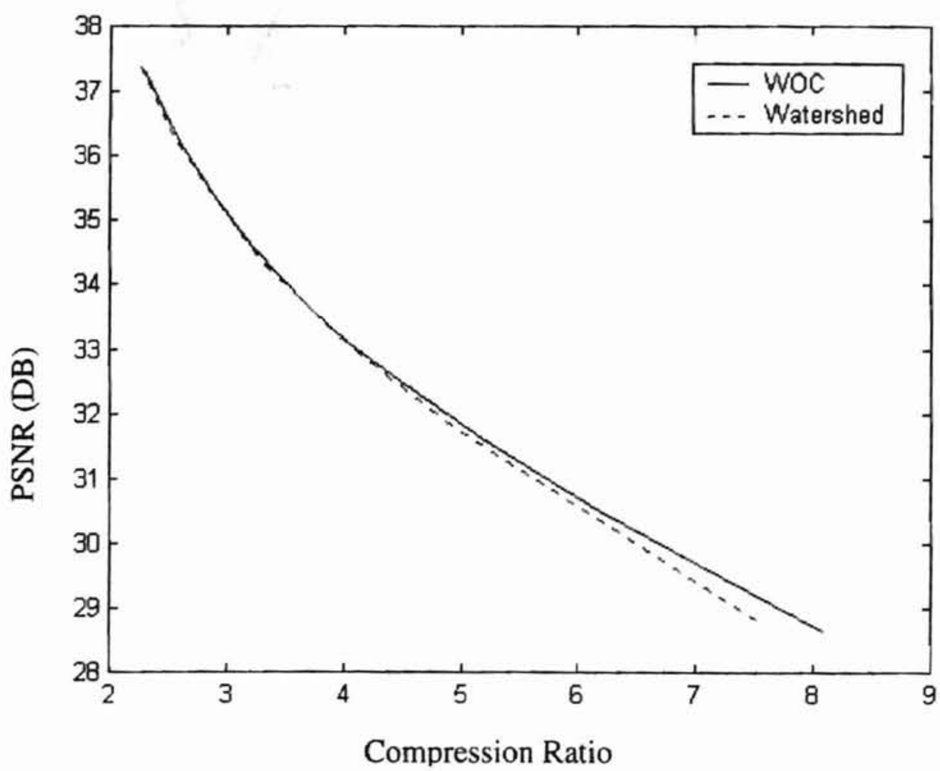
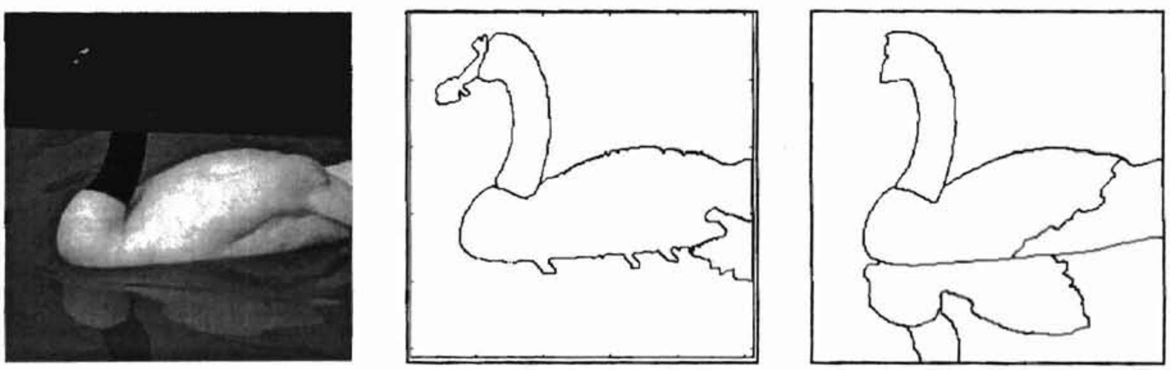


Figure 27: Rate-distortion curve for the image 'peppers'

For this image both of the segmentation methods have performed equally good. This is because this image does not contain textures and has smooth areas of different gray scale levels. Thus the performance of WOC is as good as the watershed but supersedes watershed for images that contain textures.

The third result is of the image 'swan'. The segmentation results, demonstrative example and rate distortion curves are given in figure 28, table 3 and figure 29 respectively. Here also the WOC segmentation method gives higher PSNR values for the same compression ratios. The difference is greater at higher compression ratios than at lower compression ratios. This proves that WOC is a better segmentation method for object based coding than watershed especially at high compression ratios.



