

SEGMENTATION OF HUMAN OVARIAN FOLLICLES FROM
ULTRASOUND IMAGES ACQUIRED *in Vivo* USING
GEOMETRIC ACTIVE CONTOUR MODELS AND A NAÏVE
BAYES CLASSIFIER

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Na Harrington

©Na Harrington, September 2007. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

Ovarian follicles are spherical structures inside the ovaries which contain developing eggs. Monitoring the development of follicles is necessary for both gynecological medicine (ovarian diseases diagnosis and infertility treatment), and veterinary medicine (determining when to introduce superstimulation in cattle, or dividing herds into different stages in the estrous cycle).

Ultrasound imaging provides a non-invasive method for monitoring follicles. However, manually detecting follicles from ovarian ultrasound images is time consuming and sensitive to the observer's experience. Existing (semi-) automatic follicle segmentation techniques show the power of automation, but are not widely used due to their limited success.

A new automated follicle segmentation method is introduced in this thesis. Human ovarian images acquired *in vivo* were smoothed using an adaptive neighbourhood median filter. Dark regions were initially segmented using geometric active contour models. Only part of these segmented dark regions were true follicles. A naïve Bayes classifier was applied to determine whether each segmented dark region was a true follicle or not.

The Hausdorff distance between contours of the automatically segmented regions and the gold standard was $2.43 \pm 1.46 \text{ mm}$ per follicle, and the average root mean square distance per follicle was $0.86 \pm 0.49 \text{ mm}$. Both the average Hausdorff distance and the root mean square distance were larger than those reported in other follicle segmentation algorithms. The mean absolute distance between contours of the automatically segmented regions and the gold standard was $0.75 \pm 0.32 \text{ mm}$, which was below that reported in other follicle segmentation algorithms.

The overall follicle recognition rate was 33% to 35%; and the overall image misidentification rate was 23% to 33%. If only follicles with diameter greater than or equal to 3 mm were considered, the follicle recognition rate increased to 60% to 63%, and the follicle misidentification rate increased slightly to 24% to 34%. The proposed follicle segmentation method is proved to be accurate in detecting a large number of follicles with diameter greater than or equal to 3 mm.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Mark G. Eramian, for his guidance, support, and encouragement. I would also like to thank Dr. Roger A. Pierson of Royal University Hospital for providing the ultrasound data used in this research, and for providing manual segmentation results.

All the ultrasound images used in this study were collected at the Women's Health Imaging Research Laboratory, Department of Obstetrics Gynecology and Reproductive Sciences, Royal University Hospital, Saskatoon, Saskatchewan, Canada. Special thanks to all the women who volunteered to have their ovarian images taken.

I would like to acknowledge Dr. Baris Sumengen of University of California for sharing his level set toolbox that was used to implement my research software.

To my family.

CONTENTS

Permission to Use	i
Abstract	ii
Acknowledgements	iii
Contents	v
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Problem Description	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Proposed Solution	2
1.5 Ovarian Ultrasound Imaging	2
1.6 Thesis Outline	3
2 Image Processing Preliminaries	4
2.1 Grayscale Image Representation	4
2.2 Linear Filtering	4
2.2.1 Convolution	4
2.2.2 Gradient	5
2.2.3 Gaussian Filters	5
2.3 Nonlinear Filtering	6
2.3.1 Median Filters	6
2.4 Basic Morphological Operations	7
2.4.1 Binary Dilation	7
2.4.2 Binary Erosion	8
2.4.3 Opening	8
2.4.4 Closing	8
2.5 Region Descriptors	9
2.5.1 Compactness	9
2.5.2 Eccentricity	10
3 Literature Review	11
3.1 Image Segmentation Techniques	11
3.1.1 Thresholding	11
3.1.2 Automatic Thresholding	12
3.1.3 Region Growing	12
3.1.4 Watershed Segmentation	14
3.1.5 Parametric Active Contour Models	17
3.1.6 Geometric Active Contour Models	24
3.2 Classifiers	29
3.2.1 Nearest Neighbour Classifier	30
3.2.2 k -nearest Neighbour Classifier	30
3.2.3 Minimum Distance Classifier	30
3.2.4 Tree-based Classifier	31

3.2.5	Naïve Bayes Classifier	31
3.2.6	Clustering	34
3.3	Classifier Training and Validation Methodologies	34
3.3.1	Half-and-half	34
3.3.2	<i>N</i> -way cross validation	35
3.3.3	Leave-one-out	35
3.4	Validation	35
3.4.1	Segmentation Validation	35
3.4.2	Recognition Validation	37
3.5	Existing Follicle Segmentation Techniques	39
3.5.1	Thresholding	39
3.5.2	Graph Searching	40
3.5.3	Watershed Segmentation	40
3.5.4	Region Growing	42
3.5.5	Cellular Neural Networks	42
4	Methodology	45
4.1	Scale	46
4.2	Pre-processing	46
4.2.1	Noise Reduction	47
4.2.2	Sub-image Extraction	48
4.3	Segmentation	48
4.3.1	Initial Contours	49
4.3.2	Speed Function	49
4.3.3	Evolution	50
4.4	Post-processing	50
4.4.1	Hole Filling	50
4.4.2	Remove Regions Which Touch The Fan Border	51
4.4.3	Separation of Large Regions	51
4.5	Classification of Potential Follicle Regions	52
4.5.1	Feature Extraction	54
4.5.2	The Classifier	54
4.5.3	Training and Testing	55
5	Experiments & Results	58
5.1	Experiment Design	58
5.1.1	Image Data	58
5.1.2	Hypothesis Testing: Comparing Segmented Images	58
5.1.3	Evaluation	59
5.2	Results & Analysis	61
5.2.1	Segmentation Results & Analysis	61
5.2.2	Potential Follicle Classification Results & Analysis	62
5.2.3	Performance Statistics	67
5.2.4	Analysis of Overall Results	68
5.3	Failure Case Analysis	68
5.3.1	Follicles Leaking to The Fan Border	68
5.3.2	Multiple Follicles Are Treated as One	69
5.3.3	Over-Splitting	70
5.3.4	Inexpressive Follicles	72
6	Conclusion	73
6.1	Future Work	74
	References	77

A	Sample Segmentation Results	78
B	Experiment Results	80

LIST OF TABLES

3.1	An overview of the performance of all existing follicle segmentation techniques.	44
5.1	The image recognition rate (IRR), image misidentification rate (IMR), follicle recognition rate (FRR) and follicle misidentification rate (FMR) for follicles with diameters (D) greater than or equal to different values before classification.	61
5.2	A review of the Hausdorff distance (HD), mean average distance (MAD), root mean square distance (RMSD) of existing follicle segmentation algorithms, as well as the proposed follicle segmentation algorithm.	62
5.3	The IRR, IMR, FRR, and FMR before and after classification using training set 1.	63
5.4	The IRR, IMR, FRR, and FMR before and after classification using training set 2.	64
5.5	The IRR, IMR, FRR, and FMR before and after classification using training set 3.	65
5.6	The IRR, IMR, FRR, and FMR before and after classification using training set 4.	66
5.7	The HD, MAD, RMSD between contours drawn by an expert and contours of automatically segmented regions.	67
B.1	Evaluation for automatic segmentation result and gold standard: A5 and G1 - G5.	80
B.2	Evaluation for automatic segmentation result and gold standard: A1 and G1 - G5.	80
B.3	Evaluation for automatic segmentation result and gold standard: A2 and G1 - G5.	81
B.4	Evaluation for automatic segmentation result and gold standard: A3 and G1 - G5.	81
B.5	Evaluation for automatic segmentation result and gold standard: A4 and G1 - G5.	81

LIST OF FIGURES

1.1	Segmenting follicles from ovarian ultrasound images acquired <i>in vivo</i>	2
1.2	An ultrasound image contains a fan area and a margin area.	3
2.1	An example of discrete gradient approximation.	6
2.2	An example of median filtering. The pixel $I(x, y)$ is replaced by the median of $\{I(i, j) i = x - 1, \dots, x + 1, j = y - 1, \dots, y + 1\}$	6
2.3	An example of binary dilation. Object pixels in (a) are represented as 1 and background pixels are represented as 0. The origin in (b) are marked as x.	7
2.4	An example of binary erosion. Object pixels in (a) are represented as 1 and background pixels are represented as 0. The origin in (b) are marked as x.	8
2.5	An example of opening. Object pixels in (a) are represented as 1 and background pixels are represented as 0. The origin in (b) are marked as x.	9
2.6	An example of closing. Object pixels in (a) are represented as 1 and background pixels are represented as 0. The origin in (b) are marked as x.	9
3.1	An example of thresholding.	11
3.2	An example of region growing.	13
3.3	An example of watershed segmentation	15
3.4	An example of the geodesic distance and geodesic influence zones	16
3.5	Watershed segmentation based on image gradient	16
3.6	Watershed segmentation based on a distance transform.	17
3.7	An example of marker-controlled watershed segmentation.	18
3.8	A example of external energy.	21
3.9	An example of capture range.	22
3.10	Different segmentation results using Gaussian potential force and GVF.	23
3.11	An example of a level set function.	25
3.12	An example of segmenting an object using the level set method.	26
3.13	An example of the signed distance function.	27
3.14	A tree-based classifier for deciding whether to buy a lottery ticket or not.	31
3.15	Same data being binned with different bin width.	33
3.16	An example of DCP from outer to inner contour (a) and from inner to outer contour (b). . . .	37
4.1	An overview of the segmentation and classification methodology.	45
4.2	An ultrasound image showing the scale and markers.	46
4.3	An example of filtering. An original image is filtered by an adaptive neighbourhood filter. . .	47
4.4	An example of sub-image extraction.	48
4.5	Thresholding the filtered image to obtain the initial contour	49
4.6	Holes inside regions are filled in.	51
4.7	Regions which touch the fan border are removed.	51
4.8	An example of separating large regions	52
4.9	An example of eroding a large region until it completely eroded	53
4.10	An example of eroding a large region into two regions.	53
4.11	An example of obtaining the centroid location of potential follicles.	55
4.12	Example of discretizing the compactness feature.	57
5.1	Examples of a good scan and a bad scan.	59
5.2	The cause of edge shadowing.	69
5.3	An example of edge shadowing.	69
5.4	The segmentation algorithm leaked a follicle to touch the fan border because of edge shadowing.	70

5.5	Multiple follicles in gold standard (b) were treated as one in automated segmentation algorithm (a). This happens twice in this particular image (arrows)	71
5.6	The follicle pointed by the arrow in (a) is incorrectly split because of its concavity.	71
5.7	Inexpressive follicles (pointed by arrows) in (b) are too bright to have an initial contour. They are not segmented in (a).	72
A.1	Sample segmentation results - typical good results.	78
A.2	Sample segmentation results - typical poor results.	79

CHAPTER 1

INTRODUCTION

1.1 Problem Description

Ovarian ultrasound imaging provides a non-invasive method for monitoring follicles (sites of egg development). Detecting follicles in an ultrasound image involves two steps: identifying follicle regions (deciding which dark regions are follicles and which are not) and detecting accurate boundaries for the follicle regions. Currently, this process is mainly accomplished manually, which is time consuming and sensitive to the observer's experience. Existing (semi-)automatic follicle segmentation techniques either focus only on detecting accurate boundaries of individual follicles [25, 43], or study both aspects but have limited success [9, 32, 35, 36, 37, 49].

1.2 Motivation

In gynecologic medicine, frequent *in vivo* observation of ovarian follicles is necessary for the diagnosis of ovarian diseases (polycystic ovarian syndrome [27]) and infertility treatments (gamete intrafallopian transfer [8], artificial insemination [48], in vitro fertilization [28], etc). In veterinary medicine, monitoring the growth of ovarian follicles can assist in determining when to introduce superstimulation in cattle [31], or in dividing herds into different stages in the estrous cycle [1].

In vivo ultrasound imaging provides a non-invasive and inexpensive method of acquiring images of ovaries to monitor follicles. Currently, ultrasound images are analyzed by highly trained imaging specialists, and the image assessment is tedious and time-consuming. A first step is to automatically locate and/or measure the visible follicles in order to develop computer assisted techniques for analyzing ovarian ultrasound images. This thesis presents a new automated follicle segmentation algorithm.

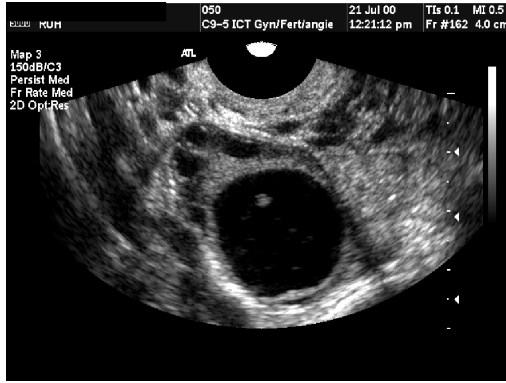
1.3 Objectives

The goal of this research is to test the hypothesis that geometric active contour models can detect a large number of visible follicles in human ovarian ultrasound images acquired *in vivo* with the accuracy greater than that of existing methods when compared to the same gold standard, and the hypothesis that a naïve

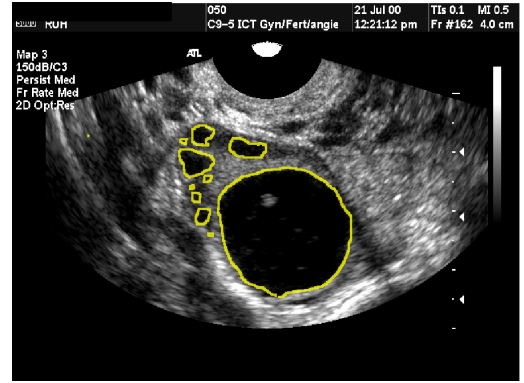
Bayes classifier can subsequently eliminate a large number of false positives to reduce the misidentification rate.

1.4 Proposed Solution

The automatic segmentation algorithm presented here detects the boundaries of **all** dark regions in the ultrasound image, then determines which ones are follicles using a naïve Bayes classifier as opposed to the manual segmentation approach (identification of follicle regions, followed by the detection of follicle boundaries). The image is smoothed using an adaptive neighbourhood median filter. All dark regions (potential follicles) from the image are coarsely initialized by thresholding, and the boundaries of these potential follicles are detected by applying geometric active contour models. A naïve Bayes classifier is applied to determine whether each dark region is a follicle or not. Figure 1.1 is an example of an ideal result.



(a) An original image generated in an ultrasound scan.

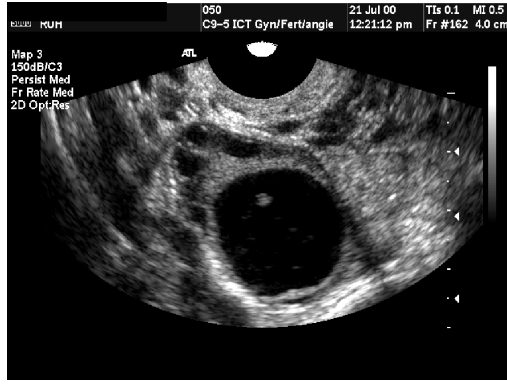


(b) The desired result of a follicle segmentation of (a).

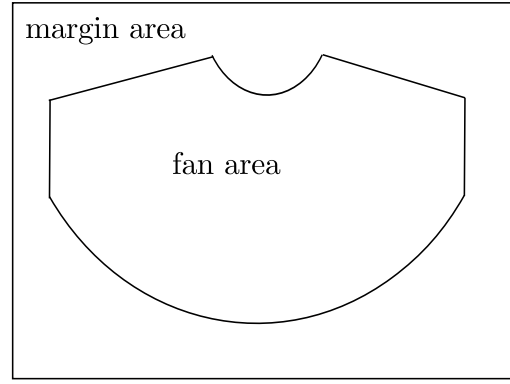
Figure 1.1: Segmenting follicles from ovarian ultrasound images acquired *in vivo*

1.5 Ovarian Ultrasound Imaging

The initial approach for obtaining ovarian images is often performed transabdominally [17]. Because an ovary changes its shape as follicles develop and regress and its position in the pelvis very frequently as intestinal contents shift, some parts of the ovary are not visible transabdominally. The highest resolution ovarian ultrasound images are obtained by transvaginal scanning with a curvilinear-array transducer [17]. Each such image contains a fan shaped area which shows the image of interest obtained by the ultrasound scan and a marginal area containing other information such as the patient's name and date that the image was taken (Fig. 1.2).



(a) An original image.



(b) An image showing fan area and margin area.

Figure 1.2: An ultrasound image contains a fan area and a margin area. The fan area in (a) shows an ovary image obtained by an ultrasound scan. The margin area shows information about the patient, the date that the image was taken, and the ultrasound machine settings

1.6 Thesis Outline

This thesis is organized as follows:

- Chapter 2 presents basic definitions and techniques used later in the thesis.
- Chapter 3 reviews, in detail, relevant image segmentation techniques, commonly used classifiers, classifier training and testing methodologies, and validation techniques used in image segmentation. Also, existing follicle segmentation algorithms are reviewed in this chapter.
- Chapter 4 presents a new follicle segmentation algorithm.
- Chapter 5 describes the ultrasound image data used in this study, and the methodology used to evaluate the proposed follicle segmentation algorithm. The algorithm is evaluated and the results are presented and discussed. Failure cases are analyzed.
- Chapter 6 provides a summary of this research and reviews the experimental results.

CHAPTER 2

IMAGE PROCESSING PRELIMINARIES

Many basic concepts in image processing will be mentioned in this thesis, and some of them are crucial to its understanding. This chapter reviews these concepts.

2.1 Grayscale Image Representation

The images used in this research are grayscale digital images. A digital image is represented as a matrix where the origin is located at the upper left corner, with the positive x-axis extending to the right and positive y-axis extending down [46]. The value of each element (a pixel) in the matrix represents the brightness of the corresponding part in the image. This value is called a graylevel, which ranges from 0 to 255 for all the images used in this research.

Given a grayscale digital image I , any pixel in I is represented as $I(x, y)$, where x and y are matrix indices in the coordinate system described above, and the value of $I(x, y)$ is that pixel's graylevel.

2.2 Linear Filtering

2.2.1 Convolution

Given a digital image, a kernel (a matrix of predetermined elements) is centered over the pixel to be replaced. The corresponding elements in the kernel and the image are multiplied and the sum of all the multiplications is assigned to the pixel to be replaced [42]. The original pixel values are used in the calculation in all cases. Let the image to be filtered be I , and the kernel be K with size $m \times n$, the convolution of I with K is written as

$$J = I * K \quad (2.1)$$

where each element in J is defined as

$$J(x, y) = \sum_{i=1}^m \sum_{j=1}^n I(x - \lceil \frac{m}{2} \rceil + i, y - \lceil \frac{n}{2} \rceil + j) K(i, j) \quad (2.2)$$

In discrete convolution filtering, each pixel in an image is replaced by a linear combination of the intensity of itself and its neighbours using the convolution operator [42]. The kernel is also known as the filter.

2.2.2 Gradient

The gradient (∇I) of a continuous 2D function I is a vector field with magnitude ($|\nabla I|$) and direction (ψ) [46]:

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (2.3)$$

$$\psi = \arg\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) \quad (2.4)$$

where $\arg\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$ represents the direction (in radians) of the vector defined by $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ [46]. When I is a discrete function, one method of approximating the gradient is to use the Sobel gradient [46]:

$$\frac{\partial I}{\partial x} \approx G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \quad (2.5)$$

$$\frac{\partial I}{\partial y} \approx G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I \quad (2.6)$$

where the symbol $*$ denotes the convolution. Then, the gradient magnitude and direction is obtained by

$$|\nabla I| = \sqrt{G_x^2 + G_y^2} \quad (2.7)$$

$$\psi = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.8)$$

An example of the gradient of an image is shown in Fig. 2.1. Image gradient is often used for edge detection [46].

2.2.3 Gaussian Filters

The Gaussian filter smooths an image by convolving a Gaussian-shaped (bell-shaped) kernel, p , with the image [46].

$$p(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2.9)$$

where σ is the standard deviation of the Gaussian function, which determines the degree of smoothing. The Gaussian-shape of the kernel means pixels closer to the pixel to be replaced have more influence on the new pixel value. Gaussian filtering reduce high spatial frequencies, therefore, it blurs the image [46].

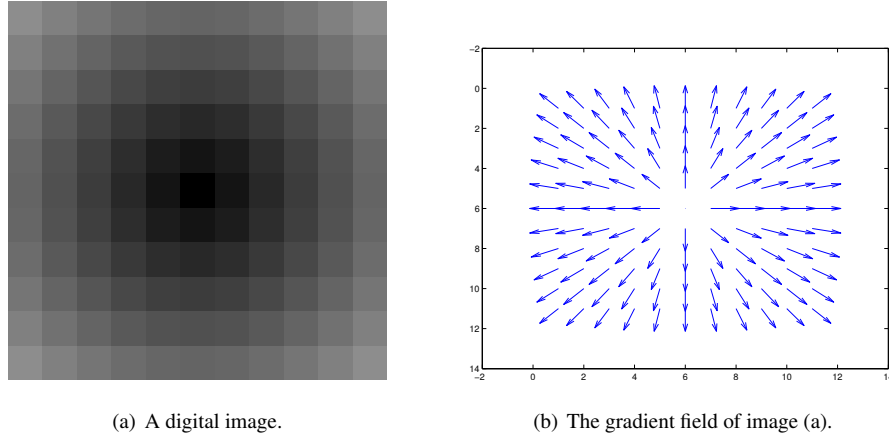


Figure 2.1: An example of discrete gradient approximation. The gradient field of image (a) is represented by arrows in (b). The length of arrows represents the gradient magnitude, and the direction of arrows represents the gradient direction.

2.3 Nonlinear Filtering

2.3.1 Median Filters

Median filtering replaces an image pixel's value with the median pixel value from some set of nearby pixels (a neighbourhood) [46]. Typically the neighbourhood is the set of pixels which fall within a square matrix centered over the pixel to be replaced (Fig. 2.2). Because the median graylevel in a neighbourhood is not affected by individual noise pixels, median filtering removes outliers (impulse noise) while preserving thick edges. However, thin lines and fine textures are removed by the median filter because they are outliers within the filter window.

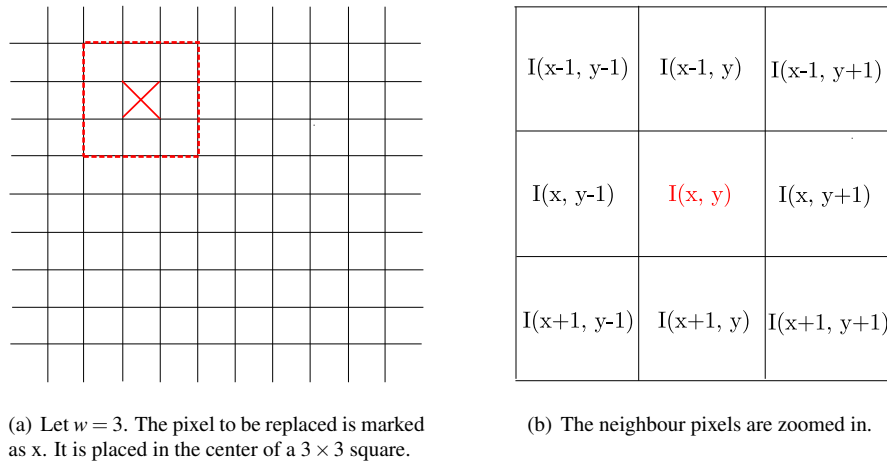


Figure 2.2: An example of median filtering. The pixel $I(x, y)$ is replaced by the median of $\{I(i, j) | i = x - 1, \dots, x + 1, j = y - 1, \dots, y + 1\}$

In traditional median filtering, the size and shape of the neighbourhood remains the same for all pixels in the image. Adaptive neighbourhood median filtering customizes the neighbourhood for every pixel by including only neighbouring pixels with similar properties (such as graylevel, texture) to the pixel to be replaced. This approach avoids including pixels which belong to different homogeneous regions as part of the neighbourhood, therefore, the resulting graylevels are closer to the true values of pixels to be replaced. Adaptive neighbourhood median filtering generally yields better results than ordinary median filtering at the cost of additional computation.

2.4 Basic Morphological Operations

This section introduces the basic morphological operations on binary images. Given a binary image B and a structuring element S (a small binary image with an arbitrary shape that decides the effect of the operations) with a reference point (origin), four basic morphological operations are defined below.

2.4.1 Binary Dilation

Intuitively, dilation uses the structuring element as a “rubber stamp” at locations specified by white pixels in the input image. The indices of black pixels in the image and structuring element are recorded, the dilation is formally defined as the set of pixels whose indices are given by all possible vector additions of the indices of pairs of black pixels, one pixel from the image B , and the other pixel from the structuring element S .

$$B \oplus S = \{c | c = b + s, b \in B, s \in S\} \quad (2.10)$$

Dilation enlarges a region, resulting in the swelling of objects and shrinking of holes inside the objects [3, 45].

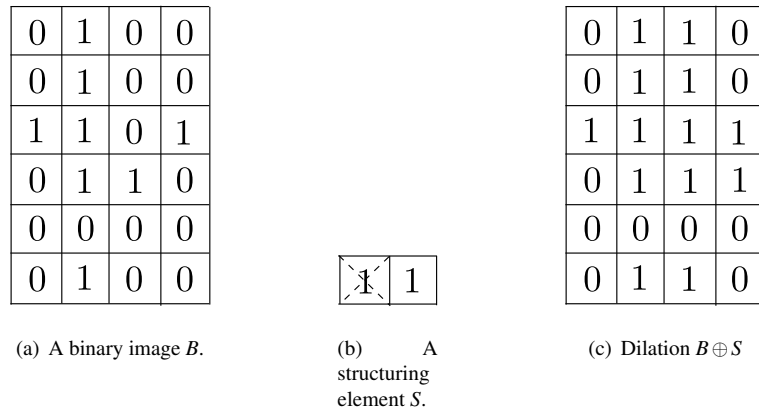


Figure 2.3: An example of binary dilation. Object pixels in (a) are represented as 1 and background pixels are represented as 0. The origin in (b) is marked as x.

2.4.2 Binary Erosion

Every black pixel from the image B is tested. If the vector addition of the index of a pixel (b) and the index of each black pixel in the structuring element S locates a black pixel in the image B , this pixel (b) is a black pixel in the result of erosion (Fig. 2.4) [46].

$$B \ominus S = \{b | b + s \in B \forall s \in S\} \quad (2.11)$$

Erosion shrinks objects and swells holes inside objects [3].

