Segmentation Of Textured Images
Using Gibbs Random Fields

By

Naji R. El-Hayek

A Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Department of Computer Science

Faculty of Natural and Applied Science

Notre Dame University – Louaize

Zouk Mousbeh, Lebanon

July 2000
Segmentation Of Textured Images
Using Gibbs Random Fields

By
Naji R. El-Hayek

Approved:

Khaldoun El-Khaldi: Assistant professor of Computer Science.
Advisor.

Fouad Chedid: Associate professor of Computer Science and Chairperson.
Member of Committee.

Hoda Maalouf: Assistant professor of Computer Science.
Member of Committee.

Amer Jajour: Assistant professor of Mathematics.
Member of Committee.

Date of thesis defence: July 6, 2000
ACKNOWLEDGEMENT

First, I would like to thank my Advisor Dr. Khaled El-Khaldi for his consistent support, and for being helpful in encouraging me and providing me with the correct information throughout a whole year in my studies and research.

Second, I would like to thank Miss Nadine Haddad for help in providing me, through the A.U.B. library, the needed books and articles.

Third, I am also thankful for the help I received from Mr. Armen Balian and family, Mr. Pierre Salim, Mr. Ephram Houjeily who have given of themselves and their time to help me in achieving my work.

Moreover, I also appreciate the help and support I have received from Mrs. Ben Saadeh (Mathematics Instructor).

Finally, I would like to express my deep feelings for the support and love of my parents Ramez and Aida, my brother Ziad and my sister Sandra that carried me through my whole life.
TABLE OF CONTENTS

LIST OF FIGURES........................................................................................................ vi

ABSTRACT................................................................................................................... viii

1. INTRODUCTION......................................................................................................1

1.1. OVERVIEW.........................................................................................................1
1.2. IMAGE DEFINITION............................................................................................1
  1.2.1. Digital Image................................................................................................1
  1.2.2. Formation.....................................................................................................2
  1.2.3. Restoration..................................................................................................2
  1.2.4. Frequency....................................................................................................3
  1.2.5. Presentation................................................................................................3
  1.2.6. Storage........................................................................................................4
  1.2.7. Digital Image Compression...........................................................................5

2. SEGMENTATION......................................................................................................6

2.1. INTRODUCTION...................................................................................................6
2.2. POINT DETECTION...........................................................................................8
2.3. LINE DETECTION..............................................................................................9
2.4. EDGE DETECTION............................................................................................10
  2.4.1. Gradient Operators....................................................................................10
  2.4.2. Laplacian ..................................................................................................12
  2.4.3. Combined Detection..................................................................................12
2.5. BOUNDARY DETECTION................................................................................13
  2.5.1. Local Processing.......................................................................................13
  2.5.2. Hough Transform.....................................................................................14
  2.5.3. Graph Theoretic Technique.......................................................................16
2.6. REGION BASED APPROACH...........................................................................18
  2.6.1. Region merging and splitting.....................................................................18
  2.6.2. Thresholding.............................................................................................22

3. TEXTURE ANALYSIS..............................................................................................24

3.1. INTRODUCTION..................................................................................................24
3.2. TEXTURE...........................................................................................................25
  3.2.1. Definition...................................................................................................25
3.3. METHODS OF TEXTURE................................................................................26
  3.3.1. Statistical Method....................................................................................27
  3.3.2. Structural Method....................................................................................30
  3.3.3. Spectral Method.......................................................................................31
  3.3.4. Multiscale Method....................................................................................32
    3.3.4.1. Wavelet Multiresolution Analysis.......................................................33

4. TEXTURAL SEGMENTATION..................................................................................36

4.1. INTRODUCTION..................................................................................................36
4.2. TEXTURE AND SEGMENTATION ................................................. 37
4.3. TEXTURAL SEGMENTATION ...................................................... 39
    4.3.1. Quad Tree Segmentation ............................................... 41
    4.3.2. Gabor wavelet scale for Unsupervised Texture Segmentation ...... 42
    4.3.3. Multiresolution Analysis For Feature Extraction Used For Texture Segmentation .......................................................... 46

5. IMPLEMENTATION OF THE ALGORITHM ...................................... 57

5.1. INTRODUCTION ....................................................................... 57
5.2. GIBBS DISTRIBUTION .............................................................. 58
    5.2.1. Definition ........................................................................... 58
    5.2.2. Class of Gibbs Distribution ................................................. 60
5.3. HIERARCHICAL IMAGE MODELING ........................................... 61
    5.3.1. Hierarchical Gibbsian Model For Image Data ......................... 61
    5.3.1.1. High Level Process ............................................................. 61
    5.3.1.2. Low Level Process .............................................................. 62
    5.3.2. Segmentation ...................................................................... 62
    5.3.3. Parameter Estimation ............................................................ 63
5.4. MAP ALGORITHM ................................................................. 63
    5.4.1. Textured Region .................................................................. 64
5.5. ESTIMATION OF PARAMETERS IN GIBBS DISTRIBUTION .......... 66
    5.5.1. Technique for Estimation ..................................................... 66
5.6. SIMULATION RESULTS ............................................................. 68

6. CONCLUSION .............................................................................. 71

BIBLIOGRAPHY ............................................................................ 73
# LIST OF FIGURES

**Figure**

1. Image Storage ........................................................................................................ 4
2. Mask for isolated points ....................................................................................... 9
3. Masks respond to different line orientations ...................................................... 9
4. Gradient operators ................................................................................................. 11
5. Sobel Edge Detector Masks .................................................................................. 11
6. Complete gradient image ...................................................................................... 12
7. Hough Transformation ......................................................................................... 15
8. Hough Transform (Polar representation) .............................................................. 15
9. Graph theoretic technique .................................................................................... 17
10. Region merging .................................................................................................... 19
11. Quad-tree representation of a binary image ....................................................... 21
12. Texture examples ................................................................................................ 27
13. Structural Method ................................................................................................ 31
14. Wavelet decomposition of 2D image .................................................................. 33
15. Pyramidal and tree structure Wavelet Decomposition ....................................... 34
16. Textural segmentation of a sitting room .............................................................. 40
17. Textural segmentation of an image of the thorax ............................................... 40
18. Quad Tree representation ..................................................................................... 41
19. Gabor wavelet representation ............................................................................. 43
20. Segmentations for two artificial images .............................................................. 44
21. Gabor wavelet scale algorithmic pipeline ......................................................... 45
22. Multiresolution wavelet transform ...................................................................... 47
23. Image and its pyramid-structured wavelet transform ......................................... 48
24. Texture image shown with its pyramid-structure wavelet transform .................. 48
25. Tree structured wavelet transform ..................................................................... 49
26. Cluster Analysis ......................................................................................... 51
27. Image segmentation using multiresolution analysis ............................... 52
28. Test image generated by combining microtextures ................................ 53
29. Segmentation process of the image containing microtextures .............. 54
30. Segmentation using Markov random field ............................................. 56
31. Neighborhoods and clique types ............................................................ 59
32. Representation of 333 block .................................................................. 66
Abstract

In this thesis, digital image is introduced as an amount of data, which is produced when a 2-D light intensity function is sampled and quantized to create a digital image. Its principle objective is to define the segmentation process, in which we can partition an image into meaningful regions that correspond to part of, or the whole of, objects within a scene. This is done by systematically dividing the whole image up into its constituent areas or regions. If the regions do not correspond directly to a physical object, or object surface, then they should correspond to some area of uniformity. Major approaches to segmentation have been introduced.

However, since texture plays an important role in image analysis and understanding. As a front end in a typical classification system, texture feature extraction is of key significance to the overall system performance. There have been many papers, proposing different approaches to this problem. This thesis also outlines the basic methods of texture analysis and comments on different issues. It also point to the connections of related methods.

Texture segmentation is also introduced. It is an interesting but difficult problem in image processing. The main difficulty of traditional texture segmentation is the lack of adequate tools to characterize different scales of texture effectively. Recent developments and researches help to overcome this difficulty. We present many approaches such as the Quad Tree Segmentation, Gabor wavelet scale for
Unsupervised Texture Segmentation, Unsupervised Texture Segmentation Using Multiresolution Analysis For feature Extraction, and Markov Random Field Model and Segmentation. The textural features are extracted from each decomposed image. The procedures results in a segmented image whose regions are distinct from one another with respect to texture characteristic content.

Finally, a new approach was presented to the use of Gibbs distribution (GD) for segmentation of textured images. Specifically, random field models were presented for textured image data based upon a hierarchy of GD. Then the dynamic programming based segmentation algorithms for textured images were presented, considering a statistical maximum a posteriori (MAP) criterion. Since the model parameters are needed for the segmentation algorithms, a new parameter estimation technique is developed for estimating the parameters in a GD.
CHAPTER 1

INTRODUCTION

1.1. OVERVIEW

Interest of scientists in digital image processing methods stems from two principle application areas: improvement of pictorial information for human interpretation, and processing of scene data for autonomous machine reception. Digital image processing techniques now are used to solve a variety of problems, commonly require methods capable of enhancing pictorial information for human interpretation and analysis.

While the second major area of application of digital image processing techniques is solving problems dealing with machine reception. That is, the scientists focus on procedure for extracting from image information in a form suitable for computer processing. This information uses little resemblance to visual features that human beings use in interpreting the content of an image. Example of the type of information used in machine reception is statistical moments, Fourier transform coefficients, and multi dimensional distance measures.

1.2. IMAGE DEFINITION

The term image refers to a two-dimensional intensity function \( f(x,y) \), where \( x \) and \( y \) denote coordinates and the value of \( f \) at any point \((x,y)\) is proportional to the brightness or gray level of the image at that point. Sometimes viewing an image in perspective with the third axis being brightness is useful.
1.2.1. Digital Image

A digital image is an image $f(x,y)$ that has been discretized in spatial coordinates and brightness. A digital image can be considered a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at that point. The elements of such a digital array are called image elements, picture elements, or pixels.

1.2.2. Formation

The first step in any image processing application is the digital image formation. Basically, it consists of an optical system, the sensor and the digitizer. The optical signal is usually transformed to an electrical signal by using a sensing device, which can be a monochrome or color T.V camera that produces an entire image of the problem domain every 1/30 sec. The imaging sensor could also be a line-scan camera that produces a single image line at a time. In this case, the object’s motion past the line scanner produces a two dimensional image. If the output of the camera or other imaging sensor is not already in digital form, then the analogue or electrical signal is transformed to a digital one by using a video digitizer. The optical image is transformed to a digital one. However, each digital image formation introduces a deformation to the digital image (ex: geometrical distortion, noise). Then, mathematical modeling is necessary in order to have a precise knowledge of the deformation introduced in the digital image.

