A NEW SEGMENTATION ALGORITHM FOR PROSTATE BOUNDARY DETECTION IN 2D ULTRASOUND IMAGES

By

Bernard Chiu

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2003

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Prostate segmentation is a required step in determining the volume of a prostate, which is very important in the diagnosis and the treatment of prostate cancer. In the past, radiologists manually segment the two-dimensional cross-sectional ultrasound images. Typically, it is necessary for them to outline at least a hundred of cross-sectional images in order to get an accurate estimate of the prostate's volume. This approach is very time-consuming. To be more efficient in accomplishing this task, an automated procedure has to be developed. However, because of the quality of the ultrasound image, it is very difficult to develop a computerized method for defining boundary of an object in an ultrasound image.

The goal of this thesis is to find an automated segmentation algorithm for detecting the boundary of the prostate in ultrasound images. As the first step in this endeavour, a semi-automatic segmentation method is designed. This method is only semi-automatic because it requires the user to enter four initialization points, which are the data required in defining the initial contour. The discrete dynamic contour (DDC) algorithm is then used to automatically update the contour. The DDC model is made up of a set of connected vertices. When provided with an energy field that describes the features of the ultrasound image, the model automatically adjusts the vertices of the contour to attain a maximum energy. In the proposed algorithm, Mallat's dyadic wavelet transform is used to determine the energy field. Using the dyadic wavelet transform, approximate coefficients and detailed coefficients at different scales can be generated. In particular, the two sets of detailed coefficients represent the gradient of the smoothed ultrasound image. Since the gradient modulus is high at the locations where edge features appear, it is assigned to be the energy field used to drive the DDC model.

The ultimate goal of this work is to develop a fully-automatic segmentation algorithm. Since only the initialization stage requires human supervision in the proposed semi-automatic initialization algorithm, the task of developing a fully-automatic segmentation algorithm is reduced to designing a fully-automatic initialization process. Such a process is introduced in this thesis.

In this work, the contours defined by the semi-automatic and the fully-automatic segmentation algorithm are compared with the boundary outlined by an expert observer. Tested using 8 sample images, the mean absolute difference between the semi-automatically defined and the manually outlined boundary is less than 2.5 pixels, and that between the fully-automatically defined and the manually outlined boundary is less than 4 pixels. Automated segmentation tools that achieve this level of accuracy would be very useful in assisting radiologists to accomplish the task of segmenting prostate boundary much more efficiently.

Acknowledgements

Contents

List of Tables

List of Figures

xi

Chapter 1

Introduction

Prostate cancer is the second most common cancer in North American men (skin cancer is the most common cancer). It is estimated that in 2002, approximately 189,000 new cases and 30,200 prostate cancer-related deaths will occur in the United States [12]. Prostate cancer is the second leading cause of cancer death in men, exceeded only by lung cancer. It accounts for 29% of all male cancers and 11% of male cancer-related deaths [19]. The disease is found in as many as 34% of men in their 50's and in up to 70% of men 80 years of age and older [20].

Early prostate cancer often does not cause any symptoms. However, if left untreated, metastasis – the spread of the cancerous cells from the prostate to nearby lymph nodes or other organs – can occur, at which point treatment may become less effective, whereas complete recovery is possible if the disease is detected and treated when it is confined to the prostate. Therefore, periodic screening of prostate cancer is important, especially for men over 50 years of age. The prostate-specific antigen (PSA) test is one of the most popular screening tests. Using the most common type of PSA test currently available in the USA, a normal, healthy, 50-year-old male is generally believed to have a PSA of less than 4.0 nanograms per milliliter of blood (4.0 ng/ml). However, since prostate cancer is not the only reason for a person to have a higher PSA value, the specificity of the PSA test is undesirably low. This low specificity results in unnecessary treatment applied on subjects who do not carry the disease. To improve the accuracy of the PSA test, doctors have developed schemes that incorporate the readings of this test with other information. One of such tests is the PSA density (PSAD) test. PSAD is a measure of the concentration of PSA in a man's prostate. It measures the PSA value per cubic centimetre of the prostate (ng/ml/cc). To obtain the PSAD value, the information about the volume of a prostate is essential.

The usefulness of the information on the prostate's volume is not restricted to the establishment of the PSAD value. The volume of a prostate, by itself, is useful in the diagnosis of prostate cancer. Also, doctors need the volume of the prostate in assigning the appropriate dose in various treatments. Segmenting the prostate boundaries in two-dimensional cross-sectional ultrasound images is the first step in determining the volume of a prostate. This task is usually accomplished manually by the radiologists. Typically, it is necessary for them to outline at least a hundred of cross-sectional images in order to get an accurate estimate of the volume of a prostate. This approach is very time-consuming. To be more efficient in accomplishing this task, an automated segmentation procedure has to be developed. This is the goal of this thesis.

Image segmentation is a well-known problem in computer vision, and there are many well-established techniques for solving this problem, some of which are discussed in Section 3.1. Although these algorithms have achieved some level of success, no

algorithm, when used alone, is suitable in detecting boundary in an ultrasound image because of the noisy and fuzzy nature associated to this type of image. For example, edge-based techniques are not applicable because of the amount of "false edges" – edges that come from different kinds of noise or imaging artifacts – in an ultrasound image. Also, the boundaries detected by edge-based techniques are generally not closed, which poses a problem when the closed prostate boundary is needed to be found. Although a model-based technique can be used to find a closed contour, the boundary determined by this method is very sensitive to the initial contour, which is difficult to be accurately defined in an ultrasound image.

In the first part of the thesis, three elements are combined in determining the accurate boundary: human observation, the discrete dynamic contour (DDC) model and the dyadic wavelet transform. In the proposed algorithm, an expert observer is required to enter four initialization points, which – in his judgement – lie on the prostate boundary. The number of initial point is chosen to be four because the performance of the proposed algorithm is planned to be compared fairly with the UWO algorithm, which will be introduced in Chapter 3. This segmentation algorithm uses four initialization points, and is considered to be the benchmark in the first part of the thesis. An initial contour is then defined by interpolating these four points. This contour is served as the initial contour of the DDC model. The DDC model is made up of a set of connected vertices. When provided with an energy field that describes the features of the ultrasound image, the model automatically adjusts the vertices of the contour to attain a maximum energy. The dyadic wavelet transform is an edge-based technique. Its role in the algorithm is to compute the gradient modulus of an image, which serves as an energy field to drive the DDC model. In other words, the DDC model is attracted to strong edges, where the gradient modulus is high. It is expected that if the initial contour is not very far from the actual prostate boundary, the final contour resulting from the DDC algorithm should approximate the boundary accurately. The end product of this phase is the semi-automatic part of proposed algorithm described in Section 5.1.

The semi-automatic algorithm reduces the radiologist's task in segmenting hundreds of prostate ultrasound image to picking four initial points on each image. Although the time required to finish the task is significantly reduced, it is still cumbersome for him to pick at least four hundred points in obtaining the prostate's volume, and therefore, a fully-automatic segmentation algorithm would be very attractive to him.

The advantage of a fully-automatic algorithm is not limited to increasing the efficiency of the segmentation process. As mentioned, the quality of the ultrasound images is usually poor. Therefore, a certain level of expertise is required to outline the prostate boundary in an ultrasound image. This level of expertise is not readily available in the health care community. If an automatic segmentation tool is available, this level of expertise is not crucial in identifying the prostate boundary. Also, if the automatic segmentation tool detects the prostate boundary accurately, it can be used to train the non-professional health care practitioner, and therefore, alleviate the shortage of the skill required for prostate segmentation.

The significance of a fully-automatic algorithm motivates us to work on its development, which constituents the second phase of this thesis. Since, in the framework of the proposed semi-automatic algorithm, only the initialization stage requires human supervision, the task of developing a fully-automatic segmentation algorithm is reduced to designing a fully-automatic initialization process. The proposed initialization process will be described in Section 4.4.

The accuracy of the semi-automatically and the fully-automatically defined boundaries are evaluated by the deviations between these boundaries and the professionally outlined boundary. The results will be presented in Section 5.3. Tested using 8 randomly chosen images, the mean absolute difference between the semi-automatically generated and the manually outlined boundary is about 2.5 pixels, and that between the automatically generated and the manually outlined boundary is less than 4 pixels.

This thesis is organized as follows: In Chapter 2, the most common techniques used in screening of prostate cancer – digital rectal examination (DRE), PSA test and imaging tests – are introduced. A brief introduction of two most popular medical imaging techniques, ultrasound and MRI, will also be given in this chapter. A survey of existing image segmentation methods is given in Chapter 3. Two existing algorithms that serve as the building blocks of the proposed algorithm, the dyadic wavelet transform and the DDC model, are also described in this chapter. In Chapter 4, we correct an error made by Mallat in the dyadic wavelet transform algorithm he proposed [1], introduce the possibility of choosing different wavelet functions in performing the dyadic wavelet transform, and introduce two building blocks that are essential in the proposed segmentation algorithm: an energy field that is essential for the semi-automatic algorithm and the fully-automatic initialization algorithm. The complete segmentation algorithm is constructed and the boundaries obtained using this algorithm are evaluated in Chapter 5.

Chapter 2

Techniques in Diagnosing Prostate Cancer

Digital Rectal Examination (DRE), Prostate-Specific Antigen (PSA) test and Transrectal Ultrasound (TRUS) are three most commonly used techniques in screening and diagnosing prostate cancer. A brief description of each technique is provided in the following:

a) Digital Rectal Examination (DRE)

A digital rectal examination (DRE) is performed by a doctor during a regular office visit. In this examination, the doctor inserts a gloved finger into the rectum and feels the prostate gland through the rectal wall to check for abnormality.

Rectal examination is inexpensive and relatively non-invasive. Also, it can be taught to non-professional health workers. However, its effectiveness depends on the skill and experience of the examiner. The effectiveness of DRE in diagnosing prostate cancer is questionable. Contradictory results have been obtained by several studies in this area (such as [14], [15] and [16]).

b) Prostate-Specific Antigen (PSA) Tests

The PSA test has revolutionized the detection of prostate cancer since its introduction in the mid 1980s. As mentioned in Chapter 1, a normal, healthy, 50-year-old male is generally believed to have a PSA value of less than 4.0 ng/ml. Over the years, physicians have developed schemes that incorporate the readings of this test with other information for a more accurate diagnosis of the prostate cancer. Examples of these schemes are free/total PSA (PSA II) test, PSA velocity (PSAV) test and PSA density (PSAD) test.

The PSA II or free/total PSA test is a new type of PSA test that can be used to help the physician discriminate between patients with relatively low standard PSA levels (say 2.5-10.0 ng/ml) who are at greater risk of having prostate cancer (and therefore need a prostate biopsy), and those patients who are more likely to have benign prostatic hyperplasia (BPH) or non-cancerous prostate enlargement. The PSA II test measures the amount of PSA that is free in the blood stream, and compares it to the total free and bound PSA found in the blood. The lower the ratio of free to total PSA, the higher the likelihood that the patient has prostate cancer.

The PSAV test measures the rate of change of an individual's PSA value. Some physicians believe that result of PSAV is related to the prostate cancer development in individual patients. PSAV is measured in nanograms per milliliter of blood per year (ng/ml/yr).

PSAD is a measure of the concentration of PSA in a man's prostate. It measures the PSA value per cubic centimetre of the prostate (ng/ml/cc). Prostate cancer is likely to be present in an individual having high PSAD value. If the PSAD value is greater than 0.15 ng/ml/cc, biopsy, a procedure in which doctor takes small samples of tissue from the prostate in order to make proper diagnosis, is highly recommended [17].

c)   Transrectal Ultrasound (TRUS) and Other Imaging Tests

Prostatic imaging is possible by ultrasound or magnetic resonance imaging, which will be introduced in the next two sections. Out of all the imaging techniques, ultrasound is most commonly used. There are two common uses of ultrasound imaging: The first is to guide the doctor in the biopsy procedure. The second use is in the establishment of the prostate's volume.

2.1.   Ultrasound

Ultrasound imaging has been used in medical diagnostics for more than 50 years. Because of its safe and non-invasive nature, ultrasound imaging today takes an outstanding position among modern imaging techniques. In this section, a brief introduction of the operating principles of ultrasound imaging is provided.[1]

2.1.1.   Ultrasound Physics

Human can hear sound waves or mechanical waves in the frequency range of 20Hz to 20kHz. Ultrasound waves are high-frequency mechanical waves human cannot hear, that is, waves having frequency greater than 20kHz. A wave is generated by a force acting on the molecules of a material, called the medium for the wave. Through interaction of neighbouring molecules, energy is passed from one location to another.

Waves are classified into two basic types: longitudinal and transverse. For longitudinal waves, the particles move along the direction of the wave energy propagation, while for transverse waves, the motion of particles is perpendicular to the direction of wave propagation. Ultrasound waves are longitudinal.

In diagnostic ultrasound, an ultrasound wave is generated by a transducer and directed into the body to interact with tissues. After these interactions, the wave is received by the transducer. The characteristic of the received waves depends on the nature of tissues and provides important information for diagnosis. Main types of interactions that occur are reflection, refraction, scattering and diffraction, interference and absorption. The following sections introduce these phenomena.

---

[1] *This introduction is based on [11].*

A) Reflection and Acoustic Impedance

The most important phenomenon in ultrasound imaging is reflection. When a sound beam is directed to an interface, it will be partially reflected back towards the sound source. Based on the amount of energy reflected, the nature of the tissue inside the body can be characterized. Acoustic impedance is a measure of the resistance to sound passing through the medium. It is defined by

$$Z = \rho v$$

(2.1)

where v denotes the velocity of sound in the medium, measured in *m/s*, ρ denotes the density of the medium, measured in *kg/m³*, and Z denotes the acoustic impedance, measured in $kg / m^2 \cdot s$.

The fraction of energy being reflected at the interface of two media depends on the difference between acoustic impedances of the two media. One is able to characterize the tissue structure with an ultrasonic beam because of the difference in acoustic impedances at a biological interface (see Figure 1). The fraction of sound energy reflected at an interface, or the reflection coefficient of an interface is expressed as

$$\alpha_R = \left(\frac{Z_2 - Z_1}{Z_2 + Z_2} \right)^2$$

(2.2)

where Z_1 denotes the acoustic impedance of medium #1, and Z_2 denotes the acoustic impedance of medium #2.

B) Refraction

If the ultrasound beam strikes an interface between two media at an angle of 90°, a fraction of the beam is reflected back to the first medium and the rest is transmitted into the second medium at the same angle. If, however, the beam strikes the interface at an angle other than 90°, the transmitted beam is refracted from the straight-line path.
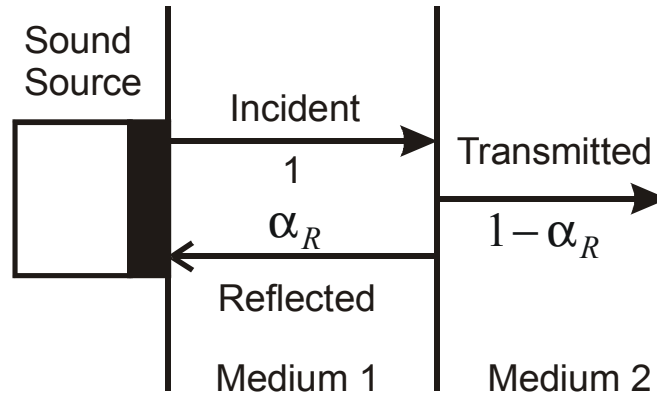
Figure 1   Reflection caused by a sound wave striking an interface at normal incidence
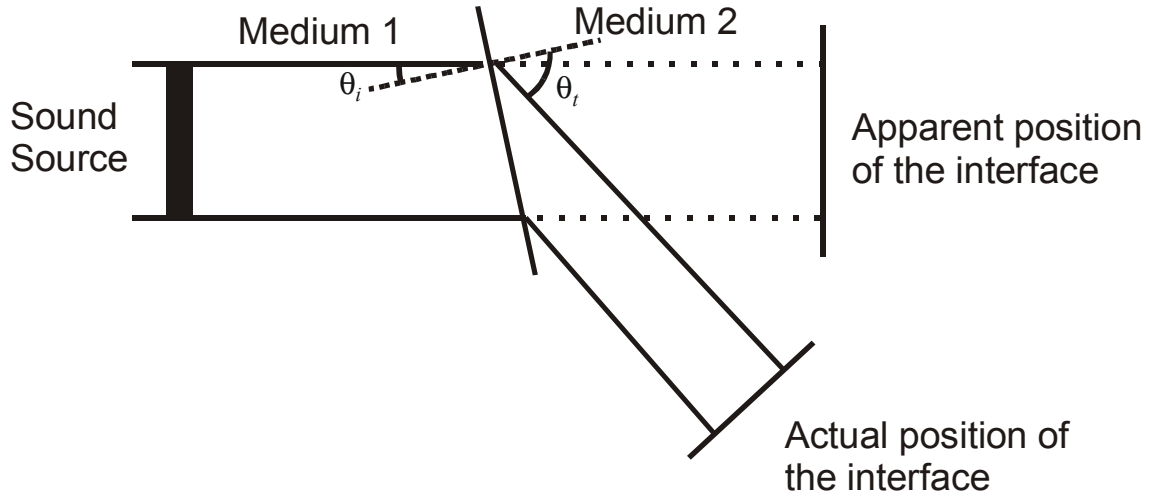
Figure 2   Refraction causes the interface to be displayed at the wrong position (Adapted from [11])

The change in the angle of transmission is related to the velocities of sound in the two media of an interface. Refraction of sound waves obeys Snell's law, which is given by

$$\frac{\sin \phi_i}{\sin \phi_t} = \frac{v_i}{v_t}$$

(2.3)

where ϕ_i is the incident angle, ϕ_t is the transmitted angle, v_i is the velocity of sound in the incident medium, and v_t is the velocity of sound in the transmitted medium.

Refraction causes some artifacts in diagnostic images. For example, in Figure 2, an object appears as if it were at the centre of the beam, whereas it is located elsewhere.

C) Scattering and Diffraction

Scattering occurs when the interfaces are smaller than the diameter of the ultrasound beam (see Figure 3). Since this phenomenon has strong frequency dependence, it makes ultrasound useful in characterizing tissue. When the frequency of the ultrasound beam changes, the same material may scatter the sound beam differently. Therefore, this effect gives important information for diagnosis.

As the ultrasound beam moves further from the source, it spreads out. This phenomenon is called diffraction (see Figure 4(a)). This divergence increases as the diameter of the sound source decreases. Diffraction also occurs after the beam passes through a small aperture on the order of 1 wavelength. This effect is shown in Figure 4(b).

D) Interference

Interference is caused by superposition or algebraic summation of waves. For two waves of the same frequency, constructive interference occurs if they are in phase, which means that the maximum and minimum of two waves occur at the same time. The amplitude of the resulting wave is magnified in constructive interference (see Figure 5(a)). Complete destructive interference occurs when two waves of same frequency are out of phase (i.e., maximum corresponds to minimum and vice versa). The resulting wave has zero amplitude (see Figure 5(b)).
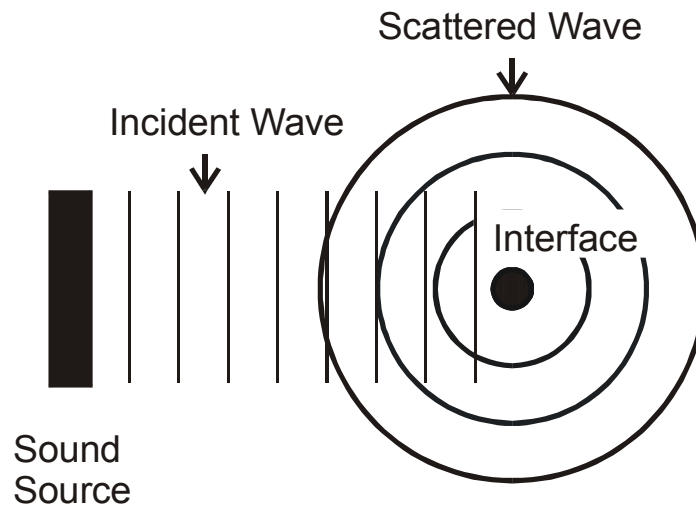
Figure 3   The diameter of the sound beam is larger than the interface. The scattered wave emits in all directions. (Adapted from [11])
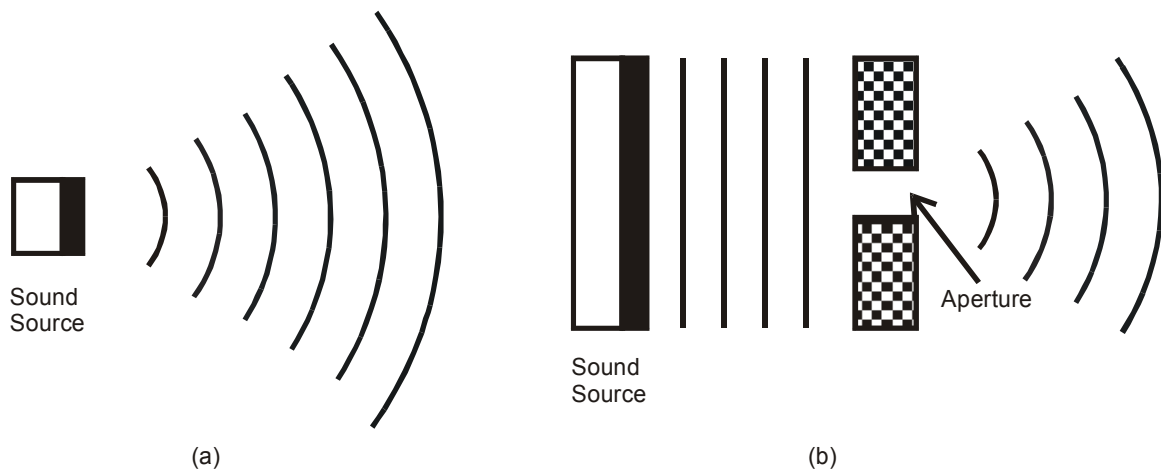
Figure 4   Principles of Diffraction. (a) Divergence of the beam from the sound source; (b) Diffraction from small aperture (Adapted from [11])
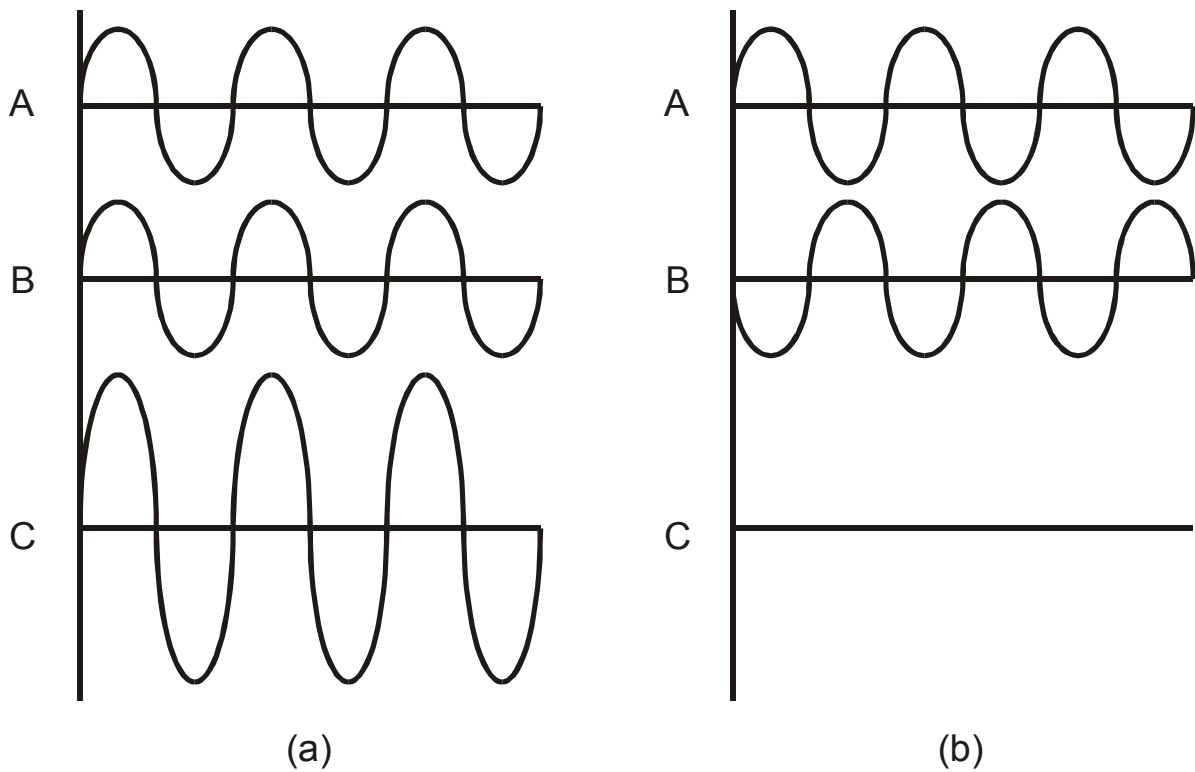
Figure 5   (a) Constructive Interference; (b) Destructive Interference. Wave C is the resultant of Wave A and B.