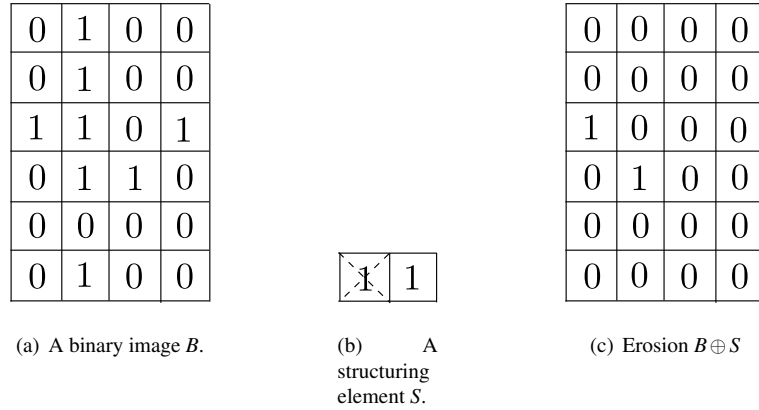


Figure 2.4: An example of binary erosion. Object pixels in (a) are represented as 1 and background pixels are represented as 0. The origin in (b) are marked as x.

2.4.3 Opening

An opening is defined as an erosion followed by a dilation using the same structuring element (Fig. 2.5) [3].

$$B \circ S = (B \ominus S) \oplus S \quad (2.12)$$

The opening operation removes small structures that do not completely enclose the structuring element. It is often used to remove thin connections between objects [3].

2.4.4 Closing

A closing is a dilation followed by an erosion with the same structuring element (Fig. 2.6) [3].

$$B \bullet S = (B \oplus S) \ominus S \quad (2.13)$$

The closing operation fills in the part of background where it does not contain the entire the structuring element. It is often used to connect objects that are close together, and fill in small holes [46].

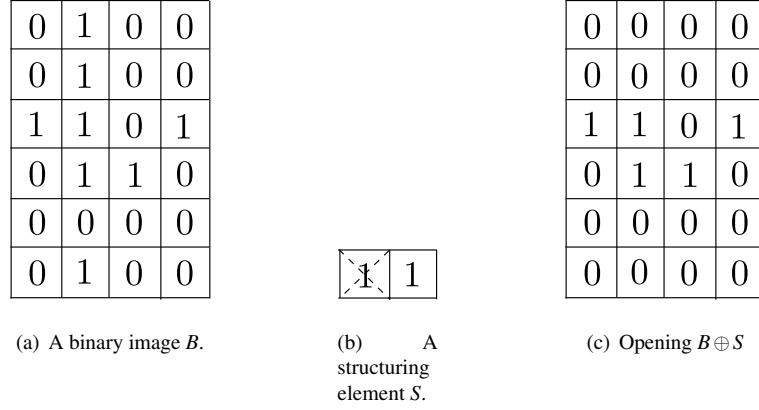


Figure 2.5: An example of opening. Object pixels in (a) are represented as 1 and back-ground pixels are represented as 0. The origin in (b) are marked as x.

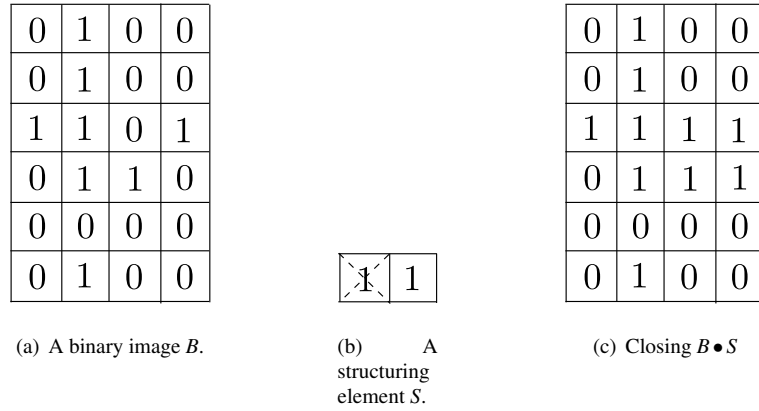


Figure 2.6: An example of closing. Object pixels in (a) are represented as 1 and back-ground pixels are represented as 0. The origin in (b) are marked as x.

2.5 Region Descriptors

Region descriptors describe a set of pixels in simpler terms such as scalar or vector quantities. Two scalar region features are introduced below.

2.5.1 Compactness

If a man was given a fixed length of rope and awarded any land that he could encircle with it, the compactness would describe his level of regret after he is shown the maximum area of land he could have encircled with the same rope. Formally, the compactness of a region is defined as the ratio of the square of the perimeter of a region to its area [46]:

$$\text{compactness} = \frac{\text{perimeter}^2}{\text{area}} \quad (2.14)$$

A lower value of compactness indicates that the region is more compact, with a circle having the lowest possible compactness value (4π in continuous case and 4 in digital images using the distance around the inner boundary as the perimeter).

2.5.2 Eccentricity

Let AB be a region's longest chord (major axes), and CD be the longest chord of the same region that is perpendicular to AB (minor axes), the eccentricity of this region is the ratio of the length of AB to the length of CD [46]

$$\text{eccentricity} = \frac{|CD|}{|AB|} \quad (2.15)$$

The value of eccentricity is between 0 and 1. A larger value of eccentricity indicates that the length of the minor axes of a region is closer to the length of the region's major axes.

CHAPTER 3

LITERATURE REVIEW

3.1 Image Segmentation Techniques

Image segmentation divides an image into meaningful regions correlated with objects in the real world [38, 46]. This section reviews a selection of image segmentation techniques in a general context.

3.1.1 Thresholding

Given a grayscale image I and a threshold value T , thresholding maps pixels with intensity higher than or equal to T to 1, and pixels with intensity lower than T to 0 (Equation 3.1). A binary image is usually used to distinguish the background and foreground (Fig. 3.1) [46].

$$J(i, j) = \begin{cases} 1 & \text{if } I(i, j) \geq T \\ 0 & \text{if } I(i, j) < T \end{cases} \quad (3.1)$$

Thresholding performs well when objects do not touch each other, and objects graylevels are distinct from background graylevels.



(a) A gray-scale image.



(b) Pixels with graylevel above than 75 are considered as part of the object (orchid flowers).

Figure 3.1: An example of thresholding. There is a considerable difference between gray levels of orchids and gray levels of the background. Orchid flowers are segmented by thresholding.

3.1.2 Automatic Thresholding

The thresholding result is sensitive to the choice of the threshold value. A high threshold value will treat some object pixels as background, and a low threshold value will treat some background pixels as objects. This sub-section introduces some methods for determining threshold values.

3.1.2.1 Ratio-based Detection

If the ratio (p) between the area of background and the area of objects is known as *a priori* knowledge, the threshold value T may be chosen so that $\frac{1}{p}$ of the image pixels have graylevel less than T [46]. This is a good method to obtain the threshold value. However, such *a priori* information is usually unavailable.

3.1.2.2 Histogram-based Detection

The threshold value could be estimated from analyzing the histogram shape of an image. If an image is composed of regions with distinct graylevel, the shape of its histogram is generally characterized by two or more peaks (bi-modal or multi-modal). In the case of bi-modal, the threshold value is at the bottom of the valley between two peaks. Otsu's method is a technique to select a threshold value for an image with a bi-modal histogram that maximizes the variance between foreground and background [34]. In the multi-modal case, the desired multiple threshold values are chosen at the bottom of the valley between each pair of neighbouring peaks, segmenting the image into multiple regions. However, noisy images generally do not have bi-modal or multi-modal histograms, making the selection of the threshold value difficult. The use of histogram-based threshold value detection method does not guarantee a correct segmentation because the pixels composed to peaks in the histogram could come from different objects.

Thresholding is a fast segmentation technique, but it is sensitive to noise and the choice of the threshold value. Thresholding only provides good segmentation results when the scene satisfies the assumption that the graylevels of the foreground and background are distinct and objects in foreground do not touch each other [46]. This assumption, however, is usually not true.

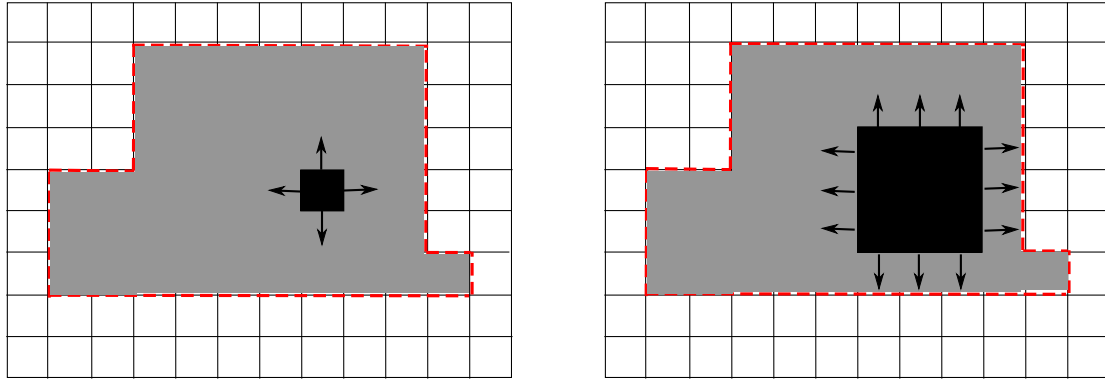
3.1.3 Region Growing

Region growing is based on the assumption that pixels belonging to the same objects in a scene have similar properties (graylevel, texture, color, etc). Region growing can be seeded or unseeded.

3.1.3.1 Seeded Region Growing

Starting from a set of seed regions or a set of seed pixels (usually provided by the user or obtained automatically), seeded region growing compares each growing region with its external boundary pixels (one pixel at a time) and expands the region by adding in similar external boundary pixels (Fig. 3.2) [45, 46]. This

process is continued until no more pixels can be added. Commonly used homogeneity criteria include the average graylevel of the existing region and texture properties.



(a) A seed pixel of the gray region is given in black. This figure shows the pixels to be considered when 4-connected neighbourhood is used.

(b) The region after grown a few iterations using the criteria that the graylevel of a newly added pixel should be similar to the mean graylevel of the existing region.

Figure 3.2: An example of region growing.

Depending on the homogeneity criteria, different seed regions/pixels for the same object may be grown into regions that are not identical. This characteristic requires the careful choice of seed regions/pixels. If there are multiple objects to be segmented, at least one seed region/pixel needs to be provided for each object. Homogeneity is tested locally, as a result, the size of the test region influences the test result. Small test regions tend to become overgrown, by including pixels from other objects. Large test regions tend to become undergrown by not including all pixels from the object.

Because homogeneity is tested locally, the size of the test region influences the test result; a small test region tends to become over-grown (include pixels from other objects), and a large test region tends to become under-grown (not include all pixels from the object) [45].

3.1.3.2 Unseeded Region Growing

Unseeded region growing starts from many small regions and merges similar ones (when they satisfy the merging criteria) until no more neighbouring regions satisfy the merging criteria. The small regions could be all the pixels from the original image, or obtained from the split and merge method [19, 46]. Seeded region growing is often used to segment particular objects in the image, given at least one pixel of each object. Unseeded region growing is often applied to obtain a complete segmentation, where each pixel in the image is in one and only one region. Both of them take local information into consideration, and therefore they are sensitive to noise [46]. Region growing is a slow segmentation technique because pixels need to be examined individually.

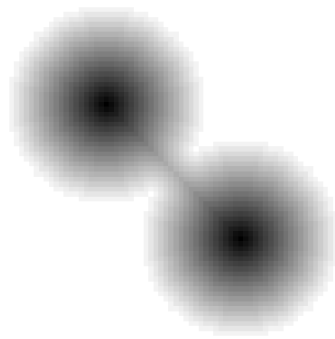
3.1.4 Watershed Segmentation

Suppose the intensity of a pixel is treated as altitude or height. Bright pixels have high altitude, dark pixels have low altitude (Fig. 3.3(a)). A regional minima of such a image is defined as a pixel “from which it is impossible to reach a point of lower altitude without having to climb” [50]. Each minima is associated with a catchment basin, which is a set of pixels such that if a water drop falls at a pixel in the basin, it will flow downwards and reach the minima [50]. Lines separating different catchment basins are called watersheds [50].

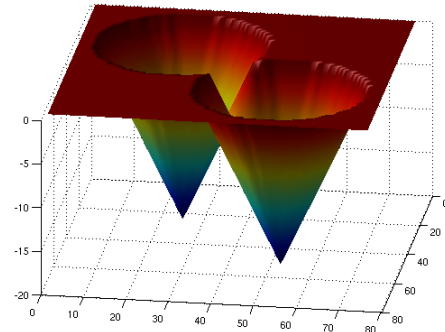
Conceptually, a hole is pierced at each regional minima of the image, and the the surface is immersed slowly into a lake (Fig. 3.3(c)). Water enters the catchment basins through pierced holes and water level starts to raise. A “dam” is built whenever a flood between two adjacent catchment basins occur (water from different catchment basins merge). The immersion is continued until the whole surface is under the water, at which point all the catchment basins are completely surrounded by dams. These dams correspond to the resulting boundaries of the segmentation. This is the basic idea of watershed segmentation (Fig. 3.3).

An efficient implementation of this algorithm was proposed by Vincent and Soille [50]. This implementation evolves two definitions: the *geodesic distance* between two pixels in a region is the length of the shortest path joining the two pixels that are entirely inside the region (Fig. 3.4(a)) [50]. Given a region and several connected components inside the region, the *geodesic influence zone* of one connected component is the set of points whose geodesic distances to this connected component are smaller than their geodesic distances to other components (Fig. 3.4(b)). This implementation of watershed segmentation has two steps:

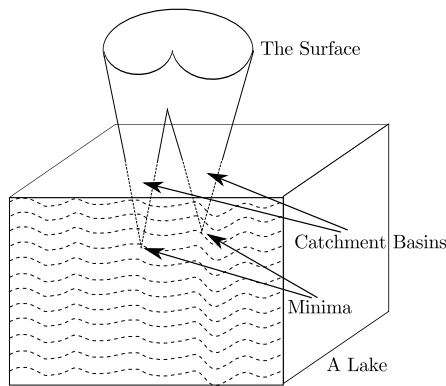
- The sorting step: The number of pixels at each graylevel is determined first for the purpose of memory allocation. After that, the image pixels are sorted in increasing order by their graylevel values.
- The flooding step: Let h be the smallest graylevel value (lowest altitude or global minima), and X_h be the set of pixels having graylevel smaller than or equal to h , which is first reached by the water. When the surface is immersed by one more level, pixels with graylevel smaller than or equal to $h + 1$ are reached by the water. The set of pixels with graylevel smaller than or equal to $h + 1$ is written as X_{h+1} . Let X_{h+1_i} be one of the connected components of X_{h+1} , there are three possible relations between X_{h+1_i} and X_h
 - $X_{h+1_i} \cap X_h = \emptyset$: X_{h+1_i} is a new minima.
 - $X_{h+1_i} \cap X_h \neq \emptyset$ and there is only one connected component in $X_{h+1_i} \cap X_h$: X_{h+1_i} belongs to the catchment basin associated with $X_{h+1_i} \cap X_h$.
 - $X_{h+1_i} \cap X_h \neq \emptyset$ and there are more than one connected component in $X_{h+1_i} \cap X_h$: X_{h+1_i} contains different minima of the image. Pixels in X_{h+1_i} are assigned to the catchment basin associated with the geodesic influence zone in which they reside.



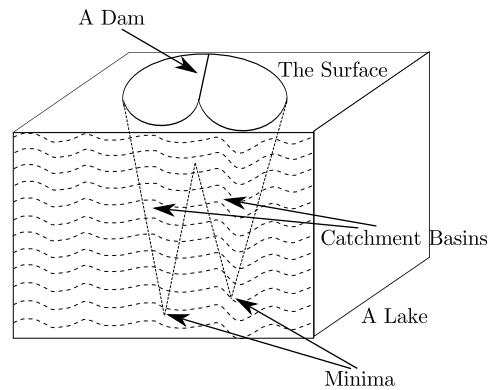
(a) Pixel intensity of this image represents altitude.



(b) The same image viewed as a surface. Bright pixels have high altitude and dark pixels have low altitude. There are two catchment basins (the cone-shaped objects) in this image.



(c) Part of the surface is immersed.



(d) The surface is completely immersed.

Figure 3.3: An example of watershed segmentation

In order to apply watershed segmentation, an original image needs to be converted to another image, whose intensity represents the altitude, and the objects to be segmented are the catchment basins of such an image. The conversion is based on the properties of objects. When segmenting objects from grayscale images, gradient information is used to create such images. When segmenting overlapping objects from binary images, distance information is used to create such images. The details of the two conversions are introduced in the rest of this section.

3.1.4.1 Watershed Segmentation Based on Gradients

Before the watershed segmentation is applied, a grayscale image needs to be preprocessed by taking the gradient magnitude (Section 2.2.2) of all the pixels. In this gradient magnitude image, pixels along object boundaries have large values, and other pixels have small values (Fig. 3.5). Because low gradient interiors correspond to homogeneous area, which tend to belong to a single object, and because low gradient interiors

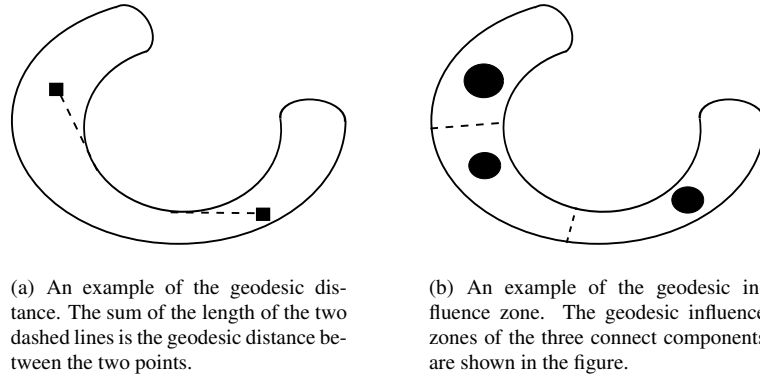


Figure 3.4: An example of the geodesic distance and geodesic influence zones

correspond to catchment basins, gradient produces catchment basins that correspond to objects. Pixels in the gradient magnitude image are sorted and flooded as it was introduced in Section 3.1.4. Objects are segmented by building “dams” along watershed lines.

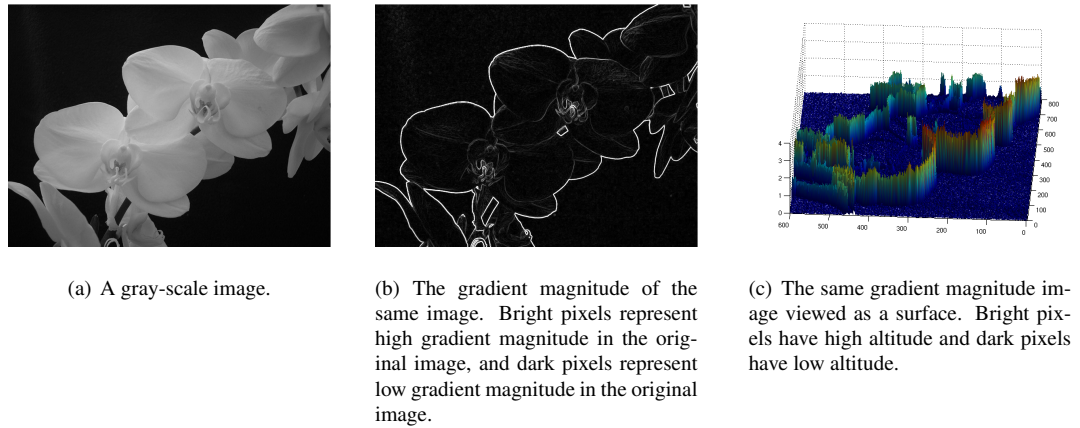


Figure 3.5: Watershed segmentation based on image gradient

3.1.4.2 Watershed Segmentation Based on Distance Transform

Watershed segmentation based on the distance transform is a morphological segmentation technique. Given a binary image with overlapping objects, this segmentation technique attempts to obtain individual objects. Given a binary image, the distance transform is the distance from every pixel to its nearest non-zero pixel.

Given a binary image with 1 representing object pixels and 0 representing background pixels, the distance transform of the binary image’s complement is calculated and negated. In the resulting negated distance image, the further away a pixel is from the object boundary, the higher value it has, and pixels belonging to the background have the value 0 (Fig. 3.6). When objects are overlapping, it is likely that it is narrow at the point where two objects are connected. Distance transform on an binary image has small values at narrow regions (most likely to be the parts where objects are connected), and large values at wide regions

(most likely to be in the center of objects), resulting in catchment basins that corresponding to individual objects. Pixels in the negated distance image are sorted and flooded (as introduced in Section 3.1.4). Objects are segmented by building “dams” along watershed lines.

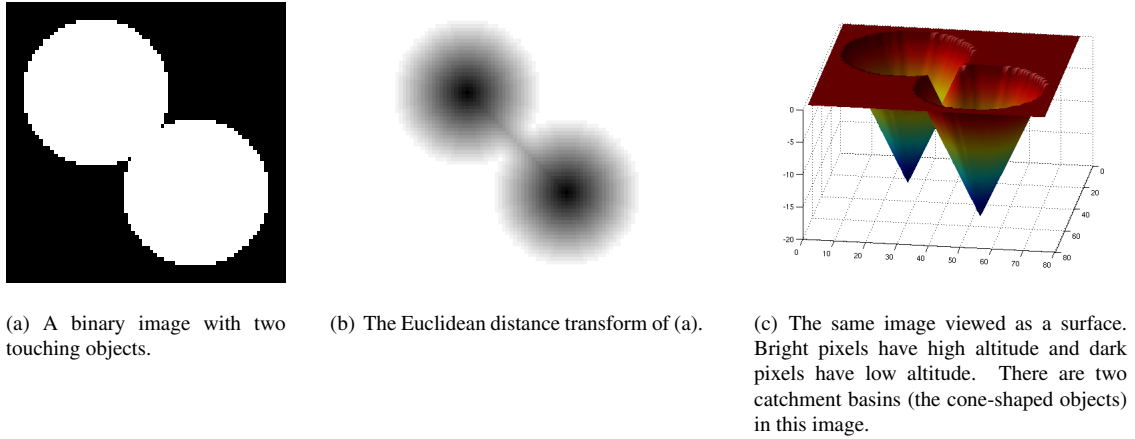


Figure 3.6: Watershed segmentation based on a distance transform.

3.1.4.3 Oversegmentation & Marker-Controlled Watershed Segmentation

In watershed segmentation, each regional minima results in catchment basin and thus a region in the output. Because of this, direct application of the watershed segmentation algorithm often causes over-segmentation due to irrelevant minima caused by noise or unrepresentative local minima (Fig. 3.7). Over-segmentation can be avoided by applying marker-controlled watershed segmentation which removes shallow minima. A marker is a connected component in an image [16]. Internal markers are situated inside the object of interest, and external markers belong to the background. The image is modified so that regional minima occur only in the regions with either internal or external markers (Fig. 3.7). Marker-controlled watershed segmentation avoids over-segmentation by controlling the number of objects in the image.

3.1.5 Parametric Active Contour Models

3.1.5.1 Active Contour Models

Active contour models recover object shapes by deforming curves [52, 53]. An initial curve, which is either provided by the user or obtained automatically, is deformed based on information extracted from the image data as well as the curve itself [23, 52]. Active contour models are either parametric [23] or geometric [5].

3.1.5.2 Basic Idea

Parametric active contour models, also known as snakes, were introduced in 1988 by Kass et al. as a deformable model to segment images [23]. A contour (snake), which is represented by a set of points,

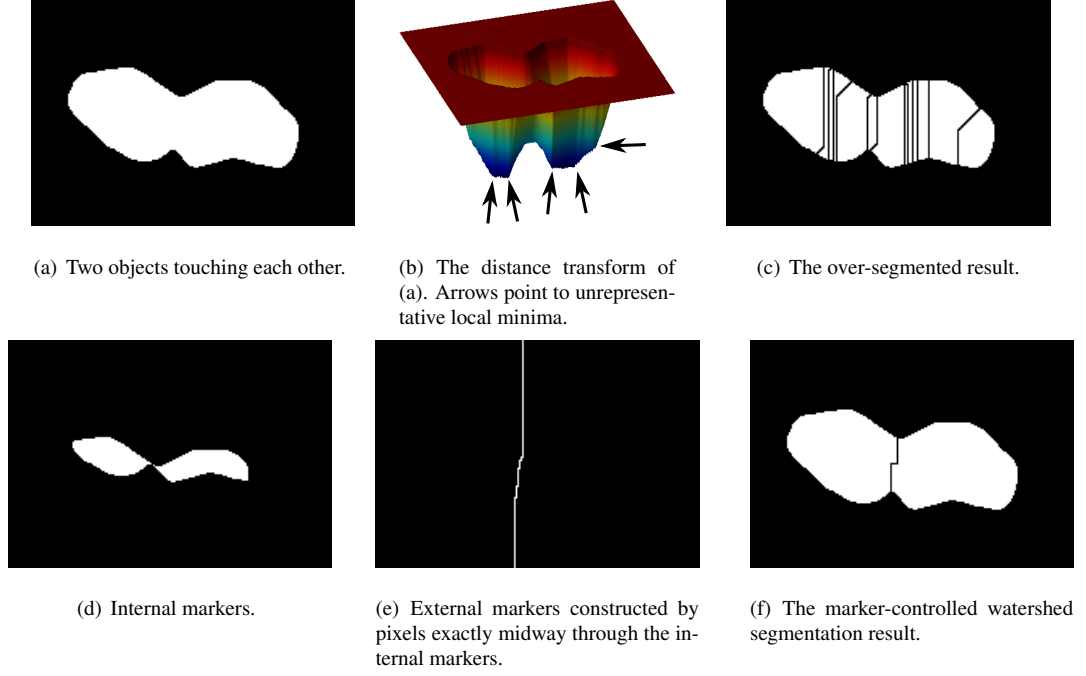


Figure 3.7: An example of over-segmentation when using the watershed segmentation technique and the proper result when using marker-controlled watershed segmentation. (b) is the distance transform of (a), and some of the local minimas are indicated by arrows. Because each local minima (instead of just global minima) forms a catchment basin, the final segmentation result shown in (c) is over-segmented. The image is properly segmented (f) with internal markers (d) and external markers (e).

is associated with an energy function, the value of which is determined by the shape of the contour and its location in the image. The energy function has the minimum value when the contour is smooth and are at the true object boundary [23]. The problem of segmentation is converted to the problem of energy minimization, which is then solved by the Euler-Lagrange differential equation.

3.1.5.3 Structure

A contour (v) is a set of points (v_i) in the image, the positions of which are parametrically represented by

$$v = \{v_1, v_2, \dots, v_n\} \quad (3.2)$$

where

$$v_i = (x_i, y_i) \quad (3.3)$$

and $i = \{1, 2, \dots, n\}$. Given an object to be segmented and an initial contour, which could be provided by a user or obtained automatically, the snakes algorithm changes the position of all the points on the initial contour iteratively until the resulting contour is as close as possible to the expected contour. To be considered the expected contour, a contour must be smooth and aligned with the object boundary.

An energy function is constructed so that its minimum occurs when the associated contour is the expected contour or very close to the expected contour. Corresponding to the two properties of an expected contour

(being smooth, and being aligned with the object boundary), an energy function is composed of two terms; internal energy E_{int} (which depends on the shape of the contour), and external energy E_{ext} (which depends on the image properties):

$$E(v) = \alpha E_{int}(v) + \beta E_{ext}(v) \quad (3.4)$$

where the parameters α and β control the importance of each energy. The contour deformation is achieved by moving points toward positions which will yield a smaller energy value for the entire contour, and stops when the energy reaches its minimum. The problem of segmenting an image has now been transformed into one of finding the minimum of the energy function $E(v)$, which could be solved by the Euler-Lagrange differential equation.