1.2.3. Restoration

Concerning the reduction of the deformation introduced during digital image formation, the digital image restoration techniques take place. The knowledge of the
mathematical model of the deformations is essential in restoration. That is, restoration techniques have strict mathematical foundation. While digital image enhancement techniques are concerned in improving the quality of the digital image. This usually involves digital image sharpening, noise reduction, and isolating regions whose texture indicates a likelihood of alphanumeric information.

1.2.4. Frequency

Moreover, digital image frequency is involved in digital noise filtering, digital image restoration and digital image compression. Digital image transforms are used to obtain the digital image frequency content. The transforms used are two-dimensional, because the digital image is a two-dimensional.

1.2.5. Presentation

Digital image can be conveniently represented by an NxM matrix A of the form

\[
A = \begin{bmatrix}
  a_{1,1} & a_{1,2} & \ldots & a_{1,M} \\
  a_{2,1} & a_{2,2} & \ldots & a_{2,M} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{N,1} & a_{N,2} & \ldots & a_{N,M}
\end{bmatrix}
\]

The matrix elements or image pixels are integers in the range \([0, \ldots, 255]\) for 8 bit images. The elements have the form \(a[i][j], b[i][j]\) where \(1 \leq i \leq N\)

\& \(1 \leq j \leq M\)
1.2.6. Storage

The storage scheme used for such arrays is of the form shown below:

\[ a \rightarrow a[1][1] \rightarrow a[1][2] \rightarrow \ldots \rightarrow a[1][M] \]
\[ \rightarrow a[2][1] \rightarrow a[2][2] \rightarrow \ldots \rightarrow a[2][M-1] \rightarrow a[N][M] \]

Fig. 1: Image Storage

This scheme corresponds to the spatial coordinates illustrated in the following figure:

(used in the digital image presentation)

This system is usually employed in a computer graphics display. The indices \( i, j \) of the image matrix \( a[i][j] \) refers to the axes \( y, x \) respectively. This system is a \(-90^\circ\) or \(270^\circ\) rotated version of the coordinate system usually used in geometry and graphics that is shown below:
1.2.7. Digital Image Compression

Nevertheless, digital images require a large amount of memory for their storage. For example, a colored image of size $1024 \times 1024$ pixels occupies about 3MB of disk or RAM space. Moreover, the Britannica Encyclopedia needs 25 gigabytes for its pictures. Therefore, the reduction of the memory requirements is of extreme importance in many applications such as in image storage or in image transmission. Therefore, the digital image coding and compression take advantage of the information redundancy existing in the image in order to reduce its information content and to compress it. Large compression ratio might reach 1:24, which can be obtained properly of the information redundancy. Excessive image compression might result in image deformation.

Therefore, image compression plays an important role in several vital applications such as image databases, digital image transmission, facsimile (exact copy), digital video and high definition T.V.

Digital image compression techniques can be of two large classes: lossless and lossy compression. Lossless compression is employed in applications where image data are difficult to obtain or contain necessary information that may be destroyed by compression. While lossy compression can be used when image data can be easily reproduced or when the information loss can be tolerated at the receiver site. [2,3,5,6,8]
CHAPTER 2
SEGMENTATION

2.1. INTRODUCTION

The next stage deals with segmentation, which identifies areas of an image that appear uniform to an observer, and subdivides the image into regions of uniform appearance. The level to which this subdivision is carried depends on the problem being solved. That is, segmentation should stop when the object of interest in an application have been isolated. Segmentation deals with techniques for extracting information from an image. We refer to this area of processing as image analysis, for example, in autonomous air-to-ground target acquisition applications, interest lies among other things, in identifying vehicles on a road. The first step is to segment the road from the image and then to segment the content of the road down to objects of a range of sizes that correspond to potential vehicle. There is no point in carrying segmentation below this scale, nor is there any need to attempt segmentation of image components that lie outside the boundaries of the road.

Segmentation algorithms for monochrome images generally are based on one of two basic properties of gray-level value: pixel-based or discontinuity, and region-based or similarity. These approaches are complementary and should produce the same results, however in practice this is rarely the case. The pixel-based approach is to partition an image based on sudden changes in gray level. It seeks to detect and enhance edges or edge elements within an image and then link them to create a boundary which encloses a region of uniformity (it includes detection of isolated points, lines and edges in an image). While the region-based approach seeks to create regions directly by grouping together pixels which share common feature into areas or
regions of uniformity. It is based on thresholding, region growing, and region splitting and merging.

There are many several techniques for detecting the three basic types of discontinuities in a digital image: points, lines, and edges. In practice, the most common way to look for discontinuities is to run a mask through the image. The idea behind mask operations is to let the value assigned to a pixel be a function of its gray level and the gray level of its neighbors. Consider the subimage area shown below:

\[
\begin{array}{ccc}
Z_1 & Z_2 & Z_3 \\
Z_4 & Z_5 & Z_6 \\
Z_7 & Z_8 & Z_9 \\
\end{array}
\]

And suppose we want to replace the value of \(Z_5\) with the average value of the pixels in a 3 \times 3 region centered at the pixels with value \(Z_5\):

\[
Z = \frac{1}{9} (Z_1 + Z_2 + Z_3 + Z_4 + Z_5 + Z_6 + Z_7 + Z_8 + Z_9) = \frac{1}{9} \sum_{i=1}^{9} Z_i.
\]

And assigning to \(Z_5\) the value of \(Z\). [8]
With the reference to the mask shown;

<table>
<thead>
<tr>
<th>$W_1$</th>
<th>$W_2$</th>
<th>$W_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_4$</td>
<td>$W_5$</td>
<td>$W_6$</td>
</tr>
<tr>
<td>$W_7$</td>
<td>$W_8$</td>
<td>$W_9$</td>
</tr>
</tbody>
</table>

The same operation can be obtained in more a general terms by centering the mask at $Z_5$ multiplying each pixel under the mask by the corresponding coefficient and adding the results:

$$Z = W_1Z_1 + W_2Z_2 + W_3Z_3 + W_4Z_4 + W_5Z_5 + W_6Z_6 + W_7Z_7 + W_8Z_8 + W_9Z_9$$

$$= \sum_{i=1}^{9} W_iZ_i$$

Proper selection of the coefficients and application of the mask at each pixel position in an image makes possible a set of useful image operations, such as noise reduction, region thinning and edge detection.

However, applying a mask at each pixel location in an image is a computationally expensive task. For example, applying a $P \times P$ mask to an $m \times n$ image requires $P^2$ multiplications, and $P^2 - 1$ additions at each pixel location, for a total of $m \times n \times P^2$ multiplications $m \times n \times (P^2 - 1)$ additions.

Therefore, the response of the mask shown above at any point in the image is

$$R = \sum_{i=1}^{9} W_iZ_i$$

where $Z_i$ is the gray level of the pixel associated with mask coefficient $W_i$. The mask's response is defined according to its centered at the boundary; the response is computed by using the appropriate neighborhood. [8]

### 2.2. POINT DETECTION

The detection of isolated points in an image is by using the following mask:
Fig. 2: A mask used for detecting isolated points different from a constant background.

If $|I_R| > T$, then a point has been detected at the location on which the mask is centered. $T$ is a nonnegative threshold.

The idea is that the gray level of an isolated point will be quite different from the gray level of its neighborhood. We are only interested in the isolated points in an image whose differences are large enough (determined by $T$). [9,10]

2.3. LINE DETECTION

If we consider the masks:

Fig. 3: Mask (a) would respond to lines oriented horizontally.

While mask (b) responds best to lines oriented at $45^\circ$.

The third mask to vertical lines; and the fourth mask to lines in the $-45^\circ$ or $135^\circ$ direction.
For example, if at a point in the image \( |R_a| > |R_j| \) where \( j = b, c, d \) that particular point is said to more likely associated with a horizontal line. [9,10]

2.4. EDGE DETECTION

Moreover, edge detection is the most common approach for detecting meaningful discontinuities in gray level. The following schemes search for edges between regions.

**Basic Formulation:** An edge is the boundary between two regions with relatively distinct gray-level. Most techniques rely on the computation of a local derivative operator as shown in the figure below:

The first derivative is positive at the leading edge, negative at trailing edge. The second derivative is positive for the part associated with the dark side of the edge, negative for that part associated with the light side of the edge.

Then, the first derivative at any point in an image is obtained by using the magnitude of the gradient at that point. The second derivative is similarly obtained by using Laplacian.

2.4.1. Gradient Operators

The image gradient \( \nabla f(x,y) \) at location \((x,y)\) is the vector:

\[
\nabla f(x,y) = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}
\]
While the magnitude of this vector can be used as an edge detector and is referred to as the gradient:

$$e(x,y) = (f_x(x,y)^2 + f_y(x,y)^2)^{1/2}$$

Alternatively, the sum of the absolute values of partial derivatives $f_x$, $f_y$ are used for computational simplicity:

$$e(x,y) = |f_x(x,y)| + |f_y(x,y)|$$

The direction of the gradient vector can be represented by:

$$\alpha(x,y) = \tan^{-1}(f_y / f_x)$$

In this case we can use the Sobel edge detector, which provides a good performance and is relatively insensitive to noise. The Sobel operators have the advantage of providing both a differencing and a smoothing effect.

![Fig.5: Sobel Edge Detector Masks](image)

Derivative based on the Sobel operator masks are:

$$\nabla_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$$

$$\nabla_y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)$$
Computation of the gradient at the location of the center of the masks using the $e(x,y)$, which gives one value of the gradient. To get the next value, the masks are moved to the next pixel location and the procedure is repeated.

2.4.2. Laplacian

Another approach to edge detection is the use of the Laplace operator. It is defined in terms of second-order partial derivative of $f(x,y)$:

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

Moreover, the coefficients associated with the center pixel be positive and the ones with the outer pixels be negative. The sum of the coefficients has to be a zero. The Laplacian is sensitive to noise, produces double edges, and is unable to detect edge direction. Therefore it plays a secondary role of detector for establishing whether a pixel is on the dark or light side of an edge.

Fig. 6: Original image + complete gradient image obtained by using $e(x,y) = |f_x(x,y)| + |f_y(x,y)|$

2.4.3. Combined Detection

Using a multimask makes it easier to develop a new method that determine whether a pixel is more likely to be an isolated point or part of a line or an edge. This development becomes better with the use of a vector formulation:
\[
W = \begin{pmatrix} W_1 \\ \vdots \\ W_9 \end{pmatrix} \quad Z = \begin{pmatrix} Z_1 \\ \vdots \\ Z_9 \end{pmatrix}
\]

\[R = \sum_{i=1}^{9} W_i Z_i = W^T Z\]

For example, assuming that we have two masks designed to detect edges and lines, and the coefficients are two vectors, \(W_1\) and \(W_2\), are orthogonal and have unit length. In this case, the product \(W_1^T Z\) and \(W_2^T Z\) equal to the projections of \(Z\) onto the vectors \(W_1\) and \(W_2\), since

\[W_1^T Z = \| W_1 \| \| Z \| \cos \Theta\]
\[\| W_1 \| = 1;\]
\[\| Z \| \cos \Theta = W_1^T Z\] which is the projection of \(Z\) on \(W_1\). Similar components apply to \(W_2\). [7,8,10]

2.5. BOUNDARY DETECTION

2.5.1. Local Processing

This part deals with images of disconnected edge elements and it needs to link these elements into straight and curved boundaries. These boundaries provide patterns, which are more appropriate for classification in any object in the scene.
Therefore, edge detection is followed by linking procedures to change edge pixels into meaningful boundaries.