E) Absorption

Absorption is the process in which sound energy is converted to another form of energy, primarily heat.

The rate of absorption is related to the viscosity of the medium, the relaxation time of the medium and the frequency of the ultrasonic beam.

The viscosity describes the resistance of relative motion between the molecules in a medium. If the viscosity of a medium is high, more energy is required to overcome the frictional forces between molecules, thus more energy is dissipated as heat.

The relaxation time is the average time required for a molecule to return to its original position after a disturbance. If the relaxation time of a media is short, molecules disturbed by a wavefront would have returned to the original position when the next wavefront arrives; otherwise, the molecules may be moving back towards its original position when the next wavefront comes. In this case, extra energy is required to reverse the direction of the molecule and more energy is dissipated.

In the same medium, the rate of absorption for a high-frequency wave is higher compared to that for a low-frequency wave. If the frequency of the wave increases, the molecule needs to move more often, and therefore, heat energy is dissipated in a higher rate. Also, the higher the frequency of the wave, the shorter interval of time is available for the molecule to return to its original position. More energy is required to redirect the molecules, and therefore, the rate of energy absorption is higher.

It is empirically observed that the absorption of the ultrasonic wave follows an exponential relationship:

$$A = A_o \exp(-\alpha x)$$

(2.4)

where A is the amplitude of the wave at a distance x, A_o is the original amplitude of the wave, α is the amplitude absorption coefficient, and x is the distance travelled by the wave.

2.1.2. Ultrasound Imaging Modes

Before discussing different modes of ultrasound imaging, an important mechanism called echo ranging is described.

Echo ranging relates the depth of a biological interface with the time at which the corresponding reflected signal is received. An ultrasonic wave is transmitted through the

body tissue. Because of the mismatch of the acoustic impedances at a biological interface, the ultrasound beam is partially reflected and received by the transducer. The reflections at different biological interfaces are received at a certain time, *t,* after the ultrasonic wave is transmitted. Knowing the velocity of sound wave propagation in the biological medium, *v,* the depth of a biological interface, *d,* can be calculated using the following formula:

$$d = vt$$

<div align="right">(2.5)</div>

Echo ranging is the fundamental operating principle of each of the imaging modes that will be discussed, except the transmission mode.

A) A-Mode

A-Mode scanning refers to amplitude-mode scanning. The operation of this scanning mode is primarily based on echo ranging. The term amplitude refers to the energy of the echo received by the transducer. A-mode scanning gives a portrait of the biological tissues in one dimension. In a typical A-mode scan, the magnitude of the reflected ultrasonic energy is represented in the Y axis and the depth is represented in X axis (see Figure 6(c)).
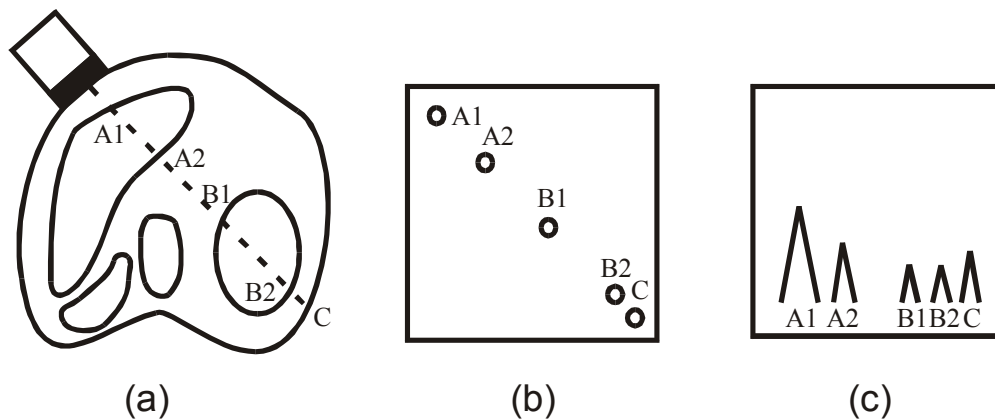
Figure 6   Comparison between A-Mode and B-Mode scan. (a) The transducer emits an ultrasound wave in a given direction; (b) B-Mode display of the interface detected; (c) A-Mode display of the interface detected (Adapted from [11])
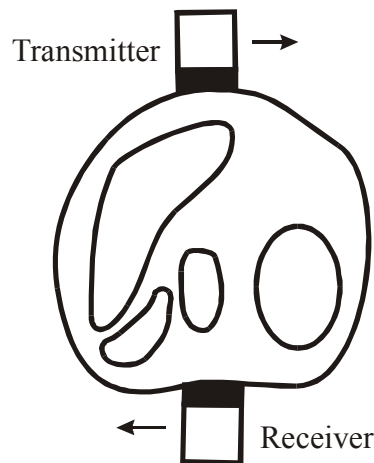
Figure 7   The Transmission Mode scanning system (Adapted from [11])

B) B-Mode

B-Mode scan can be described as the bird's-eye view of the A-Mode scan, but the amplitude of the reflected signal in an A-Mode plot is represented by the brightness of a dot in a B-Mode scan (see Figure 6(b)).

One can perform B-Mode scanning in many different directions to obtain a two-dimensional portrait of the internal structure of the body, which can then be stored as a grey-scale image. To determine the exact location of different biological interfaces detected, it is necessary to keep track of the accurate position where the ultrasonic wave emits. A computerized arm is used for this purpose.

C) Real-Time Mode

Real-time scanning can be described as continuous time B-Mode scanning. In modern real-time scanner, up to 120 stationary B-Mode scans can be produced per second to give an impression of motion. Real-time scanning is widely used in obstetrics to assess the health and development of the fetus.

D) Transmission Mode

Up to this point, all ultrasonic scanning systems described rely on the reflected echo to detect the biological structure of the body. In transmission mode scanning, the receiver is 180° apart from the transmitter of the ultrasonic wave (see Figure 7). Thus, this scanning method captures information based on the transmission, rather than the reflection, of the ultrasound wave through the patient's body. Transmission Mode scanning is used extensively in breast scanning. The measurements of the attenuation and the velocity of

the sound waves are easily obtained using this imaging mode. These data provides vital information in diagnosing tumour.

2.1.3. Instrumentation of A-Mode Scanner

In our project, the prostate images are obtained using a B-Mode scanner. Therefore, we are primarily focus on the operating principles of a B-Mode scanner. However, since the B-Mode scanner is basically an enhanced version of the A-Mode scanner, the discussion should start from introducing the operating principles of a basic A-Mode scanner. Figure 8 illustrates the block diagram of an A-Mode scanner.

The A-Mode scanner consists of four units:

1) Transducer – This unit contains a transmitter and a receiver. The transducer is responsible for the generation of the ultrasonic signal and the reception of echoes reflected from different biological interfaces. The operating principles of the transducer and some design consideration on the ultrasonic signal transmitted are presented in part A, B and C of this section.

2) Electronic processing unit – This unit enhances the received signal for displaying purposes. The mechanism for processing signals will be introduced in part D of this section.

3) Display – The enhanced signal are displayed using a cathode ray tube. A brief introduction on how the display unit works is provided in part E of this section.

4) Master Synchronizer – The master synchronizer starts the scanning process by sending a pulse signal to the transducer, the electronic processing unit and the display unit at the same time. Upon reception of the pulse signal, the transmitter emits an ultrasonic signal into the body. The same pulse signal is send to the receiver because the elapsed time from the emission of the ultrasonic signal by the transmitter to the reception of the ultrasonic echo must be known. The electronic processor and the display unit also need to know the point at which the ultrasonic signal is emitted in order for them to display and process the received echo correctly.
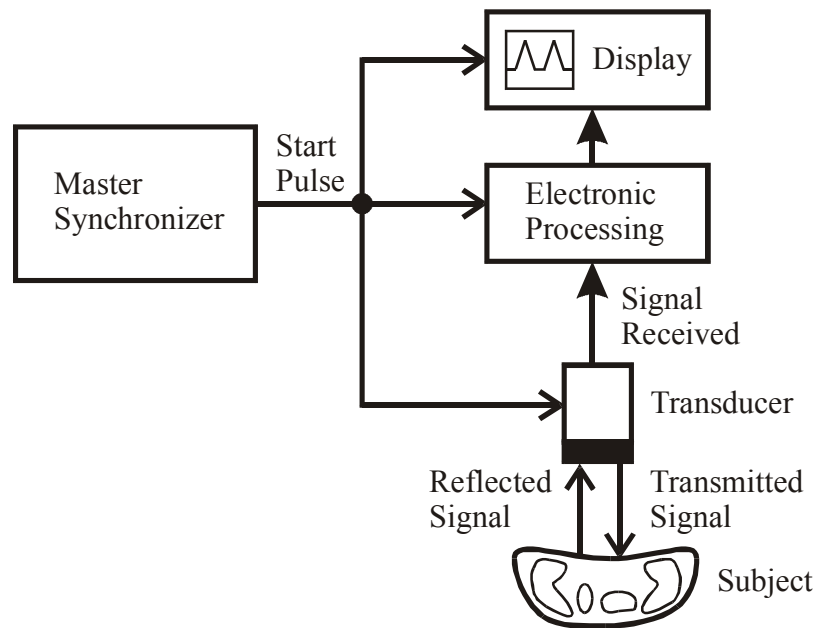
Figure 8   Block Diagram of an A-Mode scanner (Adapted from [11])

A) Transducer Construction

The basic operating principle of the transducer is a phenomenon called the piezoelectric effect, discovered by Pierre and Marie Curie in 1880. This effect is found in crystalline structures that consist of electric dipoles (i.e., molecules that have net positive charge on one end and negative charge on the opposite end).

In a normal crystalline lattice structure, the electric dipoles are arranged randomly (see Figure 9(a)). When heated above a temperature called Curie temperature, the bipolar molecules are free to move. If one applies an electric field (by placing a pair of oppositely charged plates) across the crystal, the bipolar molecule will be aligned towards a direction that is close to the direction of the electric field (see Figure 9(b)). This molecular structure is maintained after the crystal is cooled below the Curie temperature. At this point, the molecular structure unique to the piezoelectric material is constructed.

Then, a sinosoidally alternating electric field perpendicular to that used to form the piezoelectric property is applied to the crystal (see Figure 9(c)). As the bipolar molecules flip back and forth to align with the sinosoidally alternating electric field, the crystal are contracted and expanded periodically (see Figure 10). This periodical motion creates mechanical vibrations of air molecules and therefore produces ultrasonic waves. This process converts electrical energy to mechanical energy and is called the converse piezoelectric effect (or the electric-pressure effect).

On the other hand, the piezoelectric effect (or pressure-electric effect) enables the receiver to detect the ultrasonic echo. The echo reflecting from biological interfaces creates a mechanical vibration of the crystal and electric signals are generated.

The structure of a simple transducer is shown in Figure 11. It basically consists of three major components: the backing material, the crystal and the matching layer. The uses of the crystal are explained in detail previously. In the following, we focus on understanding the functions of the backing material and the matching layer.

In ultrasound imaging, the pulse is sent out in a short time interval. In the remaining time of the cycle, the transducer receives the ultrasonic echo without sending out any signal. The backing material serves two purposes. Firstly, it is used to dampen the transmitted pulse so that it dies away quickly. Secondly, it absorbs all the energy produced by the crystal in the "off" time of the transmitter. For ideal absorption, the backing material should be chosen such that it has the same acoustic impedance as the crystal.

The purpose of the matching layer is to match the acoustic impedance of the crystal and that of the tissue so that the energy reflected at the crystal-tissue interface is minimized, or in other words, most of the energy is transferred into the body.

(a)   (b)   (c)
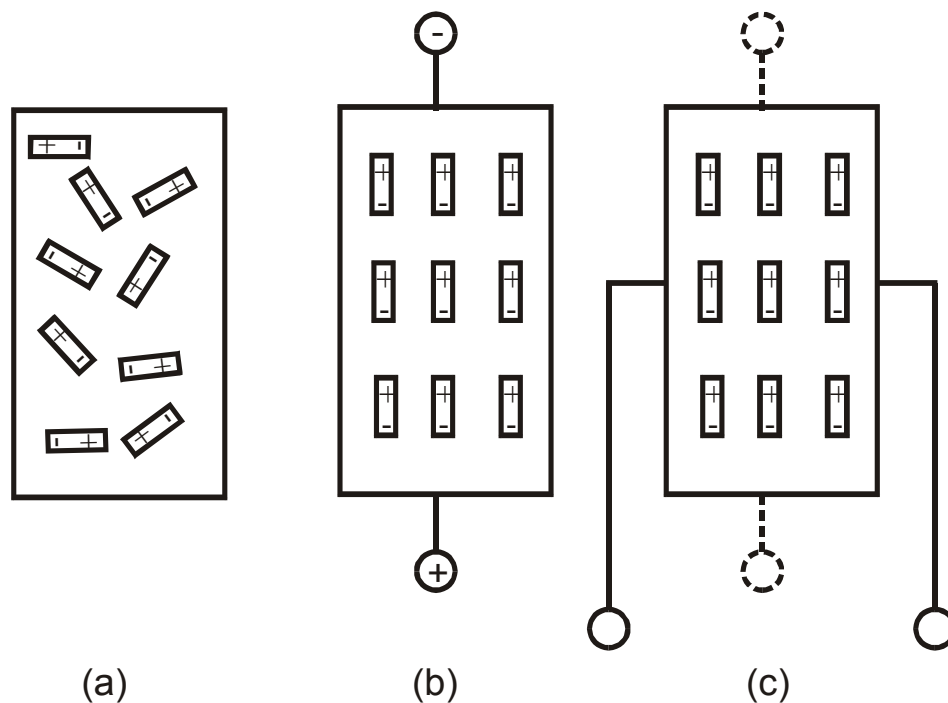
Figure 9   Construction of the piezoelectric material: (a) Electric dipoles are arranged randomly in normal crystalline structure; (b) Electric dipoles are aligned by an electric field; (c) An electric field perpendicular to that used for alignment is used to generate waves. (Adapted from [11])
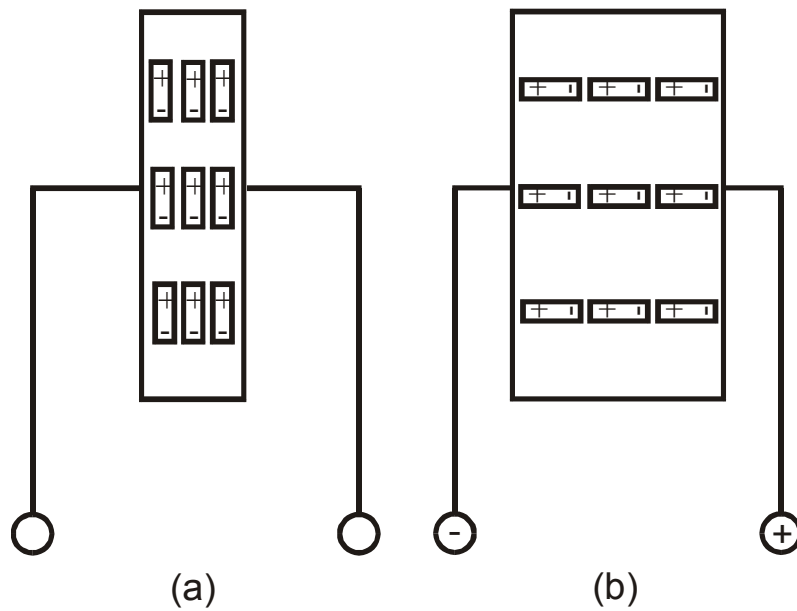
Figure 10   Generation of ultrasonic waves: (a) The normal thickness of the crystal when no electric field is applied; (b) The crystal is expanded if an electric field is present. (Adapted from [11])
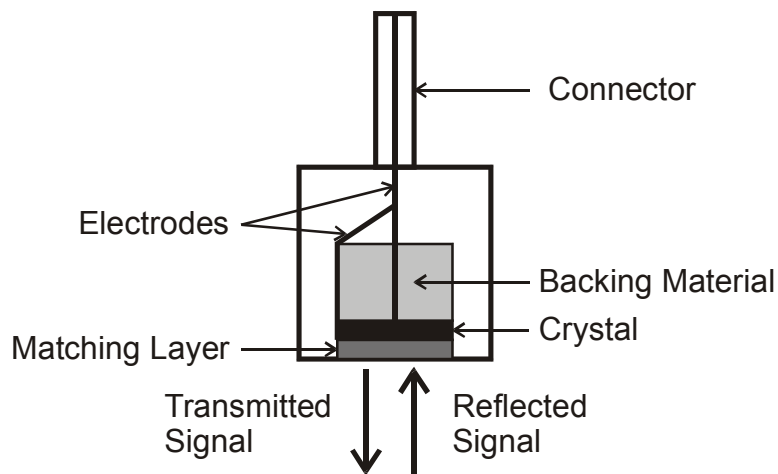
Figure 11   A simple transducer (Adapted from [11])
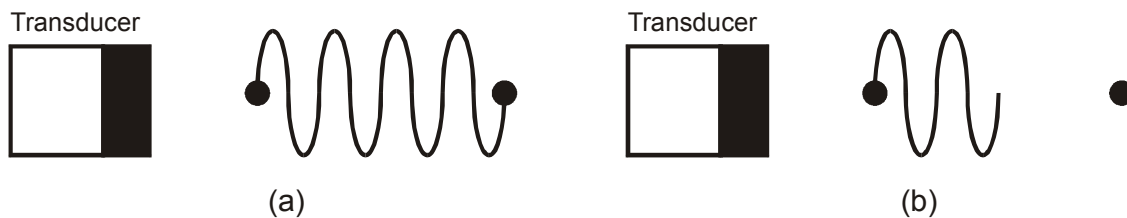
Figure 12   (a) Two objects cannot be resolved by a long ultrasound pulse; (b) A shorter ultrasound pulse resolves two objects. (Adapted from [11])

B) Factors Affecting the Sensitivity of an Ultrasonic System

The overall performance of an ultrasonic system is related to several factors: the electromechanical coupling coefficient, k, the transmission coefficient, d, the reception coefficient, g, and the mechanical coefficient, Q. These quantities are discussed below.

The transmission coefficient, d, is defined as the fraction of electrical energy that is converted to ultrasonic energy. The reception coefficient, g, specifies the fraction of ultrasonic energy that is converted to electrical energy. The electromechanical coupling coefficient equals the product of the transmission coefficient and the reception coefficient. The conversion efficiency, described by the electromechanical coupling coefficient, is the most important factor in determining the sensitivity of an ultrasonic system.

The axial resolution of an ultrasound system is also very important in determining the quality of an ultrasound image. The axial resolution is defined as the minimum distance between two objects that can be identified as distinct. The axial resolution is closely related to the length of the emitted ultrasound pulse (see Figure 12). The length of the pulse, in turn, depends on the mechanical coefficient, Q, of the crystal, which is defined as the energy stored per oscillating cycle divided by the energy lost per cycle. A low-Q crystal emits a short ultrasonic pulse, and therefore, should be chosen in diagnostic ultrasound applications for improving the axial resolution.

C) Reception and Amplification

As explained earlier, the ultrasonic waves reflected by biological interfaces exert a force on the crystal. The crystal then converts the mechanical energy to electrical energy. However, the electrical signals produced are typically in the microvolts or millivolts range. To be properly displayed on the screen, the signals are amplified to about 10 volts.

As described in Section 2.1.1, Part E, the ultrasonic energy is exponentially decreasing due to absorption when it penetrates the tissue. Therefore, echoes produced at equally reflective interfaces have different amplitude depending on the distances between the interfaces and the receiver. It is not convenient for the viewer to take the depth of interfaces in consideration when observing the displayed signal. To solve this problem, a time-gain compensation (TGC) control is used to amplify the signal according to the time it is received, or equivalently, the depth at which the signal is reflected. The TGC control has a reverse exponential profile (see Figure 13).