When an energy function reaches its minimum, the associated contour satisfies the following Euler-Lagrange differential equation [23, 53, 54]:

$$\alpha \nabla E_{int} - \beta \nabla E_{ext} = 0 \quad (3.5)$$

In order to study the physical behavior of the curve deformation, the equation 3.5 is interpreted as a force balance equation [53]. Each term is a force produced by the corresponding energy term:

$$\alpha F_{int} + \beta F_{ext} = 0 \quad (3.6)$$

At any time t , a contour is affected by internal and external forces:

$$F(t) = F_{int}(t) + F_{ext}(t) \quad (3.7)$$

The contour deforms based on the net effect of the sum of forces, and stops at equilibrium, when the sum of forces is 0. Let the force on a point v_i on the contour v be $F_i(t)$, the acceleration of the point v_i at the time t is

$$a_i(t) = \frac{1}{m_i} F_i(t) \quad (3.8)$$

where m_i is the mass of the point v_i . Since the mass of all points are the same, this term could be eliminated. The velocity of the same point v_i at the time $t + \Delta t$ is

$$v_i(t + \Delta t) = v_i(t) + a_i(\Delta t) \quad (3.9)$$

And the position of the same point v_i at the time $t + 1$ is

$$p_i(t + \Delta t) = p_i(t) + v_i(\Delta t) \quad (3.10)$$

The snakes algorithm is summarized as Algorithm 1.

Algorithm 1 Snakes

```
Obtain an initial contour
Calculate the sum of forces for the contour
while sum of forces  $\neq 0$  do
    Deform the contour based on the net effect of the forces
    Update the position of contour points according to Equation 3.10
    Calculate the sum of forces for the contour
end while
```

3.1.5.4 Internal Energy & Internal Forces

The internal energy is derived from the curve's shape. It is composed of elastic energy (the energy stored in deformed elastic materials such as a spring) which represents a contour's ability of to return to its original shape after being deformed, and rigid energy which represents the stiffness of the contour [23]. If a contour v is viewed as a continuous function, the elastic energy and rigid energy are defined as follows:

$$E_{elastic} = \frac{1}{2}w_e|v'|^2 \quad (3.11)$$

$$E_{rigid} = \frac{1}{2}w_r|v''|^2 \quad (3.12)$$

where w_e and w_r are weighting parameters that control the contribution of each energy. Corresponding to the two types of internal energy, there is an elastic force which makes a contour more compact (Section 2.5.1) and a rigid force which smooths sharp corners [23].

3.1.5.5 External Energy & External Forces

The external energy is a function of the image data which attempts to map energy minima to object boundaries. This may not always be the case, and may result in an imperfect segmentation. The construction of the external function energy is application dependent.

External forces are derived from image data, and move contours toward object boundaries. Multiple external forces may be combined for deformation by giving each force a weight. The following sub-sections introduce some commonly used external forces.

3.1.5.5.1 Gaussian Potential Force Given a bitmap image I (Section 2.1), if the object of interest in is a single black (pixel value 0) line on a white (pixel value 1) background, the image matrix itself (I) has the smallest value (0) on the object. Therefore, the external energy function for the contour is I (Fig. 3.8). If the object is a single white line on the black background, $-I$ has the minimum value (0) on the object. Therefore, the external energy function for the contour is $-I$.

$$E_{ext}(line) = \pm I \quad (3.13)$$

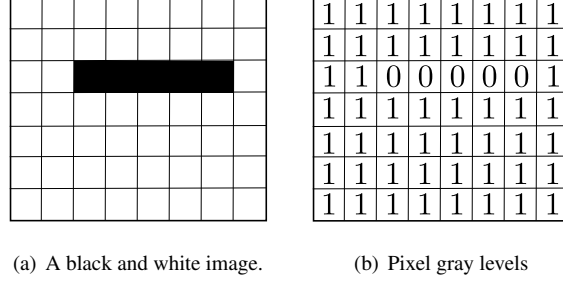


Figure 3.8: (a) shows an image of a single black line on the white background. (b) shows the image gray levels. The gray level has the smallest value 0 along the line.

If the object of interest is a homogeneous area, the gradient magnitude of the image (written as $\nabla|I|$) has large positive values on the edge, and values of close to 0 elsewhere. The negative of the square of the gradient magnitude has large negative values on the edge and values of close to 0 elsewhere. Therefore, it could be used as the external energy function for the contour.

$$E_{ext}(edge) = -|\nabla I|^2 \quad (3.14)$$

In practice, a Gaussian filter with standard deviation σ is convolved (Section 2.2.1) with the image first, where σ controls the capture range of a contour (the area a contour could be attracted, Fig. 3.9).

$$E_{ext}(line) = \pm(G_\sigma * I) \quad (3.15)$$

$$E_{ext}(edge) = -|\nabla(G_\sigma * I)|^2 \quad (3.16)$$

The corresponding Gaussian potential force ($F_{potential}$) is defined as

$$F_{potential} = -w_{potential} \nabla E_{ext} \quad (3.17)$$

where $w_{potential}$ is a weighting parameter that controls the contribution of the Gaussian potential force.

The biggest problem with Gaussian potential force is that final contours are very sensitive to the location of initial contours because the capture range is generally small (Fig. 3.9). The parameter, σ , that controls the capture range is difficult to choose because although a large σ will lead to a greater capture range, it will cause significant edge blurring [53].

3.1.5.5.2 Pressure Force The pressure force ($F_{pressure}$) was introduced by Cohen [10] in 1991 to solve the problem of small capture range. A contour is treated as a 2D balloon, and the pressure force either inflates or deflates the balloon.

$$F_{pressure} = \pm w_{pressure} \vec{n}(s) \quad (3.18)$$

where $\vec{n}(s)$ is the normal unit vector at point $v(s)$, and $w_{potential}$ is a weighting parameter that controls the contribution of the pressure force. A positive sign results in contour inflation, and a negative sign results in contour deflation.

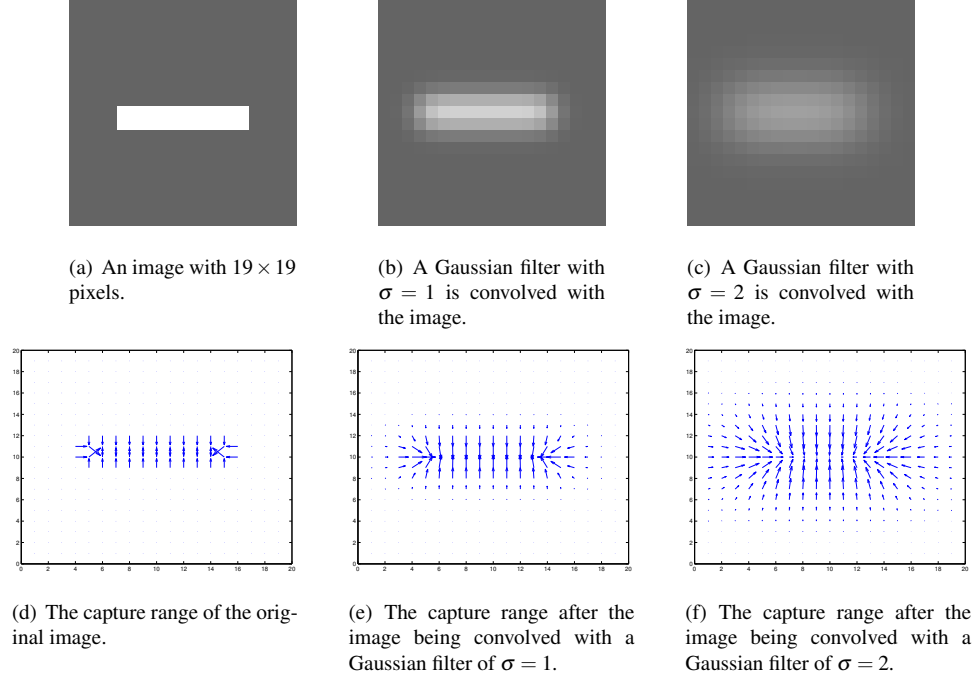


Figure 3.9: The capture range of the original image is small (d). After being convolved with a Gaussian filter, the capture range increased (e). The larger the standard deviation (σ), the larger the capture range.

The weighting parameter of pressure force ($w_{pressure}$) is chosen so that a contour will pass through weak edges and be stopped by the Gaussian force at strong edges:

$$\begin{aligned} w_{pressure}F_{pressure} &> w_{potential}F_{potential} && \text{at weak edges} \\ w_{pressure}F_{pressure} &< w_{potential}F_{potential} && \text{at strong edges} \end{aligned}$$

If $w_{pressure}$ is too low, the balloon force is simply disabled. If $w_{pressure}$ is too high, the final contours will pass through object boundaries. By using the pressure force, the initial contours no longer need to be near object boundaries.

3.1.5.5.3 Gradient Vector Flow Xu and Prince proposed another way of solving the problem of small capture range [55], namely gradient vector flow (GVF). The idea of GVF is to extend the Gaussian potential force from edges into homogeneous areas by diffusion, that is, the gradient around edges stays the same, and diffuses to further homogeneous regions to form gradient with corresponding directions and smaller magnitudes.

An edge map $f(x,y)$ is defined as a matrix derived from an image such that the matrix elements are large near the object boundaries [55]. A commonly used edge map is

$$f(x,y) = -E_{ext}^i(x,y) \quad (3.19)$$

where $i = 1, 2, 3$ or 4 , and E_{ext} is the external energy defined in Equation 3.16. The effect of i is to impose the difference among pixel values in the edge map. The diffusion process is achieved by solving the minimization problem of the following energy function:

$$E_{GVF} = \int \int \mu |\nabla v|^2 + |\nabla f|^2 |v - \nabla f|^2 dx dy \quad (3.20)$$

where μ is a weighting parameter and f is the edge map. The term that dominates E_{GVF} is different depending on how far a pixel is away from an edge. The gradient magnitude $|\nabla f|$ is small in homogeneous areas, therefore, E_{GVF} is decided by the first term. Around edge areas, the gradient magnitude $|\nabla f|$ becomes larger, so E_{GVF} is decided by the second term. To minimize the energy function E_{GVF} , the following Euler equation must be satisfied:

$$\mu \nabla^2 v - (v - \nabla f) |\nabla f|^2 = 0 \quad (3.21)$$

The GVF field (v) is then obtained from the above equation. The largest advantage of replacing Gaussian potential force with the GVF is that boundaries with concavities will be well segmented.

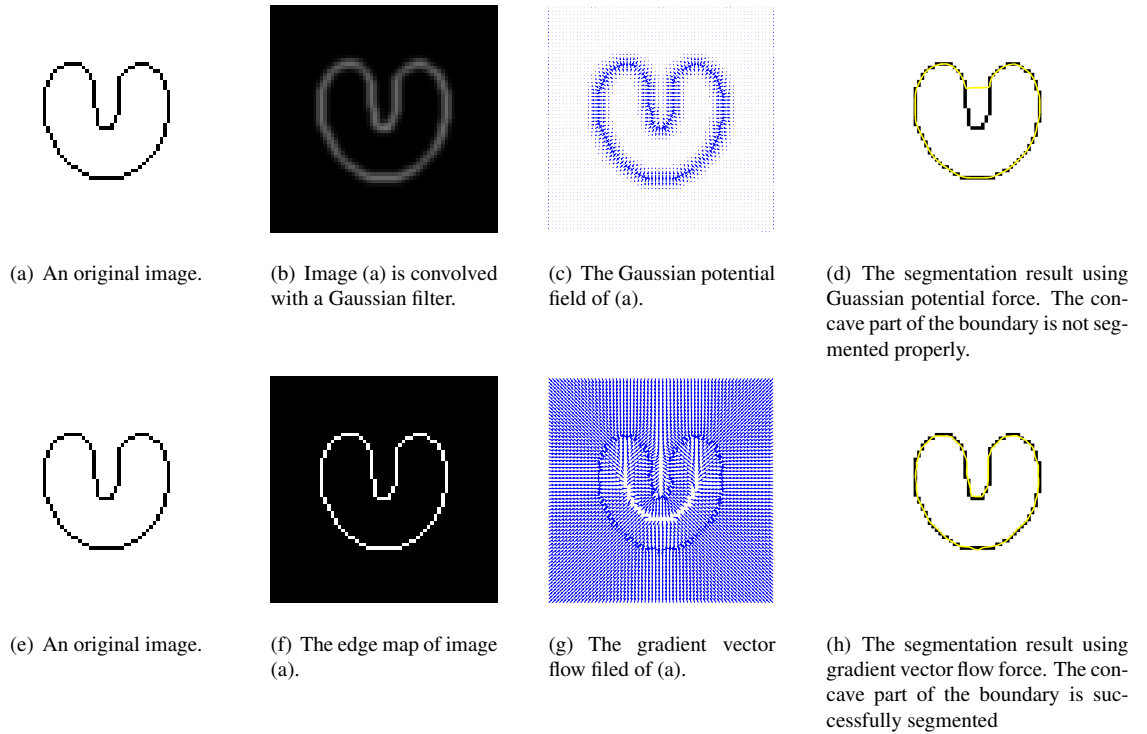


Figure 3.10: Different segmentation results using Gaussian potential force and GVF. GVF has the ability to segment boundaries with concavities properly (h). The original image is from <http://iacl.ece.jhu.edu/projects/gvf/>.

3.1.5.6 Advantages & Disadvantages

The parametric active contour models have a low computational complexity. They have been well studied, and their effectiveness has been proved. It is easy to include new internal or external forces. However,

this algorithm is sensitive to the selection of parameters and the position of initial contours. Control points representing a snake require resampling when the length of the snake increased (more control points are added) or decreased (some existing control points are removed). This resampling process is rather arbitrary, causing snakes to handle topological changes poorly.

3.1.6 Geometric Active Contour Models

Geometric active contour models are based on curve evolution theory and the level set method. An initial contour is embedded as the zero level set of a higher-dimensional function, which evolves over time t based on image data and the curvature. When the evolution is completed, the final contour is the zero level set of this high-dimensional function [33].

3.1.6.1 Intuition

Imagine people are standing very closely along a curve, and are connected with neighbours by rubber bands. Everyone is assigned a direction to run along, and a speed. They all start running at the same time, and stop after a certain time. The number of people is adjusted to keep the space between neighbours constant, and everyone's direction and speed is re-assigned at each stop. This assign \rightarrow run \rightarrow stop process continuous until everyone is assigned a 0 (zero) speed, at which point the curve evolution has completed. The resulting contour formed by rubber bands and people is the final contour.

3.1.6.2 Curve Evolution Theory

Let C be a moving curve. At any time t , the current curve is represented as $C(s, t)$, where s indexes all the positions on the curve. To obtain the next curve $C(s, t + \Delta t)$ from the current curve $C(s, t)$, each position on $C(s, t)$ is moved along its normal direction at certain speed. A pre-defined speed function F , written as $F(s, t)$ at time t , specifies the speed for all the points. The moving curve C converges (stops) when the value of speed function is 0 everywhere on the current curve, that is, $F(s, t) = 0$ for all s .

The idea of curve evolution theory is that a curve is deformed under only unit normal (a unit vector in the normal direction of the curve) and curvature (the rate of change of slope). In order to apply this theory to image segmentation, a speed function that embeds image data must be constructed so that the curve evolution stops at object boundaries [53]. The design of such a speed function will be introduced later in this section.

3.1.6.3 The Level Set Method

A curve's topology might change after it evolves for some period of time, for example, two curves might merge together or one curve may split into separate curves. The level set method, which embeds a curve in a higher dimension, is able to handle the topological changes with no special effort.

Let $z = \phi(x, y)$ be a 3D function (level set function). A level set is defined as the set of all the pairs (x, y) which share the same z value (Fig. 3.11). The zero level set is the set of all the pairs (x, y) which satisfies $\phi(x, y) = 0$. In other words, the zero level set is $\{(x, y) | \phi(x, y) = 0\}$.

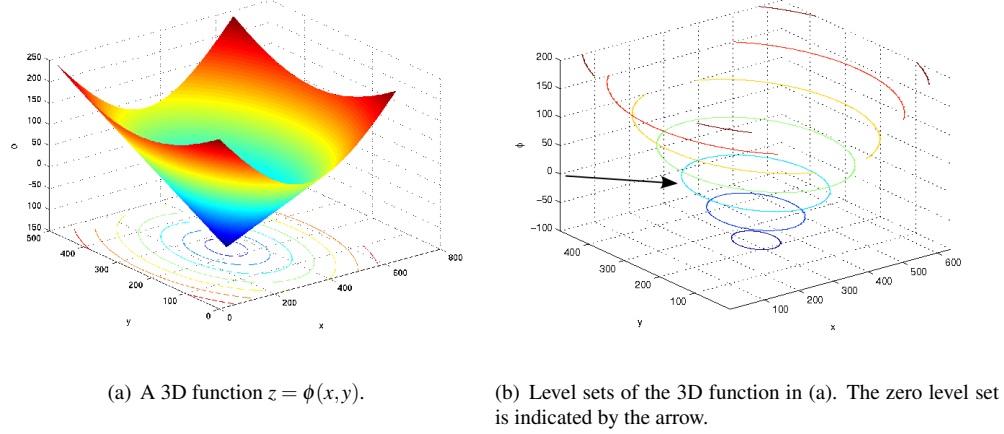


Figure 3.11: An example of a level set function.

The level set method embeds an initial contour as the zero level set of a 3D function $\phi(x, y)$ [44, 53]. That is, the function ϕ is constructed so that $\phi(x, y) = 0$ is true if and only if (x, y) is on the initial contour (at time $t = 0$). The function $z = \phi(x, y)$ appears as a surface in 3D Cartesian coordinates. Instead of evolving the curve directly, the level set method evolves this surface over time. The speed the surface evolution is determined by a known speed function (to be introduced later). In another words, the z -values of all the points are recalculated through time. At any time t , the current contour is the zero level set of the surface. When the surface evolution stops, the zero level set of the last surface is the final contour (Fig. 3.12).

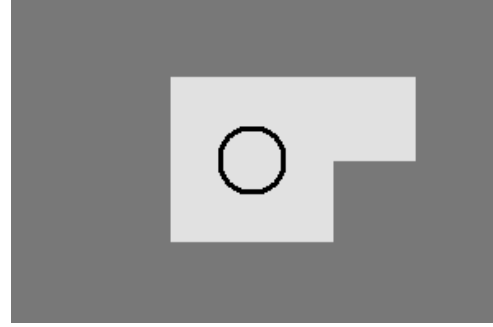
A common way to construct the initial function ϕ is to set ϕ to a signed distance function. The signed distance function $D(x, y)$ is defined as the shortest distance from a point (x, y) to the contour. Function $D(x, y)$ has negative values for points inside of the contour, and positive values for points outside of the contour (Figure 3.13). Let C be a contour, and $d((x, y), C)$ be the shortest distance from a point (x, y) to all of the points on the contour C . The signed distance function is defined as follows:

$$D(x, y) = \begin{cases} -d((x, y), C) & \text{if point } (x, y) \text{ is inside contour } C, \\ 0 & \text{if point } (x, y) \text{ is on contour } C, \\ d((x, y), C) & \text{if point } (x, y) \text{ is outside contour } C. \end{cases} \quad (3.22)$$

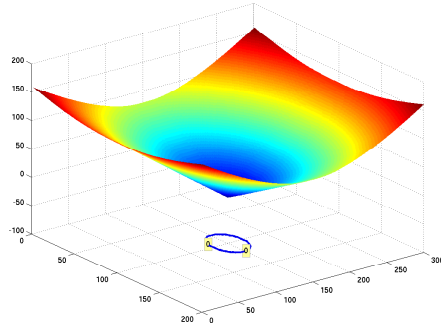
The numerical approximations of the level set method are only accurate when there are not many variations in the gradient magnitude of the surface. Because the signed distance function has the property that $|\nabla D| = 1$, it is often chosen to be the initial level set function (ϕ).



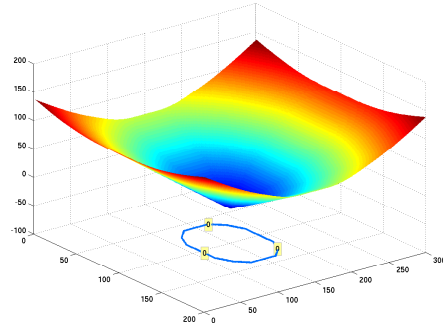
(a) An image with the centered area to be segmented.



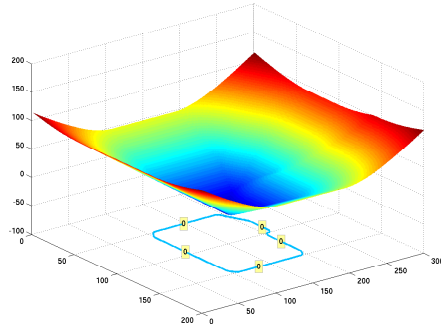
(b) An initial contour (a circle) is placed.



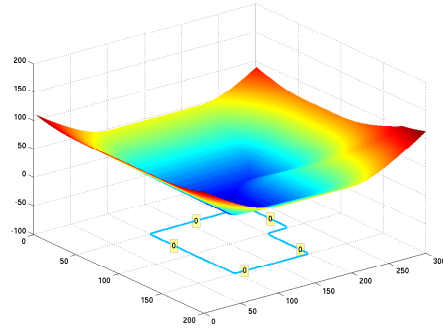
(c) The initial level set function (surface) and the zero level set representing the initial contour (projected onto the $z = -100$ plane).



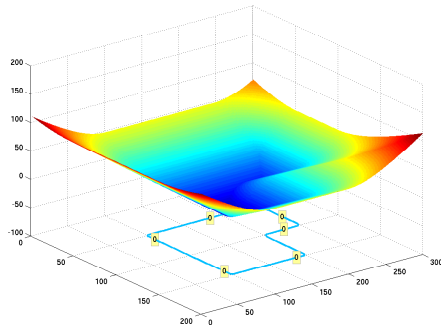
(d) The level set function after deform 200 iterations (surface) and the resulting contour (projected onto the $z = -100$ plane).



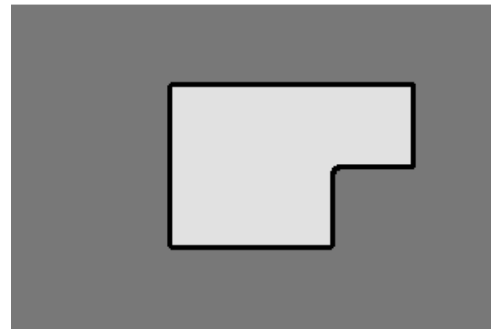
(e) The level set function after deform 400 iterations (surface) and the resulting contour (projected onto the $z = -100$ plane).



(f) The level set function after deform 600 iterations (surface) and the resulting contour (projected onto the $z = -100$ plane).

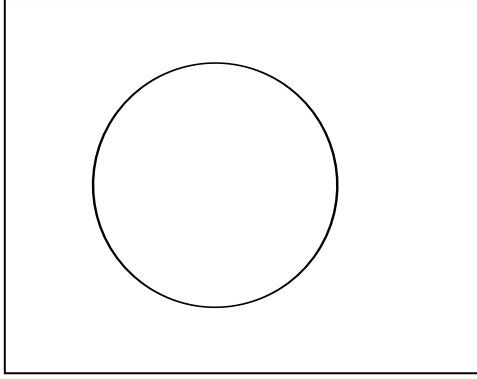


(g) The final level set function (surface) and the final contour (projected onto the $z = -100$ plane).

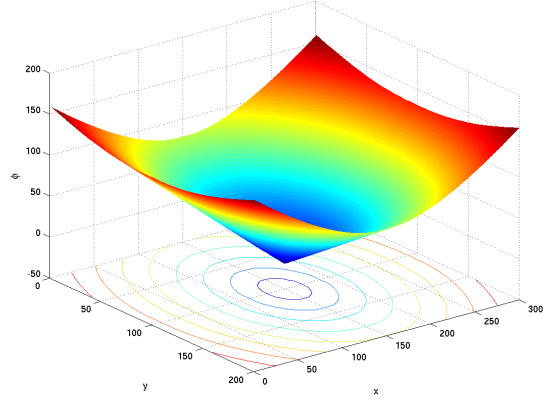


(h) The final contour on the original image (a).

Figure 3.12: An example of segmenting an object using the level set method.



(a) A contour inside an image.



(b) The signed distance of elements in the image.

Figure 3.13: An example of the signed distance function. Points inside the contour have negative values, points on the contour have the value of 0, and points outside the contour have positive values.

3.1.6.4 Speed Function for 2D Curves

In curve evolution theory, the speed function F determines the speed of curve flow in the normal direction over time. The curve evolution can therefore be characterized by the following equation

$$\frac{\partial C}{\partial t} = F \vec{n} \quad (3.23)$$

where \vec{n} is the outward unit normal.

Let the curvature at a point of the contour be κ , the basic speed function which evolves a 2D curve is based only on the normal and the curvature:

$$F = \pm 1 - \varepsilon \kappa \quad (3.24)$$

where ε is the weighting parameter that controls the contribution of curvature. The first term ± 1 inflates (+1) or deflates (-1) the curve, and the second term $\varepsilon \kappa$ smooths out the part of the contour with high curvature. The corresponding velocity function that determines both speed and direction is defined as

$$\frac{\partial C}{\partial t} = F \vec{n} \quad (3.25)$$

$$= (\pm 1 - \varepsilon \kappa) \cdot \vec{n} \quad (3.26)$$

The design of the speed function for image segmentation is based on the idea that everywhere in the image should have a certain speed except on the object boundary, where the speed should be 0. This is achieved by multiplying the basic speed function with a stopping term g_I , the value of which is close to 0 on the edge and 1 everywhere else:

$$F = g_I(\pm 1 - \varepsilon \kappa) \quad (3.27)$$

The stopping term g_I is designed as follows:

$$g_I = \frac{1}{1 + |\nabla(G_\sigma * I)|} \quad (3.28)$$

where $I(x, y)$ is the image that contains the object to be segmented, and $G_\sigma * I$ is the convolution of I with a Gaussian filter that has standard deviation σ . The gradient magnitude of the convolved image ($|\nabla(G_\sigma * I)|$) is large on edges and small everywhere else, which results in the stopping term g_I being close to 0 on edges and close to 1 everywhere else.

This speed function (Equation 3.27) was proposed by Caselles et al. [5] and Malladi et al. [30] in 1993 and 1995. This speed function works very well when objects have a continuous boundary and good contrast over the background. However, when an object has a non-continuous boundary, the curve will leak out from the gap. To address this problem, Caselles et al. [6] and Yezzi et al. [57] proposed a new design of speed function in 1995 and 1997.

$$F = g_I(\pm 1 - \varepsilon \kappa) + \nabla g_I \cdot |\nabla \phi| \quad (3.29)$$

The second term $\nabla g_I \cdot |\nabla \phi|$ acts similar to the Gaussian potential force in parametric active contour models and is able to pull back the leaking contour.