**Boundary Refining:** One of the simplest techniques for linking edges is to analyze a small neighborhood of pixels within an image that has already undergone edge detection. At each point within this edge map the gradient magnitudes and directions are checked such that:

\[(x', y') \text{ is an edge pixel in the predefined neighborhood of } (x, y), \text{ has a similar magnitude to the pixel at } (x, y):\]

\[|\nabla f(x, y) - \nabla f(x', y')| \leq T\]

\[T \text{ is non-negative Threshold}\]

Moreover an edge pixel at \((x', y')\) has an angle similar to the pixel at \((x, y)\) if

\[|\alpha(x, y) - \alpha(x', y')| < A\]

\[A \text{ is an angle threshold}\]

If both criteria are satisfied, then the predefined neighborhood of \((x, y)\) is linked to the pixel at \((x, y)\). This process is repeated for every location in the image. A record must be kept of linked points and the center of the neighborhood is moved from pixel to another. A simple bookkeeping procedure is to assign a different gray level to each set of linked edge pixels.

### 2.5.2. Hough Transform

Nevertheless, we consider linking points determined whether they lie on a curve of specified shape. For \(n\) points in an image, we want to find subset of these points that lie on straight line. According to Hough transform a point \((x_i, y_i)\) was considered. The general equation is \(y = ax + b\).
Both points produce another points in Hough space in the form \((a_i, b_i)\) and \((a_j, b_j)\). Thus when all points of interest in input space have been transformed to points in Hough space, the intersections of these points give a vote as to the best set of parameters for the unique line in the input space which will join all given points. The disadvantage in this approach is that a very large array is required to store all possible votes. In order to solve this problem, the Cartesian form was replaced with polar coordinates. That is, a straight line can thus be defined by the length \(r\) and angle \(\phi\). These parameters are related to \(x\) and \(y\) in the form

\[
R = x \cos \phi + y \sin \phi
\]

For example,
Each point in space is represented as a sinusoidal curve plotted between 0 and 2\Pi. Therefore, the position of maximum intersection can be found and the unique values of \( r_b \) and \( \phi_b \) used to define the best straight line detected in the input image which joins the given pixel points. Moreover, the curve exhibits a symmetrical positive and negative response. That is why the amount of computation can be halved, since it is possible to plot curve between 0 and 2\Pi.

Nevertheless, if objects contain circular shapes then the Hough transform can be used to detect circles such as:

\[
(x - a)^2 + (y - b)^2 = r^2
\]

\((a, b)\) : center of circle
\(r\) : radius

Therefore, in this case we have three diameters, which leads to a three-dimensional Hough space. Moreover, the complexity of Hough transform depends on the number of points in the input image and number of coefficients required to define the functional representation of the curves to be detected.

When applying Hough transform, it should be remembered that the straight lines, which have been detected, are infinitely long and the curves are complete real boundary. This can be accomplished by examining all points which fall on the best-fit curve and calculating the distance between disconnected pixels. A threshold can be set for the distance, and any points falling above the threshold are being set a disconnected.

2.5.3 Graph Theoretic Technique

According to this approach, it is based on representing edge segments in the form of a graph and searching the graph for low-cost paths that correspond to significant edges. This technique performs well in the presence of noise.
With respect to its development, a graph $G = (N,A)$ is a finite nonempty set of nodes $N$. Each pair $(n_i, n_j)$ of this set is called an arc. If an arc is directed from node $n_i$ to $n_j$, then $n_j$ is the successor of its parent node $n_i$. A cost $c(n_i, n_j)$ can be associated with arc $(n_i, n_j)$. A sequence of nodes with each pair $n_i$ being a successor of node $n_{i-1}$ is called a path from $n_1$ to $n_k$ and the cost of the path is:

$$C = \sum_{i=2}^{k} (n_{i-1}, n_i)$$

For example, according to the figure below: the outer numbers are pixel coordinates and the numbers in parentheses represent intensity. Each edge element defined by pixels $p$ and $q$ has an associated cost:

$$c(p,q) = H - [f(p) - f(q)]$$

where $H$ is the highest intensity value (7 in this case), and the $f$ is the intensity value of each.

Fig. 9: (a)

The cost of each edge element, computed by the above is the leading to it. Therefore, the below figure shows the graph to this problem. The dashed lines represented the minimum-cost path. [4,7,8]

![Graph Diagram](image-url)
2.6. REGION BASED APPROACH

The region-based approach to segmentation seeks to create regions directly by grouping together pixels, which share common features into areas or regions. This approach is too computationally expensive for typical machine vision application. Therefore, simpler forms of segmentation have been found wanting. Despite this, region-based methods of interest because they are generally less sensitive to noise than boundary-based approaches.

2.6.1. Region merging and splitting

This approach is referred to as either 'region merging' or 'region growing'. In this method the image is divided into arbitrary elementary regions often starting at the level of individual pixels, then by merging these elementary regions according to some specific schema until no more can be merged. The criteria for merging regions determine the final segmentation and it will depend on the type of data available and on the application. Moreover, it can be measured over the entire region or along the boundaries of the two regions. Nevertheless, it is important to decide when to stop the merging process.
determine the final segmentation and it will depend on the type of data available and on the application. Moreover, it can be measured over the entire region or along the boundaries of the two regions. Nevertheless, it is important to decide when to stop the merging process.

Fig. 10: Region merging

For example, refer to the figure above, the gray-level values fall between 1 and 10 and the uniformity predicate used is for regions to merge when the difference in gray-level value intensity between adjacent regions is 1. Fig 'a' shows the original distribution of gray-level values. In fig 'b' the image identifies all regions with
discrete gray-level values and hence a primary region map is produced. It can be seen that 5 and 6, 6 and 7, and 9 and 10 can be merged. Fig ‘c’ displays one light and two dark gray level regions.

Another way could be achieved while getting the same result if the starting point had been to identify one pixel in the original image as a ‘seed’, for example one of the pixels labeled 10, and then apply the uniformity predicate to ‘grow’ from this single pixel region into the major dark area, as identified in the resultant segmented image.

The techniques classified as ‘split and merge’ first split the image into a set of arbitrary regions, then a uniformity predicate is applied either to subdivide the region further or merge adjacent regions. The subdivision of the image can be viewed graphically through a ‘tree’ structure in which the ‘root’ of the tree corresponds to the whole image and the ‘leaves’ of the tree correspond to individual pixels. Then, the descending one level in the hierarchy of the tree represents dividing the image into four sub-images, hence obtaining the term ‘quad tree’. This division extends to the lowest level of the tree when the children represent leaves, or individual pixels. Moreover, the quad-tree representation of an image is a form of data compression.

(Refer to the figure 11: fig ‘b’ only uses 25 nodes to represent 64 pixels.)

A split and merge algorithms can be summarized as follows:

- Split any region into four quadrants if the uniformity predicate is false.
- Continue this subdivision for all new sub-images until stopping criteria is reached.
- Merge regions for which the uniformity is true.
- Stop when no further merging is possible. (See figure below)
Fig. 11: Quad-tree representation of a binary image.

According for the figure above, the original image is represented by a binary array. While fig 'b' presents the subdivision or splitting into four regions, which
continues when the uniformity predicate is false. Fig 'c' is the production of the
merging process when splitting is complete. Finally, the quad-tree representation
is given in fig 'd'; shaded circles represent nodes that have been labeled as the
dark region of the image, and vice-versa. [8,9]

2.6.2. Thresholding

Thresholding is one of the most important techniques for segmentation and is
a widely used tool for machine vision system. From the segmentation viewpoint,
thresholding is a method of producing regions of uniformity within an image
based on some threshold criterion, T. The thresholding operation can thus be
thought of as a test involving the function T and defined as

\[ T = T(x, y, A(x, y), f(x, y)) \]

Where \( f(x, y) \) is the gray level of the pixel at \((x, y)\), and \( A(x, y) \) denotes some local
property in the neighborhood of this pixel. A threshold image \( g(x, y) \) is defined as

\[
\begin{align*}
g(x, y) &= \begin{cases} 
1 & \text{if } f(x, y) \geq T \\
0 & \text{if } f(x, y) < T
\end{cases}
\end{align*}
\]

The value of the function \( T \) can be defined in one of three ways:

(a) Global threshold: \( T = T\{f(x, y)\} \)

where \( T \) is dependant only upon the gray-level value of the pixel at \((x, y)\).

(b) Local threshold: \( T = T\{A(x, y), f(x, y)\} \)

where \( T \) is dependant upon a neighborhood property of the pixel as is its gray-
level value.
where $T$ is dependant on the pixel coordinates, in addition to the other two criteria.

In practice the global thresholding technique is often used although the selection of the most appropriate threshold, $T$, to use for a given application in the main dilemma for the system designer. This choice will be simplified if the image is of good contrast and the quality image has been captured. If correct illumination and image acquisition strategies are followed for a specific application, then the segmentation process can be trivialized to simple thresholding and the ideal global threshold value can be easily evaluated. For a simple global threshold, where the image histogram is bimodal or has easily identifiable peaks and valleys, the selection of $T$ is fairly straightforward. If the image is noisy or there is a considerable spread in gray level values, then the selection of $T$ is problematic. In cases of difficult identification of optimum segmentation, optimal thresholding techniques rely on statistical analysis, for example, probability density, histogram entropy, or minimum error analysis.

One approach to improving the segmentation process is to consider a histogram made up of only those pixels, which lie at or near an edge or boundary of objects within the image. Such a histogram should contain sharper peaks and lower valleys. The main problem here is to decide which pixel lie on or near a boundary. The Laplacian of Gaussian operator is first applied to the image so that the edge points will be clearly identified. Once identified the average gray-level value for these edge points is calculated and used as the global threshold value. Although this calculation may take some time, it is only carried out once during the system calibration and initialization procedure and not before every threshold operation.
Moreover, in some cases, a sensor might make available more than one variable to characterize each pixel in an image. An example is color imaging, where red (R), green (G), and blue (B) components are used to form a composite color image. In this case, each pixel is characterized by three values. The basic procedure is the same for that use for one variable. For example, for three 16-level images corresponding to the RGB components, a $16 \times 16 \times 16$ grid is formed (cube), and inserted in each cell of the cube is the number of pixels whose RGB components have intensities corresponding to the coordinates defining the location of that particular cell. Each entry can then be divided by the total number of pixels in the image to form a normalized histogram.