D) Electronic Processing

The electrical signal induced by the received acoustic echo is in the radio frequency range. For an observer, the interpretation of such a high frequency signal is very confusing. Therefore, the signal is processed before it is displayed.

This first step in this procedure is the rectification of the received signal. Rectification refers to an operation in which the negative values of the signal are converted to their absolute values. Then an envelope-detection is performed on the rectified signal. Usually the area under the resulting signal is computed and recorded because it is related to the intensity of the received echo. In the display, a spike is used to represent this area. In addition, a rejection control can be used to eliminate the spike that has an amplitude lower than a certain threshold (see Figure 14).
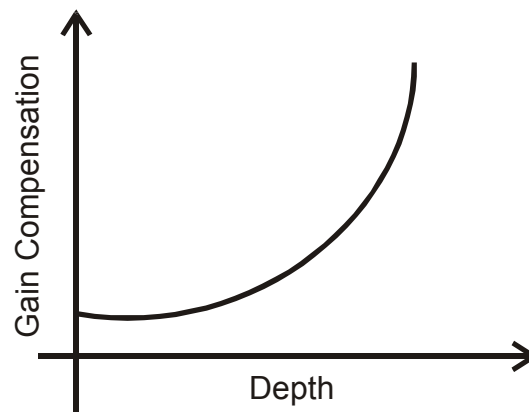
Figure 13   Time-gain compensation (TGC) control profile
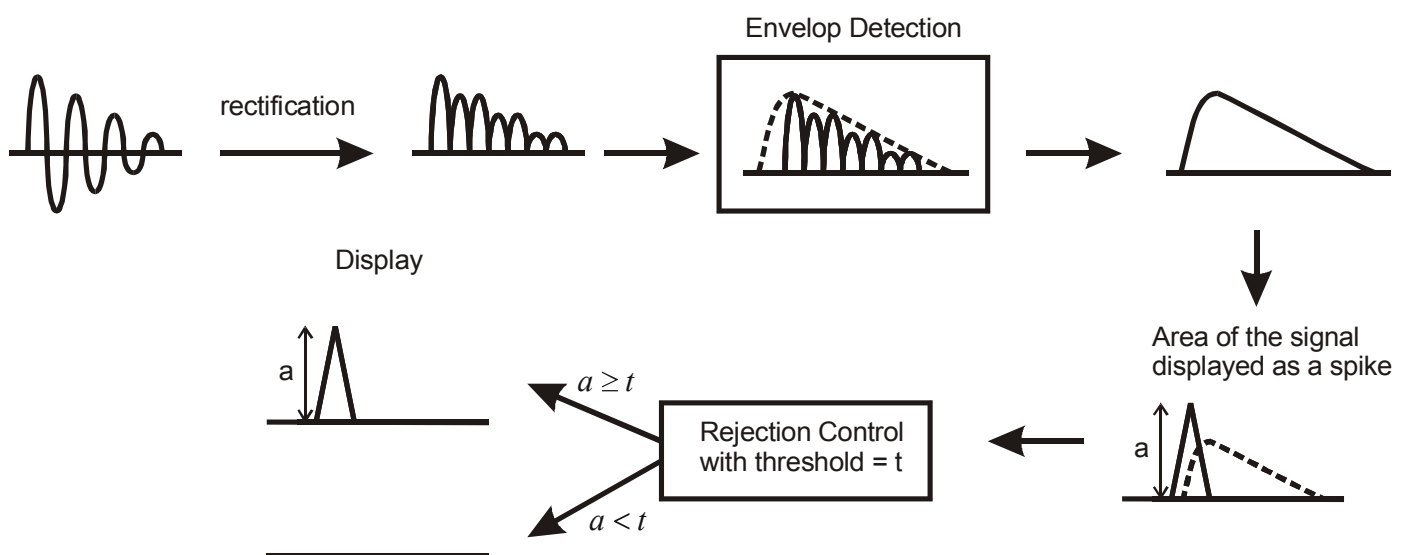
Figure 14   Procedures for processing the electrical signal

E) Display

The cathode ray tube is used for displaying purpose (see Figure 15). It is an evacuated glass tube with a large potential difference from the cathode to the screen. This potential difference accelerates the electrons emitted from the cathode. The screen is coated with a layer of phosphorescent material that emits light when hit by electrons.

There are two sets of plates that control the position at which an electron hits the screen: The X-deflection plate applies electric field in the horizontal direction and the Y-deflection plate applies electric field in the vertical direction. The X-deflection plate performs a sweep across the screen at a rate equals to the velocity of the ultrasound beam in the medium, while the Y-deflection plate is controlled by the signals resulting from the electronic processing procedure described in Part D. As a result, a graph of the intensity of the reflected echo versus the depth at which the ultrasonic wave is reflected is displayed on the screen.
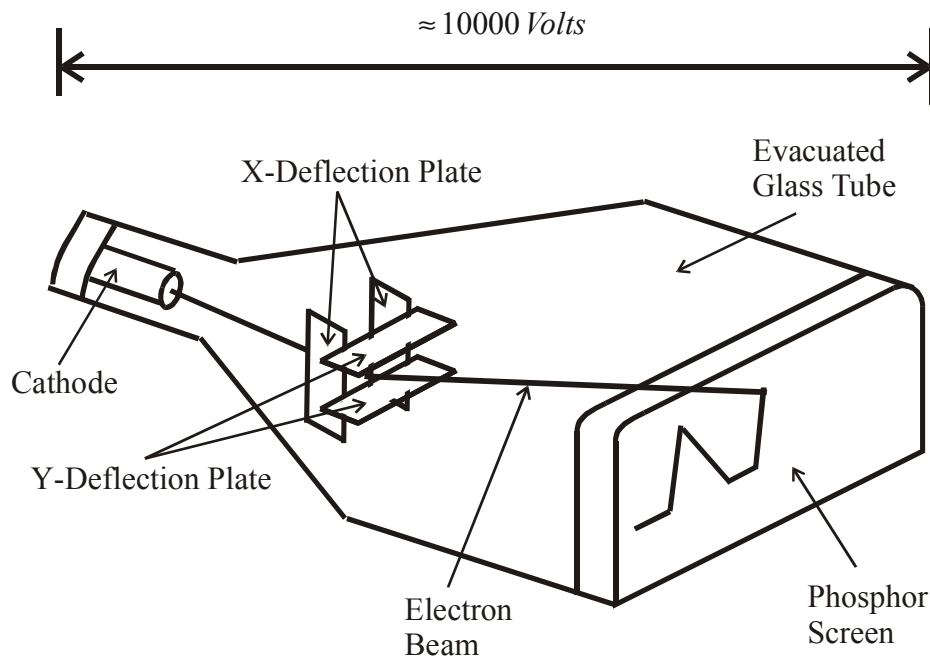
Figure 15   Cathode ray tube (Adapted from [11])
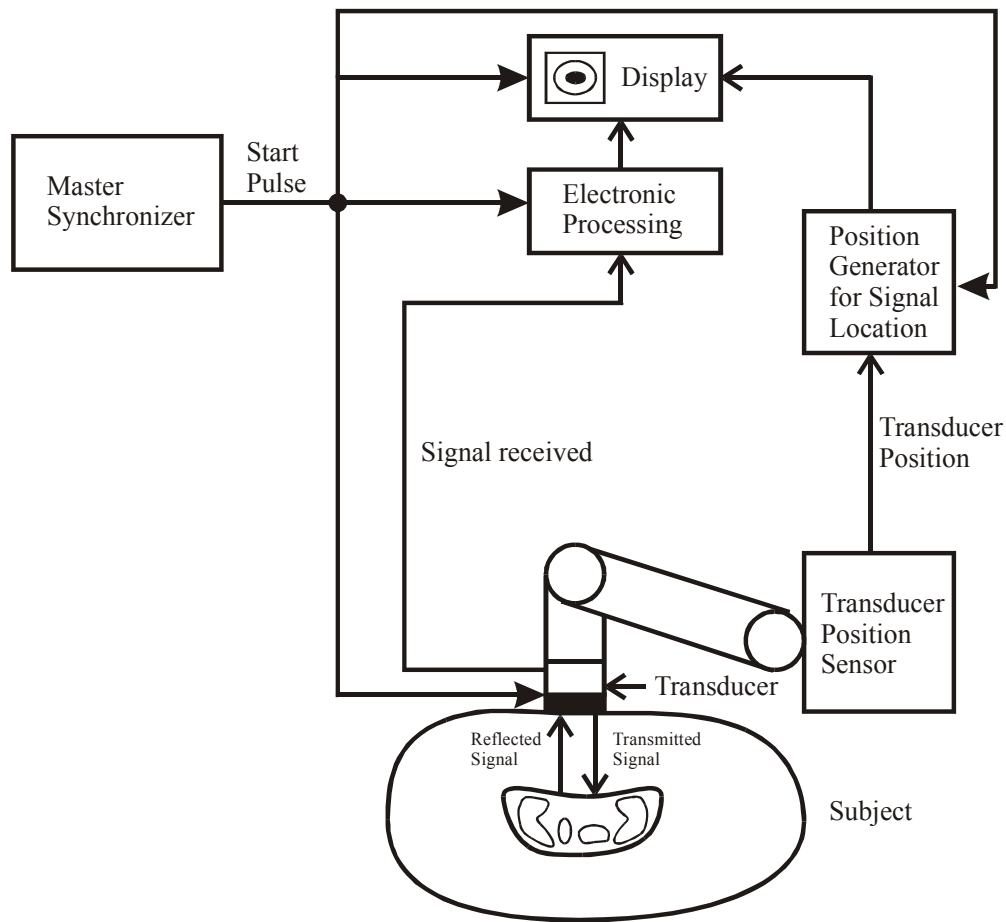
Figure 16   Block diagram of a B-Mode scanner (Adapted from [11])

2.1.4. Instrumentation of B-Mode

A B-Mode scan is very similar to an A-Mode scan, except the amplitude in an A-Mode plot is represented by the brightness of a dot in a B-Mode scan. Figure 16 shows the block diagram of a B-Mode scanner. The operating principles of a B-Mode scanner are basically the same as that of an A-Mode scanner. However, since B-Mode scanning is performed in different directions in order to obtain a two-dimensional portrait, one needs to determine the position of the transducer very accurately so that the reflected ultrasonic beam is correctly mapped in a two-dimensional image.

For this reason, the transducer is attached to one end of a special scanning arm. This scanning arm is connected to a position sensor, which accurately determines the horizontal and the vertical position of the transducer as well as the angle at which the ultrasonic beam is directed. This position information is passed to the position generator, which computes the location of the ultrasonic path. The electronically processed signal is then plotted along this path on the two-dimensional image.

2.1.5. Artifacts in Ultrasound Images

After understanding the basic operating principles of a B-Mode scanner, we are finally in a position to discuss several artifacts appeared in ultrasound images. By artifacts, we refer to physical phenomena that lead to an incorrect representation of the biological structures. Note that artifacts do not necessarily have a negative impact on medical diagnosis. Actually, experienced radiologists may be able to derive important information from the artifacts. In the following, the most common artifacts are discussed:

1) Enhancement – More energy is able to pass through cystic (liquid-like) structure compared to surrounding tissues that have higher attenuation. Since the received echo is processed by the TGC control, which compensates the energy loss by absorption, the structure behind the cystic structure appears to be brighter than the surrounding structure (see Figure 17(a)). This phenomenon, though classified as an artifact, makes it easier for us to identify the prostate boundary.

2) Shadowing – Since most of the acoustic energy is attenuated or reflected by a solid structure, a shadow appears behind the solid structure (see Figure 17(b)). This effect is helpful for radiologist to identify gallstone or renal stone.

3) Boundary shadow – Due to the refraction of the ultrasonic beam near the edge, a shadow appears behind the edge of a cystic structure (see Figure 17(c)).

4) Reverberation – This effect occurs when the sound is bounced back and forth by multiple interfaces. When the ultrasonic beam is reflected by an interface and travels towards the transducer, a fraction of the beam is reflected by another interface into the body. This beam is consequently reflected by the original interface a second time and travels towards the transducer. As a result, a single interface is represented by multiple traces on the image (see Figure 18).

5) Propagation speed error – The ultrasonic system assumes the velocity of sound is constant in all biological tissues. Since the depth of an interface is computed by the simple formula that multiplies the time at which the reflected echo is received and the velocity of ultrasound in the medium, the depth of interfaces will be incorrectly displayed if the velocity of ultrasound in a certain medium is different from the assumed constant (see Figure 19).
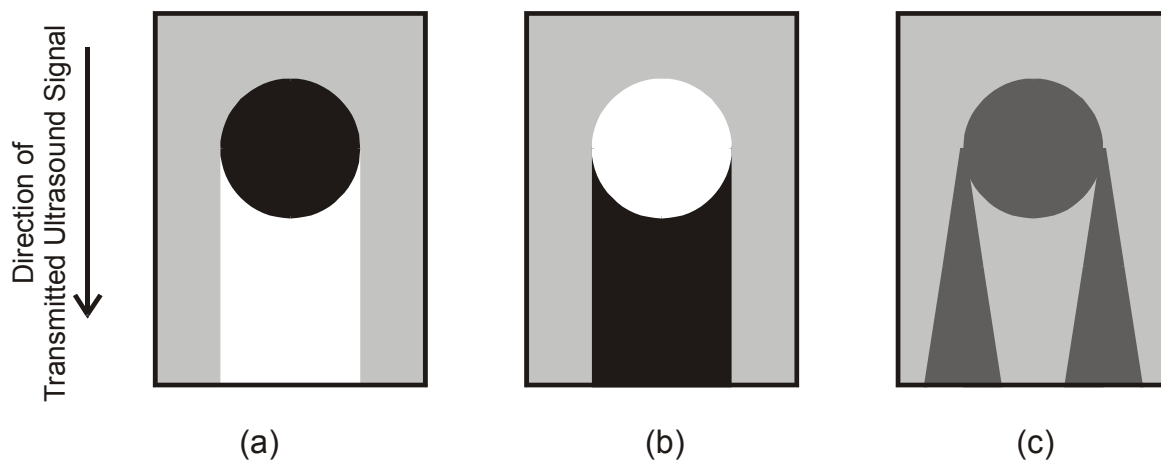
Figure 17   Artifacts in ultrasound images: (a) Enhancement; (b) Shadowing; (c) Boundary Shadow
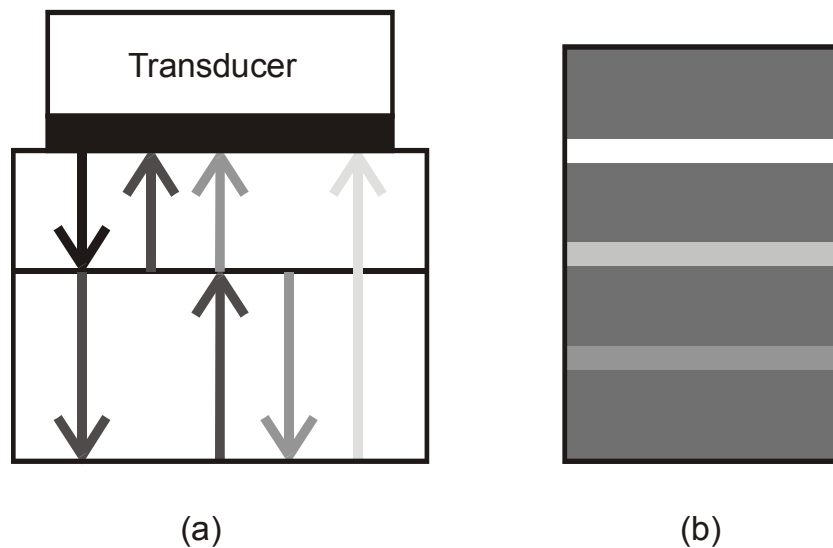
Figure 18 (a) A simple model demonstrating the reverberation artifact. The sequence of reflections starts from the left. The intensities of the reflections are represented by the darkness of the arrows; (b) the corresponding ultrasound image
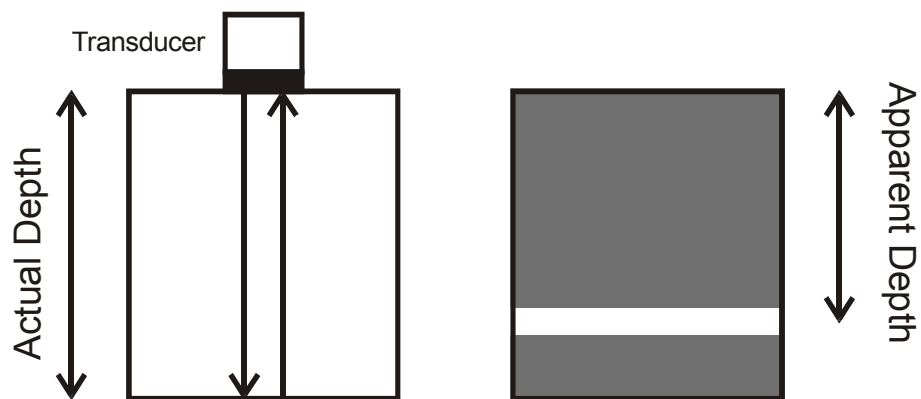
Figure 19   Ultrasound travels at a velocity higher than the assumed velocity, normally set as 1540m/s. (a) The actual depth of the interface; (b) the depth of the interface shown in the ultrasound image

2.2.   Magnetic Resonance Imaging (MRI)

This section gives a brief introduction on the operating principles and the instruments used in MRI[2].

2.2.1. MRI Physics

The human body is primarily composed of water and fat. Both water and fat have hydrogen atoms. Hydrogen atom possesses a property called spin, which can be thought of as a small magnetic field that can generate magnetic resonance signal. In this section, we will look at the microscopic and macroscopic properties of particles having net spin.

A) Microscopic Properties of Particles Having Spin

Under the influence of an external magnetic field of strength *B*, a particle with net spin can absorb a photon of frequency υ, which is expressed as

$$\upsilon = \gamma B$$

(2.6)

where γ is the gyromagnetic ratio. For the hydrogen atom, $\gamma = 42.58$ MHz/T. υ is called the Larmor frequency of the particle.

To understand why a particle with net spin only absorbs a photon with Larmor frequency, we consider the particle to be a small magnet with north pole and south pole. When placed in an external magnetic field, the particle aligns itself with the field. However, the particle can attain a low-energy level, in which its north pole is attracted to the south pole of the external magnetic field and vice versa (see Figure 20(a)), or it can have a high-energy configuration, when it is aligned in the way shown in Figure 20(b). The particle can undergo a transition from the low-energy level to the high-energy level through the absorption of a photon. The photon that causes the transition must have an energy matching the energy difference between the two energy levels. Since the energy of a proton is directly proportional to its frequency, the particle with net spin only absorbs a photon with Larmor frequency.

---

[2] *This section is based on [18].*
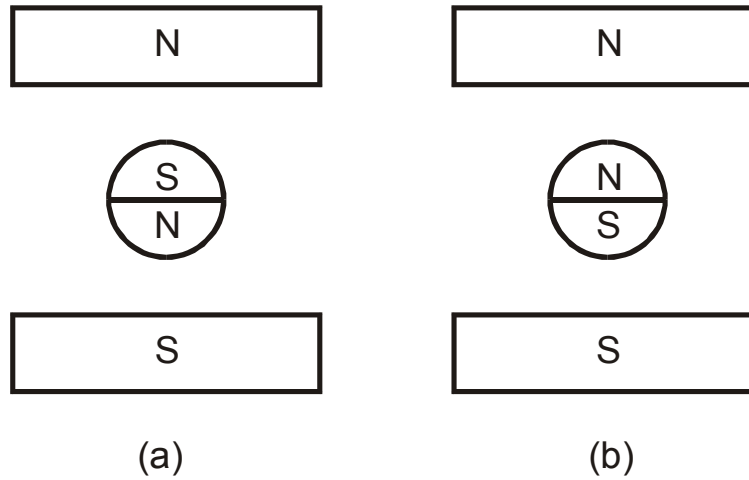
Figure 20   (a) A particle in the low-energy level; (b) A particle in the high-energy level

B) Macroscopic Properties of Particles Having Spin

Since it is too complicated to describe the behaviour of spins in a microscopic level, physicists define a group of spins under the influence of the same external field as a spin packet. The magnetic field of the spin packet is represented by a magnetization vector. The vector sum of magnetization vectors of all spin packets is called the net magnetization. From here on, effects are described in terms of the net magnetization.

In the following discussion, we adopt a coordinate system in which the external magnetic field, B_o, is directed along the z-axis.

At equilibrium, the net magnetization vector aligns to the direction of the external magnetic field (see Figure 21(a)). However, one can apply energy of a specific frequency to change the net magnetization vector. Before progressing further, we will decompose the vector into its longitudinal component, which is parallel to the z-axis, and its transverse component, which lies on the x-y plane (see Figure 21(b)). The dynamics of these two components of the magnetization vector are considered separately.

After the longitudinal magnetization vector is changed from its equilibrium value, it returns to the equilibrium value according to an exponential profile:

$$M_z(t) = M_o + \left(M_z(0) - M_o\right)e^{-t/T_1}$$

$$(2.7)$$

where M_o is the magnitude of the magnetization vector at equilibrium and T_1 is the time constant that depends on the temperature and the viscosity of the material.

On the other hand, the transverse component of the magnetization vector rotates about z-axis at the Larmor frequency of the spin packets. Since, within a material, there are a lot of spin packets that experience a slightly different magnetic field, they are rotating in a different Larmor frequency. Therefore, there exist phase differences between the magnetization vectors of spin packets. As a result, destructive interference occurs and the magnitude of the net magnetization vector decreases exponentially:

$$M_{XY}(t) = M_{XY}(0)\, e^{-t/T_2}$$

$$(2.8)$$

where $M_{XY}(t)$ is the magnitude of the net magnetization vector in the transverse direction and T_2 is the time constant that is different from T_1.

We mentioned that it is possible to change the net magnetization vector from its equilibrium value. Now, we shall find out how it could be changed.