3.1.6.5 Speed Function for 3D Surfaces

Since the level set implementation works on a surface in 3D space, a new speed function that determines speed for all the points on the surface needs to be derived. That is, a 3D speed function with direction perpendicular to the curve plane needs to be derived from the 2D speed function (Equations 3.27 and 4.2), the direction of which is normal to the curve (inside the curve plane). At any time t , the current curve is obtained by taking the zero level set of the function ϕ

$$\phi(C(s, t), t) = 0 \quad (3.30)$$

Take derivative of the above equation with respect to t . By the chain rule,

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \frac{\partial C}{\partial t} = 0 \quad (3.31)$$

Because F is speed in the outward normal direction,

$$F \cdot \vec{n} = \frac{\partial C}{\partial t} \quad (3.32)$$

where \vec{n} is the outward unit normal of the surface, and

$$\vec{n} = \frac{\nabla \phi}{|\nabla \phi|} \quad (3.33)$$

The equation 3.31 can be rewritten as

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{\partial C}{\partial t} \quad (3.34)$$

$$= -\nabla \phi \cdot (F \cdot \vec{n}) \quad (3.35)$$

$$= -F \cdot (\nabla \phi \cdot \vec{n}) \quad (3.36)$$

$$= -F |\nabla \phi| \quad (3.37)$$

This is the speed function for 3D surfaces that controls the moving speed (up and down) of points on the surface.

3.1.6.6 Re-initialization

As mentioned in section 3.1.6.3, the level set method embeds an initial curve as the zero level set of a 3D function ϕ . The initial curve is therefore represented as $\phi = 0$. The numerical approximations of the level set method are only accurate when ϕ is the signed distance function, that is, $|\phi|$ represents the closest distance from a point to the boundary $\phi = 0$. The sign of ϕ is $-$ (negative) if the point is inside of the boundary, and is $+$ (positive) if the point is outside of the boundary. After the surface evolves for a few iterations (usually around 10 iterations), ϕ will no longer represent the signed distance function of the curve. This will cause the algorithm to become inaccurate. Therefore, it is important to pause the evolution and rebuild a level set function that is a signed distance function of the current curve to replace the previous level set function. This process is called re-initialization [44].

Geometric active contour models embed an initial curve into a 3D surface, which evolves over the time. The speed of the evolution is derived from the image data. When the curve evolution stops, the contour that segmented objects of interest is obtained by taking the zero level set of the evolved 3D surface. Geometric active contour models automatically handle topological changes such as holes or isolated parts. The largest weakness is the algorithm is very slow.

3.2 Classifiers

A classifier was used to decide whether a region was a real follicle or not after the follicle segmentation algorithm developed in Section 4.3 was used to segment dark regions (potential follicles). A review of commonly used classifiers will help the reader to understand this process.

Given several classes (collections of objects with similar attributes) and an unknown pattern (described by features, which are measurable properties), a classifier is an algorithm that determines the class to which the unknown pattern belongs based on its features. A classifier can be supervised (needs to learn from a training set composed of patterns with known classes) or unsupervised (does not need to learn). The ideal features for classification are attributes which have similar values for patterns in the same class, and very

different values for patterns from different classes. Commonly used features for image region classification include shape (size, perimeter, eccentricity, compactness, etc), graylevel and color.

Given a pattern with n features, it can be represented as a point in a n -dimensional feature space (an abstract space where the number of dimension equals to the number of features used to describe the pattern, and each feature determines coordinates of points along one axis. If two features are used to describe a pattern, the feature space is a plane, whose x-axis represents one feature and y-axis represents another). The relationship between two patterns is often measured by the distance between the two corresponding points in the feature space. Depending on the application, this distance could be Euclidean distance, city block distance, or Minkowski distance [13].

3.2.1 Nearest Neighbour Classifier

The basic idea of the nearest neighbour classifier is that patterns in the same class are likely to be near each other in the feature space, therefore the class of the training pattern nearest to the unknown pattern in the feature space is assigned to that unknown pattern. When a tie occurs, it is broken at random [11, 13].

The main problem of the nearest neighbour classifier is that when the training patterns of two or more classes in the feature space are close together, unknown patterns belonging to these classes tend to be improperly classified. Because a large number of training patterns is often need to build a strong representation of each classes in the feature space, the nearest neighbour classifier is generally computationally expensive [11].

3.2.2 k -nearest Neighbour Classifier

The k -nearest neighbour classifier is similar to the nearest neighbour classifier, but uses more than one neighbour. Given an unknown pattern, the k -nearest neighbour classifier selects the k nearest training patterns to the unknown pattern, and the unknown pattern is assigned to the class that occurs the most frequently in the k nearest training patterns (majority vote) [13]. When a tie occurs, it is broken at random. The k -nearest neighbour classifier is still computationally expensive due to the requirement of a large set of training data.

3.2.3 Minimum Distance Classifier

The basic idea of the minimum distance classifier is that the center of all the patterns in the feature space (mean of one feature, centroid of two features, center of gravity of three features, etc.) of a class should represents that class. Each class is represented by a point in the feature space. The distances between an unknown pattern and all class centers are calculated and the unknown pattern is assigned to the class with the minimum distance [51].

The difference between the nearest neighbour classifier and the minimum distance classifier is that the distance calculated in the nearest neighbour classifier is between an unknown pattern and a training pattern

but the distance calculated in the minimum distance classifier is between an unknown pattern and a class (represented by a point). The nearest neighbour classifier assigns an unknown pattern to the class of its nearest training pattern while the minimum distance classifier assigns an unknown pattern to its nearest class.

The minimum distance classifier only works well when the distance between centers are large compared to inner-distance of classes (distances among points in individual classes). If a class has large inner-distances (large variance of features in this class), points representing patterns in the feature space tend to be far from the mean. This will cause patterns being misclassified.

3.2.4 Tree-based Classifier

The tree-based classifier “classifies a pattern through a sequence of questions” [13]. The next question depends on the answer to the current question (Fig. 3.14). Starting at the root node, each node checks a feature of an unknown pattern, and select a branch based on the feature value. This process is continued until a leaf node, which is a class label, is reached [26].

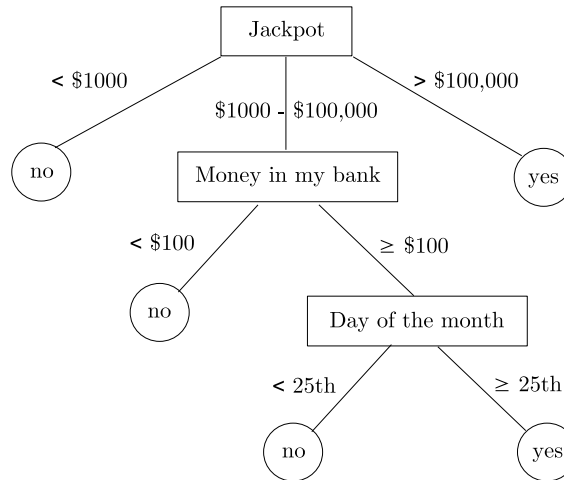


Figure 3.14: A tree-based classifier for deciding whether to buy a lottery ticket or not.

The tree-based classifier is very intuitive [26]. The key steps in constructing a tree-based classifier are to decide nodes positions (with more representative features are closer to the root), and to select the splitting rules for each node (cut-off values for different classes) which can be very complex and is beyond the scope of this work. However, the reader may find details on tree inference in [13, 51, 39, 40].

3.2.5 Naïve Bayes Classifier

The naïve Bayes classifier is a statistical classifier based on Bayes’ theorem with the assumption that features are conditionally independent of each other. The likelihood of an unknown pattern belonging to each class is calculated, and the pattern is assigned to the class with the maximum likelihood [13].

3.2.5.1 Structure

Given a set of classes $\omega = \{\omega_i | i = 1, 2, \dots, n\}$ and a pattern (\vec{x}) with m features ($\vec{x} = \{x_1, x_2, \dots, x_m\}$), the posterior probability (derived from \vec{x}) of the pattern \vec{x} belongs to a class ω_i is defined as:

$$p(\omega_i | \vec{x}) = \frac{p(\vec{x} | \omega_i) p(\omega_i)}{p(\vec{x})} \quad (3.38)$$

where $p(\vec{x} | \omega_i)$ is the class-conditional density:

$$p(\vec{x} | \omega_i) = p(x_1, x_2, \dots, x_m | \omega_i), \quad (3.39)$$

$p(\omega_i)$ is the *a priori* probability (no information about \vec{x} is taken into account), and $p(\vec{x})$ is the unconditional probability distribution:

$$p(\vec{x}) = \sum_{j=1}^n p(\vec{x} | \omega_j) p(\omega_j). \quad (3.40)$$

If one assumes that features are conditionally independent of others, Equation 3.39 may be rewritten as

$$p(\vec{x} | \omega_i) = p(x_1 | \omega_i) p(x_2 | \omega_i) \dots p(x_m | \omega_i) \quad (3.41)$$

$$= \prod_{j=1}^m p(x_j | \omega_i) \quad (3.42)$$

Combining Equations 3.38, 3.40, and 3.41, the posterior probability that a unknown pattern \vec{x} belonging to a class ω_i is

$$p(\omega_i | \vec{x}) = \frac{p(\omega_i)}{\sum_{j=1}^n p(\vec{x} | \omega_j)} \prod_{j=1}^m p(x_j | \omega_i) \quad (3.43)$$

Because the denominator $\sum_{j=1}^n p(\vec{x} | \omega_j)$ is same for all the classes (it acts as a normalization constant), the likelihood that the pattern \vec{x} belonging to a class ω_i is obtained by simplifying the posterior probability

$$lh(\omega_i | \vec{x}) = p(\omega_i) \prod_{j=1}^m p(x_j | \omega_i) \quad (3.44)$$

The naïve Bayes classifier assigns the pattern to the class with the maximum likelihood:

$$\omega = \underset{\omega_i}{\operatorname{argmax}} lh(\omega_i | \vec{x}) \quad (3.45)$$

$$= \underset{\omega_i}{\operatorname{argmax}} p(\omega_i) \prod_{j=1}^m p(x_j | \omega_i) \quad (3.46)$$

The naïve Bayes classifier assumes that all the features are conditionally independent. When features are correlated, their contributions are reduced significantly. In practice, even if features are not completely independent, the naïve Bayes classifier still yields a good result because although dependent features change the values of posterior probability (absolute values), they generally do not change the class which has the maximum posterior probabilities (relative rankings) [13]. The Bayesian classifier is simple to implement, fast to train (it only requires a small amount of training data) and easy to use.

3.2.5.2 Discretization

The Bayesian classifier requires knowledge about *a priori* probability and the class-conditional density. *A priori* probabilities are obtained by dividing the number of training patterns in a class ω_i by the total number of training patterns in all classes. Class-conditional densities are derived from feature values collected from training patterns.

Given an unknown pattern \vec{x} , each class-conditional density ($p(x_j = v_j|\omega_i)$) represents the probability that \vec{x} has the observed feature value ($x_j = v_j$) for a class (ω_i). However, because continuous valued features may take on an infinite number of values, even within a finite range (for example, there are an infinite number of real numbers between 0 and 1), there will be very few (or even zero) training patterns that have that particular feature value ($x_j = v_j$). These class-conditional densities will all be assigned to 0. The solution to this problem is to assign neighbouring values (values within a range) a single value. This process is called discretization (binning) [56].

There are many methods for discretization. The commonly used approach is to sort data into ascending order, and then group them by pre-determined intervals (bin width) of equal size. The number of bins is critical because a large number will cause the histogram to be noisy, and a small number will oversmooth the histogram and mask its true shape (Fig. 3.15).

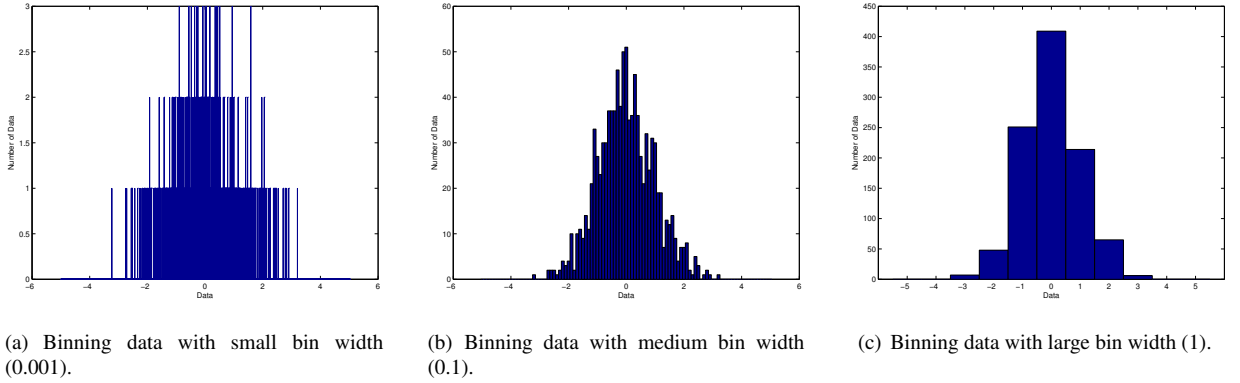


Figure 3.15: Given 1000 random numbers (Gaussian distribution with mean 0 and standard deviation 1), (a), (b), and (c) show the histogram of the same data sample using different bin width. (a) creates a noisy histogram by using a small bin width, (c) creates a oversmoothed histogram by using a large bin width. (b) uses a proper bin width, showing the data has the normal distribution.

There are two empirical rules to determine the number/width of bins. However, neither is guaranteed to be optimal. By Sturges' rule [47], the number of bins is calculated as follows:

$$\text{number of bins} = 1 + \log_2(n) \quad (3.47)$$

where n is the number of samples. It creates wide bins to reduce the chance of having empty bins, but it tends to oversmooth the histogram. Freedman-Diaconis rule focuses on the data that is in the middle of the

distribution [15]. The bin width is calculated as follows:

$$\text{bin width} = 2IRQ(x)n^{-\frac{1}{3}} \quad (3.48)$$

where $IRQ(x)$ is the interquartile range (range between the third and first quartiles) of the sample and n is the number of samples. The idea of the Freedman-Diaconis rule is that after sorting the data in ascending order, the data in the middle of the list are more representative than data at the two ends (outliers appear at the ends of the list). By considering only representative data, this calculation reduces the chance that a large width will be chosen due to outliers.

3.2.6 Clustering

Clustering is unsupervised classification, where similar patterns are grouped into classes without training [13]. In other words, clustering classifies patterns as well as defining classes. Clustering is used when features of patterns are unknown, or change over the time (e.g. the color of the same leaf is different over the year) [13]. The main disadvantage of using clustering is that classes created by clustering may lack of physical meanings.

MacQueen k -means Clustering is one of the commonly used clustering methods. Knowing there are k clusters (classes) in a data set, k points are chosen randomly from the data set as initial class centroids. All the remaining patterns in the data set are assigned to the class that has the closest centroid. After all points have been assigned, all the class centroids are re-calculated and all the points are re-assigned the closest new centroids. This process is continued until all the centroids are stabled (no more changes to centroids) [29]. MacQueen clustering is sensitive to the choice of initial centroids. This can be mitigated by running the algorithm multiple times with different initial points (centroids).

Other clustering methods include fuzzy C -means clustering [2, 14] and Hierarchical clustering [22].

3.3 Classifier Training and Validation Methodologies

Classification is achieved in two stages: training (patterns with known classes are used to learn trends in the data) and testing (determining classes of unknown patterns and estimating the performance of the classifier). Given a set of data, an algorithm is needed to decide how to divide the data into training and testing sets. This section reviews different training and validation methodologies. All the methods share one important property - the training and testing data must be disjoint. Otherwise, the performance of a classifier is biased because it has been tested with pattern-class pairs which it has already seen.

3.3.1 Half-and-half

The half-and-half method randomly divides the data into two sub-sets A and B . The classifier is trained with the sub-set A and tested with the sub-set B first, and then trained with B and tested with A . The average of

the two runs is the classifier performance [51]. This method performs well on a large data set with a low training cost.

3.3.2 N -way cross validation

N -way cross validation randomly divides a given data set into N sub-sets with equal number of data. The classifier is tested on each subset and trained on the rest of the data (all the $N - 1$ sub-sets). Common values of N are 5 or 10 for the division of the data set in cross validation techniques resulting in the testing set containing 20% or 10% of the entire data. This training and testing process is repeated N times with different testing sets, and the average performance of the N tests is the classifier performance [51]. The main advantage of this method is all the data will eventually be used to both training and testing purposes. N -way cross validation is computationally expensive because the classifier is tested N times. Therefore it is generally preferred on small data sets or when the variance of the performance needs to be reported [13, 51].

3.3.3 Leave-one-out

The leave-one-out method, also known as round-robin, is a special case of the N -way cross validation. The data set is split into a training set of size $N - 1$ and a testing set of size 1, and the classifier is trained and tested. This process is repeated until all the data in the data set are tested. N different splits are tested, whose average performance is the classifier performance [51]. This method is used when the data sets are extremely small.

3.4 Validation

3.4.1 Segmentation Validation

There are two types of methods for image segmentation evaluation [58]. Analytical methods, similar to white box testing, analyze and evaluate segmentation algorithms themselves by their principles and properties [58]. Empirical evaluation methods, similar to black box testing, measure the quality of segmentation results [58]. Because empirical evaluation methods provide easy-to-interpret objective evaluations, only empirical methods, in particular, empirical discrepancy methods based on the properties of mis-segmented pixels were considered to evaluate the segmentation algorithm.

Ideally, a segmentation result obtained from the algorithm would be compared with an expected segmentation, known as *the ground truth*. However, obtaining a ground truth segmentation in medical imaging would require an unethical, invasive procedure. The segmentation results are therefore compared with manual segmentations completed from experts, often called a gold standard.

3.4.1.1 The Number of Mis-segmented Pixels

In this method, the number of mis-segmented pixels is used as an indicator to measure the accuracy of the segmentation result [58]. The expert segmentation classifies all the pixels in an image into two classes - objects and background. The contour obtained from the segmentation algorithm classifies pixels in a different way (e.g. the contour is not identical to the expert's segmentation). Every pixel therefore belongs to one of the following classes: true positive (TP), true negative (TN), false positive (FP), or false negative (FN). The false positive rate (FPR, ratio of the number of false positive and the number of negative instances) and false negative rate (FNR, ratio of the number of false negative and the number of positive instances) are calculated to measure the accuracy of the segmentation:

$$FPR = \frac{|FP|}{|TN| + |FP|} \quad (3.49)$$

$$FNR = \frac{|FN|}{|TP| + |FN|} \quad (3.50)$$

A good segmentation result consists of both a low false positive rate (FPR) and a low false negative rate (FNR).

3.4.1.2 The Position of Mis-segmented Pixels

Segmentation accuracy could also be measured by the position of mis-segmented pixels. The comparison is achieved by establishing a correspondence between the points on the two curves, followed by calculating the error metrics that measure the difference of the curves.

The correspondence between points on two curves could be established by DCP (distance-to-the-closest point). Let contour A contain m pixels a_1, a_2, \dots, a_m , and contour B contain n pixels b_1, b_2, \dots, b_n . That is, $A = \{a_1, a_2, \dots, a_m\}$, $B = \{b_1, b_2, \dots, b_n\}$, where a_i, a_2, \dots, a_m and b_1, b_2, \dots, b_n are ordered pairs of x and y coordinates. The DCP from a pixel a_i to the contour B is the distance of a_i to the closest point on contour B :

$$DCP(a_i, B) = \min_{b_j \in B} \|b_j - a_i\|$$

where $\|b_j - a_i\|$ is the Euclidean distance between b_j and a_i [7]. Fig. 3.16 is an example of DCP correspondence.

After establishing a correspondence between two contours, error metrics are calculated to measure the distance (difference) between two contours. Commonly used error metrics are the Hausdorff distance, the average contour distance and the root mean square distance [7].

3.4.1.2.1 Hausdorff Distance The Hausdorff distance between two curves is the maximum DCP between the two curves [20].

$$HD(A, B) = \max \left\{ \max_i \{DCP(a_i, B)\}, \max_j \{DCP(b_j, A)\} \right\} \quad (3.51)$$

The Hausdorff distance measures the largest disagreement between two contours A and B .

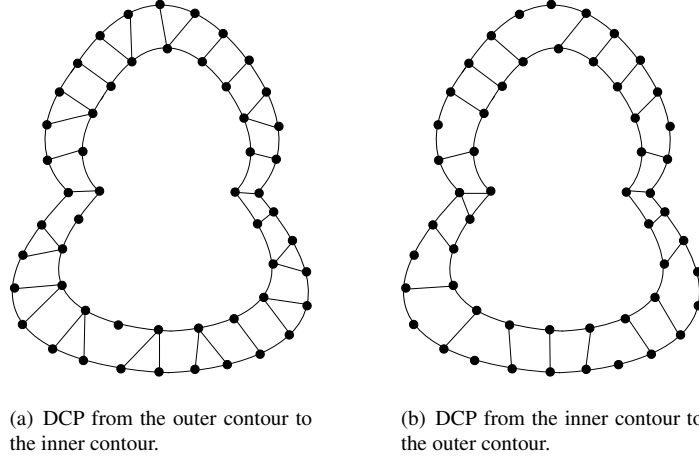


Figure 3.16: An example of DCP from outer to inner contour (a) and from inner to outer contour (b).

3.4.1.2.2 Mean Absolute Distance The average DCP of all the points on each curve is calculated individually, and the average of the two average DCPs is defined as the mean absolute distance (MAD).

$$MAD(A, B) = \frac{1}{mn} \left(\sum_{i=1}^m (DCP(a_i, B)) + \sum_{j=1}^n (DCP(b_j, A)) \right) \quad (3.52)$$

The mean absolute distance measures the average disagreement between two contours A and B .

3.4.1.2.3 Root Mean Square Distance The root mean square (RMS) distance between two contours is the root mean square distance between all the corresponding points.

$$RMSD(A, B) = \sqrt{\frac{1}{m+n} \left(\sum_{i=1}^m (DCP^2(a_i, B)) + \sum_{j=1}^n (DCP^2(b_j, A)) \right)} \quad (3.53)$$

The root mean squared distance measures the average disagreement between two contours A and B , with more penalty on points with a large DCP.

Generally, an error metric is computed for each pair of contours (a gold standard contour and an auto-segmented contour). The result is written as $m \pm sd$, where m is mean value of the metric for all the contours, and sd is the standard deviation of the same metric. A good segmentation result consists of a small mean and a small standard deviation.

3.4.2 Recognition Validation

The region classification problem in this thesis is a binary classification where there are only two classes. This section introduces some commonly used measurements for binary classification.

3.4.2.1 The Definition of Being Recognized

This sub-section introduces a method that decides whether an object is recognized or not. The correlation of the ground truth (a region) and the auto-detected region is measured using either the Jaccard or Dice coefficient, and the region is considered to be recognized if this coefficient is above a pre-determined threshold value.

3.4.2.1.1 Jaccard Coefficient Let region R_A contain m pixels $R_A = \{a_1, a_2, \dots, a_m\}$ and region R_B contain n pixels $R_B = \{b_1, b_2, \dots, b_n\}$. The Jaccard coefficient is the ratio between the number of pixels in the region where R_A and R_B intersect and the number of pixels in the region composed of the union of R_A and R_B [21]

$$s = \frac{|R_A \cap R_B|}{|R_A \cup R_B|} \quad (3.54)$$

3.4.2.1.2 Dice Coefficient Let region R_A contain m pixels $R_A = \{a_1, a_2, \dots, a_m\}$ and region R_B contain n pixels $R_B = \{b_1, b_2, \dots, b_n\}$. The Dice coefficient is twice the number of pixels in the region where R_A and R_B intersect divided by the sum of the number of pixels in each region [12]:

$$s = \frac{2|R_A \cap R_B|}{|R_A| + |R_B|} \quad (3.55)$$

The Jaccard and Dice coefficients are very similar. They both have the value 1 for two identical regions, and value 0 for regions without any overlapping. However, the Jaccard coefficient is consistently smaller than the Dice coefficient. When one coefficient is used to decide whether a region is recognized or not, a predetermined threshold value (usually 0.5) is compared.

3.4.2.2 Measurements

Measurements of recognition generally involves terms such as true positive (TP), false positive (FP), true negative (TN) and false negative (FN). These terms are defines as follows:

TP: A recognized region that is correctly determined to be a follicle;

FP: A recognized region that is incorrectly determined to be a follicle;

TN: An unrecognized region that is correctly determined to be a non-follicle; and

FN: An unrecognized region that is incorrectly determined to be a non-follicle.

3.4.2.2.1 Recognition Rate & Misidentification Rate A pair of commonly used measurements for the quality of recognition are the recognition rate (RR) and misidentification rate (MR). They are defined as

follows:

$$RR = \frac{|TP|}{|TP| + |FN|} \quad (3.56)$$

$$MR = \frac{|FP| + |FN|}{|TP| + |FP| + |TN| + |FN|} \quad (3.57)$$

The recognition rate is the probability that the classifier correctly recognize a real object. The misidentification rate is the probability that the classifier mis-classifies an object. A good recognition consists of both a high recognition rate and low misidentification rate.

3.4.2.2.2 Sensitivity & Specificity A commonly used pair of measurements for the quality of binary classification are sensitivity and specificity.

$$\text{sensitivity} = \frac{|TP|}{|TP| + |FN|} \quad (3.58)$$

$$\text{specificity} = \frac{|TN|}{|TN| + |FP|} \quad (3.59)$$

Sensitivity is the same as the recognition rate. Of all the regions that are thrown away by the classifier, specificity measures the percentage of the true negatives. A good recognition consists of both a high sensitivity and high specificity.

Generally, a pair of measurements (recognition rate and misidentification rate, or sensitivity and specificity) is computed for each image. The result is written as $m_1 \pm sd_1$ and $m_2 \pm sd_2$, where m_1 and m_2 are mean values of the measurements for all the images, and sd_1 and sd_2 are the standard deviations of the same measurements.

3.5 Existing Follicle Segmentation Techniques

Automatic follicle segmentation has been attempted using various techniques [9, 25, 35, 37, 43]. .

3.5.1 Thresholding

Potočník *et al.* detected follicles by subimage thresholding [37]. An original image was filtered by a homogeneous region growing mean filter [24], where each pixel was replaced by the mean of the pixel values from its neighbouring homogeneous region. Edges were detected by Kirsch's operator [42] from the filtered image. A binary image was constructed by thresholding the edge image using a threshold value of 40 so that only significant edges were preserved. This binary image was then thinned, and corrupted edges (discontinuous edges) were corrected by connecting the start and end points.

A subimage that contains the coarsely estimated bounding box of the ovary was obtained by examining the density of edge pixels of the reconstructed edge image. The subimage was then thresholded by a value that was determined using an optimal threshold selection method. A segmented region was retained only if

its area was larger than 150 pixels, the compactness (Section 2.5.1) was less than 0.5, and the eccentricity (Section 2.5.2) was less than 0.75. For images size of 768×576 , the average processing time of this algorithm was 6 minutes. The recognition rate of this algorithm was 70%. The recognition rate of this algorithm is sensitive to the threshold value.