(a) (b) (c)

Figure 28: (a) Swan (b) WOC segmentation (c) watershed

	Compression Ratio	PSNR	CPU time
WOC	6.46	36.54	181
watershed	6.21	36.52	145

Table 3: Example results for 'swan'

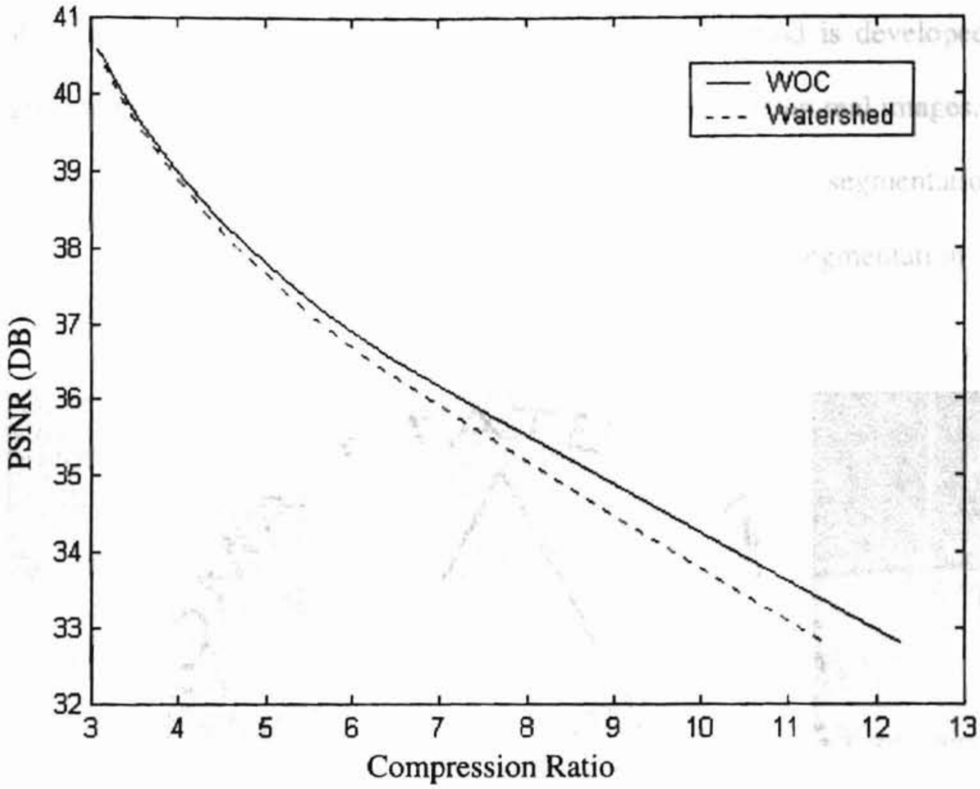
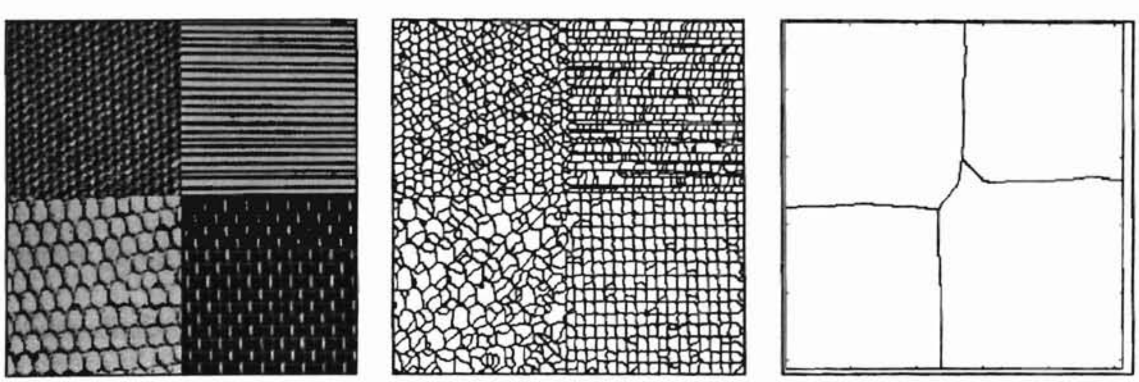