Since the transverse magnetization vector is rotating at the Larmor frequency, it is convenient for us to adopt a rotating frame of reference x'-y'-z that rotates about z-axis at Larmor frequency (see Figure 22). Suppose a coil with an alternating current is placed around x-axis. If the frequency of the alternating current is equal to the Larmor frequency, the coil provides a constant magnetic field along the x'-axis. Since the frequency of the alternating current is normally in the radio frequency or RF range, this coil is called a RF coil. In MRI, the magnetic field produced by the RF coil is called the B_1 magnetic field. Under the influence of the B_1 magnetic field, the net magnetization vector rotates about the direction of the B_1 magnetic field by an angle of θ, which can be calculated by the following formula:

$$\theta = 2\pi\gamma\tau|B_1|$$

$$(2.9)$$

where τ is the length of time the B_1 field is on. Figure 23 shows an example where the net magnetization vector is rotated by 90°. This principle is used to choose the slice for which a magnetic resonance image is taken, as we will see in next section.
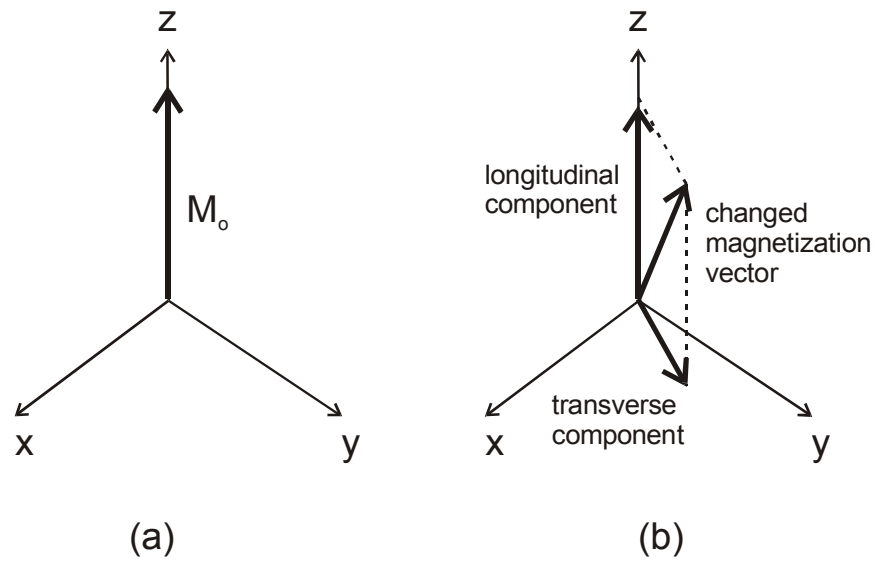
Figure 21   (a) The net magnetization vector at equilibrium, M_o; (b) the net magnetization vector is changed. This vector can be decomposed into its longitudinal and transverse components.
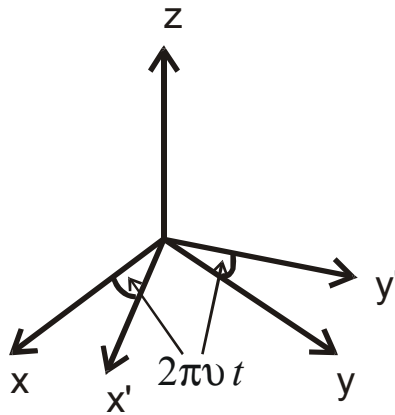
Figure 22   The rotating frame of reference compared to the stationary coordinate frame
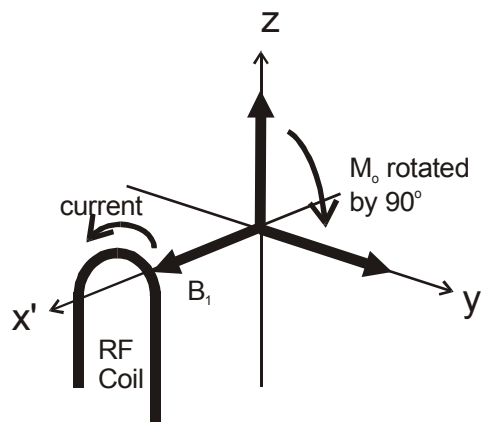
Figure 23   The B_1 field rotates the net magnetization vector by 90°. (Adapted from [18])

2.2.2. Imaging Principles

This section describes how a MRI image is obtained. Since MRI is a two-dimensional imaging technique, one needs to choose a two-dimensional slice for which the image is taken. Section A introduces the procedure for selecting a slice.

A) Slice Selection

Suppose a part of the body is modelled as a cube of small magnetization vectors (see Figure 24(a)). To take an image for a particular slice, the first step is to rotate the magnetization vector of that slice of 90° (see Figure 24(b)). The first step taken to achieve this is to apply a constant magnetic field B_o and a magnetic field gradient G_z to produce a magnetic field profile that is linearly increasing along the z-direction, i.e.,

$$B_z(z) = B_o + zG_z$$

$$(2.10)$$

Suppose the slice chosen is experiencing a magnetic field of B_s, and therefore, the magnetization vectors within the slice have a Larmor frequency equal to $f_s = \gamma B_s$. To rotate the magnetization vectors of the slice by 90°, it is reasonable to think that a pulse vector B_1 with frequency f_s and duration of τ given by (2.9) should be applied (see Figure 25(a)). Before applying this pulse vector, we should analyse the frequency content of this pulse. By using the convolution theorem, it can be shown that the Fourier transform of the pulse B_1 is a sinc function, shown in Figure 25(b). The amplitude of the sinc function is highest at the frequency f_s and the magnetization vectors of the chosen slice are rotated by 90°. However, since the sinc function has a non-zero amplitude at the neighbourhood of f_s, the magnetization vectors near the chosen slice will also rotate, but in an angle less than 90°. Therefore, a sinc-shaped angle profile results, which is not really a slice.

In order to get the ideal profile shown in Figure 24(b), the spectrum of the pulse B_1 should have a square distribution centring at the frequency f_s. However, this is not possible because it would require the pulse to be an infinite-duration, modulated sinc signal. A solution would be to use the truncated, f_s-modulated sinc function shown in Figure 26(a). The Fourier transform of this function is shown in Figure 26(b). This spectrum demonstrates the windowing effect in signal processing, which is a major problem in designing a bandpass filter. It can be observed that the spectrum demonstrates a gradual transition, rather than the ideal sharp switch, at the edge. The second problem is that the spectrum has a non-zero value out of the passband, the frequency band corresponds to the resonance frequency of the magnetization vectors within the slice, and it causes the magnetization vectors out of the slice to be slightly rotated. Although an ideal slice profile is not possible, the profile generated using this method is much better than that shown in Figure 25(c). In MRI, the slice is normally selected using the B_1 pulse described in this paragraph.
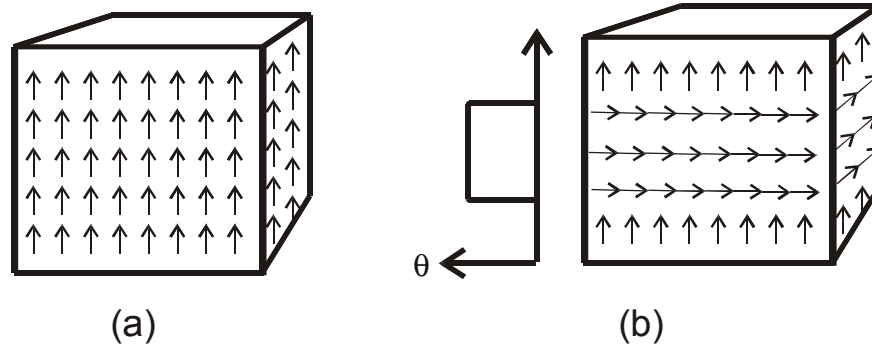
Figure 24 **(a) A cube of small net magnetization vectors; (b) the net magnetization vectors of a certain slice are rotated by $\theta = 90^{o}$.**
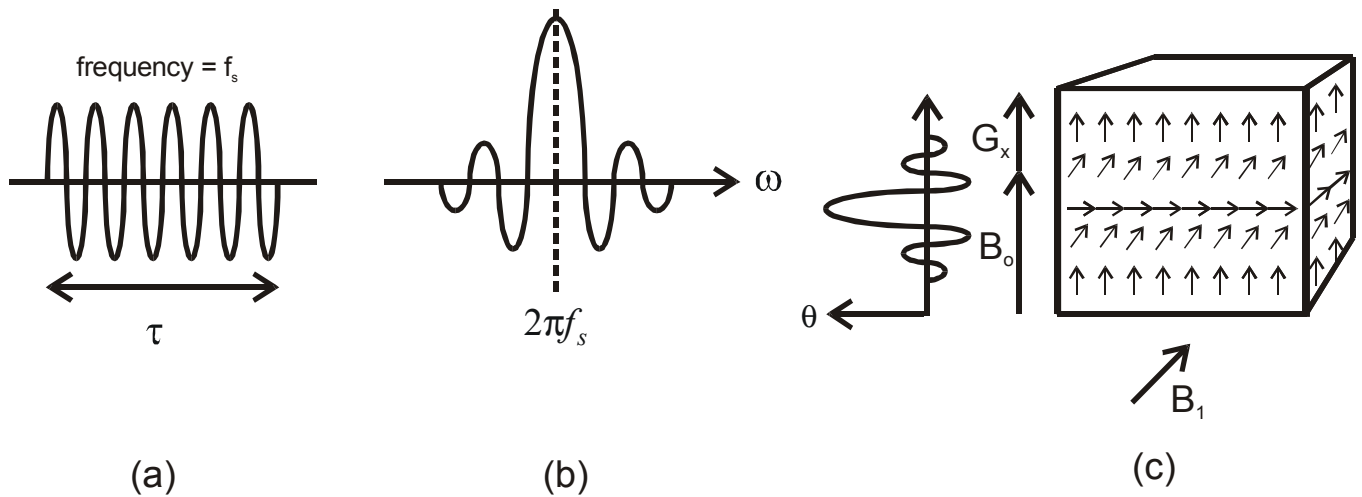
Figure 25 **(a) The pulse vector B_1 with frequency f_s and duration of τ is applied; (b) the one-sided Fourier transform of the pulse shown in (a); the resulting magnetization vectors profile (Adapted from [18])**
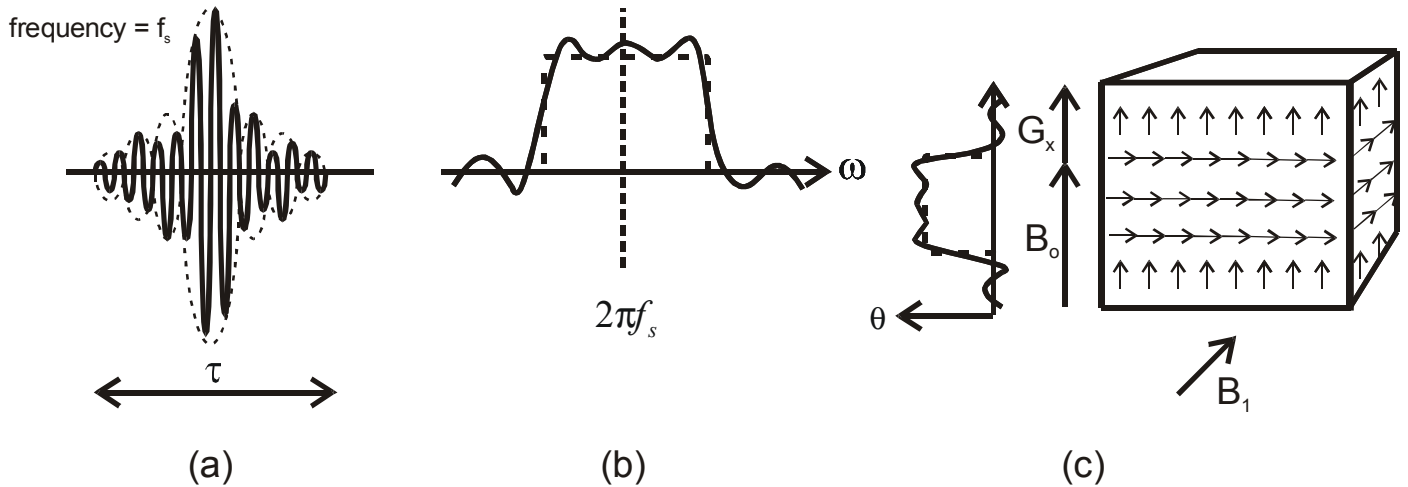
Figure 26 **(a) The pulse vector B_1 is enveloped by a sinc function; (b) the one-sided Fourier transform of the pulse shown in (a); the resulting magnetization vectors profile (Adapted from [18])**

B) Magnetic Field Gradient

After choosing the appropriate slice on which we wish to take an image, we should consider how the imaging is performed.

First, the magnetic field gradient in z direction, which was used for slicing, is removed. It is clear that all the small magnetization vectors within the chosen slice are rotating in the same frequency, since they are experiencing the same magnetic field, namely B_o. One would expect that if a one-dimensional linear magnetic field gradient is applied along the x-y plane, the frequency of the small magnetization vectors vary linearly along the direction where the gradient is applied. To demonstrate this idea, suppose a linear gradient G_x is applied along x-axis, as shown in Figure 27. The external magnetic field applied to different regions is different according their positions:

$$B_z(x) = B_o + xG_x$$

(2.11)

The resonance frequency of spin packets along x-axis can therefore be expressed as

$$\upsilon = \gamma(B_o + xG_x) = \upsilon_o + \gamma xG_x$$

(2.12)

One can detect the magnetic resonance spectrum along the direction along which the gradient is applied. This spectrum can be used to construct a one-dimensional image of the slice in question because the amplitude of the signal at the frequency υ is proportional to the number of spins at the position $x = \dfrac{\upsilon - \upsilon_o}{\gamma G_x}$.

Using this method, one-dimensional images along several angles are taken. One can use a technique called back projection imaging to map this set of one-dimensional images into a two-dimensional image. Figure 28 shows an example of this procedure taken from [18].
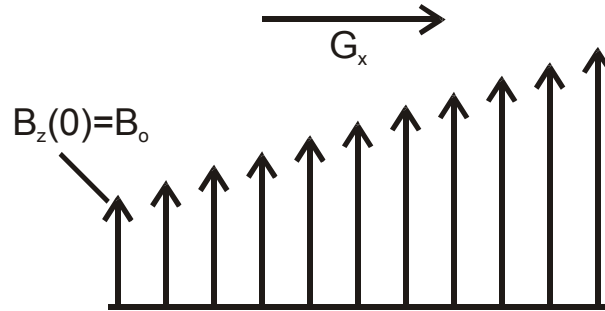
Figure 27   A linear gradient of B_o is applied along x-axis.
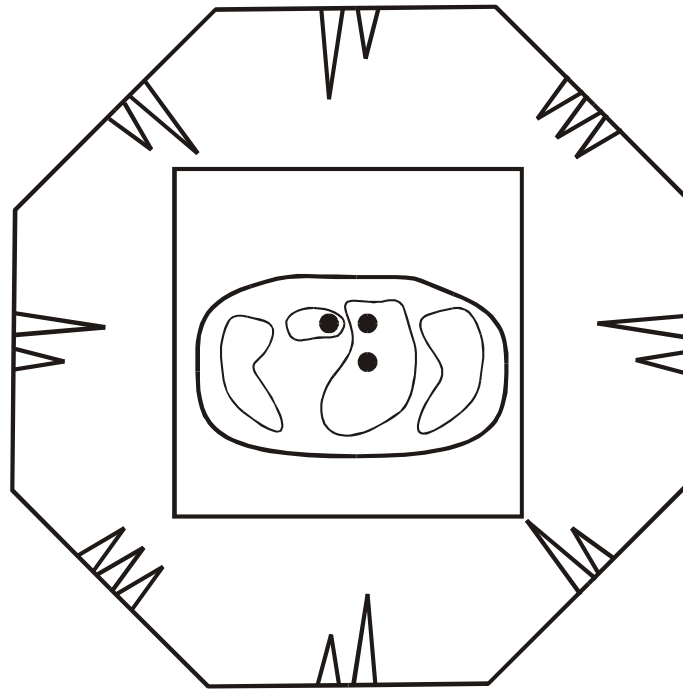
Figure 28   The back projection technique. In the example, it is assumed that there are only 3 spin packets, indicated by the dots. (Adapted From [18])

2.2.3. MRI Instrumentation

After introducing the basic operating principles of the MRI technique, we are in a position to understand the actual machine that performs MRI. Figure 29 shows the schematic of the MRI system. The function of each component is described below:

1) Magnet – The magnet produces a homogeneous B_o field necessary for the imaging procedure. Most of the magnets used in this application are superconducting magnets. These magnets are made up of superconducting wires, which have zero resistance when immersed in liquid helium having a temperature close to absolute zero.

2) Gradient Coils – The gradient coil produces a linear magnetic field gradient of B_o in X, Y and Z directions. The uses of these linear magnetic field gradients are described in Section 2.2.2.

3) RF Coil – The RF coil produces the B_1 field necessary for the slice selection procedure described in Section 2.2.2, part A. It is also responsible for receiving the magnetic resonance signals.

4) Shield – The scan room is surrounded by an RF shield and a magnetic shield. These shields prevent the RF signal and the magnetic signal from propagating out of the room.
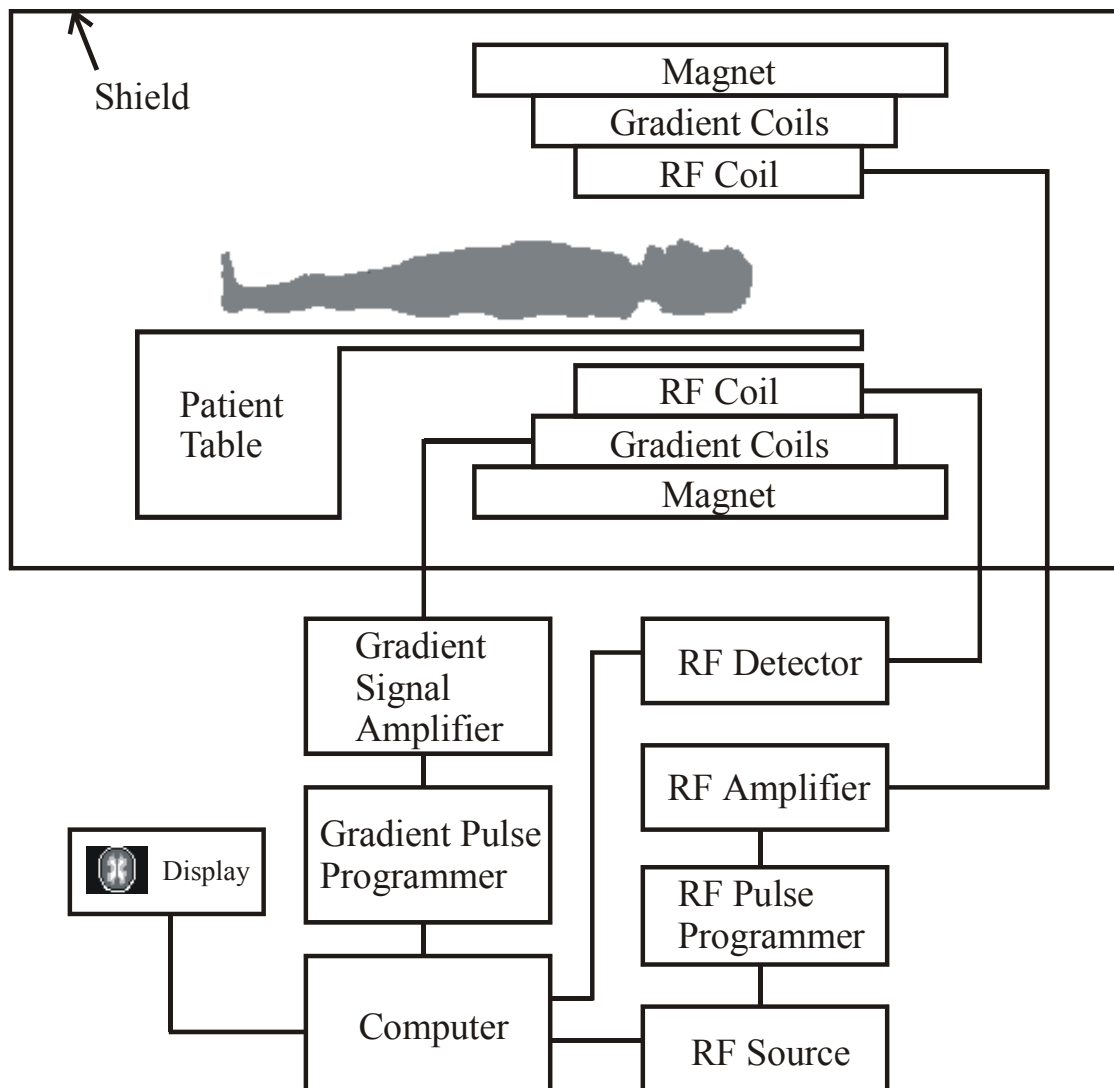
Figure 29   Schematic of the MRI system (Adapted from [18])

5) Patient Table – A computer-controlled patient table is used to position the patient accurately. A positioning accuracy of 1mm can be achieved.

6) Computer – The computer controls all components of the imager. Under the instruction of the computer, the RF source produces a sinusoidal wave of the desired frequency and the pulse programmer multiplies the sine wave by a sinc pulse. The processed pulse looks like the one illustrated in Figure 26(a). This processed pulse is amplified by a RF amplifier, which increases the power of the signal to the appropriate level. The computer also controls the gradient pulse programmer, which shapes the linear gradient profiles along the X, Y and Z directions appropriately. These gradient profiles are also amplified by a gradient amplifier before they are used to drive the gradient coils.

2.2.4. Artifacts in MRI Images

In this section, we briefly describe some artifacts that could appear in a MRI image. In the following description, the artifacts are classified according to their source:

1) B_o Inhomogeneity – MRI technique assumes a homogeneous B_o magnetic field. An inhomogeneous B_o magnetic field causes spatial or intensity distortion.

2) Gradient – Distortion could also occur when the linear magnetic field gradient is not shaped as desired. This is mostly caused by a damaged gradient coil.

3) RF Inhomogeneity – RF inhomogeneity is usually used to refer to the nonuniformity of the B_1 magnetic field. It causes variations of intensity in an image. This artifact is caused by the failure of the RF coil or the presence of nonferromagnetic material in the imaged object.

4) Motion – The movement of the imaged object may results in a blurring of the MRI image.