The concept of thresholding now becomes one of finding clusters of points in 3-D space. Suppose, for example, that $K$ significant clusters of points found in the histogram. The image can be segmented by assigning one intensity to pixels whose RGB components are certainly to more clusters. The principle difficulty is that cluster seeking becomes an increasingly complex task as the number of variable increases.[9]
CHAPTER 3

TEXTURE ANALYSIS

3.1. INTRODUCTION

After an image has been segmented into regions, the resulting aggregate to segmented pixels usually are represented and described in a form suitable for further computer processing. Basically, representing a region involves two choices:

1. We can represent the region in terms of its external characteristics (its boundary);
2. We can represent it in terms of its internal characteristics (pixels comprising the region);

However, choosing a representation scheme is only part of the task of making the data useful to a computer. The following task is to describe the region based on the chosen representation. For example, a region may be represented by its boundary where the boundary is described by features such as its length, the orientation of its straight line joining the extreme points, and the number of concavities in the boundary. Moreover, an external representation is chosen when the primary focus is on shape characteristics. An internal representation is selected when the primary focus is on reflectivity properties, such as color and texture. However, in both cases, the features selected as descriptors should be as insensitive as possible to variations such as changes in size, translation and rotation.
3.2. TEXTURE

3.2.1. Definition

An important approach to region description is to quantify its texture contents. There is no universal accepted definition for texture. However, all researchers agree on two points. First, there is significant variation in intensity levels between nearby pixels, that is, at the limit of resolution there is no homogeneity. Second, texture is a homogeneous property at some spatial scale larger than the resolution of the image. Some researchers describe texture in terms of human visual system: that texture does not have uniform intensity, but are observed as homogeneous regions by a human observer. Texture regions give different interpretations at different distances and at different degrees of visual attention. At a standard distance with normal attention, it gives the notion of macro regularity that is characteristics of the particular texture. When viewed closely and attentively, homogeneous regions and edges, sometimes constituting texels, are noticeable. According to A.K. Jain, *Fundamental of Image Processing*, the term texture generally refers to repetition of basic texture elements called texels. The texels contain several pixels, whose placement could be periodic, quasi-periodic or random. Natural textures are generally random, whereas artificial textures are often deterministic or periodic. With respect to Wilson and Spann's *Image Segmentation And Uncertainty*, textured region are spatially extended patterns based on the more or less accurate repetition of some unit cell called texton or sub pattern.

Therefore, it seems that a single, unambiguous, widely accepted definition does not exist. A reason for this is a strong intuitive concepts of texture, which are indeed hard to encompass fully in a definition. [12]
3.3. METHODS OF TEXTURE

Although no formal definition of texture exists, this descriptor provides measures of properties such as coarseness, fineness, smoothness, granulation, ripple, regularity, irregularity or linearity. The principle approaches used in image processing to describe the texture of a region are statistical, structural, spectral and multiscale.

Fig.12: Texture Examples

3.3.1. Statistical Method

It is one of the simplest techniques for describing texture, in which moments of the gray-level histogram of an image or region are used.

Let $z$ be a random variable denoting different image intensity and let $P(z_i)$, $i=1,2,3,\ldots,L$ be the corresponding histogram, where $L$ is the number of different intensity levels.

Then, the $n^{th}$ moment of $z$ about its mean is:

$$\mu_n(z) = \Sigma_{i=1}^{L} (z_i - \mu)^n P(z_i) \quad \text{where } \mu_0=1$$
smoothness. Such that, the measure \( R = 1 - \frac{1}{1+\sigma^2(z)} \) is zero for the areas with the same intensity and approach 1 for large values of second moment. The third moment is a measure of the skewness of the histogram, while the fourth moment is a measure of its relative flatness.

Measure of textures computed using only histogram carry information regarding the relative position of pixels with respect to each other. One way to solve the problem in texture analysis process is not only to consider the distribution of intensities but also the positions of pixels with equal and nearly intensity values:

Therefore, consider an image with the following gray levels, \( z_1=0, z_2=1 \) and \( z_3=2 \) such that,

\[
\begin{array}{cccc}
0 & 0 & 0 & 1 & 2 \\
1 & 1 & 0 & 1 & 1 \\
2 & 2 & 1 & 0 & 0 \\
1 & 1 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 1 \\
\end{array}
\]

Defining the position operator \( P \) as "one pixel location below and to the right" which yield an \( A_{3x3} \) matrix:

\[
A = \begin{pmatrix}
4 & 2 & 1 \\
2 & 3 & 2 \\
0 & 2 & 0 \\
\end{pmatrix}
\]

The size of \( A \) is determined strictly by the number of distinct gray levels in the input image. While its contents, for example, \( a_{11} \) is the number of times that a point with
The size of $A$ is determined strictly by the number of distinct gray levels in the input image. While its contents, for example, $a_{11}$ is the number of times that a point with level $z_1=0$ appears one pixel location below and to the right of a pixel with the same gray level, $a_{31}$ is the number of times that a point with level $z_1=0$ appears one pixel location below and to the right of a point of gray level $z_3=2$, and $a_{21}$ is the number of times that a point with level $z_2=1$ appears one pixel location below and to the right of a point with gray level $z_1=0$.

Let $n$ be the total number of point pairs in the image that satisfy $P$ ($n=16$ in the previous example). Let a matrix $C$ be formed by dividing every element of $A$ by $n$. The matrix $C$ is called a gray-level coocurrence matrix. Since $C$ depends on $P$, then the presence of given texture patterns may be detected by choosing an appropriate position operator. Such that, the preceding example is sensitive to bands of constant intensity running at $-45^\circ$. The problem is to analyze a given $C$ matrix in order to categorize the texture of the region over which $C$ was computed. Therefore, the following descriptors or features are useful for this purpose: [14,19]

(a) Maximum probability: $\max_{ij} (c_{ij})$

Gives an indication of the strongest response to $P$.

(b) Element difference moment of order $k$

$$\sum_i \sum_j (i - j) c_{ij}$$

This descriptor has a relatively low value when the high values of $C$ are near the main diagonal because the differences $(i - j)$ are smaller there.

(c) Inverse element difference moment of order $k$
Is a measure of randomness, achieving its highest value when all elements of $C$ are equal.

(e) Energy

$$\sum_{i,j} c_{ij}^2$$

(f) Inertia

$$\sum_{i,j} (i - j)^2 c_{ij}$$

3.3.2. Statistical Method

A second major category of texture description is based on structural concepts. It makes a description using texture primitives and syntactic rules. Suppose that we have a rule of the form $S \rightarrow aS$, i.e., the symbol $S$ can written as $aS$. Such that, after three application it will yield the string $aaaS$. If “$a$” represents a circle, as shown in the figure below, and the meaning is “circles to the right”, then the rule $S \rightarrow aS$ allows the generation of the texture pattern shown in (b).

Suppose $S \rightarrow bA$, $A \rightarrow cA$, $A \rightarrow c$, $A \rightarrow bS$, $S \rightarrow a$.

$b$ means “circles down” and $c$ means “circles to the left”. Therefore, $aaabccbaa$ generates a 3x3 matrix of circles. Therefore, large texture patterns can be generated as the one shown in figure (c). Moreover, patterns that are not rectangular can also be generated.
b means “circles down” and c means “circles to the left”. Therefore, aaabccbaa generates a 3x3 matrix of circles. Therefore, large texture patterns can be generated as the one shown in figure (c). Moreover, patterns that are not rectangular can also be generated.

![diagram](c)

Fig.13: Structural Method

However, we can conclude that a simpler “texture primitive” can be used to form more complex texture patterns using some rules. [11]

3.3.3. Spectral Method

Since the Fourier spectrum works good for describing the directionality of periodic 2-D patterns in an image, then it is useful for texture patterns that are difficult to detect with spatial methods but can be easily distinguished as concentrations of high-energy bursts in the spectrum.

Here, we consider three features of the Fourier spectrum that are useful for texture description:

1. Distinguishable peaks in the spectrum give the principle direction of the texture patterns.
2. The location of the peaks in the frequency plane gives the fundamental spatial period of the patterns.

3. Eliminating any periodic components via filtering leaves non-periodic image elements, which can then be described by statistical techniques.

However, since the spectrum of a real image is symmetric about the origin, then only half of the frequency plane needs to be considered.

Detection and interpretation of the spectrum features are simplified by expressing the spectrum in polar coordinates to yield a function $S(r, \theta)$.

$$S: \text{The spectrum function.}$$

$r, \theta$: Variables in the coordinate system.

For each direction $\theta$, $S(r, \theta)$ may be considered a 1-D function $S_\theta(r)$, and for each frequency $r$, $S_r(\theta)$ is a 1-D function. Analyzing $S_\theta(r)$ for a fixed value of $\theta$ yields the behavior of the spectrum along a radial direction from the origin, while analyzing $S_r(\theta)$ for a fixed value of $r$ yields the behavior along a circle centered on the origin.

Then

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r)$$

$$S(\theta) = \sum_{r=1}^{R} S_r(\theta)$$

where $R$ is the radius of a circle of center $(0,0)$. For $N \times N$ spectrum, $R$ is chosen $N/2$.

These two equations results in a pair [$S(r)$, $S(\theta)$] for each pair of coordinates $(r, \theta)$. By varying these functions, we can generate two 1-D function, $S(r)$, $S(\theta)$, that constitute a spectral-energy description of texture for an entire image or region under consideration. [12,19,20]
strong motivation for texture analysis methods. Moreover, wavelets can act as basis functions for these methods.

3.3.4.1. Wavelet Multiresolution Analysis:

Multiresolution techniques intend to transform images into a representation in which both spatial and frequency information is present. A lot of techniques where developed to accomplish this technique such as Gabor, Gaussian and Laplacian pyramids, scale space, ... Years ago, the wavelet method came into view. It provided a more formal, solid and unified approach to multiresolution representations. Moreover, it became a preferred tool for multiresolution analysis.

The wavelet decomposition of 2-D image can be obtained by applying the filtering consecutively along horizontal and vertical directions. See the figure below:

Fig. 14: One level of decomposition in three steps:

(1) Low & high filtering in horizontal directions.
(1) Low & high filtering in vertical directions.
(3) Subsampling.

It yields 4 subimages for 1 level of decomposition. To construct a multilevel decomposition, this is repeated for the low pass subimages. The result is the standard
pyramidal wavelet decomposition. When the detailed images are also decomposed further, we obtain the tree structured wavelet decomposition.