3.5.2 Graph Searching

Sarty *et al.* introduced a semi-automated follicle segmentation algorithm based on graph searching [43]. For each follicle, the user draw two concentric circles; one inside and one outside the follicle wall. A Sobel operator was applied to pixels inside this annular region of interest (AROI), and the resulting edge image was resampled in directions perpendicular to the outer circle of the AROI to obtain a rectangular edge magnitude image. A Prewitt operator was applied to obtain one of eight edge directions for each pixel $(-\frac{3}{4}\pi, -\frac{1}{2}\pi, -\frac{1}{4}\pi, 0, \frac{1}{4}\pi, \frac{1}{2}\pi, \frac{3}{4}\pi, \pi)$. An inner border detection graph (G_I) was constructed from the rectangular edge image to detect an approximate inner follicle wall boundary. The nodes of G_I were pixels in the rectangular edge magnitude image, and the cost function incorporated edge strength, edge direction and the distance to the inner circle of AROI. The graph G_I was searched using the A* algorithm [18] and the path with the minimum cost was the detected inner follicle wall boundary.

The detected inner follicle wall was then examined and corrected by an expert, and then used as the basis to detect the outer follicle wall boundary. A new region of interest (ROI) was defined by eroding 10 pixels and dilating 30 pixels of the manual-corrected inner follicle wall boundary. A new rectangular edge image for ROI was obtained by the same method introduced in the inner follicle wall detection. An outer border detection graph (G_O) was constructed from the new rectangular edge image to detect the outer follicle wall boundary. The nodes of G_O were pixels in the new rectangular edge magnitude image, and the cost function consisted of the edge flow (bright-to-dark and dark-to-bright transaction), local edge strength, neighbouring edge strength, edge direction and the correspondence of the inner follicle wall shape model. The graph G_O was searched and the path with the minimum cost was the detected inner follicle wall boundary. Using this algorithm, the (mean and standard deviation of) Hausdorff distance of the automatically generated contours and the expert's contour was $1.47 \pm 0.83 \text{ mm}$, and the root mean square distance was $0.59 \pm 0.28 \text{ mm}$. Because the accuracy of inner follicle wall borders were critical for detecting outer follicle walls, this algorithm required user intervention to repair errors in the inner follicle wall segmentation.

3.5.3 Watershed Segmentation

Krivanek and Sonka reported a watershed-based segmentation algorithm that automatically detected the coarse inner follicle wall automatically [25] based on the research done by Sarty *et al.* [43].

Given an original 640×480 image, a 400×400 subimage that contains the minimum irrelevant area was extracted. This subimage was filtered using an adaptive neighbourhood median filter (Section 2.3.1) twice and segmented using watershed segmentation based on image gradient. The resulting oversegmented image

was converted to a mosaic image by assigning the average of pixel values in each region to all the pixels in that region. A binary image was obtained by thresholding the mosaic image using a threshold value of 95. Morphological opening (Section 2.4.3) was applied to the binary image to remove small regions. Watershed segmentation based on the distance transform (Section 3.1.4.2) was then applied to separate overlapping regions. For each separator, the average edge strength and average gray level along that separator was calculated, and algorithm 2 was used to decide whether the separator was a true follicle border or not.

Algorithm 2 Deciding Whether a Separator is a True Follicle Border

```

1: if the average edge strength along boundary > 15 then
2:   true border
3: else
4:   if length of boundary between two subregions > 0.65mm then
5:     false border (merge)
6:   end if
7: else
8:   if average graylevel along the boundary < 80 then
9:     false border (merge)
10:  end if
11: else
12:  if the average edge strength along the boundary < 6.8 then
13:    false border (merge)
14:  end if
15: else
16:  true border
17: end if

```

The coarsely detected inner follicle wall was then scaled to 80% of its size to replace the inner circle in the algorithm proposed by Sarty *et al.* [43], and dilated 30 pixels to replace the outer circle. The rest of the algorithm is the same as the algorithm proposed by Sarty *et al.* except the cost function for inner border detection graph (G_I) used the correspondence of the coarsely detected inner follicle wall shape model instead of the distance to the inner circle of AROI.

Given an expert's segmentation and automatic segmentation from this algorithm, the Hausdorff distance was 1.64 ± 0.92 mm, and the root mean square distance was 0.63 ± 0.36 mm for inner follicle wall borders. The Hausdorff distance was 1.81 ± 1.03 mm, and the root mean square distance was 0.67 ± 0.41 mm for outer follicle wall borders.

3.5.4 Region Growing

Potočník and Zazula segmented follicles using region growing [35]. An original image was smoothed by an adaptive neighbourhood median filter twice (Section 2.3.1). Homogeneous regions were composed of the union of the following three types of pixels: pixels whose graylevel was below a threshold T_1 , pixels whose neighbourhood had a standard deviations less than a threshold T_2 , and pixels whose graylevel was above a threshold T_3 and not touching the image border. Outer boundary pixels of each region were considered to be potential candidates if both their intensity and weighted gradient ($\text{grad}(x) = |\nabla I_x| (e^{\frac{2 \ln 2 \cdot \sigma(N_x)}{\text{mean}(N_x)}} - 1)$, where N_x is 11×11 neighbourhood of the current pixel x) were similar to the growing region (Section 3.1.3). Such a potential candidate was added into the growing region only if it had at least four neighbours from the growing region or potential candidates from the 3×3 neighbourhood.

After regions were grown, regions that touch the image border were removed, regions with an area less than 220 square pixels were removed, regions with an area-to-bounding-box ratio less than 0.5 were removed. The remaining regions were sorted individually by area, compactness, and area-to-bounding-box ratio, and every region was assigned a index (the sum of its position in all 3 lists). The region with smallest index was marked as the first actual follicle. The center of gravity of the first follicle(s) was calculated and the region closest to that center of gravity was selected. The global center of gravity of all the selected follicles (the average of gravity centers of all selected follicles) was updated and more regions were selected. This process kept going until the minimal distance from a region to the global center of gravity exceeded a quarter of the image size.

The recognition rate of this algorithm was $78\% \pm 21\%$ and the misidentification rate was $29\% \pm 25\%$. The mean absolute distance between expert's segmentation auto-generated segmentation was $1.1 \pm 0.4 \text{ mm}$. This algorithm tends to under-segment follicles, and did not perform well on outliers (e.g. in the case that a follicle has a small size, or a large compactness, or a small area-to-bounding-box ratio). Because the region growing algorithm is slow in general

3.5.5 Cellular Neural Networks

Cigale *et al.* developed a follicle segmentation algorithm that used four cellular neural networks (CNN) [9]. A cellular neural network had three templates (feedback template, influence template, and bias template), each of which was constructed by a coefficient that defined the behavior of the cellular neural network.

The initial state and the input of the first cellular neural network (CNN1) was transformed from an original image. CNN1 was trained by images where follicles were segmented by an expert to obtain three coefficients. Expressive follicles (dark follicles) of a testing image were roughly detected (generally smaller than the true follicle region) using CNN1, resulting in a binary image whose foreground denotes the roughly detected expressive follicles.

The initial state of the second cellular neural network (CNN2) was transformed from the binary image obtained from CNN1, and the input of CNN2 was transformed from the input of CNN1. CNN2 was trained by binary images whose foreground were follicles segmented by an expert to obtain three new coefficients. This step expanded the roughly detected expressive follicles and caused merging of regions with dim borders.

The initial state and the input of the third cellular neural network (CNN3) was transformed from a reduced image, where regions detected from CNN1 were eliminated from the original images. The coefficients of feedback template and influence template were the same as they were in CNN1, but the coefficient of the bias template was reduced by 2.5. This step detected inexpressive follicles (light follicles), with the cost of introducing regions that were not follicles.

The initial state of the fourth cellular neural network (CNN4) was transformed from the binary image obtained from CNN1. The input of CNN4 was transformed from the original image. CNN4 was trained by images where ovaries were segmented by an expert to obtain three coefficients. After this step, regions that were outside of an ovary tended to grow up to the image border. The convex hull of all the remaining regions were obtained as a rough ovary.

The following rules were used to decide whether a segmented region is a candidate follicle or not.

- all expressive follicles obtained from CNN2 were candidate follicles;
- if an inexpressive follicle from CNN3 joins two or more expressive follicles obtained from CNN1, it is not a candidate follicle;
- if an inexpressive follicle from CNN3 does not touch any expressive follicles, it is a candidate follicle; and
- if an inexpressive follicle from CNN3 touches one of the expressive follicles, it is a candidate follicle.

All the candidate follicles characterized by eight-shape were eroded until they split into two or more regions. All resulting regions with area smaller than 100 pixels were eliminated and the remaining regions were dilated back. Finally, regions outside of the ovary detected in CNN4 were deleted.

The recognition rate of this algorithm was around 60% and the misidentification rate was around 30%. The main advantage of this algorithm is its speed. The algorithm proposed by Potočník *et al.* [37] reported an average of 6 minutes to segment an image using thresholding; Potočník and Zazula [35] reported an average of 12 minutes to segment one image using region growing; but this segmentation algorithm [9] only takes an average of 8 seconds to segment an image.

The performances of all follicle segmentation techniques are listed in Table 3.1.

Table 3.1: An overview of the performance of all existing follicle segmentation techniques.

Author	Segmentation Technique	RR	MR	HD (mm)	MAD(mm)	RMSD (mm)
Počnik <i>et al.</i>	optimal thresholding	70%	-	-	-	-
Sarty <i>et al.</i>	graph searching	-	-	1.47 ± 0.83	-	0.59 ± 0.28
Krivanek & Sonka	watershed	-	-	1.64 ± 0.92	-	0.63 ± 0.36
Potočník & Zazula	region growing	$78\% \pm 21\%$	$29\% \pm 25\%$	-	1.1 ± 0.40	-
Cigale & Zazula	cellular neural networks	60%	30%	-	-	-

CHAPTER 4

METHODOLOGY

The proposed follicle segmentation algorithm consists of two steps - potential follicles segmentation and potential follicles classification. Geometric active contour models were applied first, aiming to segment most dark regions (potential follicles) in an image. A naïve Bayes classifier was applied later to determine whether each dark region was a follicle or not. An overview of the algorithm is shown in Fig. 4.1.

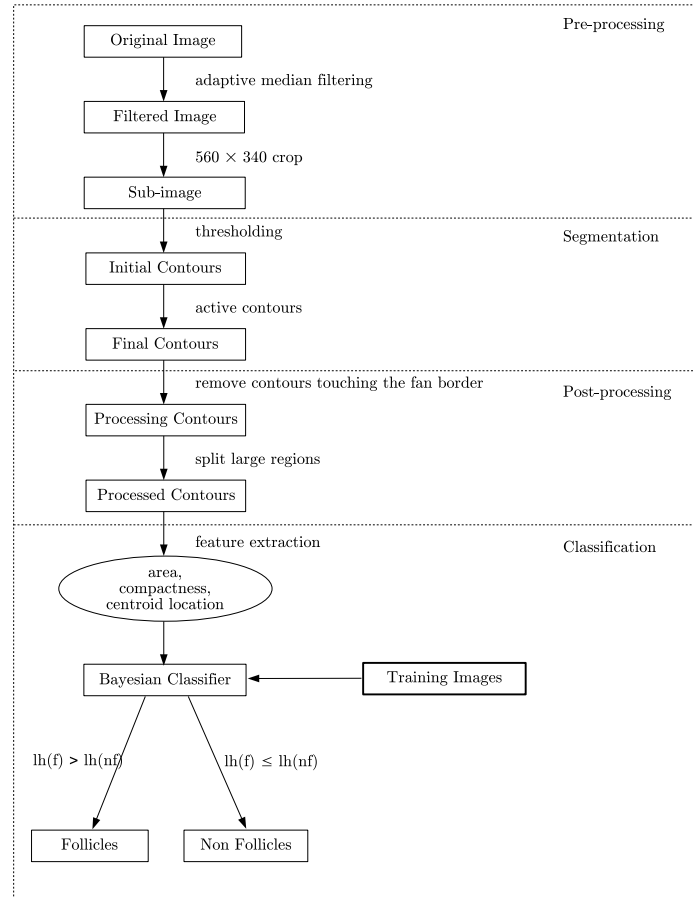
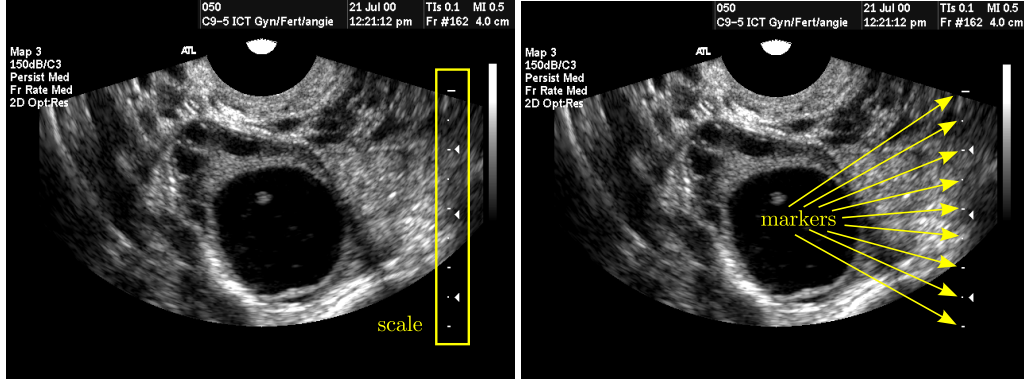


Figure 4.1: An overview of the segmentation and classification methodology.

4.1 Scale

Ultrasound images may be acquired with different zoom factors. Therefore, the physical dimensions of a pixel may differ among images. This section describes the method for obtaining the scale of the images used in this study.

The scale for the images used in this study is located on the right side of the images (Fig. 4.2). A scale contains 9 markers, which are short white lines. The distance between any two neighbouring markers is 5 mm. The number of pixels from the top marker to the bottom marker was manually counted. This number was divided by 8 (the number of gaps) to obtain the scale. Generally, the scale is the same for images within the same data set. For the images used in this study, 70 - 80 pixels represents 1 cm. This information will be used to compute validation metrics.



(a) An image showing the scale.

(b) The same image showing the scale markers.

Figure 4.2: An ultrasound image showing the scale and markers.

4.2 Pre-processing

The pre-processing step provides input images for the segmentation algorithm. The characteristics of the segmentation algorithm used in this study, that is, geometric active contours model (the choice of the segmentation algorithm will be explained in Section 4.3), are computationally expensive, and the accuracy of this algorithm depends mainly on the gradient information around object boundaries. The goal of the pre-processing step is therefore to crop the image to minimize the size, and to preserve boundaries while smoothing. This goal was achieved by reducing noise with an adaptive neighbourhood median filter followed by image cropping.

4.2.1 Noise Reduction

Generally, ultrasound images are of poor quality due to speckle noise. Applying segmentation techniques to these images directly leads to poor results. This noise reduction step reduces noise while preserving edge information.

The main segmentation technique used in this project is the active contours model (see Section 3.1.6); the details of the implementation will be given in Section 4.3. Initial contours are defined inside the follicles. Contours inflate and deform to recover the exact shape of follicles. The gradient information in the image controls how the contour deforms, and it is the main criteria for stopping the contour around the follicle boundaries.

The goal of noise reduction is to enhance the image gradient. Smoothing the interior of potential follicles stops the contour from “sticking” to noisy pixels inside the potential follicles. Preserving existing edge data ensures that the contour will stop at the potential follicle boundaries. Pixels which are outside of the follicles and are far away from the follicle boundaries are not as important during noise reduction.

A median filter was chosen to reduce noise because of its ability to preserve edges (Section 2.3.1.) In order to smooth pixels inside of potential follicles more thoroughly than pixels outside of potential follicles, an adaptive neighbourhood smoothing technique was used. Pixels inside potential follicles generally have low intensity, therefore, pixels below a threshold T_{filter} were smoothed by a large kernel k_l , and pixels above the threshold T_{filter} were smoothed by a small kernel k_s . The threshold T_{filter} was defined to be the mean gray-level of all pixels inside the fan area. This noise reduction process was repeated once to ensure that the follicle regions are well smoothed (Fig. 4.3) [25, 35].

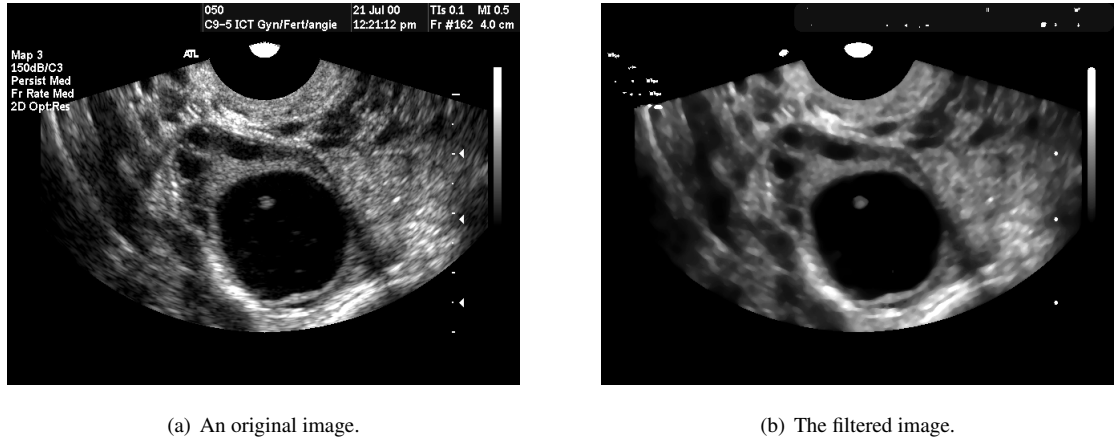


Figure 4.3: An example of filtering. An original image is filtered by an adaptive neighbourhood filter.

4.2.2 Sub-image Extraction

As was introduced in section 1.5, an ultrasound image contains a fan area which shows the object of interest, and a margin area which shows other information such as the patient's name and the date the image was taken. Because margin areas do not contain follicles and they add an extra work load to the segmentation algorithm, a sub-image which contains only the bounding box of the fan areas was extracted from original image (Fig. 4.4).

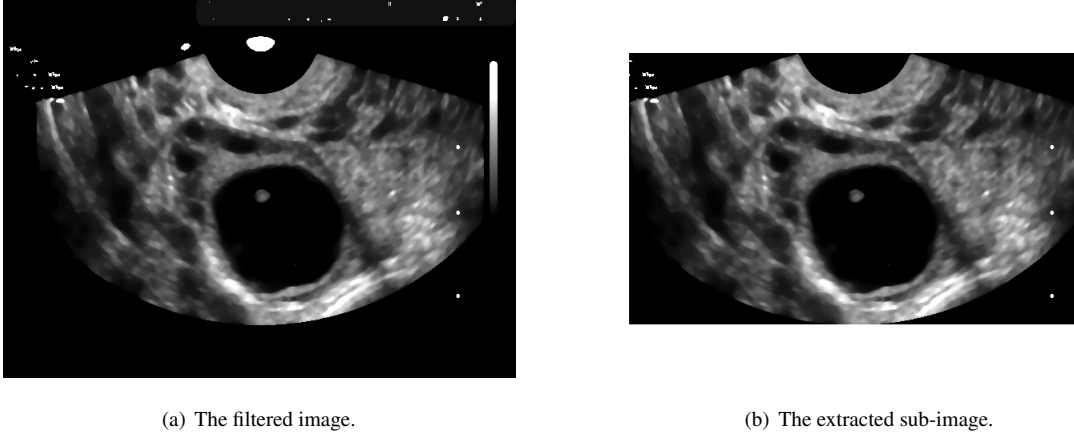


Figure 4.4: An example of sub-image extraction.

The best ultrasound imaging result is achieved when the object of interest is in the center of the image. Therefore, it is assumed that follicles will not appear close to the fan border. Although the fan areas of ultrasound images obtained from different ultrasound machines might be slightly different, a fixed fan area was used for all the images. Consequently, a fixed bounding box (560×340) was used to extract the sub-images from all of the original images.

4.3 Segmentation

Follicles appear as dark regions in ultrasound images. Therefore, the goal of this segmentation process is to identify most of the dark regions (potential follicles) in an ultrasound image. Further analysis will be applied to determine whether a potential follicle is a real follicle or not.

The choice of segmentation technique was motivated by the fact that follicles are homogeneous regions (with respect to brightness) and they are much darker than their surrounding areas. The active contour models, both parametric (Section 3.1.5) and geometric (Section 3.1.6), were considered for the choice of segmentation algorithm. In general, there is more than one follicle in one ultrasound image. Because parametric active contours (snakes) do not handle topological changes very well unless the initial contours have a bijective relationship with follicles (e.g. the number of follicles is exactly the same as the number of initial contours) they will do a poor job recovering the follicle boundaries. There is no such problem while

using the geometric active contour models. If multiple contours are used to represent one follicle in the initial contours, parametric active contour models will automatically merge together. If multiple potential follicle regions are initialized with a single contour, this contour will be separated automatically. Based on the above discussion, geometric active contour models were chosen to segment potential follicles.

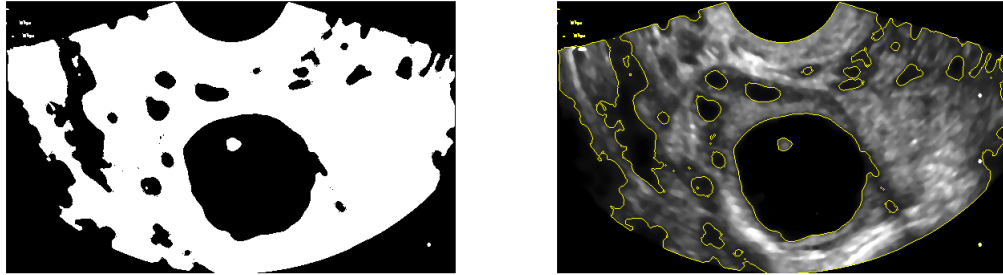
4.3.1 Initial Contours

The purpose of having an initial contour is to provide a starting contour for deformation. Theoretically, the position of an initial contour should not affect the final contour. In practice, the closer the initial contour to the final contour, the less time the segmentation process will take. Because the contour deformation uses normal force (whose sign is pre-determined by the location of the initial contours), the initial contours need to be either entirely inside of potential follicles (normal force is positive), or entirely outside of potential follicles (normal force is negative).

Because pixels inside follicle regions have consistently low intensity, it is easy to obtain initial contours that are entirely inside of potential follicles by thresholding the image with a small threshold value $T_{initialContour}$. The value of $T_{initialContour}$ is the mean gray-level of all pixels inside the fan area

$$T_{initialContour} = \frac{1}{|\Gamma|} \sum_{x \in \Gamma} f(x) \quad (4.1)$$

where Γ is the set of all the pixels in the fan area and $f(x)$ represents the values of the pixel x . Figure 4.5 shows the resulting initial contours. The initial level set function is constructed using the signed distance function of the initial contour.



(a) A resulting black and white image after thresholding

(b) The contour is shown on the filtered image

Figure 4.5: Thresholding the filtered image to obtain the initial contour

4.3.2 Speed Function

Although follicles generally have good contrast (resulting in continuous boundaries) inside ovaries, poor quality images may contain follicles with discontinuous boundaries. The speed function proposed by

Caselles et al. and Yezzi (Equation 4.2) was employed in this segmentation algorithm because its ability of segmenting objects which have discontinuous boundaries (Section 3.1.6.4). Let I be the input image, ϕ be the level set function 3.1.6.3, G_σ be a Gaussian filter that has width σ , and ε be the weighting parameter, the speed function (F) is defined as:

$$F = g_I(\pm 1 - \varepsilon \kappa) + \nabla g_I \cdot |\nabla \phi| \quad (4.2)$$

where

$$g_I = \frac{1}{1 + |\nabla(G_\sigma * I)|} \quad (4.3)$$

and

$$\kappa = \nabla \frac{\nabla \phi}{|\nabla \phi|} \quad (4.4)$$

4.3.3 Evolution

The derivative of the level set function was calculated using the 1st order accurate scheme [44]. To assure the accuracy of the level set method, a level set function needs to be a signed distance function of the current curve. This assumption was satisfied by re-initializing the level set function (Section 3.1.6.6) every $g = 10$ iterations.

Theoretically, the speed function will eventually become 0 (zero) everywhere on the surface, at which point the curve evolution stops. The resulting contours obtained from the zero level set are the final segmentation results. However, due to the numerical implementation of discretization, it is rare to have the speed function actually become 0 everywhere at the same time.

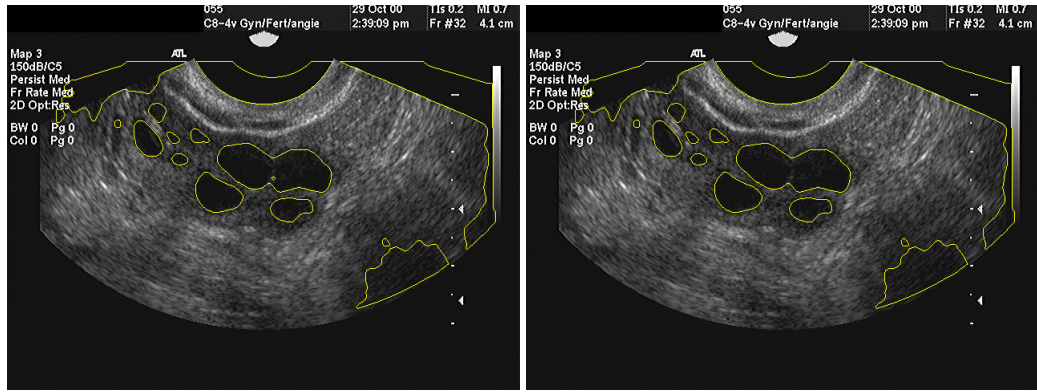
This project determines the stop point for each image dynamically - after each iteration of curve evolution, the regions under the current contours were compared with the regions under the contours obtained from the previous iteration. If the overlap of these two regions is more than 99.99% of the current regions, the curve evolution stops.

4.4 Post-processing

The main goal of the post-processing is to remove regions that are obviously not follicles based on *a priori* knowledge, and separate regions that were merged together during the noise reduction process.

4.4.1 Hole Filling

High intensity pixels inside potential follicle regions (such as noise or oocytes) were segmented automatically during the curve evolution process, and appear as holes in the segmented image. Such holes are not of interest, and were simply filled in (Fig. 4.6).