5) Flow – Flow artifact is caused by the flow of body fluid. This causes problem in the slicing process described in section 2.2.2, part A): A slice is chosen by applying a B_1 magnetic field on it, but the slice may move away from its original position due to the flow of body fluid when the image is taken.

2.3.   Comparison between Ultrasound Imaging and MRI

In this section, several techniques for diagnosing or screening prostate cancer are introduced, which include digital rectal examination (DRE), prostate-specific antigen (PSA) test and medical imaging test. The basic operating principles of two major imaging techniques, ultrasound and MRI, are described.

A comparative study of MRI and ultrasonography in staging early prostate cancer [21] shows that the sensitivity, the percentage of patients correctly identified as having the disease that spreads beyond the prostate, and the specificity, the percentage of patients correctly identified as having a localized tumour, of the MRI technique is higher than the corresponding values obtained using ultrasonography. However, because of its relatively low installation and operating costs, and its portability, ultrasonography is dominant in prostate-cancer-related applications. Since components required for ultrasonography are basically a monitor, a transducer, a scanning arm and a computer that coordinates the functions of the components in an ultrasonic system, it is not hard to understand that the cost of installing such a system is much lower than the installation cost of a MRI system that consists of a specially maintained superconducting magnet, which is the most expensive component of the MRI system, gradient coils and RF coils. Furthermore, the MRI system needs to be put in a room that is equipped with RF shield, which prevents

the high power RF pulses from radiating out through the hospital and interfering with the operations of other medical equipments, whereas the ultrasonic system weights only about thirty pounds and can be moved around. In addition, it may be dangerous to scan someone who has internal implants using MRI. For example, people with pacemakers cannot be safely scanned because the strong magnetic field applied during the scan may affect the function of the pacemaker. Although people with orthopaedic implants, such as screws and artificial joints, can often be scanned because the implants are firmly embedded in the bone, the presence of metallic implants may result in an inhomogeneous magnetic field, causing a severe distortion of the MRI image. Also, some imaging tasks are not possible using MRI. For example, the transrectal ultrasound (TRUS) can be used to guide biopsy because of the small physical size of the transducer.

Despite the popular use of ultrasonography, it is shown in [21] that the sensitivity and the specificity in staging early prostate cancer are disappointing using conventional TRUS examination. As a first step towards improving the sensitivity and specificity, the medical imaging group in the University of Waterloo is interested in enhancing the quality of the ultrasound image by some image processing techniques. The enhanced image is believed to be more useful in extracting information about the anatomical structure of the prostate, with which the accuracy of cancer identification is expected to improve. Segmentation of the prostate boundary is a crucial step in extracting information about the shape and the volume of the prostate, and is the focus of this thesis.

Chapter 3

Image Segmentation Methods

Section 1 of this chapter gives a survey of image processing techniques that have been used for segmentation. The rationale behind using the discrete dynamic contour (DDC) model and the wavelet analysis in the proposed prostate boundary segmentation algorithm is also stated. Section 2 introduces a wavelet-based multiscale edge detection technique. This technique is based on dyadic wavelet transform, which is different from the conventional pyramidal wavelet transform. We will explain the differences between these two wavelet-transform techniques, and describe the reason for choosing the dyadic wavelet transform as a building block in the proposed algorithm. After the multiscale edge detection algorithm is formalized, the wavelet basis has to be chosen before the algorithm can be implemented. Although the quadratic spline wavelet is chosen for the segmentation problem, any orthogonal or biorthogonal wavelet can be used and the choice would depend on the characteristics desired to be captured. Section 3 describes the operation mechanism of the DDC model implemented by the University of Western Ontario. This model is used as a building block of our proposed algorithm.

3.1.   Methods used for Segmentation

Image segmentation is the grouping of a set of pixels that share similar characteristics, such as intensity and texture. In our study in segmenting the prostate, segmentation can be described as a procedure that separates the foreground of the image, which refers to the set of pixels that is mapped from structures inside the prostate, and the background.

Segmentation techniques widely used can be classified into three categories: threshold techniques, edge-based techniques and model-based techniques. In the following, a brief introduction of these techniques is given, along with the pros and cons of each segmentation technique.

3.1.1. Threshold Techniques

This technique is very simple: Given an image A, which could be the original image itself or the image resulted after applying a transform operator on the original, if the greyscale value of the pixel $A[m,n]$ is lower than a preset threshold θ, the pixel will be classified as belonging to a predefined region. Otherwise, the pixel falls into another region [29].
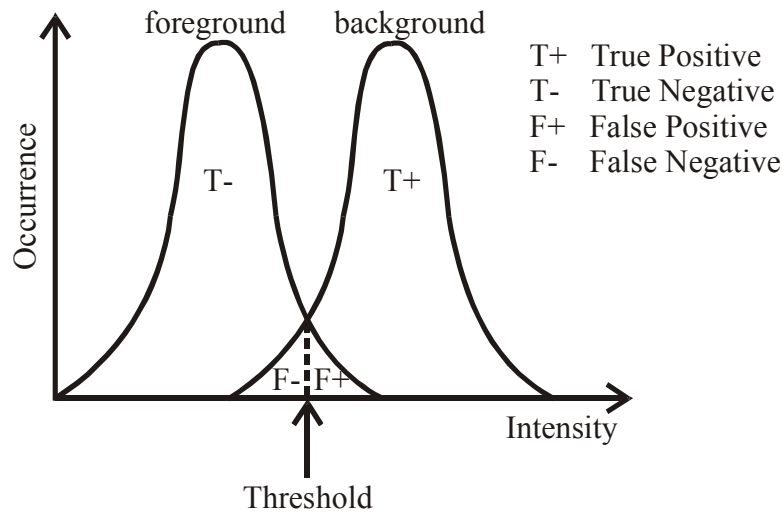
Figure 30   Distribution models of the foreground and the background

The most important issue in this technique is the choice of the threshold θ. Normally, distribution models associated with different regions, which describe the occurrence of a range of pixel value in the corresponding region, are established before the classification (see Figure 30). An appropriate choice of threshold is derived from these distribution models.

The threshold technique is efficient in obtaining a segmentation if different regions of the image have a high contrast in their intensities or other measurable features. However, as could be observed in Figure 30, if two distributions have a significant overlapping region, detection error (i.e., false positive and false negative shown in Figure 30) will be large. A partial solution to this problem may be to define the threshold to be a function of the position, based on pixel value distributions of different regions in the neighbourhood of the position in question. This method is based on the idea that the local variations of the pixel values representing a particular type of tissue is smaller comparing to the variations in the whole image, and therefore, the standard deviation of the distributions decrease, reducing the overlapping area in different distribution. Also, the thresholding technique is sensitive to noise or intensity inhomogeneity. As a result, regions will become disconnected, and a closed segment is often hard to obtain.

3.1.2. Edge-Based Techniques

Edge-based techniques find the edge of the desired object as opposed to finding the whole object in the threshold technique. This class of techniques finds the edge by locating regions where pixel values change significantly in a neighbourhood. Two widely used edge-detection techniques are the gradient-based procedure and the zero-crossing procedure.

However, both the gradient-based method and the zero-crossing method are sensitive to undesirable fluctuations caused by noise. To reduce the impact of noise on the segmented contour, one needs to smooth the image. The smoothing filter should be chosen such that it is localized in the frequency domain and the spatial domain – it should be localized in frequency domain because it should act as a low-pass filter to suppress high-frequency noise, and it should be spatially localized because the location of the edge can not be precisely identified if the image is filtered by a wide filter. The Gaussian filter is very popular in this application because it has a minimum space-bandwidth product.

A) Gradient-Based Procedure

An abrupt change can be detected by finding the first derivative of a one-dimensional function or the gradient of a two-dimensional function. For example, in a one-dimensional function, the edge can be located by finding the maximum of the first derivative (see Figure 31).

B) Zero-Crossing Procedure

An edge can also be found by locating the zero-crossing of the second derivative of a one-dimensional function or the Laplacian of an image (see Figure 31).

However, this approach has a significant drawback. Finding the zero-crossing of the second derivative of a function is equivalent to finding the inflection point, where the first derivative attains either a maximum or minimum. Sharp variation occurs at inflection point where the first derivative is at a local maximum (see Figure 31). On the other hand, slow variation occurs at inflection point where the first derivative is at a local minimum (see Figure 32). Solely using the zero-crossing technique, it is impossible to distinguish between these two kinds of inflection point, and therefore edge will be incorrectly located.
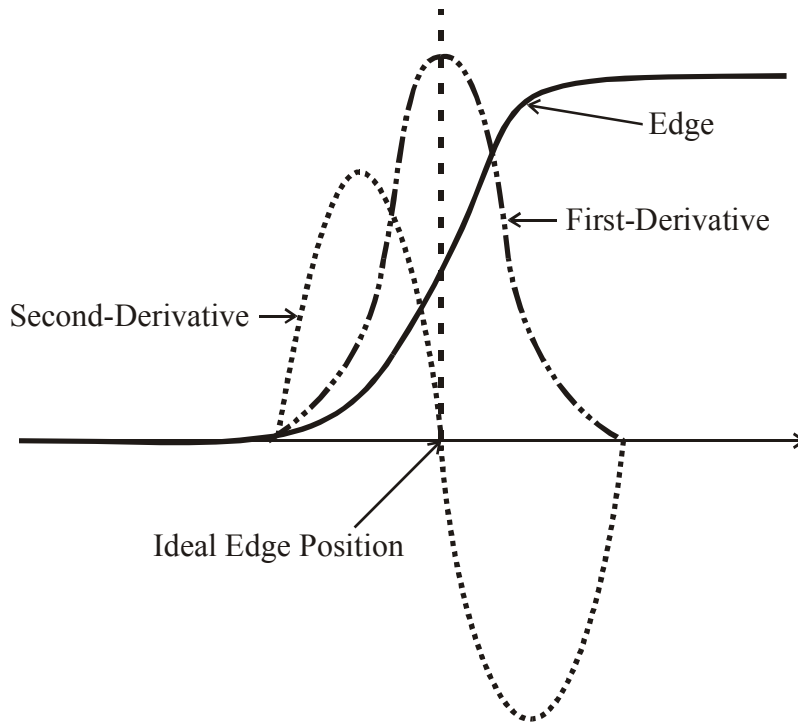
Figure 31   An edge structure and its first and second derivatives
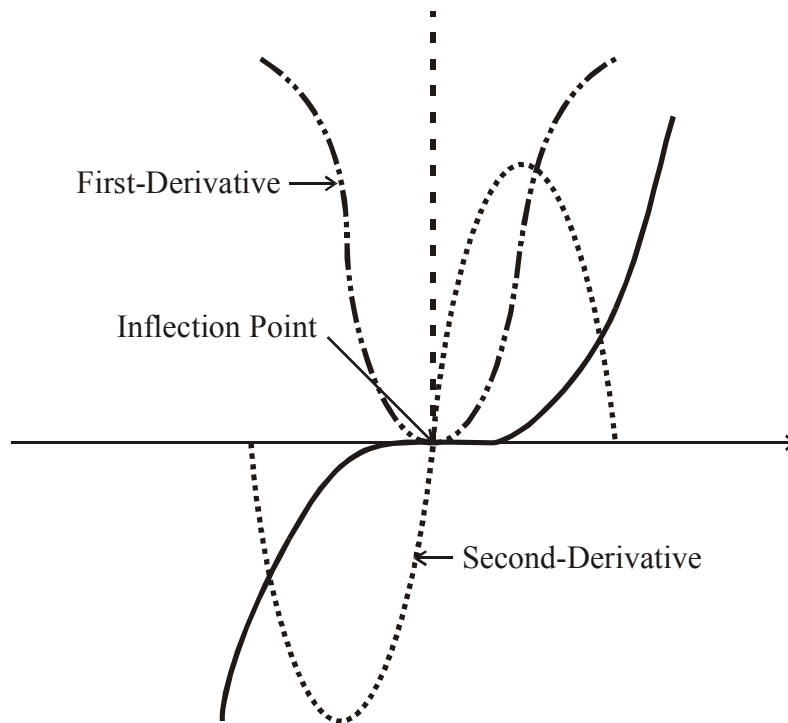
Figure 32   A slow variation point. Note that the second derivative at the point is zero, meaning that it is an inflection point, and will be falsely identified as an edge by the zero crossing technique.

3.1.3. Model-Based (or Deformable Model) Techniques

The deformable model consists of a closed parametric curve that moves under the influence of the internal force and the external force. The role of the internal force is to maintain the smoothness of the curve, whereas the external force is defined according to some image feature, such as the intensity or the gradient. The closed contour is first placed near the desired boundary. Then the forces update the contour iteratively until it converges to a final contour that is considered – by the algorithm – to be closest to the desired edge.

The most significant advantage of the deformable model is its ability to form a closed boundary regardless of the structure of the object. Since the smoothness of the contour is constrained by the internal force, the deformable is not susceptible to local variation caused by noise. A disadvantage of the model is that it requires an initial contour, and the final contour is often sensitive to how the initial contour is chosen.

3.1.4. Segmentation Methods Used in the Proposed Algorithm

The proposed segmentation algorithm combines the idea of the segmentation techniques introduced above. Before applying the edge-detection technique described in Section 3.1.2, it is required to choose a smoothing function that is localized in space and frequency. It is mentioned that the Gaussian function is an appropriate choice in most applications. There is no exception in the prostate boundary identification problem. However, the formulation of the edge-detection part of the proposed algorithm is different from the conventional gradient-based approach. In the proposed algorithm, Mallat and Zhong's [5] wavelet-based formulation is adapted. In [5], it was shown that if the quadratic spline wavelet is used, the wavelet transform produces a result that is equivalent to that generated by conventional gradient-based approach using Gaussian function. However, it is reasonable to guess that other smoothing functions, such as the rectangular (or indicator) function, may be a better choice in terms of extracting the edge of a cancerous tumour. In other words, by using different wavelet bases and the same wavelet transform framework, it is possible to find a better segmentation of the tumour. This idea will be investigated in Section 4.2. Since the wavelet-based framework allows the choice of different wavelet bases that suit different purposes, its range of applications is wider than that of conventional gradient-based approach.

After applying the wavelet-based edge-detection technique on the image, we use the DDC model, in which the external force is assigned according wavelet-transformed image. To further enhance the resulting boundary, a modified threshold technique is used, which will be described in Section 4.3.

Before progressing further, two major building blocks in the proposed algorithm – Mallat and Zhong's wavelet-based edge-detection technique and the DDC model – are described

in Section 3.2 and 3.3 respectively. The introduction of the proposed algorithm is postponed to Section 5.1.

3.2. Multiscale Edge Detection and Its Relation with Dyadic Wavelet Transform

In this section, the edge detection algorithm proposed by Mallat and Zhong [5] is introduced. In Section 3.2.1, the connection between the edge identification problem and the wavelet transform is made. Section 3.2.2 discusses a problem of the conventional pyramidal wavelet transform[3], the translational variance property. Section 3.2.3 defines an algorithm, dyadic wavelet transform, which solves the translational variance problem. An edge detection algorithm based on it will also be presented.

3.2.1. Multiscale Edge Detection

Multiscale edge detectors smooth the signal at various scales and detect sharp variation points from their first-derivative. This section explains how the gradient of a function is related to the wavelet transform.

We define a smoothing function $\theta(x)$ to be any function that integrates to a positive value and converges to 0 at infinity. Suppose $\theta(x)$ is differentiable. We define a wavelet function $\psi(x)$ to be the first-derivative of $\theta(x)$:

$$\psi(x) = -\frac{d\theta(x)}{dx}$$

$$(3.1)$$

The wavelet transform of $f(x)$ at the scale 2^j and position u computed with respect to the wavelet $\psi(x)$ can be written as

$$W_{2^j} f(u) \equiv \; < f(x), \frac{1}{\sqrt{2^j}} \psi^a\left(\frac{x-u}{2^j}\right) > = \left\{ f * \underline{\psi_{2^j}} \right\}(u)$$

$$(3.2)$$

where $\psi_{2^j}(x) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{x}{2^j}\right)$ and $\underline{\psi_{2^j}}(x) = \overline{\psi_{2^j}}(-x)$.

---

[3] *The pyramidal wavelet transform is introduced in textbooks on general wavelet theory, such as [1] and [3].*

Using (3.1), one can prove that

$$\underline{\psi}_{2^j}(x) = 2^j \frac{d\theta_{2^j}(x)}{dx}$$

(3.3)

Substituting this result into (3.2) yields

$$W_{2^j} f(u) = 2^j \frac{d}{du} \left\{ f * \underline{\theta}_{2^j} \right\}(u)$$

(3.4)

(3.4) demonstrates that the wavelet transform is the first-derivative of the function f smoothed at a scale 2^j. If j is large, the convolution with $\theta_{2^j}(x)$ removes small signal fluctuation and we therefore only detect the sharp variations of large structures.

This formulation can be easily extended in two dimensions. Suppose there is a 2-D smoothing function $\theta(x, y)$, which has similar characteristics as its 1-D correspondence, and there are two wavelets, which are the partial derivatives of $\theta(x, y)$ in x and y direction respectively:

$$\Psi^1(x, y) = -\frac{\partial \theta(x, y)}{\partial x} \quad \text{and} \quad \Psi^2(x, y) = -\frac{\partial \theta(x, y)}{\partial y}$$

(3.5)

Suppose $f \in L^2(R^2)$, the wavelet transforms of f with respect to Ψ^1 and Ψ^2 can be written as

$$W_{2^j}^k f(u, v) = < f(x, y), \Psi_{2^j}^k(x - u, y - v) > = f * \underline{\Psi}_{2^j}^k(u, v) \quad \text{for } k = 1, 2$$

(3.6)

Similar to the 1-D case, we denote

$$\theta_{2^j}(x, y) = \frac{1}{2^j} \theta\left(\frac{x}{2^j}, \frac{y}{2^j}\right) \quad \text{and} \quad \underline{\theta}_{2^j}(x, y) = \overline{\theta_{2^j}}(-x, -y)$$

(3.7)

The two scaled wavelets can be written as

$$\underline{\Psi}_{2^j}^1 = 2^j \frac{\partial \theta_{2^j}}{\partial x} \quad \text{and} \quad \underline{\Psi}_{2^j}^2 = 2^j \frac{\partial \theta_{2^j}}{\partial y}$$

(3.8)

With these notations defined, we can conclude that the wavelet transform components are proportional to the corresponding components of the gradient vector of $\left\{ f * \theta_{2^j} \right\}$, that is,

$$\begin{bmatrix} W_{2^j}^1 f(u,v) \\ W_{2^j}^2 f(u,v) \end{bmatrix} = 2^j \begin{bmatrix} \dfrac{\partial}{\partial u}\left\{ f * \theta_{2^j} \right\}(u,v) \\ \dfrac{\partial}{\partial v}\left\{ f * \theta_{2^j} \right\}(u,v) \end{bmatrix} = 2^j \nabla\left(f * \theta_{2^j} \right)(u,v)$$

(3.9)

The modulus of this gradient vector is proportional to the wavelet transform modulus

$$M_{2^j} f(u,v) = \sqrt{\left| W_{2^j}^1 f(u,v) \right|^2 + \left| W_{2^j}^2 f(u,v) \right|^2}$$

(3.10)

M_{2^j} can be used to identify significant variations in different scales: If $M_{2^j} f(u,v)$ is large for some j, u and v, it indicates that the function f has a significant variation at the neighbourhood of the point (u,v). The size of this neighbourhood is proportional to the scale 2^j.

For the ultrasound prostate images, this method would identify small-scaled variation mostly caused by different kinds of noise if j is small. It is not desirable in our segmentation problem. However, if j is too large, the outline of the prostate being identified would be too coarse and inaccurate. Therefore, j must be appropriately chosen by experimentation.

In this section, the connection is made between the gradient vector of $\left\{ f * \theta_{2^j} \right\}$, a smoothed version of a continuous function f, and its wavelet transform with respect to a certain wavelet function defined by (3.2). In Section 3.2.3, it will be demonstrated how the same concept can be applied to a digital image. A fast method to compute the modulus of the gradient vector of a digital image, called dyadic wavelet transform, will also be described. However, there is an important reason for introducing dyadic wavelet transform, rather than adapting the conventional pyramidal wavelet transform. The following section provides a detailed explanation on that.

3.2.2.   Translation Invariance

Translational invariance is important in pattern recognition. When a pattern is translated, its representations should be translated but not modified. Patterns are more difficult to be identified if its representation depends on its location. The main objective of this section is to show that the traditional pyramidal multiresolution representations are not translational invariant.

It can be easily proven that the continuous wavelet transform is shift invariant. Let $f_\tau(t) = f(t - \tau)$ be the translation of $f(t)$. The wavelet transform of f at scale s and position u can be written as a convolution product, that is,

$$W_s f(u) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) dt = f * \underline{\psi}_s(u)$$

(3.11)

where $\underline{\psi}_s(t) = \dfrac{1}{\sqrt{s}} \psi\left(\dfrac{-t}{s}\right)$, as previous defined.

The wavelet transform of f_τ at scale s and position u is

$$W_s f_\tau(u) = \int_{-\infty}^{+\infty} f(t - \tau) \psi_s(t) \, dt = W_s f(u - \tau)$$

(3.12)

which is the shift invariant property we wish to prove.