Figure 29: Rate-distortion curve for the image 'Swan'

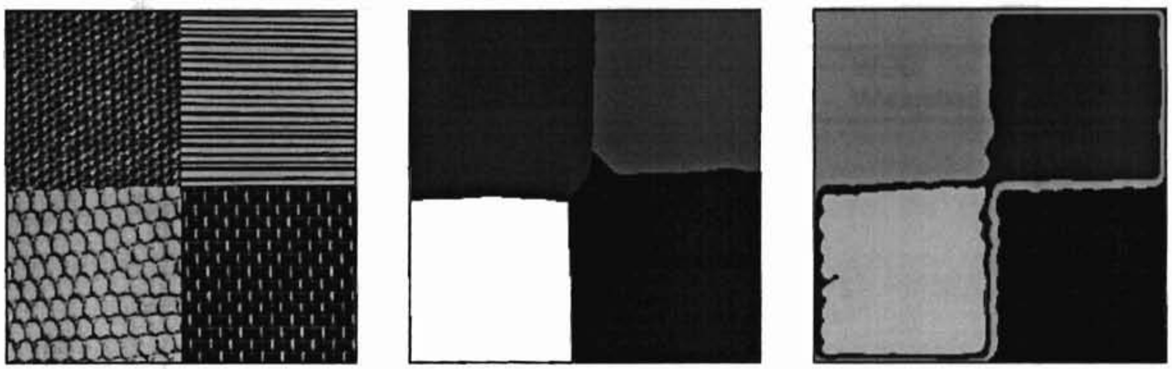
Watershed segmentation is not used for images with pure textures, since it produces false edges and so does not give acceptable results. To prove this point watershed was applied on a texture mosaic that was created using four Brodatz textures [22]. WOC was also applied on the same image and the results are given in figure 30.



(a) (b) (c)

Figure 30: (a) Original texture mosaic (b) watershed segmentation (c) WOC segmentation

Gabor segmentation is used for texture images. This method is developed for texture segmentation so does not give perceptually acceptable results on real images. To compare the performance of WOC segmentation and Gabor texture segmentation the same Brodatz texture mosaic was used. The results of the two segmentation methods are shown in figure 31.



(a) (b) (c)

Figure 31: (a) Original texture mosaic (b) WOC segmentation
(c) Gabor segmentation

	Compression Ratio	PSNR	CPU time
WOC	2.89	27.58	57
Watershed	2.63	27.60	160

Table 4: Example results for 'texture mosaic'

Table 4 gives an example result of object based coding the texture mosaic. Figure 32 is the rate-distortion curve of the two segmentation methods for the texture mosaic.

The result shows that WOC segmentation gave higher PSNR for the same compression ratio and the difference is higher than 1DB. Also the segmentation itself is more semantically meaningful.