![First two steps of pyramidal Wavelet decomposition.](image1)

![First two steps of tree structure Wavelet decomposition.](image2)

Fig. 15: Pyramidal and tree structure Wavelet Decomposition

The wavelet image decomposition provides a presentation that is easy to interpret. Every subimage contains information of a specific scale and orientation. Moreover, spatial information is kept within the subimages.

To obtain features that reflect scale-dependant properties, one can extract a feature from each subimage separately. The concept of using more than one feature from each subimage is being preferred. However, when the number of features tends to become larger, especially for the tree structured decomposition is a major problem in wavelet texture analysis. A larger number of features may carry more information, but this will make segmentation and classification more difficult. Nevertheless, feature reduction methods existed for dealing with this. Moreover, the scales that carry the most useful information can differ from one texture to another, which is a problem. Therefore, to limit the number of features at the level of their generation, where the nature of the features can be taken into account, can be advantageous.

Another strategy is to use an extra criterion that evaluates features. For a tree-structured transform, a criterion can be used to decide if a subimage needs to be decomposed further. This is known as adaptive wavelet transform.
problem. Therefore, to limit the number of features at the level of their generation, where the nature of the features can be taken into account, can be advantageous.

Another strategy is to use an extra criterion that evaluates features. For a tree-structured transform, a criterion can be used to decide if a subimage needs to be decomposed further. This is known as adaptive wavelet transform.

Moreover, segmentation also raises several difficulties since the properties of the textured regions and the number of different textured are not known in advance for an unsupervised problem. This becomes more difficult for when there are many regions. None-the-less, subsampling is often omitted in segmentation tasks, since a complete representation is then obtained in which the redundancy improves robustness. In the tree-structured decomposition, the none-subsampled is referred to as "wavelet frame decomposition". In general, omission of subsampling will improve results, but increase computation time.

Wavelet texture analysis can be extended to color texture, although there are no much-advanced researches about this topic. Color images are typically represented by RGB tristimulus values which correspond to the three color bands. A direct way to process color textures is by performing a gray level decomposition on every component image. The number of features will triple compared to the gray level one.[13,15,23]
CHAPTER 4

TEXTURAL SEGMENTATION

4.1. INTRODUCTION

A dictionary-like definition of texture segmentation would be the following:
"the partitioning of an image into regions, each of which contains a single texture distinct from its neighbors".

It is clear, then, that there is no one segmentation of an image that can be considered to be "right". The "right" segmentation exists only in the mind of the observer, which can change not only between observers, but within the same observer at different times.

We wish to partition an image into regions of homogeneous texture, but we cannot always agree on when two texture samples are similar to each other. Furthermore, two different objects with a common boundary may have the same texture and be lumped together, which may or may not be desired. A texture segmentation will invariably break up certain objects which contain multiple textures, and group together pieces of different objects into one region.

Texture segmentation is an essential component of many image processing, computer graphic and computer vision techniques. The ability to discriminate textures is generally dependent of scale, rotation, and changes in illumination.
Whenever a machine vision system is expected to perform the same texture discrimination task, it has to solve the problem of segmenting the image into uniformly textured regions, recognizing similar textures even if they are differently shaded or rotated. Texture is characterized not only by the gray value at a given pixel, but also by the gray value of the pattern within a neighborhood surrounding the pixel.

Texture analysis is an interesting but difficult problem in image processing. It requires unique mathematical operators that discriminate between an unlimited number of textures while providing precise identification of borders between textured regions. Most methods share one common weakness, which is that they primarily focus on the coupling between image pixels on a single scale. One difficulty of traditional texture analysis is the lack of tools that characterizes different scales of textures.

Recent developments in spatial analysis such as the Gabor transform, and wavelet transform provide good multiresolution analytical tools and should help to overcome this difficulty. A large class of natural textures can be detected by highly concentrated spatial frequencies and orientations. Recent study of the human vision system indicates that the spatial representation, which preserves both global and local information, is sufficient for characterizing texture properties, which motivated researchers to develop multiresolution texture models. However, the wavelet theory is a new spatial analysis, and the wavelet basis function, which includes a library of modulated waveform orthonormal basis called wavelet packets, is also a recent analysis. The wavelet and wavelet packet transform can be implemented efficiently with pyramid algorithms and tree-structured algorithms.

4.2. TEXTURE AND SEGMENTATION
Domains constitute an intermediate level between pixels and images. The image is a set of pixels; domains are spatially compact subsets of the image. They may represent interesting parts of a scene and it is the task of segmentation to isolate them. If domains are well characterized by the properties of their pixels, such the grey value, segmentation will be easy. If two domains have a common border, pixel properties at both sides of the border are different and the border can be detected. When the difference is too small, the border can be changed into clearer border pieces, if some knowledge about the domain shape is available.

Mainly, human operators perform image segmentation using their visual system. They easily detect domains that are characterized by gray value, or of which the borders are characterized by difference in gray value, but they see the borders between domains of different texture just as easily. Texture is a domain property rather than a pixel property. There is a smallest area on which it can be defined, which is called texture tile or "coarse grain". In order to detect a texture a test window of at least tile size must be analyzed. There is no problem if the test window lies within a domain. While if the test window is positioned over a domain border the detection will be doubtful or will totally fail. The human visual system suppresses this and delivers as crisp borders as between gray value domains. Moreover, the size of the texture tiles is not known before and may differ from domain to another.

A similar suppression is the detection of gray value domains in the presence of noise. If a test window contains a sufficient number of pixels statistics can be applied. The average gray value of a texture tile within a domain will be far less noisy than the gray values of single pixels. The behavior at domain borders is very noise dependent, but the human visual system suppresses this.
While the size of the texture tiles may differ from domain to domain the shape of the texture tiles will also differ. Noise suppression in narrow elongated domains can only be explained assuming the use of narrow elongated test windows that fit within them and consist of sufficient pixels to give a reliable statistical measure. Therefore, test windows of different sizes and shapes at all positions and orientations exist.

First, it is expected that the individual objects have to be segmented and characterized. The statistical distribution of the object characteristics may be used to segment the formations. Once a minimal set of relevant texture attributes has been determined, they have to be used as input for a multivariate segmentation procedure. Segmentation procedures can be based on unsupervised approaches such as clustering and linking algorithms, or supervised classification procedures. The latter procedures need a training set. Of particular importance is the preciseness of the boundary location: many approaches suffer from boundary shifts due to smoothing operations. [14,16,25,26]

4.3. TEXTURAL SEGMENTATION

Textures may be used to describe content of many real-world images: for example, clouds, trees, bricks, hair, fabric all have textural characteristics. Particularly when combined with color and shape information, where these details are important for human vision.

For Example:
Fig. 16: Textural segmentation of a sitting room. (a) An image of a sitting room. (b) A contrast-based image segmentation with a region merging algorithm. (c) A texture segmentation with four segments. The image partitioning is visualized in (d) - (g).

**Example 2:**

- blue: lung
- red: bones
- green: noise
- magenta: blood flow noise
Therefore, it is necessary to find a meaningful measure of the similarity of textures (used for texture discrimination) and to develop a procedure for segmenting images based on textural content (texture segmentation). [17]

4.3.1. Quad Tree Segmentation

In a top-down quad-tree decomposition, a full quad-tree data structure is formed by splitting a single parent node into four children. Then all descendants are recursively split into children until some minimum bound is reached. The quad-tree decomposition is used to perform spatial segmentation by assigning a condition by which nodes are split. A post-processing routine for adjoining similar spatially adjacent nodes with different parents is also added. A final block grouping stage can be added to merge all similar blocks to obtain arbitrarily shaped regions.

The quad-tree can also be modified to allow each parent node to have two, three or four children, as shown in the figure below.

Fig. 18: Quad-tree representation. Each tree parent can have two, three or four children.

When all four children cannot be merged together, subsets of the children may be paired horizontally or vertically depending on which arrangements group the most similar children.
Using the top-down approach, the texture feature extraction is performed on the spatial blocks pointed to by each children node. Based on the values of the texture features, all, some or none of the children are generated. For each spatial block, this requires feature extraction, computation of the distance between the children and the parent, and comparison to a distance threshold to test condition for merging children.[8,21]

4.3.2. Gabor wavelet scale for Unsupervised Texture Segmentation

The segmentation problem can be informally described as the task of partitioning an image into homogeneous regions. For textured images one of the main conceptual difficulties is the definition of a homogeneity measure in mathematical terms. This approach to unsupervised texture segmentation is based on four consecutive design decisions, concerning the questions of image representation, texture homogeneity, objective functions and optimization procedures.

1. A Gabor wavelet scale, space representation with frequency, tuned filters as a natural representation image.

For the segmentation of textured images, a Gabor Wavelet image representation as a natural multi-scale representation for textures is used. We use 4 orientations at 3 scales.

In the following figures a Gabor-Wavelet representation is being described. As can be seen the different texture segments are discriminated at different scales and orientations.
Gabor wavelet representation for different scales at horizontal orientation: (a) High frequency. (b) Middle frequency. (c) Low frequency.

Gabor wavelet representation for different scales at 45 degree orientation: (a) High frequency. (b) Middle frequency. (c) Low frequency.

Gabor wavelet representation for different scales at vertical orientation: (a) High frequency. (b) Middle frequency. (c) Low frequency.

Gabor wavelet representation for different scales at 135 degree orientation: (a) High frequency. (b) Middle frequency. (c) Low frequency.

Fig.19: Gabor wavelet representation for different scales at different orientations.

2. Homogeneity between pairs of texture patches can be measured by a non-parametric statistical test applied to the empirical feature distribution functions of
locally sampled Gabor coefficients. A sparse dissimilarity matrix is computed to adequately model the segmentation problem.

3. Due to the nature of the pairwise proximity data, it is derived a family of pairwise clustering objective functions based on sparse data to formalize the segmentation problem. The objective functions are designed to possess important invariance properties.

The usual approach to pairwise data clustering uses a cost function, which is quadratic in the assignment variables. It is not invariant to additive shifts, as illustrated in the following figure. Depending on the shift the unnormalized cost function often completely misses several texture classes. There may not even exist any parameter value to find all five textures. Even worse the optimal value depends on the data at hand and varies for different images.

![Segmentations for two artificial images.](image)

Segmentations for two artificial images containing five texture types each. From the left segmentations with a mean dissimilarity of \{-0.05, 0, 0.05, 0.1, 0.15, 0.2, 0.25\} are depicted. Segments collapse for negative shifts. For large positive shifts the obtained segmentations become random, because the sampling noise induced by the random neighborhood system dominates the data contributions.
4. Finally, an optimization technique known as deterministic annealing is applied to derive heuristical algorithms to efficiently minimize the clustering objective functions. [22]

The following figure displays the above algorithmic pipeline:

![Algorithmic Pipeline Diagram](image)

Fig.21: Gabor wavelet scale algorithmic pipeline.
4.3.3. Multiresolution Analysis For Feature Extraction Used For Texture Segmentation

1. Multiresolution Analysis Based on Wavelet Transform:

The multiresolution wavelet transform decomposes a signal into the coarser resolution representation, which consists of the low frequency and high frequency information. During the decomposition, the resolution decreases exponentially by 2.