The conventional pyramidal wavelet transform, defined by

$$d_j[n] = \langle f, \psi_{j,n} \rangle$$

(3.13)

where $\psi_{j,n}(t) = \dfrac{1}{\sqrt{2^j}} \psi\left(\dfrac{t - 2^j n}{2^j}\right)$, is equivalent to the continuous wavelet transform sampled at time intervals 2^j. It can be written as

$$d_j[n] = W_{2^j} f(2^j n)$$

(3.14)

The effect of this sampling is shown in Figure 33. In general, the sampled coefficient of $W_{2^j} f_\tau(2^j n)$ is very different from a translation of $W_{2^j} f(2^j n)$. The wavelet transform is

shift invariant only if τ is a multiple of 2^j (i.e., $\tau = k2^j$), in which case $W_{2^j} f_\tau(2^j n) = W_{2^j} f(2^j(n-k))$.

$$W_{2^j} f(u)$$

$$W_{2^j} f_\tau(u)$$

$$2^j$$

$$\tau$$
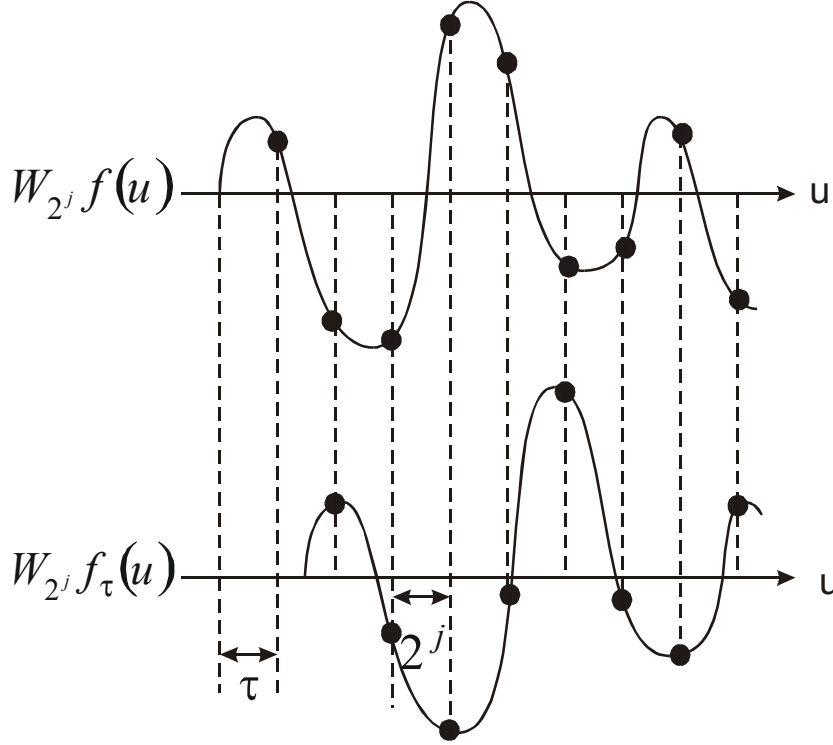
Figure 33   **If** $f_\tau(t) = f(t-\tau)$ **then** $W_{2^j} f_\tau(u) = W_{2^j} f(u-\tau)$. **Uniform sampling of** $W_{2^j} f_\tau(u)$ **and** $W_{2^j} f(u)$ **at** $u = 2^j n$ **may yield very different value if** $\tau \neq k2^j$.

In digital image processing, the discrete dyadic wavelet transform is used to overcome this problem.

3.2.3. Discrete Dyadic Wavelet Transform

In practical application, the input signal is measured at a finite resolution. In other words, the input signal, denote it as a_o, is a discrete signal. The nth sample of a_o can be considered as averages of some continuous function *f(t)* weighted by a scaling kernel $\varphi(t-n)$, where $\varphi(t)$ is some smoothing function with integral equals to 1.

Since a_o is a discrete signal, one can only translate it by an integer value τ. Since $a_o[n] = < f(t), \varphi(t-n) >$, it implies that the underlying continuous function *f* can only be translated by some integer τ. Therefore, if the continuous wavelet transform, defined in (3.2), is sampled at an interval equals to 1, the shift invariant property can be preserved.

This is the underlying idea that motivates the introduction of the discrete dyadic wavelet transform, which is defined in the following subsection.

A) One-Dimensional

Let $a_o[n]$ be the discrete input signal. It can be expressed as

$$a_o[n] = A_{2^0} f(n) = < f(t), \varphi(t-n) >$$

(3.15)

where we have defined $A_{2^j} f(u) = < f(x), \varphi_{2^j}(x-u) >$.

For any $j \geq 0$, denote

$$a_j[n] = A_{2^j} f(n) = < f(t), \varphi_{2^j}(t-n) >$$

(3.16)

The dyadic wavelet coefficients are computed for $j > 0$ over the integer grid

$$d_j[n] = W_{2^j} f(n) = < f(t), \psi_{2^j}(t-n) >$$

(3.17)

Note that (3.17) is the integer-sampled value of the continuous wavelet transform, defined in (3.2).

Mallat derived a fast algorithm to compute the discrete dyadic wavelet transform in [1].

In wavelet theory, a scaling sequence *h* and a wavelet sequence *g* characterizes the scaling function φ and the wavelet ψ respectively. They are defined by

$$\varphi(x) = \sum_{k \in Z} h[k] \varphi_{-1,k}(x) = \sum_{k \in Z} h[k] \sqrt{2} \varphi(2x-k),$$
$$\psi(x) = \sum_{k \in Z} g[k] \varphi_{-1,k}(x) = \sum_{k \in Z} g[k] \sqrt{2} \varphi(2x-k)$$

(3.18)

We denote $h_j[n]$ to be the filter obtained by inserting $2^j - 1$ zeros between each sample of $h[n]$; its Fourier transform is $h^f(2^j \theta)$. Also we let $\underline{h_j}[n] = \overline{h_j[-n]}$. Then for any $j \geq 0$,

$$a_{j+1}[n] = \{a_j * \underline{h_j}\}[n] \quad \text{and} \quad d_{j+1}[n] = \{a_j * \underline{g_j}\}[n]$$

(3.19) [4]

---

[4] *Proof of this proposition is provided in pp. 157-158 of [1].*

In Mallat and Zhong's paper [5], the quadratic spline scaling function was used. The Fourier transform of the scaling function φ, the wavelet ψ and the smoothing function θ are listed below (see Figure 34):

$$\varphi^F(\omega) = e^{-j\frac{\omega}{2}} \left[sinc\left(\frac{\omega}{2\pi}\right) \right]^3$$

(3.20)

$$\psi^F(\omega) = -\frac{j\omega}{4} e^{-j\frac{\omega}{2}} \left[sinc\left(\frac{\omega}{4\pi}\right) \right]^4$$

(3.21)

$$h^f(\theta) = \sqrt{2} e^{-j\frac{\theta}{2}} \left(cos\frac{\theta}{2} \right)^3$$

(3.22)

$$g^f(\theta) = -j\sqrt{2} e^{-j\frac{\theta}{2}} sin\frac{\theta}{2}$$

(3.23)

Since $\psi(t) = -\dfrac{d\theta}{dt}$,

$$\theta^F(\omega) = \frac{1}{4} e^{-j\frac{\omega}{2}} \left[sinc\left(\frac{\omega}{4\pi}\right) \right]^4$$

(3.24)

It can be observed from (3.24) that the smoothing function θ is closely related to the cubic spline function. In fact, by properties of Fourier transform, it can be determined that the smoothing function θ is proportional to $B_3(2(x-1/2))$, where B_3 denotes the cubic spline function.
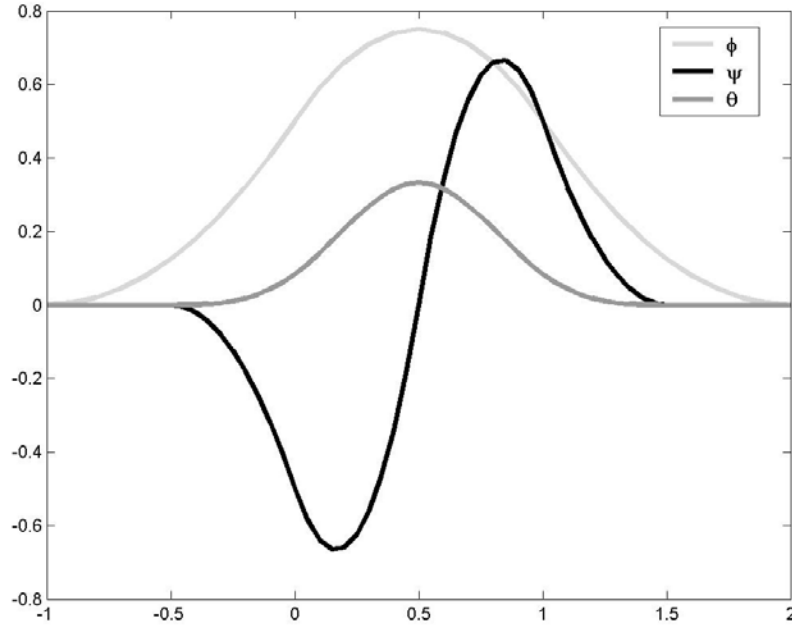
Figure 34   φ,ψ and θ used in Mallat and Zhong's paper [5]
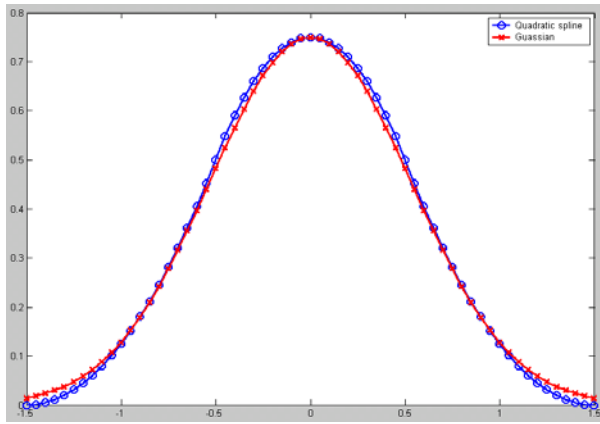
Figure 35   The quadratic spline function closely approximates the Gaussian function having a standard deviation of $4/3\sqrt{2\pi}$.
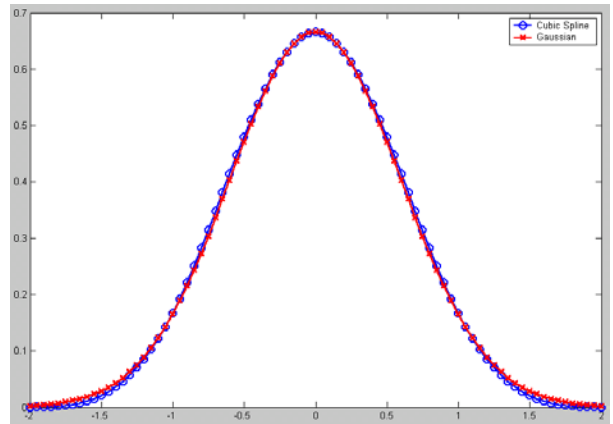
Figure 36   The cubic spline function closely approximates the Gaussian function having a standard deviation of $3/2\sqrt{2\pi}$.

Figure 35 and Figure 36 show that both the quadratic and cubic spline function closely approximates the Gaussian functions with standard deviations $4/3\sqrt{2\pi}$ and $3/2\sqrt{2\pi}$ respectively. It can be implied that the smoothing function θ can be approximated by the Gaussian function with standard deviation of $3/4\sqrt{2\pi}$ shifted by $1/2$.

From (3.4) and (3.17), it can be observed that d_j, the sequence of detailed coefficient at scale j, is the first derivative of $f * \theta_{2^j}$. Since θ_{2^j} can be closely approximated by the Gaussian function with standard deviation $2^j \left(3/2\sqrt{2\pi}\right)$, finding the local maxima of d_j is equivalent to Canny's multiscale edge detection algorithm [8].

For implementation of the dyadic wavelet transform, we only need the discrete filter $h^f(\theta)$ and $g^f(\theta)$. Coefficients of the filters are tabulated below:

n	$h[n]/\sqrt{2}$	$g[n]/\sqrt{2}$
-1	0.125	0
0	0.375	-0.5
1	0.375	0.5
2	0.125	0

Table 1   Coefficients of the discrete filter used in implementing dyadic wavelet transform

B) Two-Dimensional

In image processing, we are interested in a 2-D image, which can be represented by $I = \{i_{m,n}\}_{(m,n) \in Z^2}$.

Suppose there are an underlying continuous function $f(x,y)$ and a scaling function $\Phi(x,y)$ such that the original image I can be expressed as

$$i_{m,n} = a_0[m,n] = A_{2^0} f(m,n) = < f(x,y), \Phi(x-m, y-n) >$$

(3.25)

where we have defined $A_{2^j} f(u,v) = < f(x,y), \Phi_{2^j}(x-u, y-v) >$.

For any $j \geq 0$, we define the larger scale approximation a_j by

$$a_j[m,n] = A_{2^j} f(m,n) = < f(x,y), \Phi_{2^j}(x-m, y-n) >$$

(3.26)

As we have understood in Section 3.2.1, we need the wavelet transform of f with respect to Ψ^1 and Ψ^2, where Ψ^1 and Ψ^2 are defined according to (3.5). We define the integer samples of these two sets of wavelet transform as d_j^1 and d_j^2. Mathematically, we write

$$d_j^k[m,n] = W_{2^j}^k f(m,n) = < f(x,y), \Psi_{2^j}^k(x-m,y-n) > \quad \text{for } k = 1,2$$

(3.27)

Suppose $\Phi(x,y), \Psi^1(x,y)$ and $\Psi^2(x,y)$ can be written as separable products of one-dimensional scaling function φ and wavelet ψ in the following way:

$$\Phi(x,y) = \varphi(x)\varphi(y)$$

(3.28)

$$\Psi^1(x,y) = -\frac{\partial \theta^1(x,y)}{\partial x} = \psi(x)\sqrt{2}\varphi(2y) \quad \Rightarrow \quad \theta^1(x,y) = \theta(x)\sqrt{2}\varphi(2y)$$

(3.29)

$$\Psi^2(x,y) = -\frac{\partial \theta^2(x,y)}{\partial y} = \sqrt{2}\varphi(2x)\psi(y) \quad \Rightarrow \quad \theta^2(x,y) = \sqrt{2}\varphi(2x)\theta(y)$$

(3.30)

In (3.29) and (3.30), the scaling function φ contracts by a factor of 2. $\Psi^1(x,y)$ and $\Psi^2(x,y)$ are defined in this way so that $\theta^1(x,y)$ and $\theta^2(x,y)$ approximates a two-dimensional Gaussian function that has approximately equal standard deviations in both the x and y directions.

We can extend the 1-D fast dyadic wavelet transform algorithm in two dimensions with convolutions along the rows and columns of the image. In the following, we denote the separable convolution of the rows and columns of the 2-D image a_j with the 1-D filter u and v, respectively, by $a_j * (u,v)$. It can be proven[5] that a_{j+1}, d_{j+1}^1 and d_{j+1}^2 can be computed from a_j using the following filter bank structure:

$$a_{j+1} = a_j * \left(\underline{h_j}, \underline{h_j}\right),$$

(3.31)

$$d_{j+1}^1 = a_j * \left(\underline{g_j}, \delta\right),$$

(3.32)

$$d_{j+1}^2 = a_j * \left(\delta, \underline{g_j}\right)$$

(3.33)

---

[5] *The proof of the fast dyadic wavelet transform for 2-D signals is provided in the Appendices, Section A.1.*

where we have denoted the discrete Dirac filter by $\delta[n]$. $\underline{h}_j$ and $\underline{g}_j$ are defined the same way as we did before the introduction of (3.19). This filter bank structure is illustrated in Figure 37.

row by row
operation

column by column
operation

$* \underline{h}_j$

$* \underline{h}_j$

a_{j+1}

a_j

$* \underline{g}_j$

d_{j+1}^1

$* \underline{g}_j$

d_{j+1}^2
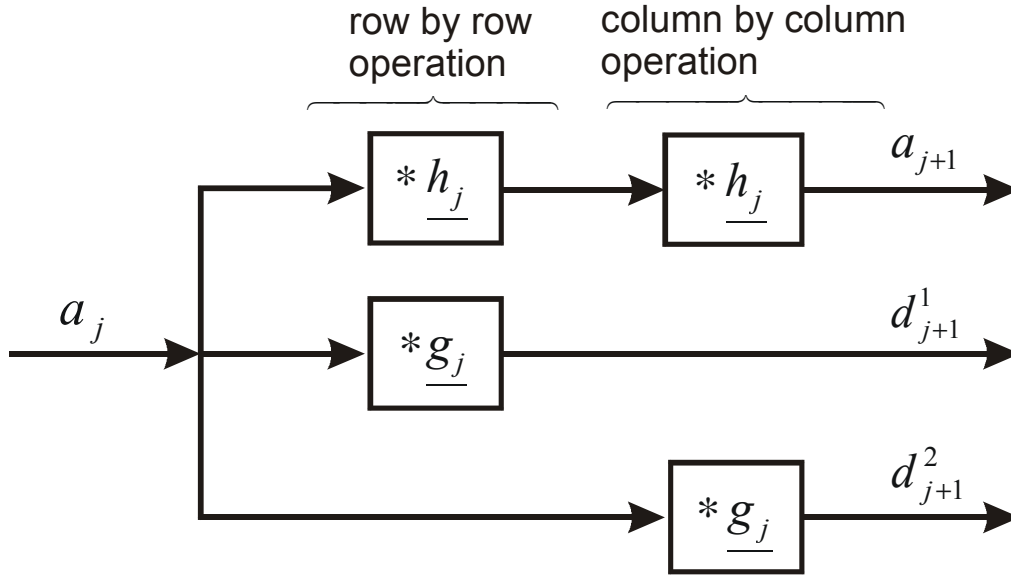
Figure 37   2-D Fast Dyadic Wavelet Transform

3.3.   Discrete Dynamic Contour (DDC)

In this section, the ideas behind the DDC algorithm implemented by the University of Western Ontario (referred to as the UWO program hereafter) are explained. As mentioned in Section 3.1.4, it is an important building block of the proposed segmentation algorithm. Section 3.3.1 describes the initialization model of the contour. Section 3.3.2 describes the structure of the discrete contour model and defines the forces that control its behaviour. A description of the dynamics of the deformation process is given in Section 3.3.3, and a technique for resampling the contour is presented in Section 3.3.4. The deformation process stops at a specific time when certain conditions are satisfied. In Section 3.3.5, these termination conditions will be discussed.

3.3.1. Initialization

The DDC initialization routine in the UWO program requires user to enter four initial vertices or control vertices. These vertices must be set close enough to the boundary determined by the naked eye.

Suppose the initial shape of the DDC is represented in parametric form as

$$p(u) = (x(u), y(u))$$

(3.34)

where p is the position of a point on the DDC and the parameter u varies between some minimum value u_0 to some maximum value u_m. In the UWO program, the minimum value is chosen to be 0 and the maximum value is chosen to be 4.

In general, the x-coordinates $x(u)$ of points on a curve are determined solely by the x-coordinates of the control vertices, and similarly for the y-coordinates. Since $x(u)$ and $y(u)$ are treated in the same way, it suffices to explain how $x(u)$ is determined only. By entering four initial points, the user specifies four data points $x(0), x(1), x(2)$ and $x(3)$. Since the initial contour is closed, one should add a fifth data point $x(4)$, which has the same value as the first data point $x(0)$. Between the five data points, there are four intervals between adjacent data points. Each interval is represented by a cubic polynomial $P_j(u)$, for $j = 0,1,2,3$. It is convenient to change the parameter u to s, such that $s = 0$ corresponds to the lower bound of an interval. In terms of the parameter s, we have the following constraint in defining the polynomials:

$$\begin{aligned} P_j(0) &= x(j) \\ P_j(1) &= x(j+1) \end{aligned} \qquad \text{for} \qquad j = 0,1,2,3$$

(3.35)

At each interior point $\{x(j): j=1,2,3\}$, the first and second derivatives of P_{j-1} and P_j are also set to be equal:

$$P_{j-1}^{(1)}(1) = P_j^{(1)}(0)$$
$$P_{j-1}^{(2)}(1) = P_j^{(2)}(0) \qquad \text{for} \quad j = 1,2,3$$

(3.36)

(3.35) and (3.36) specify *4m - 2* conditions, where *m* = *number of intervals* = *4*. Since there are four constants needed to be determined in each cubic polynomial and there are *m* polynomials, two more conditions are required. These two conditions could be chosen in a number of ways [6]. In the UWO program, the first and the second derivatives at the first point and the last point are matched:

$$P_3^{(1)}(1) = P_0^{(1)}(0)$$
$$P_3^{(2)}(1) = P_0^{(2)}(0)$$

(3.37)

At this point, the information required for calculating the four polynomials that define the x-coordinates, $\{P_j(s): j=0,1,2,3\}$, has been defined. de Boor [6] presented an algorithm of how these polynomials are actually calculated. This procedure is easily carried out by using the Spline Toolbox in MATLAB. The polynomials that define the *y*-coordinates are calculated using exactly the same method. For later discussion, it is more convenient to express the function that describes the x-coordinates (or *y*-coordinates), a piecewise polynomial, as a single function $x(u)$ (or $y(u)$):

$$x(u) = \begin{cases} P_0(u) & 0 \le u < 1 \\ P_1(u-1) & 1 \le u < 2 \\ P_2(u-2) & 2 \le u < 3 \\ P_3(u-3) & 3 \le u \le 4 \end{cases}$$

(3.38)

$y(u)$ is expressed in the same way, but the four polynomials that define the y-coordinates are, of course, different from that define the *x*-coordinates.

However, since only a finite number of vertices on the polynomials define the discrete contour, one should start to consider the spacing between adjacent vertices. In the UWO program, the spacing is closely related to the resolution of the contour model. The actual position of the vertices on the contour is determined by three steps.

In Step 1, the program uses the four initial points that users defined to estimate the length of the contour. This estimation is very rough; it is done by summing the distances between adjacent points. The number of points on the contour, *N*, is computed by

$$N = \left\lfloor \frac{length\ estimate}{resolution} \right\rfloor$$

(3.39)

where the $\lfloor x \rfloor$ takes the biggest integer that is smaller than x.

In Step 2, *N* vertices are chosen. Since $x(u)$ and $y(u)$ are defined only if $u \in [0,4]$, vertices are assigned starting from u = 0 to u = 4 in a uniform interval of u, according to the following algorithm:

Δu, the difference in *u* between adjacent vertices $= 4/(N - 1)$;
$\underline{u}$, the vectors containing all *u* at which *x* and *y* will be evaluated
$= [0, \Delta u, 2\Delta u, \dots, 4 - 2\Delta u, 4 - \Delta u, 4]$;

for *i = 1* to *N*
        vx(i) = x_coordinate of vertex *i* = $x(\underline{u}(i))$;
        vy(i) = y_coordinate of vertex *i* = $y(\underline{u}(i))$;
end

In the UWO algorithm, the four initial points user defined are assumed to be a point in final contour and will not be moved during the deformation process. Therefore, these four points must be included in the initial contour. In general, the method for assigning vertices described in last paragraph does not guarantee this because the integer 1, 2 and 3 may not be an element of the vector $\underline{u}$. Step 3 makes sure all user-defined points are vertices on the initial contour. The following algorithm basically inserts the vertices obtained in Step 2 one by one. However, if a particular vertex is too close to a user-defined point, the user-defined point is inserted in place of that vertex. Note that the first vertex is immediately inserted because it is the first user-defined point. The last vertex is not inserted because it is identical to the first one.

contour_x = a vector containing x-coordinates of the contour;
contour_y = a vector containing y-coordinates of the contour;
initial_point_index = a vector containing the index of the four user-defined initial points.