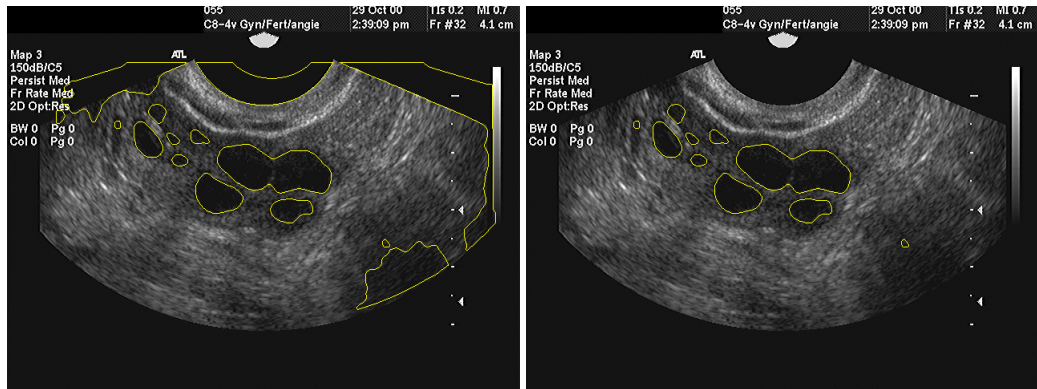
(a) Before filling in holes

(b) After filling in holes

Figure 4.6: Holes inside regions are filled in.

4.4.2 Remove Regions Which Touch The Fan Border

A fixed fan mask was used for all the images. Therefore, the segmentation algorithm might treat part of the margin areas of an ultrasound image as relevant image data. Because the margin area is mostly dark, the segmentation algorithm will segment these areas as potential follicles. Segmented regions were checked, and regions which were within 5 pixels of the fan border were removed (Fig. 4.7). This step did not remove any real follicles because our assumption that follicles do not appear near the fan border was true.



(a) Before removing regions which touch the fan border.

(b) After removing regions which touch the fan border.

Figure 4.7: Regions which touch the fan border are removed.

4.4.3 Separation of Large Regions

When two potential follicles were close together and the boundaries between them were not strong, the active contour models tended to segment them as one large region. Such a large region was generally not composed of more than two potential follicles because it was rare that all the boundaries among multiple

follicles were weak at the same time. However, when this rare case did happen, it would have caused mis-segmentation (will be analyzed in Section 5.3.2).

Among all the regions obtained from the previous step (Section 4.4.2), those with size larger than 1000 pixels (roughly twice of the area of the smallest visible follicle) were checked individually to determine whether such a region needs to be separated or not. For each such large region, a binary image that contains only the region was formed and eroded (Section 2.4.2) with a disk of radius 1 (Equation 4.5). The origin is at the center of the disk (Section 2.4.2).

$$\text{structure element} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.5)$$

The erosion was applied until the large region was completely eroded, or until the large region was eroded into more than one region, whichever occurred first. In the first case (Fig. 4.9), the large region was considered to be a large potential follicle and no separation was applied; in the second case (Fig. 4.10), the large region was considered to be composed of multiple potential follicles and was separated using watershed segmentation based on distance transform (Section 3.1.4.2 and Section 3.1.4.3) using the sub-regions obtained from erosion as internal markers, and pixels “exactly midway between the internal markers” as external markers (Fig. 4.8) [16]. The external markers were obtained by applying the watershed segmentation based on distance transform on the internal marker image [16].

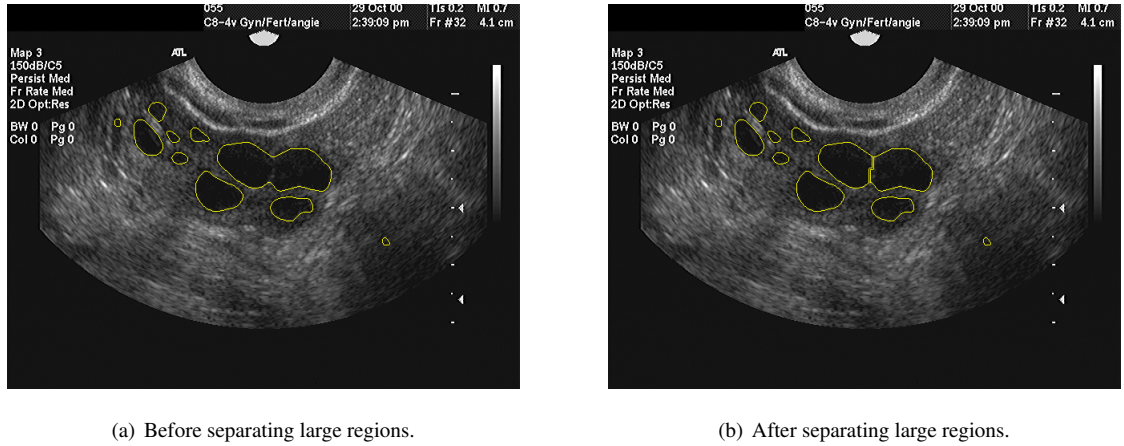
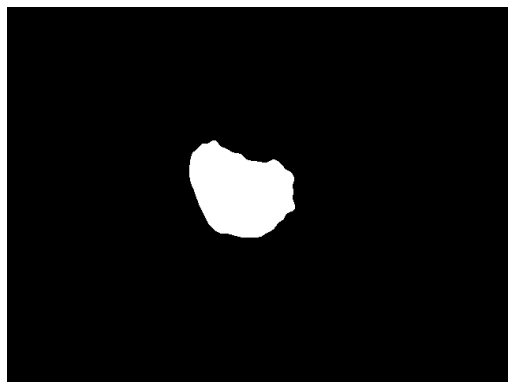


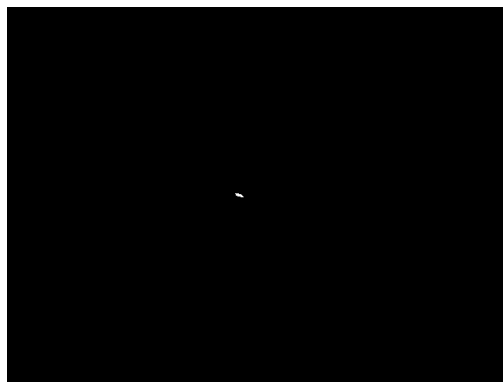
Figure 4.8: An example of separating large regions

4.5 Classification of Potential Follicle Regions

After removing the regions which touched the fan border and separating any large regions, all the potential follicles were classified using a naïve Bayes classifier (Section 3.2.5).

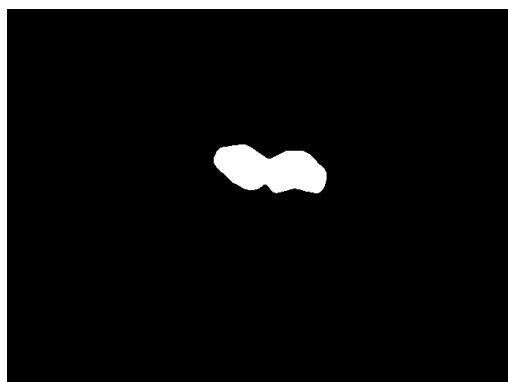


(a) One large region.

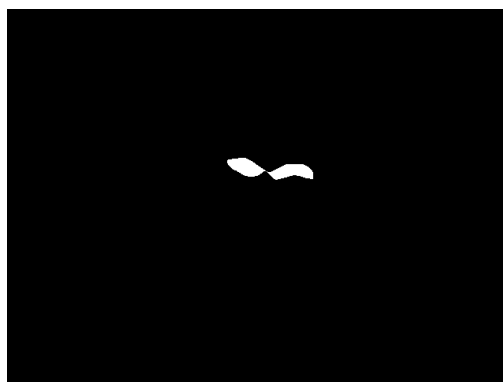


(b) The remaining region before it is completely eroded.

Figure 4.9: An example of eroding a large region until it completely eroded. Because there is always just one region in the image, the original large region is one single potential follicle.



(a) Two overlapping regions.



(b) The erosion operation separates the two regions in (a).

Figure 4.10: An example of eroding a large region into two regions. The large region is composed of two potential follicles and will be separated by watershed segmentation.

4.5.1 Feature Extraction

The goal of feature extraction is to obtain representative features that can be used to classify the object of interest. This section introduces the features which were used in this study along with the reason they were chosen.

4.5.1.1 Area

Area is the number of pixels inside a potential follicle. The diameter of a normal visible follicle is between 2 mm and 25 mm [35]. Assuming follicles are roughly circular in cross-section, the area of a normal follicle (using the formula area of circle) is between 3 mm² and 490 mm². The area of a potential follicle is a good feature to differentiate follicles from regions that are not follicles, particularly very small regions.

4.5.1.2 Compactness

As was defined in Section 2.5.1, the compactness of a potential follicle is the ratio of the square of the perimeter to the area: [46]:

$$\text{compactness} = \frac{\text{perimeter}^2}{\text{area}} \quad (4.6)$$

where the perimeter is the distance around the inner boundary of a potential follicle, and the area is the number of pixels inside the potential follicle. Because follicles are roughly spherical structures with roughly circular projections in ultrasound images and have a low compactness value (typical values from 14 - 20), region compactness is a good indicator of the likelihood that a potential follicle is a true follicle.

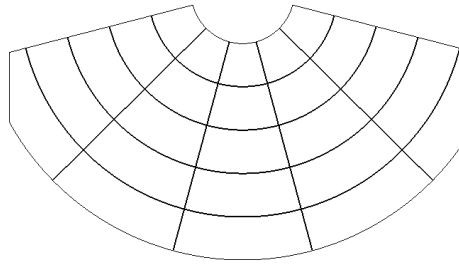
4.5.1.3 Centroid Location

The ultrasound image was divided into 5 × 5 sectors, and the centroid location of a potential follicle was defined as the index of the sector to which the centroid of the potential follicle belongs (Fig. 4.11). The number of the sectors (5 × 5) was selected so that it was small enough to limit the occurrence of 0 probabilities in the training set, and large enough to encode meaningful spatial information.

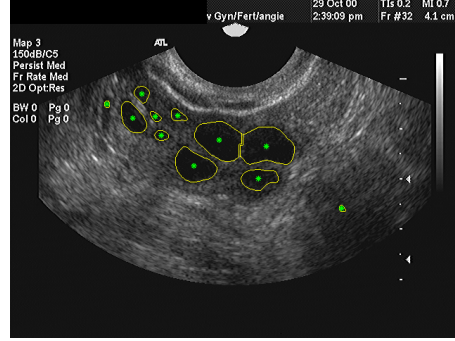
Objects of interest in an ultrasound image have the best visual quality when they are located in the center of the field of view. This fact leads to the assumption that follicles tend to be in the center of the fan area. If a potential follicle is located on the edge of the fan area, it is unlikely that it is a follicle. Centroid location was chosen as a feature since follicles are more likely to occur in centrally located sectors.

4.5.2 The Classifier

Clustering does not take advantage of the two well-defined classes (follicles / non-follicles). All the classifiers based on the distance are sensitive to every single features because the position of the point representing a pattern depends on individual features of that pattern. The naïve Bayes classifier calculates the likelihoods of a pattern being in any classes and assigns the pattern to the class with the maximum likelihood (Section



(a) A fan area is divided into 5×5 sectors.



(b) Potential follicles with marked center of mass.

Figure 4.11: An example of obtaining the centroid location of potential follicles.

3.2.5). With a naïve Bayes classifier, an unrepresentative feature may be offset by a strong representative feature of the same pattern. It is simple to implement, fast to train and easy to use. Potential follicles were determined to be follicles or non-follicles using a naïve Bayes classifier.

4.5.3 Training and Testing

4.5.3.1 Training and Testing Methodology

Gold standard follicles were manually segmented by experts, which was tedious and time-consuming. This constrained the amount of available testing data to an acceptable number (40 images) for which the expert was willing to provide a manual segmentation, resulting in a fixed testing set (the 40 manually segmented images). Standard training and testing methodologies introduced in Section 3.3 selected testing set randomly, and therefore none of them were applicable for this research. A new training and testing method was introduced in this research.

The forty testing images were randomly selected from the data. The rest of the data was randomly and evenly divided into $N = 4$ training data sets. The classifier was trained with each training set and tested on the same testing data. The performance of the 4 tests were reported individually. This training and testing method reduced the expert's work load while preserving the objectivity of testing. However, because the testing set was fixed, the reported performances were biased towards that particular testing set.

4.5.3.2 Training

Each training set consisted of images with potential follicles obtained from the segmentation algorithm (Section 4.3). Each potential follicle in a training set was manually decided to be a true follicle or not based on the observer's experience. The area, compactness and centroid location of each potential follicle was recorded as well as its class (follicle or non-follicle). The following probabilities were obtained:

- The *a priori* probability of an arbitrary potential follicle being a follicle ($p(f)$, the percentage of follicles in the training set) and a non-follicle ($p(nf)$, the percentage of non-follicles in the training set)

$$p(f) = \frac{|F|}{|F| + |NF|} \quad (4.7)$$

$$p(nf) = \frac{|NF|}{|F| + |NF|} \quad (4.8)$$

where F is the set of follicles from the training set, NF is the set of non-follicles from the training set, and $|\cdot|$ denotes the number of elements in the set.

- The probability distribution of a follicle having a certain area $p(A|f)$, and the probability of a non-follicle having certain areas $p(A|nf)$
- The probability distribution of a follicle having a certain compactness $p(C|f)$, and the probability of a non-follicle having a certain compactness $p(C|nf)$
- The probability of a follicle having a certain centroid location $p(L|f)$ (given n locations, $L = \{l_i, i = 1, 2, \dots, n\}$), and the probability of a non-follicle having a certain centroid location $p(L|nf)$

$$p(l_i|f) = \frac{\sum |F_{l_i}|}{|F|} \quad (4.9)$$

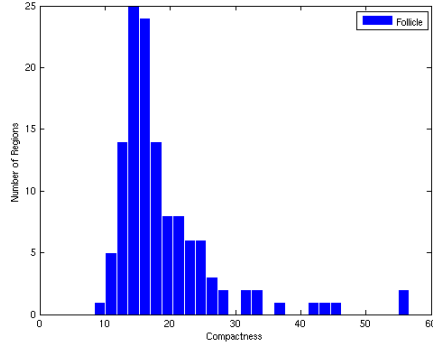
$$p(l_i|nf) = \frac{\sum |NF_{l_i}|}{|NF|} \quad (4.10)$$

where F_{l_i} was the set of follicles with location l in the training set, and NF_{l_i} was the set of non-follicles with location l in the training set.

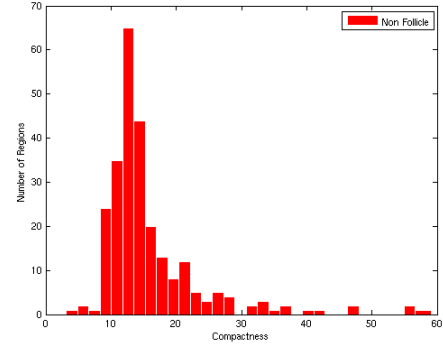
In order to obtain the probability distribution of the area and the compactness, area values and compactness values obtained from the training set needed to be discretized (binned, Section 3.2.5.2). Although Sturges' rule (Section 3.2.5.2) tries to minimize the number of empty bins, it tends to create bins with large width, which over-smooth the histogram and miss the features of the distribution. Freedman-Diaconis rule (Section 3.2.5.2) takes the data into consideration, and therefore was selected to determine the bin width ($2IRQ(x)n^{-\frac{1}{3}}$, where $IRQ(x)$ was the interquartile range of the sample and n was the number of sample) to perform the discretization (Fig. 4.12).

The binning process was applied separately for both follicles and non-follicles. For each feature (area or compactness), feature values were sorted in ascending order with the minimum value v_{min} and the maximum value v_{max} , and values within the range of $v_{min} + i \frac{n}{2IRQ(x)n^{-\frac{1}{3}}}$ and $v_{min} + (i+1) \frac{n}{2IRQ(x)n^{-\frac{1}{3}}}$ were assigned a single value. The follicle area distribution, follicle compactness distribution, non-follicle area distribution and non-follicle compactness distribution, were obtained separately.

Because the *a priori* probability depends heavily on the segmentation algorithm, the training of the classifier is biased by the specific segmentation algorithm. Replacing with another segmentation algorithm requires retrain the classifier, and may lead to a different classification/misidentification rate.



(a) Discretizing compactness feature for follicles.



(b) Discretizing compactness feature for non-follicles.

Figure 4.12: Example of discretizing the compactness feature.

4.5.3.3 Testing

Given a potential follicle with area a , compactness c , and centroid location l , the likelihood of its being a follicle $p(f|a, c, l)$ and the likelihood of its being a non-follicle $p(nf|a, c, l)$ is obtained by

$$lh(f|a, c, l) = p(f)p(a|f)p(c|f)p(l|f) \quad (4.11)$$

$$lh(nf|a, c, l) = p(nf)p(a|nf)p(c|nf)p(l|nf) \quad (4.12)$$

where the notation $p(a|f)$ denotes the probability that a follicle having an area of the range that a was contained, and the probability distribution were obtained from the training set. Given two classes ($\omega = \{f, nf\}$), the naïve Bayes classifier assigns the potential follicle (\vec{x}) to the class with the maximum likelihood (Section 3.2.5):

$$\omega = \underset{\omega_i}{\operatorname{argmax}} lh(\omega_i|\vec{x}) \quad (4.13)$$

$$= \underset{\omega_i}{\operatorname{argmax}} p(\omega_i)p(a|\omega_i)p(c|\omega_i)p(l|\omega_i) \quad (4.14)$$

In other words, if $lh(f|a, c, l) > lh(nf|a, c, l)$, this potential follicle is determined to be a follicle, otherwise this potential follicle is determined not to be a follicle. When $p(a) = 0$ or $p(c) = 0$ (when there was an empty bin in the area or compactness distribution), the probability was replaced by the average of two closest neighbours if it was not at the end of the distribution, or remained 0 if it was.

CHAPTER 5

EXPERIMENTS & RESULTS

5.1 Experiment Design

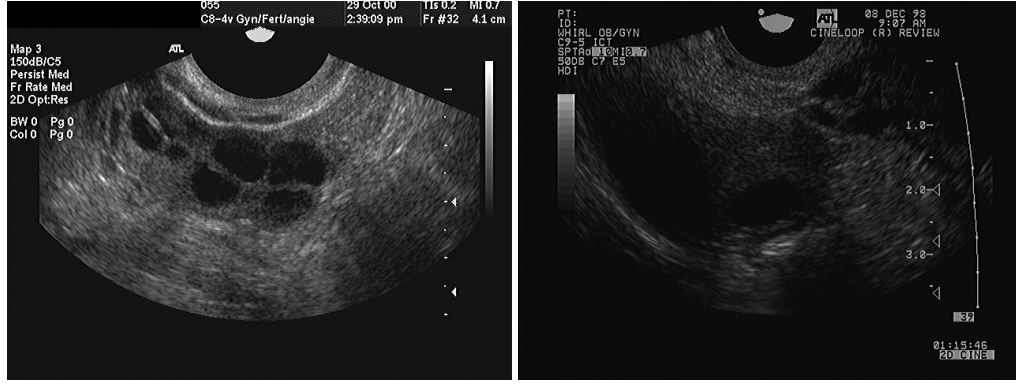
An experiment was designed to test the hypothesis that geometric active contour models can detect a large number of visible follicles in human ovarian ultrasound images acquired *in vivo* with the accuracy greater than that of existing methods when compared to the same gold standard and the hypothesis that a naïve Bayes classifier can subsequently eliminate a large number of false positive regions to reduce the misidentification rate.

5.1.1 Image Data

All the ultrasound images used in this study were collected at the Women’s Health Imaging Research Laboratory, Department of Obstetrics Gynecology and Reproductive Sciences, Royal University Hospital, Saskatoon, Saskatchewan, Canada. All the images are size of 640×480 pixels in 256 gray-levels. The scale of these images is about 70 – 80 pixels/cm. Three hundred and sixty images from 9 women with no known follicle disorders were selected, and divided into 2 sets, good scans (containing 192 images) and bad scans (containing 168 images) according to how easily an expert could interpret them (Fig. 5.1). Clinically, bad scans were discarded due to their poor quality. Therefore, bad scans were not used in this study. Forty testing images were randomly selected from the good scans set. The remaining 152 images in the good scans set were evenly divided into 4 disjoint training sets (each with 38 test images).

5.1.2 Hypothesis Testing: Comparing Segmented Images

All the good scans (192 images) were segmented by the algorithm proposed in Section 4.3. Four runs of the training-testing process were performed. For each run, potential follicles in one set of training images were determined to be a follicle or a non-follicle manually, and potential follicles in all the testing images were classified by a naïve Bayes classifier using the probability distribution obtained from that training set (Section 4.5). Five sets of images with automatically segmented regions were generated: regions obtained by segmentation and classification using training set 1 (A1); regions obtained by segmentation and classification using training set 2 (A2); regions obtained by segmentation and classification using training set 3



(a) An example of a good scan.

(b) An example of a bad scan.

Figure 5.1: Examples of a good scan and a bad scan.

(A3); regions obtained by segmentation and classification using training set 4 (A4), and regions obtained by segmentation, but before classification (A5).

The set A5 tests the performance of active contour models; the A1 - A4 sets test the performance of the naïve Bayes classifier. In order to test the algorithm's performance on follicles with different diameters (average of the major and minor axis, Section 2.5.2), the gold standard image set was further processed to create five additional sets (filtered by the follicle size): the original gold standard (G1); gold standard containing only follicles with diameters greater than or equal to 2 mm (G2); gold standard containing only follicles with diameters greater than or equal to 3 mm (G3); gold standard containing only follicles with diameters greater than or equal to 4 mm (G4); gold standard containing only follicles with diameters greater than or equal to 5 mm (G5); and gold standard containing only follicles with diameters greater than or equal to 6 mm (G6)

Each set of automatically segmented regions (A_i) was compared with each gold standard set (G_i), and the evaluation was reported for each combination.

5.1.3 Evaluation

The evaluation had two aspects - to test how close the contour of the automatically segmented regions were to the contours drawn by the expert (boundary accuracy), and to test how many follicles (as a percentage) were correctly and incorrectly detected (follicle detection).

5.1.3.1 Boundary Accuracy

A follicle was considered to have been recognized by the proposed algorithm (Section 4) if the Dice coefficient (Section 3.4.2) between the automatically segmented region and the gold standard (expert segmented region) was greater than or equal to 0.5. For every recognized follicle, the Hausdorff distance (HD), mean absolute distance (MAD) and root mean square distance (RMSD) (Section 3.4.1) between the contours of

the automatically segmented regions and the gold standard (contours drawn by an expert) were calculated (Section 3.4.1). The average (m) and standard deviation (sd) of the HD, MAD, and RMSD among all the testing images in each run were reported in the form $m \pm sd$.

5.1.3.2 Follicle Detection

Follicle detection was evaluated using four rates - the image recognition rate (IRR), the image misidentification rate (IMR), the follicle recognition rate (FRR), and the follicle misidentification rate (FMR).

- IRR is the ratio between the number of true positives (TP, recognized follicles, Section 3.4.2.2) in an image and the number of follicles segmented by the expert (GS) from the same image:

$$IRR = \frac{|TP|}{|GS|} \quad (5.1)$$

- IMR is the ratio between the misidentified regions (both false positives and false negatives, Section 3.4.2.2) in an image and the number of follicles segmented by the active contour models (AT) in the same image:

$$IMR = \frac{|FP| + |FN|}{|AT|} \quad (5.2)$$

(AT does not include regions which touch the fan border)

- FRR is the ratio between the number of true positives (TP_a , recognized follicles, Section 3.4.2.2) in **all testing images** and the number of follicles segmented by the expert (GS_a) from **the same image set**:

$$FRR = \frac{|TP_a|}{|GS_a|} \quad (5.3)$$

- FMR is the ratio between the misidentified regions (both false positives and false negatives, Section 3.4.2.2) in **all testing image** and the number of follicles segmented by the active contour models (AT_a) in **the same image set**:

$$FMR = \frac{|FP_a| + |FN_a|}{|AT_a|} \quad (5.4)$$

(AT_a does not include regions that touch the fan border)

The IRR and IMR of each image were calculated. The average (m) and standard deviation (sd) of the IRR and IMR for among all the testing images is reported in the form $m \pm sd$. The average (m) of the FRR and FMR is also reported.

For each comparing image sets (A_i and G_j , Section 5.1.2), the HD, MAD, RMSD, IRR, IMR, FRR, FMR were calculated. The complete results are listed in Appendix B, and notable results are analyzed in the following section.

5.2 Results & Analysis

5.2.1 Segmentation Results & Analysis

5.2.1.1 Recognition Rate & Misidentification Rate

After segmentation, but before classification, if all the segmented regions are considered follicles, the IRR is $43\% \pm 26\%$, the IMR is $67\% \pm 20\%$, the FRR is 40% and the FMR is 69% . If only follicles from the gold standard with diameter above a threshold (D) are considered, the observed values of the IRR, IMR, FRR, and FMR are reported in Table 5.1.

Table 5.1: The image recognition rate (IRR), image misidentification rate (IMR), follicle recognition rate (FRR) and follicle misidentification rate (FMR) for follicles with diameters (D) greater than or equal to different values before classification.

	all follicles	$D \geq 2 \text{ mm}$	$D \geq 3 \text{ mm}$	$D \geq 4 \text{ mm}$	$D \geq 5 \text{ mm}$	$D \geq 6 \text{ mm}$
IRR	$43\% \pm 26\%$	$54\% \pm 27\%$	$71\% \pm 31\%$	$77\% \pm 33\%$	$85\% \pm 28\%$	$88\% \pm 28\%$
IMR	$67\% \pm 20\%$	$67\% \pm 21\%$	$69\% \pm 21\%$	$74\% \pm 19\%$	$75\% \pm 18\%$	$76\% \pm 18\%$
FRR	40%	52%	70%	77%	86%	90%
FMR	68%	69%	71%	75%	78%	82%

The values of IRR and FRR for follicles with diameters greater than or equal to 3 mm (IRR = 71% , FRR = 70%), 4 mm (IRR = 77% , FRR = 77%), 5 mm (IRR = 85% , FRR = 86%) and 6 mm (IRR = 88% , FRR = 90%) demonstrates that the segmentation algorithm (Section 4.3) can successfully detect most medium/large follicles with diameter greater than or equal to 3 mm , 4 mm , 5 mm , and 6 mm . The low IRR value and low FRR value for all follicles (IRR = 43% , FRR = 40%), and follicles with diameters greater than or equal to 2 mm (IRR = 54% , FRR = 52%) shows the segmentation algorithm (Section 4.3) had difficulty detecting small follicles (diameter smaller than 3 mm), which are not usually visible to the untrained naked eye.

All the IMR values ($67\% - 76\%$) and FMR values ($68\% - 82\%$) are high, indicating that further analysis is required to remove the non-follicle regions. A naïve Bayes classifier was used in this research, to remove these regions (Section 4.5).

5.2.1.2 Distance Metrics

The Hausdorff distance (HD), mean absolute distance (MAD), and root mean square distance (RMSD) of the proposed segmentation algorithm is compared to those of the existing follicle segmentation algorithms (Section 3.5) in Table 5.2.