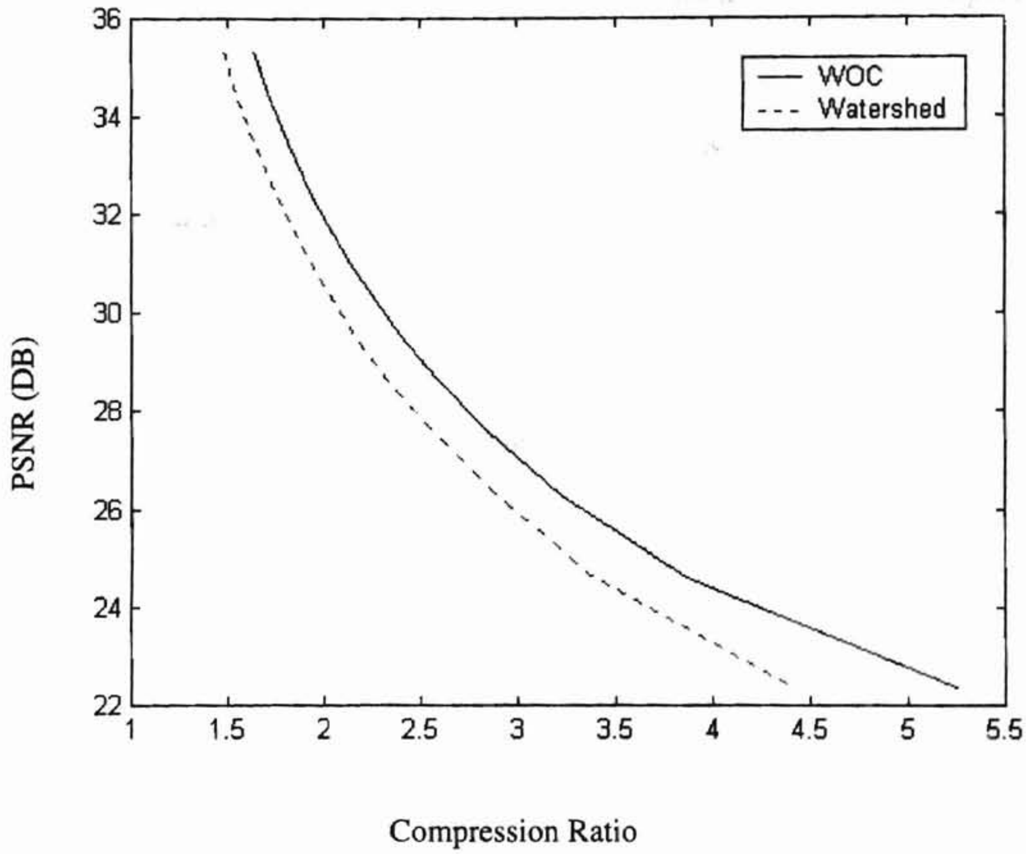


Figure 32: Rate distortion curves for the texture mosaic image

CHAPTER V

CONCLUSION

Results proved that, in general, watershed for object based coding (WOC) segmentation method segments an image such that when its object-based coded it gives higher PSNR for the same compression ratio, compared to watershed. This is true for images that contain texture (around 1DB improvement) and for images that do not contain textures and are composed of smooth gray level regions this method matches the performance of watershed at low compression ratios and exceeds them at high compression ratios. Another improvement of WOC over watershed is that it is able to segment different textures while the same is not possible with the classical watershed. The reason is that a classical watershed algorithm looks at the rate of change in the immediate vicinity of every pixel while WOC looks at the rate of change in a whole block, *e.g.* 8x8 in the examples.

Textures are segmented using Gabor segmentation methods. For the case of Brodatz texture mosaic in Figure 28(a) WOC gives perceptually better result than Gabor segmentation method also when object based coded the segmentation gives higher PSNR for the same compression ratio. The difference in the results of the two segmentations is higher that 1DB.

Another purpose of the thesis was to introduce object based coding as a method for quantitative comparison of two segmentation methods. When a segmented image is object-based coded quantization can be changed giving us different compression ratios.

This way rate-distortion curves can be generated. These rate distortion curves show how well the results of a segmentation method performed at different compression ratios.

Thus this thesis has fulfilled its two main objectives namely, introduce a new segmentation method that is tuned towards object based coding and introduce object based coding as a measure of segmentation quality.

References

1. Mahmoud R. El-Sakka and Mahamed S. Kamel *"Adaptive image Compression Based on Segmentation and Block Classification,"* Proc. Intl. Conf. On Image Processing, vol.2, pp. 555-559 IEEE 1998.
2. Jacques Vaisey and Allen Gersho *"Image Compression with Variable Block Size Segmentation,"* IEEE Trans. on Signal Processing, vol 40 8, pp. 2040–2060, Aug 1992.
3. Thomas Sikora and Bela makai *"Shape Adaptive DCT for Generic Coding of Video,"* IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5 1, pp. 59–62, Feb 1995.
4. B I Mi and Cham Wai Kuen AND Zheng Zhi Hang *"Discrete Cosine Transform On Irregular Shape For Image Coding,"* Proc. TENCON '93. IEEE Conf. (Beijing), vol.3, pp. 402–405. Oct 1993.
5. Hiroyuki Katata, Norio Ito, Tomoko Aono and Hiroshi Kusao, *"Object Wavelet Transform for coding of Arbitrarily Shaped Image Segments"*, IEEE Trans. on Circuits and Systems for Video Technology Vol. 7 1, pp. 234-237, Feb 1997.
6. John G. Apostolopoulos and Jae S. Lim *"Representing Arbitrarily Shaped Regions: A Case Study of Overcomplete Representation"* Proc. Intl. Conf. on Image Processing, vol.2, pp. 426–429, IEEE 1995.
7. Andre Kaup *"Object-Based Texture Coding of Moving Video in MPEG-4,"* IEEE Trans. on Circuit and Systems for Video Technology, vol. 9 1, pp. 5-15, Feb. 1999.