Insert $x(0)$ into *contour_x*;
Insert $y(0)$ into *contour_y*;
Insert *0* into *initial_point_index*;
index1 = 1;

for *i = 2:N - 1*

if *index1 <= 3*

$$d = \sqrt{(vx(i) - x(index1))^2 + (vy(i) - y(index1))^2} \; ;$$

end

if (*index1 <= 3*) & (*d <= resolution* / 1.25)
   Insert $x(index1)$ into *contour_x*;
   Insert $y(index1)$ into *contour_y*;
   Insert *i* into *initial_point_index*;
   index1 = index1 + 1;
else
   Insert $vx(i)$ into *contour_x*;
   Insert $vy(i)$ into *contour_y*;
  end
end

Since the user-defined initial point is not allowed to be moved during deformation, one should always keep track of the index of the initial points on the contour, which is what the variable *initial_point_index* for.

Figure 38 illustrates the initial contour defined using the described algorithm. The contour on the background is the prostate boundary defined by an expert observer. The four initial points required in the initialization algorithm are chosen from the set of points on the manually defined contour. It can be observed that the initial contour does not approximate the manually defined contour well enough. As we will demonstrate later, using the UWO program, it is impossible for the boundary to converge to a contour that closely approximates the manually defined contour.
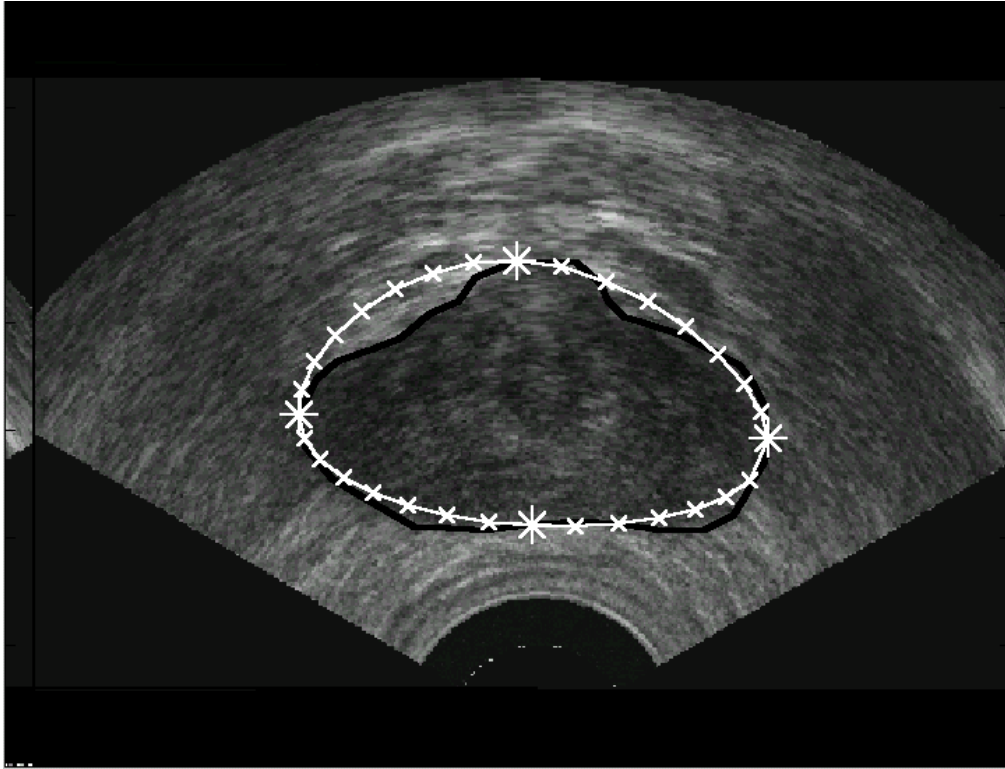
Figure 38   The vertices on the initial contour are represented by the cross symbol ($\times$). The four user-defined initial points are marked with asterisks (∗). The contour on the background (drawn in black) is the manually outlined boundary of the prostate.
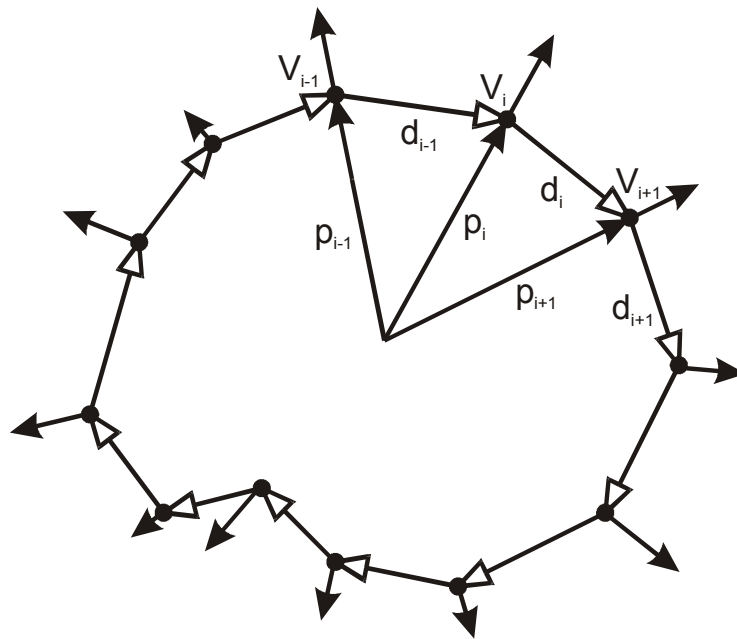
Figure 39   The model consists of a set of vertices V_i, which are connected by edges d_i.

3.3.2. Forces and Force Field

The UWO program uses the model presented in [7]. The basic structure of the model is illustrated in Figure 39; the closed contour is formed by joining the vertices with straight line segments. The vector p_i represents the position of a vertex V_i, while the vector d_i represents the vector starting from the vertex V_i and terminating at the vertex V_{i+1}. In the deformation process, different kinds of force act on the vertices. The resulting acceleration and velocity in vertex V_i is denoted by the vector a_i and v_i respectively.

A) Internal Forces

The internal forces defined in the model are related to the local contour curvature. The internal forces act as a constraint on the local curvature, and therefore, keeping the contour relatively smooth under the influence of external forces, which are defined according to all the variations of the image feature.

According to the conventional definition, local curvature is zero on the straight edge segments, while it is not defined at the exact position of a vertex. Therefore, one needs to find a new definition of the curvature, which must be proportional to the rate of change of the tangent's direction. Lobregt [7] defined the local curvature at the position of the vertex V_i to be the difference between the unit vectors $\hat{d}_i$ and $\hat{d}_{i-1}$:

$$c_i = \hat{d}_i - \hat{d}_{i-1}$$

$$(3.40)$$

Lobregt [7] also defined the vectors that represent the radial and tangential directions at the position of a vertex. The tangential unit vector $\hat{t}_i$ is defined by

$$\hat{t}_i = \frac{\hat{d}_i + \hat{d}_{i-1}}{\left\| \hat{d}_i + \hat{d}_{i-1} \right\|}$$

$$(3.41)$$

and the radial unit vector $\hat{r}_i$ is derived from rotating $\hat{t}_i$ by $-\pi/2$ radians[6]:

$$\hat{r}_i = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \hat{t}_i$$

$$(3.42)$$

---

[6] *A negative rotation represents a clockwise one.*
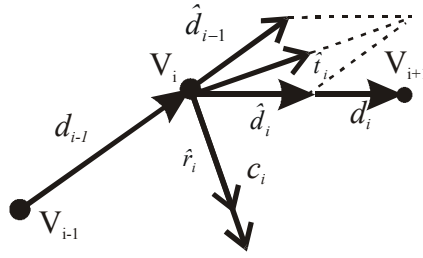
Figure 40   The local tangential vector $\hat{t}_i$ and the radial vector $\hat{v}_i$ at a vertex V_i

The vectors $\hat{r}_i$ and $\hat{t}_i$ represents a local coordinate system at the position of vertex V_i, which is useful in the calculation of internal as well as external forces.

It can be shown [7] that c_i is pointing either in the direction of $\hat{r}_i$, or in its opposite direction. In other words, c_i is a vector along the local r-axis with length equal to the dot product $(c_i \bullet \hat{r}_i)$. Expressing c_i in terms of the local r-t coordinate system will be found useful in the calculation of internal force later on:

$$c_i = (c_i \bullet \hat{r}_i)r_i$$

(3.43)
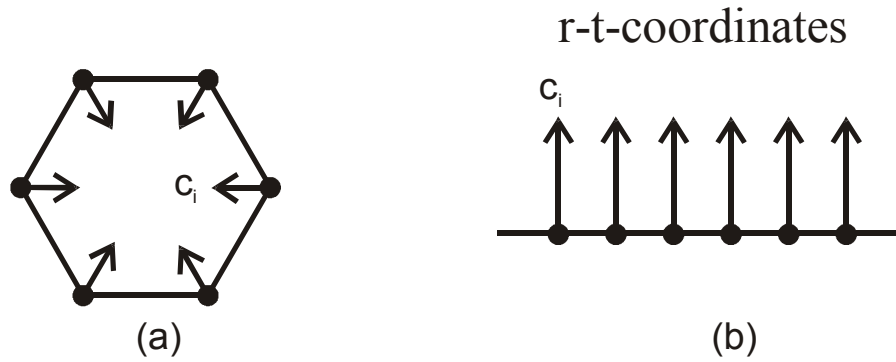
Figure 41   Curvature vectors c_i for a regular polygon in Cartesian representation and in local r-t-coordinates

At this point, the notion of local curvature has been defined. We should proceed in defining the internal forces. One would be tempted to think that the internal forces should be defined as being proportional to the local curvature vectors. Before we make the decision, we should get a clear understanding of how the internal forces affect the deformation dynamics. Assuming the external forces are completely absent, any closed shape would be deformed into a circle (or its discretized version, i.e., a regular polygon) because it has the smallest overall curvature. Since the purpose of the internal forces is to minimize the overall curvature, the deformation should stop at this point. However, as shown in Figure 41(a), it is not the case. If the internal force is defined as being proportional to the local curvature, it will act on the contour, without further decreasing the local curvature, until the contour implodes into a single point.

Lobregt [7] proposed an algorithm that reduces the local curvature without affecting parts of the contour with constant curvature. Since in the local r-t-coordinate system, the local curvature c_i is a one-dimensional variable in the radial direction, one can treat the local curvatures as function of a discrete variable, namely, the position i. The magnitude of the internal force can be represented by convolving this function with a discrete filter k_i, which should be a filter that eliminates the constant component of the function. In other words, this filter must have a Fourier transform of zero at zero frequency. Lobregt [7] chose k_i to be

$$k_i = \begin{cases} 1 & i = 0 \\ -\dfrac{1}{2} & i = 1 \; or \; -1 \\ 0 & otherwise \end{cases}$$

(3.44)

The magnitude of the internal force can therefore be expressed as

$$\left| f_{in,i} \right| = \left(c_i \bullet \hat{r}_i \right) * \widetilde{k}_i$$

(3.45)

where $\widetilde{k}_i$ is the periodic extension of k_i having period equals to the number of vertices. $\widetilde{k}_i$, rather than k_i, is used because the contour is closed, and therefore, the first point of the contour is adjacent to the last point and vice versa.

The direction of the internal force should be the same as that of the local curvature. Therefore,

$$f_{in,i} = \left| f_{in,i} \right| \hat{r}_i$$

(3.46)

As a result of using this definition of internal forces, the vertices of the contour having a regular polygonal shape shown in Figure 41(a) experience internal forces that is zero everywhere. Therefore, the shrinking problem has been solved.

In the next section, the external forces used in the UWO program will be discussed. The notion of an energy field that is closely related to the definition of the external force will be introduced.

B) External Forces

At this point of discussion, assume there is an external potential energy distribution E_{im}. This energy field represents the strength of some kind of image feature or combination of image features. But before we discuss the type of energy field used in the UWO program, we shall define how it is related to the external force.

The deformation process is implemented in such a way that the vertices will be pulled to the local maximum of the energy field E_{im}. Therefore, the external force should be defined in a way such that it is proportional to ∇E_{im}. Ladak [8] defined it by

$$f'_{ext} = \frac{2\nabla E_{im}}{max\left\| \nabla E_{im} \right\|}$$

(3.47)

where the factor $2/max\left\| \nabla E_{im} \right\|$ is included because it is more convenient to define the external force of a vertex in a way such that the range of its magnitude is from 0 to 2, equalling that of the internal force.

However, if one applies this definition of external force on the discrete dynamic contour model, he may find that the vertices on the contour are clustered in local maximum of the energy distribution. Since ideally the vertices on the contour should be equally spaced according to the resolution of the contour model, clustering of vertices is a very undesirable effect. To overcome this problem, Lobregt [7] proposed that only the locally radial component of f'_{ext} defined in (3.47) should be used to drive the vertices of the contour model:

$$f_{ext} = \left(f'_{ext} \bullet \hat{r}_i \right) \hat{r}_i$$

(3.48)
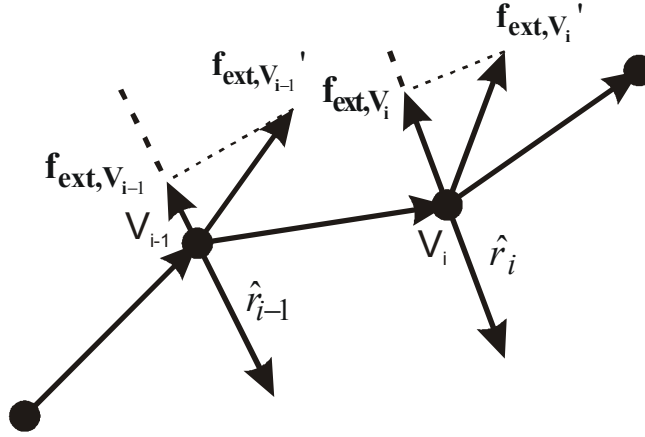
Figure 42   Only the locally radial component of f$_{ext}$' is used for deformation.

After discussing how the energy distribution E_{im} related to the external force, it is time now to discuss how this energy distribution is defined in the UWO program.

In the UWO program, the energy distribution E_{im} is defined by [8]

$$E_{im} = \left\| \nabla \left(G_\sigma * I \right) \right\|$$

(3.49)

where $*$ represents the two-dimensional convolution, I denotes the original image and G_σ denotes the Gaussian kernel having σ as the standard deviations in both the horizontal and the vertical directions.

The 2-D Gaussian kernel is separable. Therefore, it can be expressed as

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2}{2\sigma^2}\right)\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{y^2}{2\sigma^2}\right) = g_\sigma(x)g_\sigma(y)$$

(3.50)

where g_σ is used to denote a 1-D Gaussian function with standard deviation σ. With this separable property in mind, the computation of $G_\sigma * I$ can be decomposed to 1-D row-by-row convolutions, followed by column-by-column convolutions, as depicted in Figure 43.
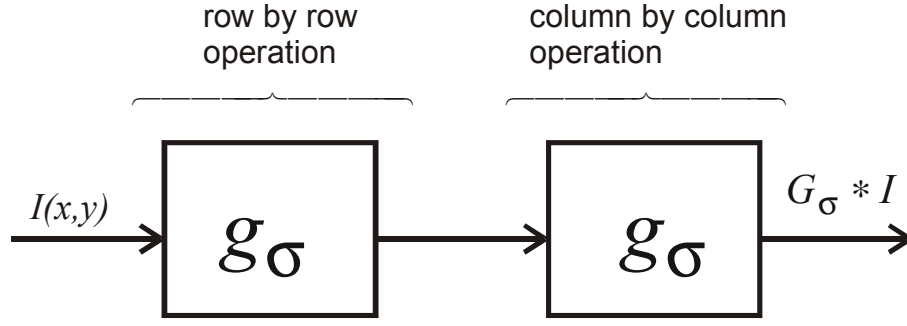
Figure 43   The 2-D convolution $G_\sigma * I$ can be decomposed to 1-D row-by-row convolutions, followed by column-by-column convolutions.

Although $g_\sigma(x)$ and $g_\sigma(y)$ are theoretically defined for all x and y in R respectively, they are represented only by a finite number of elements in a computer program. Therefore the result of the 2-D convolution obtained is only an approximation; the accuracy of which depends on the size of the Gaussian kernel, or equivalently, the length of the 1-D Gaussian functions $g_\sigma(x)$ and $g_\sigma(y)$.

In the UWO program, the sizes of $g_\sigma(x)$ and $g_\sigma(y)$ are specified by a parameter "*radiusfactor*". The function g_σ is represented by a vector $\underline{g}$, which has a length of $2 \times radiusfactor \times \sigma + 1$, defined by

$$\underline{g} = [g(-(radiusfactor \times \sigma)), \cdots\cdots, g(-1), g(0), g(1), \cdots\cdots, g(radiusfactor \times \sigma)]$$

$$(3.51)$$

The purpose of taking the convolution $G_\sigma * I$ is to smooth the image so that the effect of local variations, usually caused by noise, on the DDC model is made less significant. One would expect the magnitude of the gradient of this smoothed image is high at locations where edges appear. Therefore, by defining the energy distribution E_{im} according to (3.49), the external forces will pull the vertices towards the edge features of the image.

After having defined the internal and the external forces that drive the contour, it is suitable to describe the deformation dynamics of the contour model used in the UWO program. We will focus on this in the next section.

3.3.3. Deformation Dynamics

The total force f_i acting on a vertex is a weighted combination of the external force, the internal force and the damping force, described by

$$f_i = w_{ex} f_{ext,i} + w_{int} f_{in,i} + f_{damp,i}$$

$$(3.52)$$

where w_{ex} and w_{int} are user-defined relative weighting, normally chosen between 0 and 1. The damping force $f_{damp,i}$ is defined to be proportional to the velocity of the vertex v_i, that is,

$$f_{damp,i} = w_{damp} v_i$$

$$(3.53)$$

The weighting factor w_{damp} is negative. The damping force ensures the stability of the deformation process.

The position of each vertex V_i is calculated every Δt unit of time. Suppose the position, velocity and acceleration of the vertex V_i at a specific time t are denoted by $p_i(t), v_i(t)$ and $a_i(t)$ respectively. Then these quantities at $t + \Delta t$ are calculated using the following equations:

$$p_i(t + \Delta t) = p_i(t) + v_i(t)\Delta t,$$

$$(3.54)$$

$$v_i(t + \Delta t) = v_i(t) + a_i(t)\Delta t,$$

$$(3.55)$$

$$a_i(t + \Delta t) = \frac{1}{m_i} f_i(t + \Delta t).$$

$$(3.56)$$

The value of $f_i(t + \Delta t)$ has been calculated using (3.52) and (3.53). However, in the UWO program, the four initial vertices user defined are designated to be fixed. Therefore, the accelerations of those four vertices are always assigned a zero value. The value of m_i is the "mass" of the vertex in physical context. In this discrete dynamic contour model, the masses of all vertices are assumed to be the same. All vertices are assigned a mass of unity value in the UWO program and the time increment Δt has a value of one unit.

3.3.4. Resampling

In problem like prostate boundary detection in ultrasound image, the boundary we want to identify may not be clearly identifiable everywhere. In this situation, the best solution may be to define the boundary by joining two closest vertices. Therefore, it is important to define the distances between adjacent vertices so that the image feature between two neighbouring vertices does not affect the behaviour of the contour model significantly. However, since the position of the contour model is constantly updated, it is impossible to fix the distances between adjacent vertices. In order to keep this variation small, one could periodically resample the contour model along its path.

Since the four initial vertices are always fixed in the UWO program, special care has to be taken to ensure that the four vertices are not moved during the resampling process.

The process of resampling can be divided into four steps:

Step 1) Since the first vertex of the contour at any point of time is the first user-defined initial point, the resampling algorithm starts with inserting the first vertex of the input contour (i.e., the contour before resampling) into the output contour (i.e., the contour after resampling). The index of the first vertex, 0, is inserted into an index tracking array, *initial_point_index*.

Step 2) The positions of the remaining vertices on the output contour depend on the length of segments between adjacent vertices on the input contour. For each segment, points are sampled at an interval equals to the resolution of the contour. However, it is unlikely for the length of a segment to be equal to a multiple of the contour resolution. The "*residual*" is stored and added to the length of next segment. This is done to ensure the distance between the last point sampled in a certain segment and the first point sampled in next segment is approximately equal to the contour resolution. This process is illustrated in Figure 44.

Step 3) If the ending vertex of a certain segment on the input contour is one of the three remaining user-defined fixed vertices, it must be inserted into the output contour. Although it is impossible to set the distance between the last point sampled on a segment and the fixed point (called *d* hereafter) to be equal to the resolution of the contour, one should consider whether they are too close together. If *d* is smaller than a specific threshold, say one-third of the resolution, the resampling algorithm deletes the last point sampled and insert the fixed point into the output contour. On the other hand, if *d* is larger than one-third of the resolution, the last point sampled is kept. However, in this case, there is a possibility that the "*residual*" is larger than two-third of the resolution. As a consequence, if the "*residual*" is not changed, the distance between the first sampled point on the next segment and the fixed vertex will be small than one-third of the contour resolution. Therefore, the "*residual*" is reset to 0 in this case. In either case, the index of the fixed vertex is inserted into *initial_point_index*.

Step 4) It is known that the last segment starts at the last vertex and ends at the first vertex of the input contour. The first vertex has already been added to the output contour and does not need to be appended again. One only needs to consider whether the last point sampled on this segment is too close to the first vertex of the input contour. If this is the case, the last sampled point is simply not appended to the output contour.