Both the HD ($2.43 \pm 1.46 \text{ mm}$) and the RMSD ($0.86 \pm 0.49 \text{ mm}$) are larger than the best performance in the previous work (HD = $1.47 \pm 0.83 \text{ mm}$, and RMSD = 0.59 ± 0.28) [43], but both the mean and the

Table 5.2: A review of the Hausdorff distance (HD), mean average distance (MAD), root mean square distance (RMSD) of existing follicle segmentation algorithms, as well as the proposed follicle segmentation algorithm.

Author	Segmentation Technique	HD (mm)	MAD(mm)	RMSD (mm)
Počnik et al.	optimal thresholding	-	-	-
Sarty et al.	graph searching	1.47 ± 0.83	-	0.59 ± 0.28
Krivanek & Sonka	watershed	1.64 ± 0.92	-	0.63 ± 0.36
Potočník & Zazula	region growing	-	1.1 ± 0.40	-
Cigale & Zazula	cellular neural networks	-	-	-
Harrington	active contour models	2.43 ± 1.46	0.75 ± 0.32	0.86 ± 0.49

standard deviation of MAD ($0.75 \pm 0.32 \text{ mm}$) for the proposed algorithm are smaller than in the previous work ($1.1 \pm 0.40 \text{ mm}$) [35]. The large HD and large RMSD indicates that when the proposed algorithm erred in detecting a boundary, it was very far from the gold standard. The small MAD indicates that the proposed algorithm is able to accurately detect boundaries on average.

5.2.2 Potential Follicle Classification Results & Analysis

5.2.2.1 Recognition Rate & Misidentification Rate

The image recognition rate (IRR), image misidentification rate (IMR), follicle recognition rate (FRR) and follicle misidentification rate (FMR) before and after classification are listed in Tables 5.3, 5.4, 5.5 and 5.6. These tables correspond to the four training sets which were used. The IRR, IMR, FRR, and FMR are also reported, in the same tables when considering only follicles from the gold standard with medium and large diameters (D).

All four tables show that both IMR and FMR are reduced dramatically (35% – 51% percentage points) while IRR and FRR are reduced slightly (3% – 11% percentage points), indicating that the naïve Bayes classifier (Section 4.5) removes a large number of the non-follicle regions present in the segmentation result, at the cost of removing a small number of true follicles. The four tables also show that the classifier is able to generate repeatable results.

5.2.2.2 Distance Metrics

The distance metrics before and after classification are reported in Table 5.7. The distance metrics did not change significantly after classification. This is due to the fact that only corresponding regions were used when calculating distances. The non-follicle regions which were removed by the classifier did not affect the

Table 5.3: The image recognition rate (IRR), image misidentification rate (IMR), follicle recognition rate (FRR) and follicle misidentification rate (FMR) before and after classification using training set 1. The number of percentage points dropped after the classification is also reported.

A1	Type	all follicles	$D \geq 2 \text{ mm}$	$D \geq 3 \text{ mm}$	$D \geq 4 \text{ mm}$	$D \geq 5 \text{ mm}$	$D \geq 6 \text{ mm}$
IRR	Before	$43\% \pm 26\%$	$54\% \pm 27\%$	$71\% \pm 31\%$	$77\% \pm 33\%$	$85\% \pm 28\%$	$88\% \pm 28\%$
	After	$36\% \pm 26\%$	$46\% \pm 29\%$	$61\% \pm 33\%$	$69\% \pm 37\%$	$76\% \pm 34\%$	$77\% \pm 35\%$
	Reduction	7%	8%	10%	8%	9%	11%
IMR	Before	$67\% \pm 20\%$	$67\% \pm 21\%$	$69\% \pm 21\%$	$74\% \pm 19\%$	$75\% \pm 18\%$	$76\% \pm 18\%$
	After	$28\% \pm 22\%$	$27\% \pm 22\%$	$27\% \pm 22\%$	$30\% \pm 23\%$	$32\% \pm 24\%$	$36\% \pm 25\%$
	Reduction	39%	40%	42%	44%	43%	40%
FRR	Before	40%	52%	70%	77%	86%	90%
	After	33%	44%	60%	68%	76%	79%
	Reduction	7%	8%	10%	9%	10%	11%
FMR	Before	68%	69%	71%	75%	78%	82%
	After	27%	26%	26%	28%	30%	33%
	Reduction	41%	43%	45%	47%	48%	49%

Table 5.4: The image recognition rate (IRR), image misidentification rate (IMR), follicle recognition rate (FRR) and follicle misidentification rate (FMR) before and after classification using training set 2. The number of percentage points dropped after the classification is also reported.

A2	Type	all follicles	$D \geq 2 \text{ mm}$	$D \geq 3 \text{ mm}$	$D \geq 4 \text{ mm}$	$D \geq 5 \text{ mm}$	$D \geq 6 \text{ mm}$
IRR	Before	$43\% \pm 26\%$	$54\% \pm 27\%$	$71\% \pm 31\%$	$77\% \pm 33\%$	$85\% \pm 28\%$	$88\% \pm 28\%$
	After	$39\% \pm 28\%$	$49\% \pm 31\%$	$63\% \pm 35\%$	$70\% \pm 38\%$	$77\% \pm 36\%$	$78\% \pm 37\%$
	Reduction	4%	5%	8%	7%	8%	10%
IMR	Before	$67\% \pm 20\%$	$67\% \pm 21\%$	$69\% \pm 21\%$	$74\% \pm 19\%$	$75\% \pm 18\%$	$76\% \pm 18\%$
	After	$31\% \pm 24\%$	$32\% \pm 24\%$	$32\% \pm 24\%$	$36\% \pm 25\%$	$37\% \pm 24\%$	$41\% \pm 23\%$
	Reduction	36%	35%	37%	38%	38%	35%
FRR	Before	40%	52%	70%	77%	86%	90%
	After	35%	47%	63%	71%	79%	82%
	Reduction	5%	5%	7%	6%	7%	8%
FMR	Before	68%	69%	71%	75%	78%	82%
	After	30%	30%	31%	34%	37%	40%
	Reduction	38%	39%	40%	41%	41%	42%

Table 5.5: The image recognition rate (IRR), image misidentification rate (IMR), follicle recognition rate (FRR) and follicle misidentification rate (FMR) before and after classification using training set 3. The number of percentage points dropped after the classification is also reported.

A3	Type	all follicles	$D \geq 2 \text{ mm}$	$D \geq 3 \text{ mm}$	$D \geq 4 \text{ mm}$	$D \geq 5 \text{ mm}$	$D \geq 6 \text{ mm}$
IRR	Before	$43\% \pm 26\%$	$54\% \pm 27\%$	$71\% \pm 31\%$	$77\% \pm 33\%$	$85\% \pm 28\%$	$88\% \pm 28\%$
	After	$37\% \pm 25\%$	$47\% \pm 28\%$	$62\% \pm 33\%$	$69\% \pm 36\%$	$77\% \pm 33\%$	$81\% \pm 32\%$
	Reduction	6%	7%	9%	8%	8%	7%
IMR	Before	$67\% \pm 20\%$	$67\% \pm 21\%$	$69\% \pm 21\%$	$74\% \pm 19\%$	$75\% \pm 18\%$	$76\% \pm 18\%$
	After	$25\% \pm 20\%$	$25\% \pm 20\%$	$26\% \pm 21\%$	$30\% \pm 22\%$	$30\% \pm 22\%$	$33\% \pm 21\%$
	Reduction	42%	42%	43%	44%	45%	43%
FRR	Before	40%	52%	70%	77%	86%	90%
	After	35%	45%	62%	68%	78%	82%
	Reduction	5%	7%	8%	9%	9%	8%
FMR	Before	68%	69%	71%	75%	78%	82%
	After	23%	23%	24%	27%	29%	31%
	Reduction	45%	46%	47%	48%	49%	51%

Table 5.6: The image recognition rate (IRR), image misidentification rate (IMR), follicle recognition rate (FRR) and follicle misidentification rate (FMR) before and after classification using training set 4. The number of percentage points dropped after the classification is also reported.

A4	Type	all follicles	$D \geq 2 \text{ mm}$	$D \geq 3 \text{ mm}$	$D \geq 4 \text{ mm}$	$D \geq 5 \text{ mm}$	$D \geq 6 \text{ mm}$
IRR	Before	$43\% \pm 26\%$	$54\% \pm 27\%$	$71\% \pm 31\%$	$77\% \pm 33\%$	$85\% \pm 28\%$	$88\% \pm 28\%$
	After	$40\% \pm 28\%$	$50\% \pm 30\%$	$63\% \pm 34\%$	$70\% \pm 36\%$	$76\% \pm 34\%$	$77\% \pm 37\%$
	Reduction	3%	4%	8%	7%	9%	11%
IMR	Before	$67\% \pm 20\%$	$67\% \pm 21\%$	$69\% \pm 21\%$	$74\% \pm 19\%$	$75\% \pm 18\%$	$76\% \pm 18\%$
	After	$34\% \pm 23\%$	$34\% \pm 23\%$	$35\% \pm 24\%$	$39\% \pm 23\%$	$40\% \pm 24\%$	$46\% \pm 23\%$
	Reduction	33%	33%	34%	35%	35%	30%
FRR	Before	40%	52%	70%	77%	86%	90%
	After	35%	47%	63%	70%	76%	79%
	Reduction	5%	5%	7%	7%	10%	11%
FMR	Before	68%	69%	71%	75%	78%	82%
	After	33%	33%	34%	38%	41%	44%
	Reduction	35%	36%	37%	37%	37%	38%

distance metrics at all because they were not originally considered (they did not correspond to a region in the gold standard).

A true follicle which is removed by the classifier will worsen (enlarge) a distance metric if the value of the follicle's distance metric is better (smaller) than the average, or improve (shrink) a distance metric if the value of the follicle's distance metric is worse (larger) than the average.

Table 5.7: The Hausdorff distance (HD), mean absolute distance (MAD), root mean square distance (RMSD) between contours drawn by an expert (filtered with follicle diameters) and contours of automatically segmented regions before (A5) and after classification using training set 1 (A1), training set 2 (A2), training set 3 (A3), and training set 4 (A4). The number of percentage points dropped after the classification is also reported.

	Type	all follicles	$D \geq 2 \text{ mm}$	$D \geq 3 \text{ mm}$	$D \geq 4 \text{ mm}$	$D \geq 5 \text{ mm}$	$D \geq 6 \text{ mm}$
HD	A5	2.43 ± 1.46	2.37 ± 1.51	2.44 ± 1.55	2.56 ± 1.63	2.64 ± 1.67	2.80 ± 1.74
	A1	2.29 ± 1.36	2.24 ± 1.34	2.25 ± 1.37	2.38 ± 1.42	2.43 ± 1.45	2.56 ± 1.51
	A2	2.33 ± 1.30	2.25 ± 1.29	2.28 ± 1.32	2.36 ± 1.38	2.41 ± 1.40	2.53 ± 1.45
	A3	2.41 ± 1.43	2.33 ± 1.42	2.36 ± 1.45	2.48 ± 1.51	2.53 ± 1.54	2.68 ± 1.59
	A4	2.36 ± 1.32	2.28 ± 1.31	2.31 ± 1.35	2.43 ± 1.40	2.50 ± 1.44	2.65 ± 1.48
MAD	A5	0.75 ± 0.32	0.69 ± 0.32	0.70 ± 0.33	0.70 ± 0.35	0.74 ± 0.36	0.78 ± 0.38
	A1	0.74 ± 0.34	0.68 ± 0.33	0.68 ± 0.34	0.68 ± 0.35	0.69 ± 0.35	0.73 ± 0.38
	A2	0.73 ± 0.28	0.67 ± 0.29	0.67 ± 0.29	0.66 ± 0.32	0.70 ± 0.32	0.73 ± 0.36
	A3	0.77 ± 0.36	0.71 ± 0.35	0.70 ± 0.36	0.70 ± 0.38	0.71 ± 0.37	0.75 ± 0.38
	A4	0.76 ± 0.28	0.69 ± 0.28	0.69 ± 0.29	0.69 ± 0.32	0.73 ± 0.32	0.77 ± 0.35
RMSD	A5	0.86 ± 0.49	0.82 ± 0.53	0.83 ± 0.54	0.86 ± 0.58	0.89 ± 0.58	0.95 ± 0.60
	A1	0.82 ± 0.46	0.79 ± 0.48	0.78 ± 0.49	0.82 ± 0.51	0.83 ± 0.51	0.89 ± 0.53
	A2	0.82 ± 0.42	0.77 ± 0.45	0.78 ± 0.46	0.80 ± 0.49	0.81 ± 0.47	0.86 ± 0.49
	A3	0.84 ± 0.44	0.79 ± 0.45	0.80 ± 0.46	0.83 ± 0.49	0.83 ± 0.47	0.89 ± 0.48
	A4	0.85 ± 0.45	0.80 ± 0.46	0.81 ± 0.48	0.84 ± 0.50	0.85 ± 0.49	0.90 ± 0.51

5.2.3 Performance Statistics

This experiment was run on a 2.2GHz Intel Xeon processor with 1G ram. The algorithm was implemented in Matlab (R2007a). It took an average of 5 minutes to segment one image (times ranged from 1.5 minutes to 10.5 minutes). In terms of the number of iterations of the level set method, it took an average of 350 iterations to segment one image (the number of iterations ranged from 110 to 750).

5.2.4 Analysis of Overall Results

The proposed segmentation-classification algorithm (Section 4.3) was tested on a very ambitious data set (even the good scans have poorer quality than those shown in examples in previous work on follicle segmentation), given detailed gold standard containing follicles smaller than 2 mm (difficult for anyone but an experimented human expert to detect). It performed poorly on small follicles (diameter smaller than 3 mm), but it has proved to be accurate in detecting medium to large sized follicles with diameter greater than or equal to 3 mm, 4 mm, 5 mm, and 6 mm. The geometric active contour models can detect most medium to large sized follicles, with generally accurate boundaries, at the cost of segmenting many non-follicle regions. These non-follicle regions may be removed by using a naïve Bayes classifier at the cost of losing a small number of true follicles. Both the hypothesis that geometric active contour models can detect a large number of visible follicles in human ovarian ultrasound images acquired *in vivo* with the accuracy greater than that of existing methods when compared to the same gold standard, and the hypothesis that a naïve Bayes classifier can subsequently eliminate a large number of false positives to reduce the misidentification rate were accepted.

5.3 Failure Case Analysis

In general, the proposed algorithm (Section 4) can accurately segment potential follicles and recognize true follicles. However, in some particular cases, the algorithm fails to segment follicles. This section investigates these failure cases.

5.3.1 Follicles Leaking to The Fan Border

Ultrasound imaging is based on some underlying assumptions, which are not always true in practice resulting in some ultrasound images which do not reflect the true tissue structures. These misrepresentations are called artifacts.

One assumption is that the path of ultrasound propagation from the transducer to the object being imaged and from the object back to the transducer is always straight. This assumption does not hold when the object is spherical or elliptic (such as an ovary). When the angle between ultrasound wave propagation and the normal direction of the object surface is large, the wave will not be reflected back to the transducer, and the edge will not show up in the ultrasound image (Fig. 5.2). This phenomenon is known as edge shadowing [41]. An example of edge shadowing in an ultrasound image is shown in Fig. 5.3.

When edge shadowing occurs, if a follicle overlaps with the shadow, the segmentation algorithm treats the shadow as part of the follicle (Fig. 5.4). If the shadow region extends to the fan border, the segmented follicle leaks into the fan border as well. Such follicles are removed under the assumption that no follicles

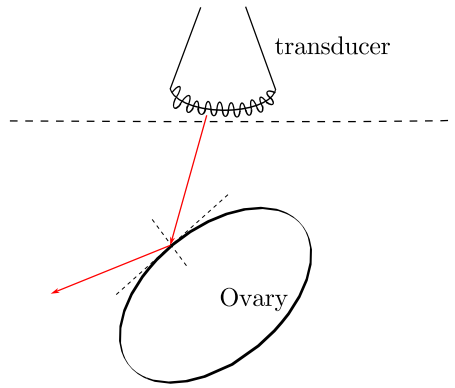


Figure 5.2: Arrows indicate the directions of wave propagation and reflection. Edge shadowing occurs when the angle of wave propagation and the normal direction of the object (ovary) surface is large. The wave will not be reflected back to the transducer.

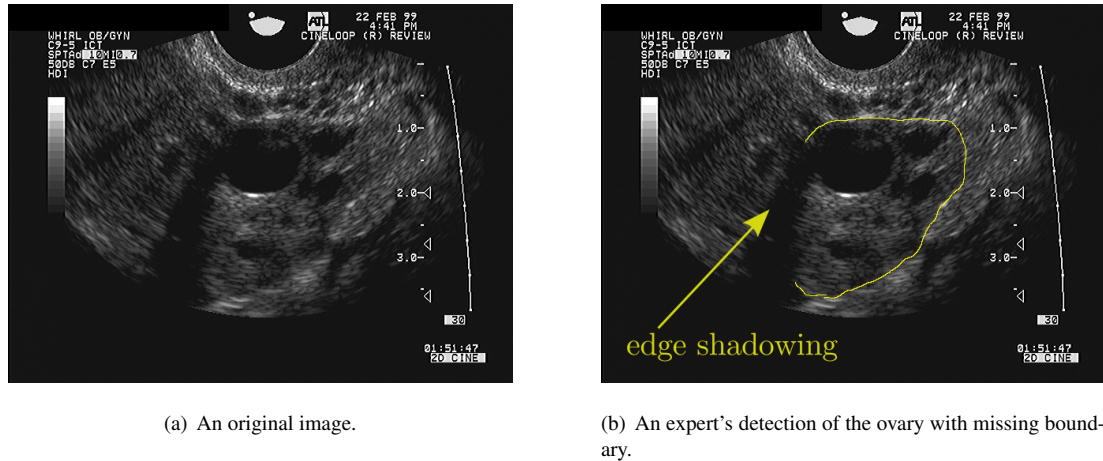


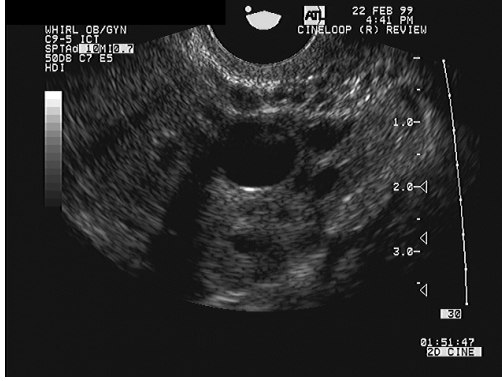
Figure 5.3: An example of edge shadowing in an ultrasound image. The ovary in (a) is partially detected by an expert in (b). Edge shadowing occurs on the left side of the ovary, making it difficult for the expert to detect the left half of the ovary's boundary.

touch the fan border (Section 4.4.2), resulting in a reduction in the number of true positives. The recognition rates are reduced by such follicles, but the misidentification rate is unaffected.

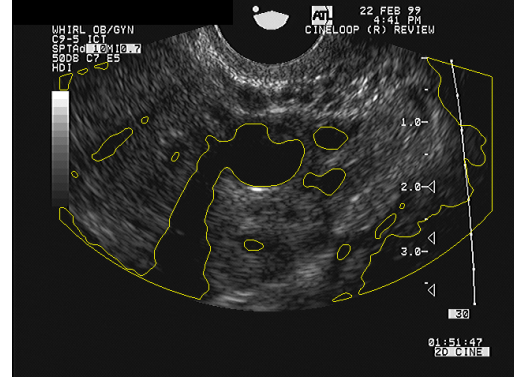
5.3.2 Multiple Follicles Are Treated as One

As introduced in Section 4.4.3, when two potential follicles are segmented as one during the segmentation process, the post processing separates them based on their shape. However, in the rare case that more than two potential follicles are close together and the boundaries between them are not outstanding, they are segmented as one potential follicle. If all the potential follicles that are close together are small, or if the combined region does not have narrow sections (Section 3.1.4.2), they will not be separated (Fig. 5.5).

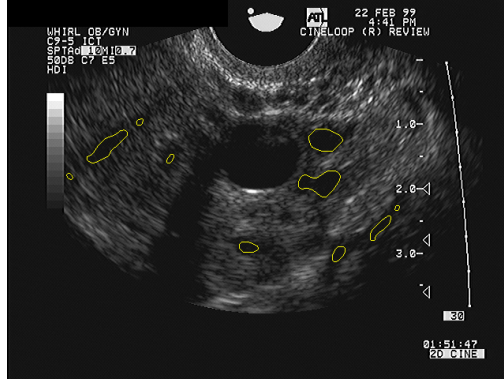
When multiple potential follicles are treated as one, any true follicles among the potential follicles are not considered recognized, resulting in a reduction in true positives. Depending on whether such a region



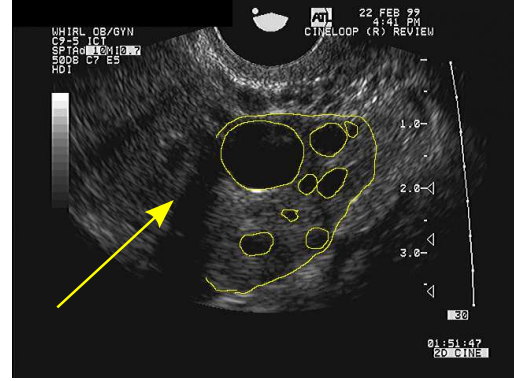
(a) The original image.



(b) Segmentation from the automated algorithm.



(c) After regions touching the fan border were removed. One follicle was removed.



(d) The expert's segmentation. The ovary edge is interpretable by the expert because of edge shadowing (pointed by an arrow).

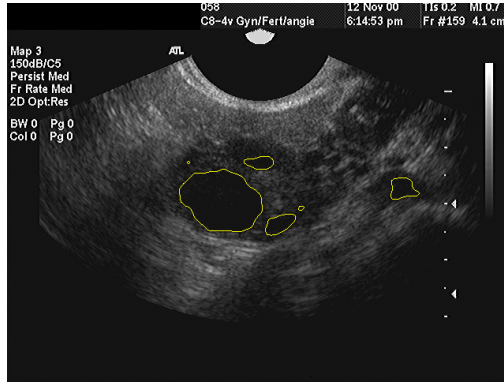
Figure 5.4: The segmentation algorithm leaked a follicle to touch the fan border because of edge shadowing.

is determined to be a follicle or a non-follicle by the naïve Bayes classifier, the misidentification rate might increase (if it is determined to be a follicle, it is considered a false positive), or might not be affected (if it is determined to be a non-follicle, it is neither a false positive nor a false negative). Such regions do not affect the segmentation evaluation result because they do not have a corresponding gold standard contour (Dice coefficient > 0.5) to be compared against.

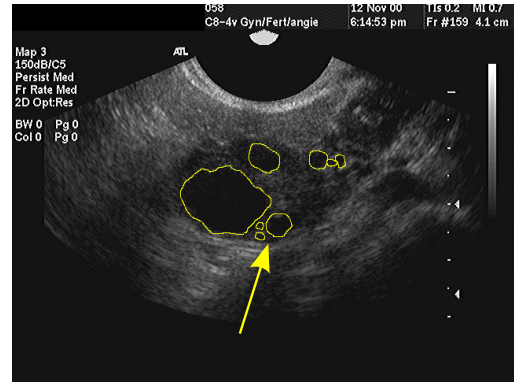
5.3.3 Over-Splitting

The purpose of applying the watershed segmentation technique is to separate overlapping potential follicles. However, if a single potential follicle is similar in shape to a region of overlapping potential follicle (narrow regions or a concavity on the boundary), either because of its natural shape or because of deficiencies in the segmentation process, it would be incorrectly split (Fig. 5.6).

If the splitting occurs at the center of a true follicle, two regions of similar area would be created, neither of which being large enough to have a corresponding region in the gold standard. The recognition rate is

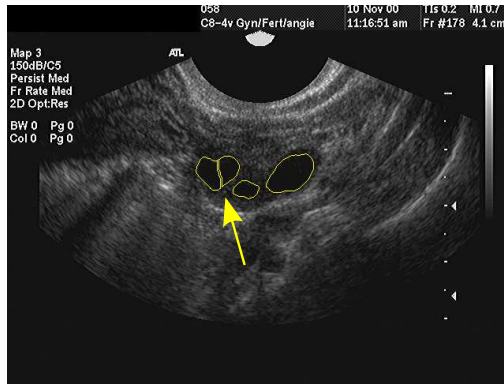


(a) The segmentation from the automated algorithm, before classification.

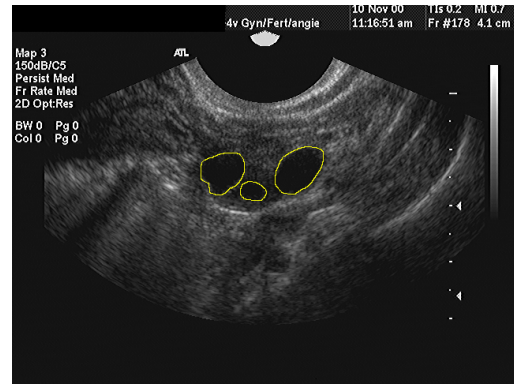


(b) The expert's segmentation.

Figure 5.5: Multiple follicles in gold standard (b) were treated as one in automated segmentation algorithm (a). This happens twice in this particular image (arrows)



(a) The segmentation from the automated algorithm.



(b) The expert's segmentation.

Figure 5.6: The follicle pointed by the arrow in (a) is incorrectly split because of its concavity.

thus reduced and the misidentification rate is increased, the distance metrics become either better or worse depending on whether the potential follicle before splitting has distance metrics worse or better than the average. If the splitting occurs around the edge of a true follicle, one small region and one large region are created. If the large region has a great enough area, it will correspond to the associated region in the gold standard, and the recognition rate is reduced, while the misidentification rate is unaffected. The distance metrics are increased since the DCP (Section 3.4.1) is incremented. If the large region's area is too small to have a corresponding region in the gold standard, the recognition rate is unaffected, the misidentification rate is not reduced, and the distance metrics become either better or worse depending on whether the potential follicle before splitting has distance metrics worse or better than the average.

5.3.4 Inexpressive Follicles

The regions which are used to provide the initial contours for the active contour models are obtained by thresholding the original images using the mean gray-level of all pixels inside the fan area 4.3.1. In some cases, this threshold value is too high to initialize inexpressive follicles (where follicles are brighter than average follicles), resulting in no final contours for such follicles. Inexpressive follicles generally cause a reduction in the recognition rate (true positive is reduced), but do not affect the misidentification rate.