8. Homer H. Chen, M. Reha Civanlar and Barry G. Haskell *"A Block Transform Coder for Arbitrarily Shaped Image Segments"* Proc. ICIP-94 Vol. 1, pp. 85-89 IEEE Nov 1994.

9. Andre kaup and Til Aach *"Coding of Segmented Images Using Shape-Independent Basis Functions,"* IEEE Trans. on Image Processing Vol. 7 7, pp. 937-947, July 1998.

10. Sobel, I.E., *"Camera Models and Machine Perception"*, AIM-2, Stanford Artificial Intelligence Lab, Palo Alto. 1970.

11. Prewitt, J., *"Object Enhancement and Extraction,"* Picture Processing and Psychopictorics, edited by B. Limpkin and A Rosenfield, Academic Press, New-York, pp. 1674-75. 1970.

12. Gonzalez, R.C. and R.E. Woods, "Digital Image Processing", Addison Wesley Publishing Co, New York. 1993.

13. T. V. Robertson, *"Extraction and classification of objects in multisopectral images,"* Machine Processing of Remotely Sensed Data, IEEE 73 CHO 837-2GE, Purdue University, West Lafayette, Indiana Oct 16-18, 1973, pp 3B-27-3B34

14. Gauch, J.M.; Pizer, S.M., *"Multiresolution analysis of ridges and valleys in grey-scale images,"* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol: 15 6, June 1993 , pp: 635 –646.

15. Pattichis, M.S.; Christodoulou, C.I.; Pattichis, C.S.; Bovik, A.C., *"Non-stationary texture segmentation using an AM-FM model,"* Intl Conf. on Neural Networks, vol: 3, 1997 , pp: 1552 –1557.

16. Saito, T.; Kudo, H.; Suzuki, S. *"Texture image segmentation by optimal Gabor filters,"* 3rd Intl Conf. on Signal Processing vol: 1 , 1996 , pp: 380 –383.

17. Chung-Tao Chu, Dimitris Anastassiou, Shih-Fu Chang *"Bi-Directional Object-Based Coding in Video Compression at Very Low Bitrate,"* 3rd Intl. Conf. on Signal Processing, vol.2, pp. 986–989. ICSP' 96.

18. Dam LeQuang and Andre Zaccarin *"A New Object-Oriented Approach for Video Compression at Very Low Bit-Rate,"* Canadian Conf. on Electrical and Comp. Eng. vol.2, pp. 823–826, IEEE 1995.

19. Ostermann, J., Jang, E.S., Jae-Seob Shin, Chen, T. *"Coding of Arbitrarily Shaped Video objects in MPEG-4,"* Intl. Conf. On Image Processing, vol.1, pp. 496-499, IEEE 1997.

20. Andre Kaup and Til Aach "Segment-Oriented Coding of Textured Images Based on Successive Approximation", Intl. Symp. on Speech, Image Processing and Neural Networks, vol.1, pp. 197-200, IEEE 1994.

21. Ujjaval Y Desai, *"DCT and Wavelet Based Representations of Arbitrarily Shaped Image Segments,* " Intl. Conf. On Image Processing, vol.1 pp. 558-561, IEEE 1995.

22. P. Brodatz, *"Textures: A Photographic Album for Artists and Designers,"* New York: Docer, 1966.

23. Bovik, A.C.; Gopal, N.; Emmoth, T.; Restrepo, A., *"Localized measurement of emergent image frequencies by Gabor wavelets,"* IEEE Trans. on Information Theory, Vol: 38 2 2, pp: 691 –712, , March 1992.

VITA 2

Junaid Ahmed

Candidate for the Degree of

Master of Science

Thesis: IMAGE SEGMENTATION FOR OBJECT BASED IMAGE COMPRESSION

Major Field: Electrical Engineering

Biographical:

Education: Bachelor of Science Degree in Electrical and Electronics Engineering from Middle East Technical University, Ankara Turkey in June 1997. Completed the requirements for the Master of Science Degree at Oklahoma State University in July 2000.

Professional Experience: Graduate Research Assistant, Oklahoma Imaging Laboratory School of Electrical and Computer Engineering, Oklahoma State University, June 1999 to June, 2000. Software Engineer, Crescent Software, Islamabad, Pakistan. Feb 1998 to Dec 1998.

Professional Memberships: Institute of Electrical and Electronics Engineers (IEEE), Eta Kappa Knu Electrical Engineering Honor Society.