The approximation of a two-dimension finite energy function $f(x,y)$ at resolution $2^j$ is:

$$A_{2^j} f = ((f(x,y) \times \phi_{2^j}(-x) \ \phi_{2^j}(-y)) \ (2^j n, 2^j m)_{(n,m) \in Z^2}$$

$j$ : decomposition level

$m, n$ : integers

$\phi(x)$ : is one dimension scaling function where $\phi_{2^j}(x) = 2^j \phi(2^j x)$. It is a smoothing function whose is concentrated in low frequencies.

The difference between approximation information at two consecutive resolution $2^j$ and $2^{j-1}$, are characterized by $A_{2^j} f$ and $A_{2^{j-1}} f$. Therefore, $A_{2^j} f$ can be perfectly reconstructed from $A_{2^{j-1}} f, D_{2^{j-1}} f, D_{2^j} f$ and $D_{2^{j+1}} f$ where:

$$D_{2^{j-1}} f = ((f(x,y) \times \psi_{2^{j-1}}(-x) \ \psi_{2^{j-1}}(-y)) \ (2^{j-1} n, 2^{j-1} m)_{(n,m) \in Z^2}$$

$$D_{2^j} f = ((f(x,y) \times \psi_{2^{j}}(-x) \ \psi_{2^{j}}(-y)) \ (2^j n, 2^j m)_{(n,m) \in Z^2}$$

$$D_{2^{j+1}} f = ((f(x,y) \times \psi_{2^{j+1}}(-x) \ \psi_{2^{j+1}}(-y)) \ (2^{j+1} n, 2^{j+1} m)_{(n,m) \in Z^2}$$

$\psi(x)$ : one dimensional wavelet function $\psi_{2^j}(x) = 2^j \psi(2^j x)$

It is a band-pass filter.
Decomposition of the frequency support of the image $A_{2j+1}f$ into $A_{2j}f$ and the detail images $D_{2j}^k f$ is shown in the figure below.

![Diagram of multiresolution wavelet transform](image)

The approximation image and detail images derived from decomposition are usually organized as shown in the figure below:

![Organized detail images](image)

(a) $A_{2j+1}f$

(b) $A_{2j}f$

(b) Approximation image derived from decomposition.

The wavelet decomposition can be interpreted as signal decomposition in a set of independent spatially oriented frequency channels. The image $A_{2j}f$ correspond to the lowest frequencies, $D_{2j}^1 f$ gives the vertical high frequencies, $D_{2j}^2 f$ gives the horizontal high frequencies, $D_{2j}^3 f$ gives the high frequencies in both directions.

When the decomposition level $j$ decreases, the resolution decreases in the spatial domain and increases in the frequency domain. This variation of the
wavelet resolution enables the wavelet transform to zoom into the irregularities of the signal and characterize them locally.

Since the most significant information of a texture often appears in the middle frequency channels, further decomposition just in the lower frequency region, such as the wavelet transform does, may not help for the purpose of segmentation.

In the figure below, the pyramid-structured wavelet transform is applied to two different images:

![Fig.23: “lady” image: Image and its pyramid-structured wavelet transform](image1)

![Fig.24: “Texture” image: Texture image shown with its pyramid-structure wavelet transform](image2)

From figure 24, the lady image is clear from its low frequency channel (the upper left corner). While in the second figure, it is not clear to recognize a similar texture pattern in the same low frequency channel, rather, some horizontal and vertical line patterns are observed clearly in the middle frequency region. Therefore, the low frequency region of textures may not necessarily contain significant information.

Then, the best way to perform the wavelet transform for textures is to detect the significant frequency channels and then decompose them further. It is not
necessary to decompose all subimages in each scale to achieve a full decomposition. To avoid a full decomposition, a maximum criterion of textural measures to locate dominant information in each frequency channels and to decide whether a decomposition is needed for a particular output. With this transform, we are able to zoom into any desired frequency channels for further decomposition.

Fig.25: (a) Decomposition is no longer simply applied to the low frequency channels recursively. It can be applied to any other frequency channel.

Resolution

\[ j = 1 \]

\[ j = 2 \]

\[ j = 3 \]

Fig.25: (b) Quad tree structure with respect to each tree-structured wavelet transform
2. Clustering:

Clustering or unsupervised classification is defined as finding “natural” grouping in a set of measurement where a certain measurement vector represents properties or attributes of some underlying set of classes. The data from a multichannel image tend to cluster within the corresponding class in a feature-space diagram. The objective of cluster analysis is to divide a given data set into subsets or clusters. However, there is no empty cluster, and every sample must be classified. In hard c-partition clustering, a sample belongs to one and only one cluster, without any distinction between samples close to their cluster center and mixed samples, which may be close to the cluster border. While the fuzzy-C-partition methods are based on a notion of similarity, which is called membership between samples and clusters. They can deal with mixed samples. The membership values calculated using the fuzzy C-means algorithm would all be real numbers between 0 and 1. The closer the value comes to 1, the higher the membership of the object. By using the fuzzy C-means method, there could be a classification of the input data based on Euclidean distance between them. However, the use of the Euclidean norm implies easier calculations than other norms.

(a) :Original Image
(b) :Attribution Space
The input pattern is considered as N-dimensional vectors. Then the C cluster centers $v_i, i=1 \ldots C$ is considered as initial classes. The Euclidean distance $d(x_j, v_i)$ between pixels $x_j$ and cluster centers $v_i$ is computed by:

$$d(x_j, v_i) = ((x_{jl}, x_{il})^2 + (x_{jl}, x_{il})^2 + \ldots \ldots)^{1/2}$$

The membership $P_j$ for each pixel is equated as follows:

$$g_{ij} = \frac{1}{d^2(x_j, v_i)}$$

$$G_j = \sum_{i=1}^{K} g_{ij}$$

$$\mu_{ij} = \begin{cases} 
g_{ij} / G_j & \text{if } x_j \in v_i \\
1 & \text{if } x_j \in v_i \end{cases}$$

Then, recompute the estimates for new cluster centers $v_i$:

$$v_i = \frac{\sum_{j=1}^{M} \mu_{ij}^2 x_j}{\sum_{j=1}^{M} \mu_{ij}^2}$$

These processes are repeated until the $v_i$'s are consistent.

3. Image Segmentation Procedure Using Multiresolution Analysis:

A window size image is extracted from the input image. The extracted window-size image is decomposed into four subimages using a bank of filters. Then, a statistical parameter is applied as the textural measure to each subimage. If the textural measure of a subimage is greater than others, then the decomposition is being
preceded in this region since it contains more information. The decomposition is
repeated until the minimal size of the subimage is being exceeded. The size of the
smallest subimage should not be less than $8 \times 8$ pixels, since if they have narrow size;
the location and the feature may vary widely so that the feature may not be robust.

Next, as shown in the figure below, the window image is moved one
pixel, and the decomposition process is performed again. These steps are repeated
continuously to cover the whole image. This results in a set of feature images that
contains a set of feature vectors. These feature vectors are assumed to capture and
characterize different scales of textures from the input image.

Then, all the input pixels of the feature images are classified based on
their associated vector values by using clustering algorithm. This will result in a
segmented image.

![Image segmentation using multiresolution analysis](image)

*Fig. 27: Image segmentation using multiresolution analysis.*
4. Application:

The selection of the moving window depends on the size of the repetitive structure in the textural region. The smaller window size will make the textural feature not robust, while the greater window size will lead to higher computational cost.

The use of $32 \times 32$ pixels moving window size gives a very satisfactory result, and a 2-level tree-structure wavelet transform is appropriate. Moreover, the results are largely independent of the choice of filter banks.

![Fig.28: A test image 256 x 256 pixels in size, generated by combining four Brodatz-like microtextures](image)

Nevertheless, the average of the energy distribution feature as the textural measure gives the maximum value always on the low frequency channel. The feature images with a depth of two are shown in the figure below.
The figure below shows the final result of the segmentation process into four classes. The segmentation process is evident from the single cluster representation and is visually satisfying. All textures have been correctly identified. The small incorrectness is shown at the joint border of different textures, which is caused by the consequence of the shifting window. When it reaches to the middle of different textures, the energy features calculated are taken from more than one texture. The result will correspond to the texture with the nearest feature. [24]

4.3.4. Markov Random Field Model and Segmentation

The segmentation with directed trees combined with textural features provides us with a robust method of documented results. It is expanded by identifying a suitable measure of uncertainty and a sensitivity analysis of data changes and thresholds.

Markov Random Field model parameters as textural features will be used to do classification of different levels of textural hierarchy. It is possible not only to
identify segments but areas made of segments. For example, it is possible to derive a division between suburban area and cultivated land area by constructing a hierarchical MRF model (Markov Random Field).

The segmentation algorithm used is as follows:

The following segmentation method is adapted from the segmentation method by directed trees (Narendra and Goldberg, 1980) to multispectral SPOT and Landsat TM images by introduction of the relative calibration of the different channels and a selection of boundary templates for optimal local edge enhancement.

- **THE CALIBRATION OF THE CHANNELS**

\[
I_k(x, y) = \frac{I_{0k}(x, y) - m_{0k}}{\sigma_{0k}} \ast \sigma + m
\]

Calibration of the different channels to identical mean and standard deviation

- channel nr, \( I_0 \): original image, \( m_0, \sigma_0 \): mean and standard deviation of original image, \( m_r, \sigma_r \): mean and standard deviation of resulting image

- **EDGE MAGNITUDE CALCULATION**

The maximal edge magnitude over different channels, over different boundary templates and over different dates, if multiday imagery used

\[
G(x, y) = \max_{k=1, m=1}^{nch, nm} \left| G_{k,m}(x, y) \right|
\]

\[
G_{k,m}(x, y) = \Sigma_{i=-1}^{1} \Sigma_{j=-1}^{1} (M_{m}(i, j) \ast I_k(x+i, y+j))
\]

\( M \): boundary template, \( nm \): number of different boundary templates, \( nch \): number of channels
• **THE LINKING OF THE PIXELS**

\[
D(x, y) = \max_{k=1, m-1} \left| G(x, y) - G(x+1, y+1) \right|
\]

If \( D(x, y) \leq -e \): evident root pixel, no linking

\( D(x, y) \geq e \): evident boundary pixel, linked to the neighbouring pixel with minimum edge magnitude

\(-e < D(x, y) < e \): linked to the neighbouring pixel with minimum edge magnitude, without creating loops

\( e \): user given threshold

• **THE LABELING OF THE SEGMENTS**

The resulting trees are traced and individual labels assigned to the pixels of the trees, i.e.: segments.