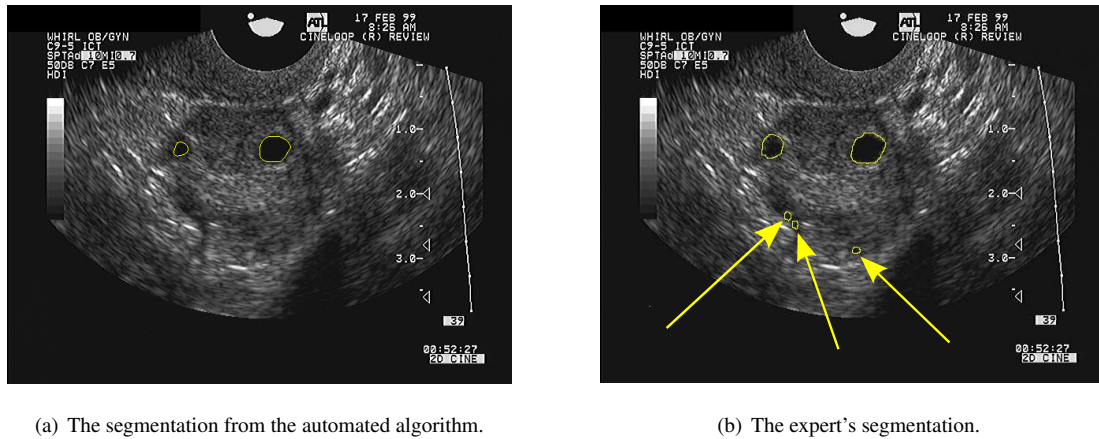


Figure 5.7: Inexpressive follicles (pointed by arrows) in (b) are too bright to have an initial contour. They are not segmented in (a).

CHAPTER 6

CONCLUSION

A new automatic follicle segmentation algorithm was proposed in this thesis. This new method consisted of two steps: segmentation of all dark regions in an image, and removal of the non-follicle regions from the set of segmented regions. The segmentation was achieved using geometric active contour models, and resulted in segmenting a larger number of follicles with diameter greater than or equal to 3 mm (follicle recognition rate = 70%), but also segmenting a larger number of non-follicles (follicle misidentification rate = 70%). A large number of non-follicle regions were subsequently removed using a naïve Bayes classifier which reduced the misidentification rate to 33% – 44% (35% – 51% percentage points reduction) with only a small (3% – 11% percentage points) reduction in recognition rate. When only follicles with diameter greater than or equal to 6 mm were considered, the recognition rate improved to 77% – 82%, and the misidentification rate increased to 33% – 46%.

Due to the poor quality of ultrasound images and a detailed gold standard (containing follicles with a diameter smaller than 2 mm, which are difficult for anyone but an experienced human expert to detect), the proposed algorithm performed poorly for small follicles (diameter smaller than 3 mm). The image/follicle recognition rate and misidentification rate were not as good as those of previously studied follicle segmentation techniques. However, it was shown capable of detecting a large number of follicles with diameter greater than or equal to 3 mm, and most follicles with diameter greater than or equal to 6 mm. The image/follicle recognition rate and the misidentification rate for follicles with diameter greater than or equal to 3 mm or larger were comparable to those of existing follicle segmentation techniques.

The Hausdorff distance (HD) and root mean square distance (RMSD) between contours of regions segmented by geometric active contour models and the gold standard were $2.43 \pm 1.46 \text{ mm}$ and $0.86 \pm 0.49 \text{ mm}$, which were larger than the best HD ($1.47 \pm 0.83 \text{ mm}$) and the best RMSD ($0.59 \pm 0.28 \text{ mm}$) among all the previous work. This indicated that when the proposed algorithm erred in detecting a boundary, it was further from the gold standard than that of existing follicle segmentation techniques. The mean absolute distance (MAD) between contours of regions segmented by geometric active contour models and the gold standard was $0.75 \pm 0.32 \text{ mm}$, which was smaller than the best MAD ($1.10 \pm 0.40 \text{ mm}$) among all the previous work. This demonstrated follicle boundaries obtained by the proposed algorithm were, on average, closer to the gold standard than those of existing follicle segmentation techniques.

The results partially support the hypothesis that geometric active contour models can detect a large number of visible follicles in human ovarian ultrasound images acquired *in vivo* with the accuracy greater than that of existing methods when compared to the same gold standard, and the hypothesis that a naïve Bayes classifier can subsequently eliminate a large number of false positives to reduce the misidentification rate.

The main disadvantage of the proposed segmentation algorithm is its speed. Due to the iterative nature of the geometric active contour models, it is not possible to use common techniques such as parallel computing to achieve a speed-up. The only option would be to move to higher performance sequential instruction hardware. Therefore, the proposed algorithm is currently not suitable for real time, affordable clinical use. However, the segmentation step of this algorithm could easily be replaced by a faster segmentation algorithm with a comparable recognition rate.

6.1 Future Work

The proposed follicle segmentation algorithm has several areas for possible improvement. The segmented non-follicle regions outside of an ovary could easily be eliminated if the ovary boundary were known. Due to the inconsistent shape and intensity of ovaries (in ultrasound images), there is no known effective ovary segmentation algorithm. However, an estimated ovary boundary which contains the entire ovary and a small portion of other regions (over-segmented) could still eliminate most of the non-follicle regions. Pixels that are outside of such an estimated ovary boundary do not need to be considered, and the speed of the segmentation step could be increased due to the reduction in the image data.

The initial contours were obtained by thresholding, which tends to miss inexpressive follicles because they generally have higher intensity. Inexpressive follicles could be initialized by detecting higher intensity pixels whose neighbourhood have similar intensities (such higher intensity pixels are most likely to be inside homogeneous areas).

The features used in the naïve Bayes classifier represent a general case. If *a priori* knowledge is provided about follicles in a particular data set, it could be added as an additional feature to increase the recognition rate and/or reduce the misidentification rate. Since follicles tend to be spatially located near each other, features derived from the distribution of potential follicle centroids can be incorporated into the classifier.

The segmentation step of this algorithm could be replaced by other segmentation algorithms such as graph cuts [4]. The graph cuts segmentation technique provides better spatial coherence in segmenting follicles with fast implementation, and it is easy to set global and local constraints on segmentation.

REFERENCES

- [1] A Ali, A.K. Abdel-Razek, S Abdel-Ghaffar, and P.S. Glatzel. Ovarian follicular dynamics in buffalo cows (*Bubalus bubalis*). *Reproduction in Domestic Animals*, 38:214 –218, 2003.
- [2] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, 1981.
- [3] R. K. Bock and W. Krischer. *The Data Analysis BriefBook*. Springer-Verlag, 1998.
- [4] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70:109 – 131, 2006.
- [5] V. Caselles, F. Catt, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1 – 31, 1993.
- [6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active cotnours. *International Journal of Computer Vision*, 22:61 – 79, 1997.
- [7] V. Chalana and Y. Kim. A methodology for evaluation of boundary detection algorithms on medical images. *IEEE Transactions on Medical Imaging*, 16:642 – 652, 1997.
- [8] M.Y. Chang, C.H. Chiang, T.T. Hsieh, Y.K. Soong, and K.H. Hsu. Use of the antral follicle count to predict the outcome of assisted reproductive technologies. *Fertility and Sterility*, 69:505 – 510, 1998.
- [9] B. Cigale and D. Zazula. Segmentation of ovarian ultrasound images using cellular neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 18:563 – 581, 2004.
- [10] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211 – 218, 1991.
- [11] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall International, 1982.
- [12] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297 – 302, 1945.
- [13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. A Wiley-Interscience Publication, 2001.
- [14] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.
- [15] D. Freedman and P. Diaconis. On the histogram as a density estimator: l_2 theory. *Springer-Verlag*, 57:453 – 476, 1981.
- [16] R. C. Gonzales, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using Matlab*. Pearson Education, Inc., 2004.
- [17] S. L. Hagen-Ansert. *Textbook of Diagnostic Ultrasonography*. Mosby, Inc, 2001.
- [18] P. E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100 – 107, 1968.

- [19] S. L. Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. In *Proceeding of the 2nd international Joint Conference on Pattern Recognition*, pages 424 – 433, 1974.
- [20] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850 – 863, 1993.
- [21] P. Jaccard. Contribution au problème de l’immigration post-glaciaire de la flore alpine. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 36:87 – 130, 1900.
- [22] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967.
- [23] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321 – 331, 1988.
- [24] J. I. Koo and S. B. Park. Speckle reduction with edge preservation in medical ultrasonic images using a homogeneous region growing mean filter (HRGMF). *Ultrasonic Imaging*, 13:211 – 237, 1991.
- [25] A. Krivanek and M. Sonka. Ovarian ultrasound image analysis: Follicle segmentation. *IEEE Transactions On Medical Imaging*, 17:935 – 944, 1998.
- [26] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
- [27] M. J. Lawrence, M. G. Eramian, R. A. Pierson, and E. Neufeld. Computer assisted detection of polycystic ovary morphology in ultrasound images. In *Proceedings of the 4th Canadian Conference on Computer and Robot Vision*, pages 105–112. IEEE Computer Society, 2007.
- [28] N. S. Macklon and B. C. J. M. Fauser. Regulation of follicle development and novel approaches to ovarian stimulation for IVF. *Human Reproduction Update*, 6:307 – 312, 2000.
- [29] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1:281 – 297, 1967.
- [30] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:158 – 175, 1995.
- [31] R. J. Mapletoft, K. B. Steward, and Gregg P. A. Recent advances in the superovulation in cattle. *Reproduction Nutrition Development*, 42:601 – 611, 2002.
- [32] R. Muzzolini, Y. Yang, and R. Pierson. Classifier design with incomplete knowledge. *Pattern Recognition*, 31:345 – 369, 1998.
- [33] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2000.
- [34] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62 – 66, January 1979.
- [35] B. Potočník and D. Zazula. Automated analysis of a sequence of ovarian ultrasound images. part i: Segmentation of single 2D images. *Image and Vision Computing*, 20:217 – 225, 2002.
- [36] B. Potočník and D. Zazula. Automated analysis of a sequence of ovarian ultrasound images. part ii: Prediction-based object recognition from a sequence of images. *Image and Vision Computing*, 20:227 – 235, 2002.
- [37] B. Potočník, D. Zazula, and D. Korže. Automated computer-assisted detection of follicles in ultrasound images of ovary. *10th IEEE Symposium on Computer-Based Medical Systems (CBMS’97)*, 1997.
- [38] W. K. Pratt. *Digital Image Processing: Paks Inside*. John Wiley & Sons, Inc., 2001.
- [39] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [40] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80(3):227–248, 1989.

- [41] R. Steel, T. L. Poepping, R. S. Thompson, and C. Macaskill. Origins of the edge shadowing artefact in medical ultrasound imaging. *Ultrasound in Medicine and Biology*, 30:1153 – 1162, 2004.
- [42] J. C. Russ. *The Image Processing Handbook*. CRC Press, 1999.
- [43] G. E. Sarty, W. Liang, M. Sonka, and R. A. Pierson. Semiautomated segmentation of ovarian follicular ultrasound images using a knowledge-based algorithm. *Ultrasound in Medicine and Biology*, 24:27 – 42, 1998.
- [44] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2005.
- [45] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, Inc., 2001.
- [46] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks/Cole Publishing Company, 1999.
- [47] H. A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21:65 – 66, 1926.
- [48] J. Szöllösi, G. Ugocsai, and M. Sas. Monitoring of artificial insemination. *International Urology and Nephrology*, 20:403 – 408, 1988.
- [49] B. Viher, A. Dobnikar, and D. Zazula. Cellular automata and follicle recognition problem and possibilities of using cellular automata for image recognition purposes. *International Journal of Medical Informatics*, 49:231 – 241, 1998.
- [50] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:583 – 598, 1991.
- [51] A. Webb. *Statistical Pattern Recognition*. John Wiley & Sons Ind., 2002.
- [52] C. Xu, A. Yezzi Jr., and J. L. Prince. On the relationship between parametric and geometric active contours. In *34th Asilomar Conference on Signals, Systems, and Computers*, pages 483 – 489, 2000.
- [53] C. Xu, D. L. Pham, and J. L. Prince. Medical image segmentation using deformable models. In M. Sonka and J. Beutel, editors, *Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis*, chapter 3, pages 129 – 174. SPIE, 2000.
- [54] C. Xu and J. L. Prince. Gradient vector flow: A new external force for snakes. *Proceedings of Computer Vision and Pattern Recognition (CVPR'97)*, pages 66 – 71, 1997.
- [55] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions On Image Processing*, 7(3):359 – 369, March 1998.
- [56] Y. Yang and G. I. Webb. A comparative study of discretization methods for naive-Bayes classifiers. In T. Yamaguchi, A. Hoffmann, H. Motoda, and P. Compton, editors, *Proceedings of the 2002 Pacific Rim Knowledge Acquisition Workshop (PKAW'02)*, pages 159–173. Japanese Society for Artificial Intelligence, 2002.
- [57] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16:199 – 209, 1997.
- [58] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335 – 1346, 1996.

APPENDIX A

SAMPLE SEGMENTATION RESULTS

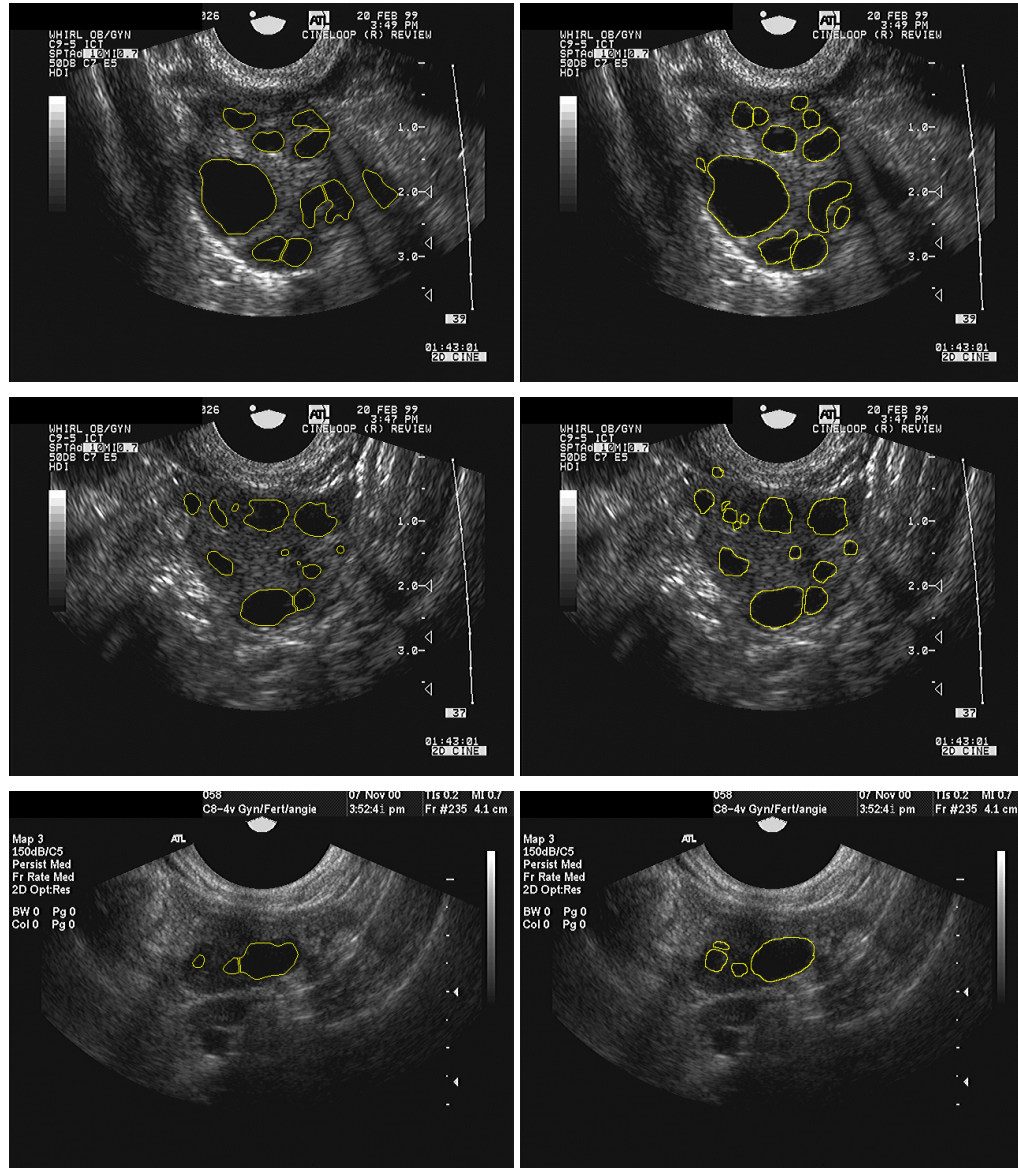


Figure A.1: Sample segmentation results - typical good results. Images on the left hand side are from the proposed automatic segmentation algorithm. Images on the right hand side are from an expert's segmentation.

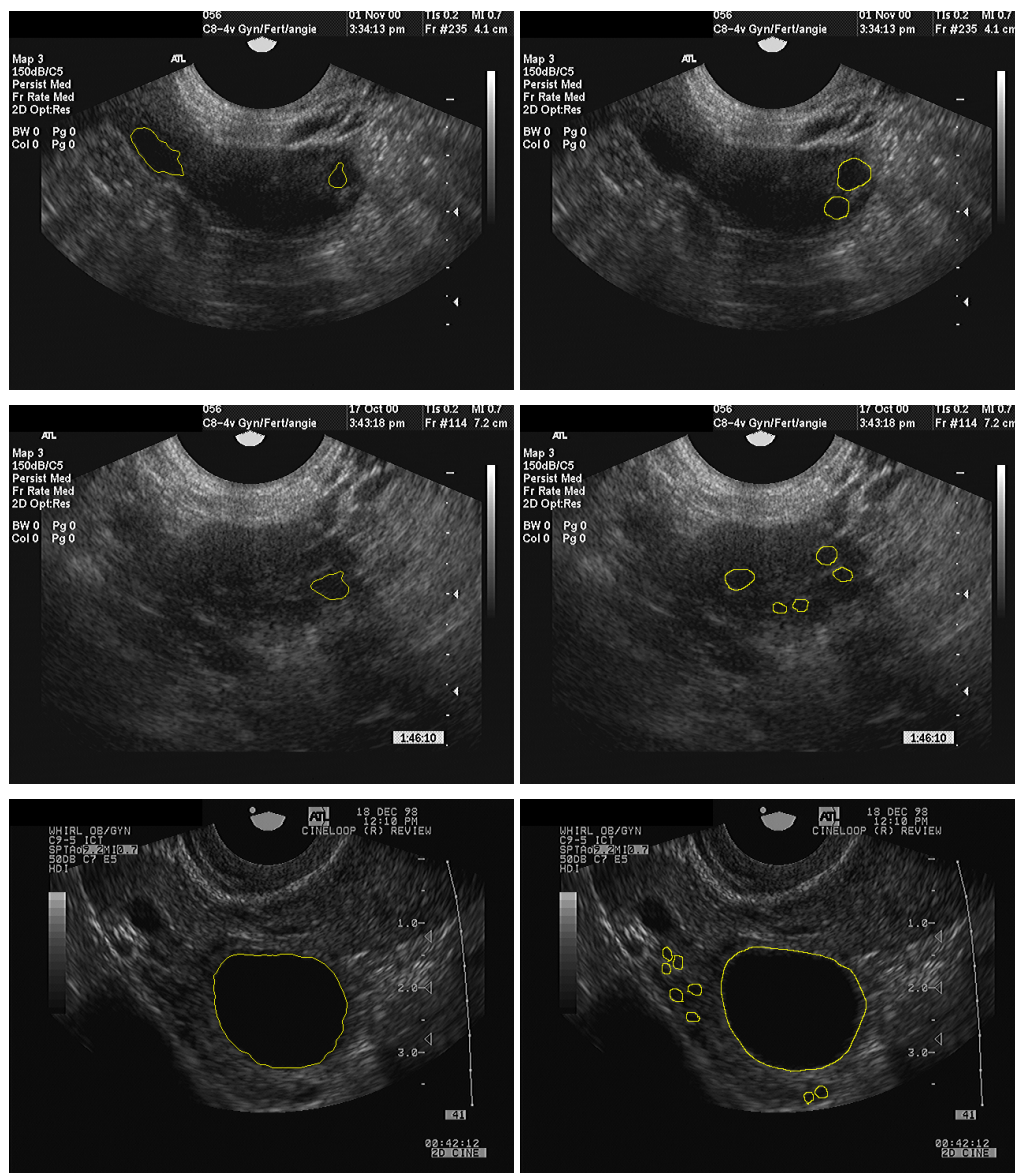


Figure A.2: Sample segmentation results - typical poor results. Images on the left hand side are from the proposed automatic segmentation algorithm. Images on the right hand side are from an expert's segmentation.

APPENDIX B

EXPERIMENT RESULTS

A1 - A5 are sets of images with automatically segmented regions.

- A1: regions obtained by segmentation and classification using training set 1
- A2: regions obtained by segmentation and classification using training set 2
- A3: regions obtained by segmentation and classification using training set 3
- A4: regions obtained by segmentation and classification using training set 4
- A5: regions obtained by segmentation, but before classification

G1 - G6 are processed the gold standard image sets (filtered for follicle size).

- G1: the original gold standard
- G2: gold standard containing only follicles with diameters greater than or equal to 2 *mm*
- G3: gold standard containing only follicles with diameters greater than or equal to 3 *mm*
- G4: gold standard containing only follicles with diameters greater than or equal to 4 *mm*
- G5: gold standard containing only follicles with diameters greater than or equal to 5 *mm*
- G6: gold standard containing only follicles with diameters greater than or equal to 6 *mm*

Table B.1: Evaluation for automatic segmentation result and gold standard: A5 and G1 - G5.

	IRR	IMR	FRR	FMR	HD (<i>mm</i>)	MAD (<i>mm</i>)	RMSD (<i>mm</i>)
A5 & G1	43% \pm 26%	67% \pm 20%	40%	68%	2.43 \pm 1.46	0.75 \pm 0.32	0.86 \pm 0.49
A5 & G2	54% \pm 27%	67% \pm 21%	52%	69%	2.37 \pm 1.51	0.69 \pm 0.32	0.82 \pm 0.53
A5 & G3	71% \pm 31%	69% \pm 21%	70%	71%	2.44 \pm 1.55	0.70 \pm 0.33	0.83 \pm 0.54
A5 & G4	77% \pm 33%	74% \pm 19%	77%	75%	2.56 \pm 1.63	0.70 \pm 0.35	0.87 \pm 0.58
A5 & G5	85% \pm 28%	75% \pm 18%	86%	78%	2.64 \pm 1.67	0.74 \pm 0.36	0.89 \pm 0.58
A5 & G6	88% \pm 28%	76% \pm 18%	90%	82%	2.80 \pm 1.74	0.78 \pm 0.38	0.95 \pm 0.60

Table B.2: Evaluation for automatic segmentation result and gold standard: A1 and G1 - G5.

	IRR	IMR	FRR	FMR	HD (<i>mm</i>)	MAD (<i>mm</i>)	RMSD (<i>mm</i>)
A1 & G1	36% \pm 26%	28% \pm 22%	33%	27%	2.29 \pm 1.36	0.74 \pm 0.34	0.82 \pm 0.46
A1 & G2	46% \pm 29%	27% \pm 22%	44%	26%	2.24 \pm 1.34	0.68 \pm 0.33	0.79 \pm 0.48
A1 & G3	61% \pm 33%	27% \pm 22%	60%	26%	2.25 \pm 1.37	0.68 \pm 0.34	0.78 \pm 0.49
A1 & G4	69% \pm 37%	30% \pm 23%	68%	28%	2.38 \pm 1.42	0.68 \pm 0.35	0.82 \pm 0.51
A1 & G5	76% \pm 34%	32% \pm 24%	76%	30%	2.43 \pm 1.45	0.69 \pm 0.35	0.83 \pm 0.51
A1 & G6	77% \pm 35%	36% \pm 25%	79%	33%	2.56 \pm 1.51	0.73 \pm 0.38	0.89 \pm 0.53

Table B.3: Evaluation for automatic segmentation result and gold standard: A2 and G1 - G5.

	IRR	IMR	FRR	FMR	HD (<i>mm</i>)	MAD (<i>mm</i>)	RMSD (<i>mm</i>)
A2 & G1	39% \pm 28%	31% \pm 24%	35%	30%	2.33 \pm 1.30	0.73 \pm 0.28	0.82 \pm 0.42
A2 & G2	49% \pm 31%	32% \pm 24%	47%	30%	2.25 \pm 1.29	0.67 \pm 0.29	0.77 \pm 0.45
A2 & G3	63% \pm 35%	32% \pm 24%	63%	31%	2.28 \pm 1.32	0.67 \pm 0.29	0.78 \pm 0.46
A2 & G4	70% \pm 38%	36% \pm 25%	71%	34%	2.36 \pm 1.38	0.66 \pm 0.32	0.80 \pm 0.49
A2 & G5	77% \pm 36%	37% \pm 24%	79%	37%	2.41 \pm 1.40	0.70 \pm 0.32	0.81 \pm 0.47
A2 & G6	78% \pm 37%	41% \pm 23%	82%	40%	2.53 \pm 1.45	0.73 \pm 0.36	0.86 \pm 0.49

Table B.4: Evaluation for automatic segmentation result and gold standard: A3 and G1 - G5.

	IRR	IMR	FRR	FMR	HD (<i>mm</i>)	MAD (<i>mm</i>)	RMSD (<i>mm</i>)
A3 & G1	37% \pm 25%	25% \pm 20%	35%	23%	2.41 \pm 1.43	0.77 \pm 0.36	0.84 \pm 0.44
A3 & G2	47% \pm 28%	25% \pm 20%	45%	23%	2.33 \pm 1.42	0.71 \pm 0.35	0.79 \pm 0.45
A3 & G3	62% \pm 33%	26% \pm 21%	62%	24%	2.36 \pm 1.45	0.70 \pm 0.36	0.80 \pm 0.46
A3 & G4	69% \pm 36%	30% \pm 22%	68%	27%	2.48 \pm 1.51	0.70 \pm 0.38	0.83 \pm 0.49
A3 & G5	77% \pm 33%	30% \pm 22%	78%	29%	2.53 \pm 1.54	0.71 \pm 0.37	0.83 \pm 0.47
A3 & G6	81% \pm 32%	33% \pm 21%	82%	31%	2.68 \pm 1.59	0.75 \pm 0.38	0.89 \pm 0.48

Table B.5: Evaluation for automatic segmentation result and gold standard: A4 and G1 - G5.

	IRR	IMR	FRR	FMR	HD (<i>mm</i>)	MAD (<i>mm</i>)	RMSD (<i>mm</i>)
A4 & G1	40% \pm 28%	34% \pm 23%	35%	33%	2.36 \pm 1.32	0.76 \pm 0.28	0.85 \pm 0.45
A4 & G2	50% \pm 30%	34% \pm 23%	47%	33%	2.28 \pm 1.31	0.69 \pm 0.28	0.80 \pm 0.46
A4 & G3	63% \pm 34%	35% \pm 24%	63%	34%	2.31 \pm 1.35	0.69 \pm 0.29	0.81 \pm 0.48
A4 & G4	70% \pm 36%	39% \pm 23%	70%	38%	2.43 \pm 1.40	0.69 \pm 0.32	0.84 \pm 0.50
A4 & G5	76% \pm 34%	40% \pm 24%	76%	41%	2.50 \pm 1.44	0.73 \pm 0.32	0.85 \pm 0.49
A4 & G6	77% \pm 37%	46% \pm 23%	79%	44%	2.65 \pm 1.48	0.77 \pm 0.35	0.90 \pm 0.51