![Image](image.png)

**Fig.30: Segmentation using Markov random field**

Segmentation was applied to the Axholm (small field agriculture). The result of the Axholm image show good delineation for the agricultural fields, but problems exists with seminatural vegetation delineation with slow land cover transitions. [18,28]
CHAPTER 5

IMPLEMENTATION OF THE ALGORITHM

5.1. INTRODUCTION

Recently, the computer vision literature had found a way to make a full use of the Gibbs distribution for characterizing Markov Random Field in image processing. New approaches were presented to the application of Gibbs distribution for problems in image processing. A method for modeling image data using such models is presented. Then an algorithm is derived for segmenting textured images using a maximum a posteriori (MAP) criterion.

The model that is used here for image data is hierarchical in nature. At higher levels a Gibbs distribution is used to characterize the clustering of image pixels into regions with similar features. While at lower level, a second set of Gibbs distribution is used to model the textural properties. Moreover, these models can be used for very regular and periodic looking textures.

The main purpose is to develop a programming approach to segmenting images whose region sizes and shapes are characterized by a Gibbs distribution. The algorithm requires a scan over the image and produces an approximation to the MAP estimate.

However, a difficulty in using the Gibbs distribution as a region model or texture model is in estimating the parameters of the model from specific realizations. A new method was developed which allows for use of standard, linear, least-squares estimation.
5.2. GIBBS DISTRIBUTION

The basic definition of Gibbs distribution and a particular class of it that is used in the image models is presented below.

5.2.1. Definition

The focus will be over a finite $N_1 \times N_2$ rectangular lattice of pixels where:

$$L = \{(i,j): 1 \leq i \leq N_1, 1 \leq j \leq N_2 \}$$

- First Definition: A collection of subsets of $L$ such as
  $$\eta = \{\eta_{ij}: (i,j) \in L, \eta_{ij} \subseteq L\}$$
  where
  1. $(i,j) \not\in \eta_{ij}$, i.e.: A site is not neighboring to itself.
  2. If $(u,v) \in \eta_{ij}$, then $(i,j) \in \eta_{uv}$ for any $(i,j) \in L$.

  i.e.: The neighboring relation is mutual.

In the first order neighborhood system, also called the 4-neighborhood system, every (interior) site has four neighbors, as shown in Figure 31(a) where x denotes the considered site and 0's its neighbors. In the second order neighborhood system, also called the 8-neighborhood system, there are eight neighbors for every (interior) site, as shown in Figure 31 (b). The numbers $n=1, \ldots, 5$ shown in Figure 31 (c) indicate the outermost neighboring sites in the $n^{th}$ order neighborhood system. The shape of a neighbor set may be described as the hull enclosing all the sites in the set.
Fig.3.1: Neighborhood (a,b,c)---(d,e) are clique types in \( \eta_1 \) where (a) is \( \eta_1 \) neighborhood system---
(d,e,f,g,h) are clique types in \( \eta_2 \) where (b) is \( \eta_2 \) neighborhood system.

- **Second Definition**: A clique of the \((L, \eta)\), denoted by \( c \), is a subset of \( L \) such that
  1. \( c \) consists of a single pixel
  2. or for \((i,j)\) not equal to \((u,v)\) / \((i,j) \in c \) and \((u,v) \in c\)

\[
(i,j) \in \eta_{\infty} \]

All the cliques collection of \((L, \eta)\) is denoted by \( C = C(L, \eta) \).
The type of cliques associated with \( \eta_1 \) and \( \eta_2 \) are shown in the above figure.

- **Third Definition (Definition of Gibbs Distribution)**:
  A random field \( X = \{X_{ij}\} \)
defined on \( L \) (lattice) has Gibbs distribution with respect to \( \eta \) if and only if its joint distribution is of the form

\[
P(X = x) = \frac{1}{Z} e^{-U(x)}
\]

Where

\[
U(x) = \sum V_c(x) \text{ energy function.}
\]
$V_c(x) = \text{potential associated with clique } c.$

$Z = \sum e^{-U(x)} \text{ partition function.}$

The clique potential $V_c(x)$ depends only on the pixel values in clique $c$. While $U(x)$ is the sum of clique potential over all possible cliques $c$. Frequently, the exponent is expressed as $-(1/T) U^I(x)$ where $T$ is called the “temperature” and $U^I(x)$ is the energy function. When $T$ is large (system is hot), the distribution $P(X=x)$ is less peaked (many realizations is highly probable). [29,30,31]

Gibbs distribution is an exponential distribution. However, by choosing the clique potential function $V_c(x)$ properly, a wide variety of distributions both for discrete and continuous random fields can be formulated as Gibbs distribution.

Any random field is viewed as Markov random field, and consequently as Gibbs random field. However, the smallest neighborhood system that indicates GRF (or MRF) is very flexible and powerful.

5.2.2. Class of Gibbs Distribution

A particular class of Gibbs distribution is presented in this section. It is used to model the region process and textures. We assume that the random field $X$ consists of $M$-valued discrete random variables $\{X_{ij}\}$ taking value in $Q=\{q_1, q_2, q_3, \ldots, q_M\}$. To specify the neighborhood system $\eta$, cliques and clique potentials is sufficient to define Gibbs distribution. However, the random fields are assumed to be homogeneous, that is, the clique potentials depend on the clique type and the pixel values in the clique, but not on the position of the clique in the lattice (specified in terms of $\eta^3$).

A parameter is assigned to each clique type:

(i) ** : $\beta_1$, (ii) * : $\beta_2$, (iii) ** : $\beta_3$, (iv) * : $\beta_4$. 
The clique potentials are:

\[ V_c(x) = -\xi \] if all \( x_{ij} \) in \( c \) are equal.

Otherwise,

\[ V_c(x) = \xi. \]

Where \( \xi \) is a parameter specified for the clique type \( c \).

According to the single pixel cliques, the clique potential is:

\[ V_c(x) = \alpha_u \text{ for } x_{ij} = q_u \]

Where \( \alpha_u \) parameters control the percentage of the pixels region type. [30]

5.3. HIERARCHICAL IMAGE MODELING

The hierarchical Gibbsian model is introduced to represent textured image data. Nevertheless, the problem of segmentation of these images and estimation of model parameters are also introduced.

5.3.1. Hierarchical Gibbsian Model For Image Data

As we previously introduced that a digital image \( y = \{ y_{ij} \} \) as an \( N_1 \times N_2 \) matrix of observation. The objective is to specify the random field \( Y \) so that it characterizes the class of textured image.

5.3.1.1. High Level Process: Each region type can occur in more than one location within the lattice. For example, a Land-sat image can consist of numerous bodies of water all characterized by the single region type: water. On this level the Gibbs distribution is used to characterize the spatial clustering of pixels into regions. In this distribution the statistical information refers to size, shape, orientation, and frequency
of regions. The distribution is used to emphasize a spatial continuity. That is, if a pixel is of a certain region type, then neighboring pixels should also have a high probability of being of the same type. Therefore, in this paper, we worked with the second order Gibbs distribution. The pair clique parameter $\beta_i$ are chosen to be nonzero and taken to be positive. [27]

5.3.1.2. Low Level Process: Each region is assumed to consist of each correlated intensity. We intend to interpret the correlation from coming from an underlying region texture. Therefore, we model this spatial dependency by a second level of Gibbs distribution. It is assumed that there are $M$ textures denoted by $T^m = 1, 2, 3, \ldots, M$ each one a discrete valued GRF defined on a lattice. Each random field $T^m$ is assumed to be a member of the class of Gibbs distribution. $r^m_i$ denotes the number of gray-levels in texture $T^m$ and $\{g^m_i\}_{i=1}^{r^m_i}$ are the actual gray-levels in $T^m$. While $t^m$ denotes a particular realization from $T^m$. Moreover, it is assumed that the region process $X$ and the textures $T^m$ are mutually independent fields. [27,30]

Then the two levels of Gibbs distribution were combined to arrive at the hierarchical Gibbsian model that is being proposed for textured images. Then the realization $y$ is specified as follows: at each pixel it takes the value of the texture specified by the value of $x$ at that pixel, which concludes the construction of the hierarchical Gibbsian model used for textured images.

5.3.2. Segmentation

The aim of this paper is to develop a segmentation algorithm for textured images that are represented by the hierarchical Gibbsian model. That is, given a
textured image realization $y$, which determines the scene realization $x$, although it is not observed and cannot be determined from $y$. Therefore, the problem is to obtain an estimate $x^* = X^*(y)$ of the scene $X$ which is based on a realization $y$. We adopted the maximum a posteriori (MAP) estimation as a statistical criterion. So, the objective is to have an algorithm which yields $x^*$ that maximizes the a posteriori distribution $P(X=x | Y=y)$ for a given $y$. [28]

5.3.3. Parameter Estimation

The segmentation algorithms presented in the following section require that the model parameters are known or estimated prior to segmentation, such as the parameters of the high level and low level Gibbs distribution, and the region intensities (gray levels).

5.4. MAP ALGORITHM

The MAP segmentation involves $x^*$, an estimate for the scene that maximizes the a posteriori distribution $P (X=x | Y=y)$ with respect to $x$. According to Bayes' rule:

$$P(X=x | Y=y) = \frac{P(Y=y | X=x) P(X=x)}{P(Y=y)}$$

Since $y$ does not affect the maximization process, it is equivalent to maximize the process of the right hand side of the above equation.

That is:

$$\ln P(X=x | Y=y) = \ln P(X=x) + \ln P(Y=y | X=x)$$

The first term in the right hand side of the above formula is due to high level spatial interaction model and the second is due to the textured data. Therefore, in determining
\( x^* \) is due to the fact that the maximization of the above formula is done over \( M^{N_1N_2} \) possible configurations, where \( M=4 \).

### 5.4.1. Textured Region

Here, it represents the MAP segmentation algorithm. The objective is to find \( x^* \). However, one has to revert to approximation in determining \( x^* \), the scene estimate that maximizes \( P(\mathbf{X}=x, \mathbf{Y}=y) \).

Then, by processing narrow strips of \( D \) rows and \( N_2 \) columns (\( D \) is optimum at 3) of the image, and the algorithm is run over a \( D \)-row strip of the image of a manageable \( M^D \) state dynamic programming, obtaining a MAP estimate over the strip based on the image data over the strip. To obtain segmentation, the processor of the strip is used as follows. First \( D \)-row strip (3-row strip) is processed and an estimate for the strip is obtained. The segmentation of the first row is kept and the rest discarded. Then the strip consisting of rows 2 through \( D+1 \) is processed with the estimate of row 1 used as a boundary condition. Then, the estimate of row 2 is kept and the rest discarded. Then we go to the next strip with row 2 as the boundary condition and so on until the strip consisting of the last \( D \) rows is processed of which the whole estimate is kept. Then. The segmentation of the whole image is obtained. Such that:

1. **Step 1:** Let \( D = 3 \).
2. **Step 2:** Set \( I = 1 \).
3. **Step 3:** Apply the strip processor to rows \( I \) through \( I + D - 1 \) (i.e.: \( I + 2 \))
4. **Step 4:** If \( I + D - 1 = N_1 \), store the segmentation for rows \( I \) through \( I + D - 1 \)
   - And go to step 8.
5. **Step 5:** Store the segmentation for row \( I \) and discard the rest.
Step 6: Set \( I = I + 1 \).

Step 7: Go to step 2.

Step 8: Stop.

However, we have to obtain an expression for \( P(Y = y, X = x) \) that is easy to control in a recursive formulation.

\[
P(Y = y | X = x) = \frac{P(X = x | Y = y) P(Y = y)}{P(X = x)}
\]

The following assumptions are done:

i. \( P(X = x) = \prod_{(i,j) \in L} P(X_{ij} = x_{ij}) \)

ii. \( P(X = x | Y = y) = \prod_{(i,j) \in L} P(X_{ij} = x_{ij} | Y = y) \)

iii. \( P(X_{ij} = x_{ij}, Y = y) = \prod_{(i,j) \in L} P(X_{ij} = x_{ij}, Y_{ij} = y_{ij}) \)

\( v_{ij} \) is a neighborhood of \((i,j)\) including \((i,j)\). \( X_{v_{ij}} \) and \( y_{v_{ij}} \) are restrictions of the image random field and its realization at \( v_{ij} \).

Equation (iii) is very reasonable and accepts the assumption that considers that the distribution of \( X_{ij} \), given \( Y = y \), depends only on the data that are close to pixel \((i,j)\).

By substitution:

\[
P(Y = y | X = x) = P(Y = y) \prod_{(i,j) \in L} \frac{P(X_{ij} = x_{ij} | Y_{ij} = y_{ij})}{P(X_{ij} = x_{ij})}
\]

By multiplying and dividing each term by \( P(Y_{v_{ij}} = y_{v_{ij}}) \) in the above equation:

\[
P(Y = y | X = x) = \frac{P(Y = y)}{P(Y_{v_{ij}} = y_{v_{ij}})} \prod_{(i,j) \in L} P(Y_{v_{ij}} = y_{v_{ij}} | X_{ij} = x_{ij})
\]

Then the log-likelihood for the data component is:

\[
\ln P(Y = y | X = x) = \ln k_0 + \sum_{m=1}^{M} \sum_{(i,k) \in S_m} \ln P(Y_{v_{ij}} = y_{v_{ij}} | X_{ij} = x_{ij})
\]
Since in the above equation the fraction term is independent from \(\{x_{ij}\}\), that is why it was treated as a constant \(k_0\) (for maximization with respect to \(x\)).

Then the algorithm is:

\[
I_0 = -\ln Z + \ln k_0
\]

\[
I_k = I_{k-1} - \sum_{c \in C^{k-1,k}} V_c(x) + \sum_{m=1}^{M} \sum_{(i,k) \in S^m} \ln P(Y_{ij} = y_{ij} | X_{ij} = m)
\]

where \(C^{k-1,k} = \{c: c \text{ is a clique with pixels only in column } k \text{ or in columns } k-1 \text{ and } k\}\). and \(S^k_m = \{(I,k): X_{ik} = m, 1 \leq i \leq N\}\)

Since \(I_{N2} = \ln P(X=x|Y=y)\), therefore we are looking to determine \(x^*\) that maximizes \(I_{N2}\). In the algorithm above, \(I_0\) can be initialized to 0, since it does not effect the maximization. \(\ln P(Y_{ij} = y_{ij} | X_{ij} = x_{ij})\) is also calculated by making use of Gibbs distribution that characterizes the \(M\) distinct texture types. Moreover, \(v_{ik}\) is a relatively small region, for example a \(3 \times 3\) block, where the computation of the partition function \(Z_m\) is a manageable task.

However, the number of gray-levels is not a major restriction, but the number of texture type is restricted to 4, that is, \(M \leq 4\) since segmentation for \(M > 4\) is not applicable computationally.

Moreover, in the above-mentioned algorithm the data window used is a \(3 \times 3\) block that appeared to be satisfactory. However, larger windows, such as, \(5 \times 5\) block can decrease the isolated single pixel errors but it will perhaps increase the errors at the region boundaries, and will use a noticeable increase in computational time.

The classification of each pixel is done based on a likelihood of two components, one due to spatial continuity and the other due to textural coloring on the \(3 \times 3\) data window, which will consist a single region type, while in reality it might consist more than one region type. [29]
5.5. ESTIMATION OF PARAMETERS IN GIBBS DISTRIBUTION

In the following section, it is presented an estimation procedure for Gibbs distribution parameters. This procedure is applicable on samples of textures and not on textured images to be segmented. However, it is assumed that the realization is a noise-free and consists of several textures with unspecific boundaries in between.

5.5.1. Technique for Estimation

The objective in this section is to estimate the parameters of a Gibbs distribution using a realization from this distribution. Therefore, it is proposed a formulation in terms of a second order neighborhood system $\eta_i^2$.

Consider a site $(i,j)$ and its neighboring $\eta_{ij}$. Let $s$ represent $x_{ij}$, and $t^1$ represent the vector of the neighboring values of $x_{ij}$ such that,

$$I = [u_1, u_2, u_3, u_4, v_1, v_2, v_3, v_4]^T$$

i.e.:

<table>
<thead>
<tr>
<th>$v_1$</th>
<th>$u_2$</th>
<th>$v_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>$s$</td>
<td>$u_3$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>$u_4$</td>
<td>$v_3$</td>
</tr>
</tbody>
</table>

Fig.32: Represents $x_{ij}$ and $\eta_{ij}$

$$I(z_1, z_2, \ldots, z_k) = \begin{cases} 
-1 & \text{if } z_1 = z_2 = z_3 = \ldots = z_k \\
1 & \text{otherwise}
\end{cases}$$

and

$$J_m(s) = \begin{cases} 
1 & \text{if } s = q_m \\
0 & \text{otherwise}
\end{cases}$$
Let \( V(s,t^1,\Theta) \) be the sum of all the potential functions of all the cliques that contain \((i,j)\) such that
\[
V(s,t^1,\Theta) = \Sigma_{c\in c} V_c(x)
\]
where \( \Theta \) is a parameter vector of the form:
\[
\Theta = [\alpha_1, \alpha_2, \ldots, \alpha_M, \beta_1, \ldots \beta_4, \gamma_1, \ldots, \gamma_4, \xi_1]^T
\]
By using the clique potentials for this class of Gibbs distribution we can write \( V(s,t^1,\Theta) \) as:
\[
V(s,t^1,\Theta) = \phi^T(s,t^1) \Theta
\]
where:
\[
\phi (s,t^1) = [J_1(s), J_2(s), J_3(s), J_4(s), \ldots \ldots, J_M(s),
(I(s,u_1) + I(s,u_3) + I(s,v_4)), (I(s,u_2) + I(s,u_4) + I(s,v_2))].
\]
\[
(I(s,v_1) + I(s,v_3)), (I(s,u_2, v_2) + I(s, u_1, u_3) + I(s, u_1, v_1)),
(I(s,u_4, v_3) + I(s, u_2, u_3) + I(s, u_1, v_1)),
(I(s,u_2, v_1) + I(s, u_1, u_4) + I(s, u_3, v_3)),
(I(s,u_4, v_4) + I(s, u_1, u_2) + \ldots \ldots + I(s, u_3, u_4))]
\]
\[
(I(s, u_1, v_1, u_2) + I(s, u_2, v_2, u_3) + I(s, u_3, v_3, u_4) + I(s, u_4, v_4, u_1))^T
\]
However, by using prior knowledge, one may set the parameters to zero wishing to shorten the length of \( \Theta \) and \( \phi (s,t^1) \). For example, the third and fourth pixel cliques can be discarded, such that, as if we are working with the first order neighborhood system, in which the vectors are shortened. [29]

5.6. SIMULATION RESULTS

In this part, I applied the segmentation algorithm for textured images modeled and generated according to the hierarchical model described above.
However, it is obvious from the figure below that the segmentation algorithm for textured images is effective in obtaining good segmentation results. Moreover, as previously mentioned, that this algorithm is implemented with known or assumed Gibbs distribution parameters and for noise-free textured images.

fig. 33: (a) Original image; (b) textured image with $\beta_1=\beta_3=1, \beta_2=1.2, \beta_4=1.3$; (c) textured image with $\beta_1=\beta_2=\beta_3=\beta_4=2$; (d) textured image with $\beta_1=\beta_3=0, \beta_2=0.2, \beta_4=0.3$;

Nevertheless, as we decrease the values of the $\beta$'s and approach them to zero, we will get vague images. While when the $\beta$'s are around 1, the images will be in a better presentation. However, when the $\beta$'s are around 2, we will get a very good representation as we can observe in figure 33. For $\beta$'s above 2, we will get almost the same results as when the $\beta$’s are equal to 2 and all of this is due to the energy function.
Another example was an image of a sitting room. This image contains a little bit of noise which makes it harder for the simulation to obtain a good image.

Fig. 34: (a) an image of a sitting room. (b) Texture segmentation of four segments. (c) Texture segmentation where $\beta_1=3$. (d) Texture segmentation where $\beta_1=6$. (e) Texture segmentation where $\beta_1=1$. (f) Texture segmentation where $\beta_1=0$. (Note that in each test different colors were used.)
CHAPTER 6

CONCLUSION

In this research I discussed the theoretical foundations of digital image and its segmentation, and described several known segmentation methods. Image segmentation is an essential preliminary step in most automatic pictorial pattern recognition and scene analysis problems. I also introduced texture analysis, which was described with some several known methods. The results of the studies prove that the methods that were discussed could be of high interest in practical segmentation and classification.

Moreover, I combined the two subjects together and described the textural segmentation and went over some of its methods. The choice of one method over another is dictated mostly by the peculiar characteristics of the problem being considered.

Since texture does exist, it can be segmented to some extent. My aim through this research was to represent a new approach to modeling and segmentation of textured images. This approach assumes the pixel values belong to a probability distribution and to estimate the parameters of the distribution. It involves local computations over large windows in an image. The hierarchical Gibbs distribution model specifies a class of image models, which can be used to represent a wide range of textured images. The parameter estimation procedure allowed us to model specific images within the class.
The segmentation algorithm defined show how such models could be used to obtain solutions to important image processing problems such as the segmentation of textured images.
BIBLIOGRAPHY


[22] Computer Vision Group, *Unsupervised Texture Segmentation*, University Bonn.