(a)

(b)
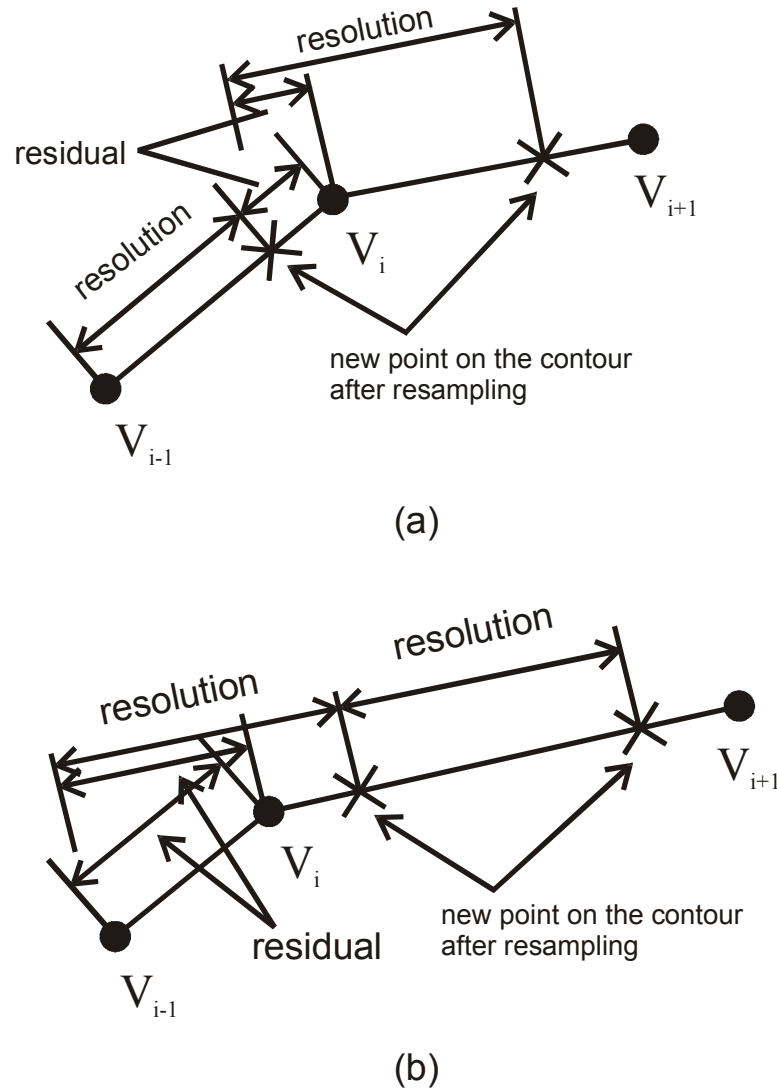
Figure 44   Two examples illustrating the effect of the resampling process: (a) The ith segment of the input contour is longer than the resolution of the contour model, resulting in a small residual; (b) the ith segment of the input contour is shorter than the resolution of the contour model, resulting in a residual that is comparable to the resolution. Note that in both cases, the residual after segment i-1 is processed is assumed to be zero.

3.3.5. Termination Criteria

During the deformation process, the position, velocity and acceleration of each vertex are calculated iteratively according to (3.54) to (3.56). After every iteration, the DDC is resampled using the procedure described in Section 3.3.4 and the algorithm also decides whether the iterations should stop. The program does this by picking the vertex with highest acceleration and one with highest velocity out of all vertices on the contour. The value of the maximum acceleration and velocity are then recorded and compared with the maximum acceleration and velocity recorded in the last iteration. If the difference between these two values of maximum acceleration is lower than a user-defined parameter, called the "convergence criteria" in the UWO program, and, at the same time, the difference between the two values of maximum velocity is lower than the same parameter, the program terminates.

3.4.   Summary and Remarks

This chapter starts by an introduction of segmentation techniques used in the field of image processing. Based on the pros and cons of each technique described in the introduction, the gradient-based technique, the deformable model technique and the threshold technique are chosen as building blocks of the proposed segmentation algorithm.

It is explained in Section 3.2.3 that the wavelet-based edge-detection technique developed by Mallat and Zhong [5] is equivalent to the conventional gradient-based technique that uses Gaussian filter as the smoothing filter. However, as pointed out at the beginning of this chapter, one can use different wavelet basis – either orthogonal or biorthogonal – rather than the cubic spline wavelet used by Mallat and Zhong for a better characterization of the edge features. In Section 4.2, different classes of wavelet that may be helpful in emphasizing the edge features are introduced. However, it must be emphasized that the primary goal of this thesis is to demonstrate the flexibility of the wavelet-based edge-detection formulation rather than finding a best basis for the segmentation problem. Therefore, rather than providing an analytical evaluation of different wavelet bases for the segmentation problem, the performance of different wavelet bases are judged according to the numerical results – that is to say that we will first apply different dyadic wavelet transforms, constructed using different wavelet bases, on the original images and evaluate the performances based on how close the final contour generated by the proposed algorithm, which will be introduced in Section 5.1, to the contour outlined by an expert observer.

In Section 3.3, the operating principles of the DDC model implemented by the University of Western Ontario are explained. This DDC model is used as a building block of the proposed algorithm.

Chapter 4

Adaptation of Mallat and Zhong's Algorithm and the DDC Model to the Prostate Boundary Detection Problem

In Chapter 3, two most important building blocks of our segmentation algorithm – Mallat and Zhong's multiscale edge detection algorithm and the DDC model – are introduced. In this chapter, several modifications of the Mallat-Zhong algorithm are proposed. In Section 4.1, we determine how one should approximate the modulus of the gradient of the smoothed image using the detailed coefficients obtained from (3.32) and (3.33). Because of the way the cubic spline wavelet was constructed as introduced in Section 3.2.3, the detailed coefficient must be shifted appropriately before the modulus could be calculated. In Section 4.2, the method used for constructing dyadic wavelet bases is introduced. This method is used to construct different wavelets that may produce better result in segmenting the prostate's boundary. In Section 5.3, the performance of the wavelets constructed in this section will be compared with that of the quadratic spline wavelet used by Mallat and Zhong. In Section 4.3, we will define an energy field, which can be used to drive the DDC model, based on the approximate coefficients generated in dyadic wavelet transform. This energy field provides information about the greyscale value near the boundary, which is not available in the energy field defined previously, based on the modulus of the gradient of the smoothed image.

Up to this point, our model depends very heavily on the initial points user enters. As mentioned in Chapter 1, an algorithm is designed to determine the initial points automatically. This initialization method is introduced in Section 4.4.

4.1. Shifting of Coefficients Obtained from Dyadic Wavelet Transform

Since the smoothing function corresponding to this type of wavelet is not centred at the origin, the coefficients generated must be shifted before they can be used in defining the gradient vector. Although Mallat briefly mentioned that the detailed coefficients obtained from (3.32) and (3.33) required to be shifted before the modulus is computed in (6.66) of [1][7], he didn't explain the underlying reason for the shift and he defined the amount of shift incorrectly. In Section 4.1.1, we will justify this claim and compute the correct shift required for accurate edge detection. In Section 4.1.2, we demonstrate that this incorrect shift results in a distorted edge map. To a certain extent, this blurring artifact has an adverse impact on the performance of the algorithm based on Mallat's dyadic wavelet transform [22]-[28].

---

[7] *The same amount of shift is also specified in (6.74) together with the equation*

$$M_{2^j} f = \sqrt{\left| d_j^1[m,n] \right|^2 + \left| d_j^2[m,n] \right|^2} \quad \text{in [2].}$$

4.1.1. The Correct Shift

(3.9) stated $d_j^1[m,n] = 2^j \dfrac{\partial}{\partial x}\left\{f * \theta_{2^j}\right\}[m,n]$. The purpose of computing d_j^1 or d_j^2 is to obtain the integer-sampled gradient vector of the function f smoothed by a smoothing function θ_{2^j}. However, if the smoothing function θ_{2^j} is not centred at the origin, undesired shifting results. If this is the case, this undesired shift should be compensated by a translation in the opposite direction when computing the gradient vector.

Before we could decide how much we should shift $\left\{W_{2^j}^k : k = 1,2\right\}$, or its discrete version $\left\{d_j^k : k = 1,2\right\}$, we must consider whether we should align to f or $A_1 f$. Since the scaling function Φ centres at $(1/2,1/2)$, $A_1 f$ is shifted by $-1/2$ in both x and y directions with respect to f and our decision on whether to align to f or $A_1 f$ will affect how we shift $\left\{W_{2^j}^k : k = 1,2\right\}$. Since we consider the discrete version of $A_1 f$ to be the original image, we should align to $A_1 f$ in theory. However, in discrete domain, we cannot shift by $1/2$, and therefore, it will not make a difference whether to align to f or $A_1 f$. In the following analysis, we will align $\left\{W_{2^j}^k : k = 1,2\right\}$ with respect to f.

From Figure 34, it can be observed that both φ and θ are centred at $1/2$. Then, according to (3.29), $\theta^1(x,y)$ centres at $(1/2,1/4)$, which implies its dilation $\theta_{2^j}^1$ centres at $\left(2^{j-1},2^{j-2}\right)$. Since $W_{2^j}^1 f$ is defined by (3.6), it is shifted -2^{j-1} in the x direction and -2^{j-2} in the y direction with respect to f. Repeating this analysis, we find that $W_{2^j}^2 f$ is shifted -2^{j-2} in the x direction and -2^{j-1} in the y direction with respect to f. $\left\{d_j^k : k = 1,2\right\}$, the discrete version of $\left\{W_{2^j}^k : k = 1,2\right\}$, should be shifted to cancel these shifting effect before the modulus of the gradient vector is computed, i.e.,

$$M_j f[m,n] = \sqrt{\left|d_j^1\left[m - 2^{j-1}, n - 2^{j-2}\right]\right|^2 + \left|d_j^2\left[m - 2^{j-2}, n - 2^{j-1}\right]\right|^2}$$

(4.1)

However, when we shift d_j^1 and d_j^2 according to (4.1), we find that when $j = 1$, d_j^1 and d_j^2 are required to shift $1/2$, in the y direction and x direction respectively, which is not possible in the discrete domain. In this case, the solution would be not to shift. (4.1) is modified to be

$$M_j f[m,n] = \begin{cases} \sqrt{\left| d_1^1[m-1,n] \right|^2 + \left| d_1^2[m,n-1] \right|^2} & \text{for } j = 1 \\ \sqrt{\left| d_j^1\left[m-2^{j-1},n-2^{j-2}\right] \right|^2 + \left| d_j^2\left[m-2^{j-2},n-2^{j-1}\right] \right|^2} & \text{otherwise} \end{cases}$$

(4.2)

4.1.2. Effects of Mallat's Inappropriate Shift

In this section, we analyse the effect on the gradient modulus, $M_{2^j}f$, when Mallat's shifting scheme stated in (6.66) of [1] is applied.

We first look at (3.9) and the smoothing functions used in the fast dyadic transform, $\underline{\theta}^1$ and $\underline{\theta}^2$. Two implicit assumptions had been made for $\theta^1_{2^j}$ and $\theta^2_{2^j}$ in (3.9): First, there was no distinction made between $\theta^1_{2^j}$ and $\theta^2_{2^j}$ in (3.9) because they should be equal in theory, that is, the image should be smoothed by convolving with the same smoothing function before the partial derivatives in different directions are taken. The distinction between $\theta^1_{2^j}$ and $\theta^2_{2^j}$ was made when constructing the fast dyadic wavelet transform, because the structure of which does not allow them to be equal. However, if the shifting scheme introduced in Section 4.1.1 is applied, $\theta^1_{2^j}$ and $\theta^2_{2^j}$ are numerically close to be considered as equal as shown in Figure 45. Second, $\theta^1_{2^j}$ and $\theta^2_{2^j}$ is centred at the origin, so that the smoothed images $\left\{ f * \theta^1_{2^j} \right\}$ or $\left\{ f * \theta^2_{2^j} \right\}$ would not be inappropriately shifted. This assumption will be satisfied if the shifting compensation introduced in Section 4.1.1 is applied.

If (6.66) of [1] is followed, we pointed out very clearly in Section 4.1.1 that the second assumption would not be satisfied. However, if only the second assumption is violated, while both $\underline{\theta}^1$ and $\underline{\theta}^2$ are centred at the same location, the edge pattern represented by $M_{2^j}f$ will only be shifted – but without any change – in an amount determined by the location of the centre. In (6.66) of [1], Mallat made compensating shifts in a way such that $\underline{\theta}^1$ is centred at $(0,-1/4)$, and $\underline{\theta}^2$ is centred at $(-1/4,0)$. At the location of a certain pixel, say $[m,n]$, the smoothed functions $\left\{ f * \theta^1_{2^j} \right\}$ and $\left\{ f * \theta^2_{2^j} \right\}$ represents averages taken in two different neighbourhoods: $\left\{ f * \theta^1_{2^j} \right\}[m,n]$ represents the average of the function f in a scale-2^j neighbourhood centred at $\left[m,n+2^{j-2}\right]$, whereas $\left\{ f * \theta^2_{2^j} \right\}[m,n]$ represents the average of the function f in a scale-2^j neighbourhood centred at $\left[m+2^{j-2},n\right]$. Therefore, the vector defined in (3.9) is not a gradient vector in any neighbourhood, and the modulus of this gradient vector would represent a distorted edge map. The significance of this

distortion increases as the scale increases. Figure 46 shows the effect of the distortion. The scale-2^5 and scale-2^6 gradient modulus obtained using the shifting scheme proposed in Section 4.1.1 are illustrated in Figure 46(b) and (d) respectively, while those obtained using Mallat's shifting scheme are illustrated in Figure 46(c) and (e). Note that the vertical edges of the square are shifted upwards, by 2^{j-2} pixels in theory, while the horizontal edges are shifted to the left by 2^{j-2} pixels, where 2^j denotes the scale at which the gradient modulus is obtained (i.e., $j = 5$ in Figure 46(b) and (c), and $j = 6$ in Figure 46(d) and (e).)

(a)   **(b)**

(c)
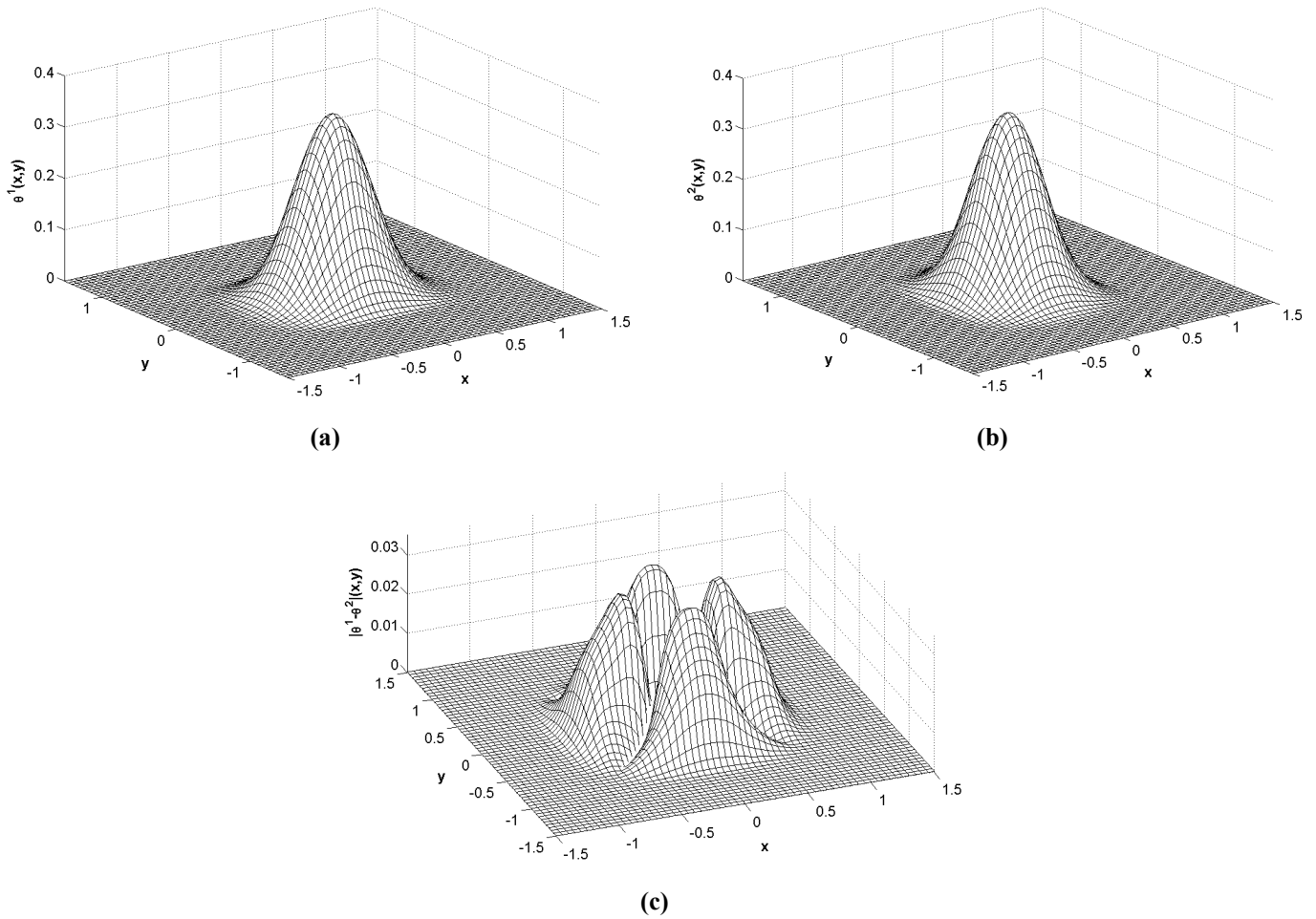
Figure 45   The origin-centred θ^1 and θ^2 and the difference between them: (a) θ^1, (b) θ^2, and (c) $\left|\underline{\theta}^1 - \underline{\theta}^2\right|$. Note that the maximum absolute difference between θ^1 and θ^2 is about 0.03 – about 10% of the peak value of θ^1 or θ^2. In a first approximation, θ^1 and θ^2 can be considered to be equal.
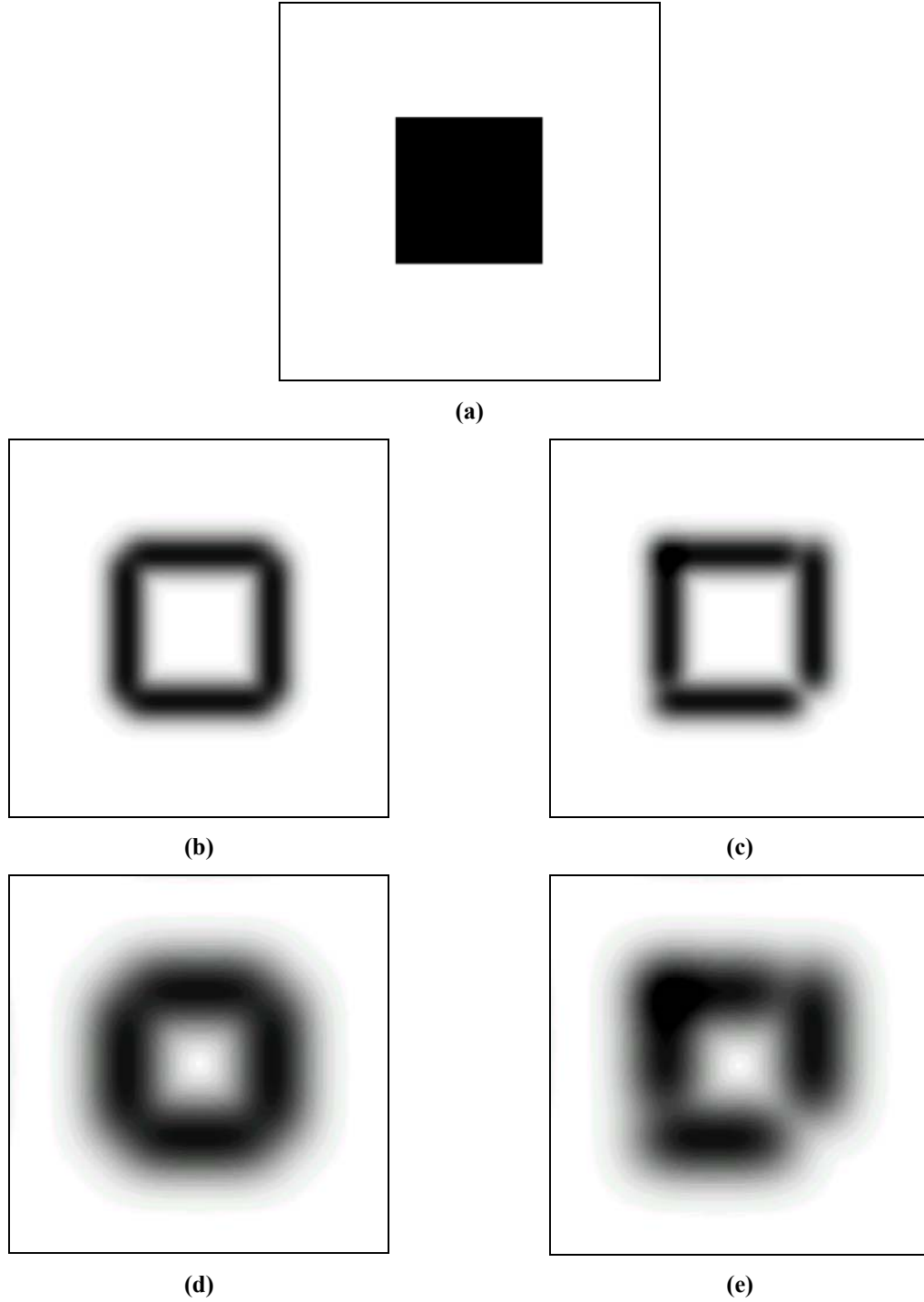
**Figure 46   Comparison between the shifting scheme introduced in Section 4.1.1 and Mallat's shifting
scheme [1] using a synthetic square image: (a) Original image; (b), (d) the gradient modulus at scale
2^5 and 2^6 obtained using the shifting scheme introduced in Section 4.1.1; (c), (e) the gradient modulus
at scale 2^5 and 2^6 obtained using Mallat's scheme [1]**

4.2.   Wavelet Construction for Dyadic Wavelet Transform

Mallat studied the properties of scaling and wavelet functions used in dyadic wavelet transform in [1]. He then made use of these properties to construct a family of wavelets called the spline wavelets. In this section, we first state his results and introduce the method of constructing the spline wavelets. Using a similar method, we found that other dyadic wavelets can be constructed. In this section, we adapt existing wavelet bases, such as Symlets and Coiflets, to the dyadic wavelet transform framework.

4.2.1.   Properties of Wavelets in Dyadic Wavelet Transform

Before describing the properties of the wavelet and scaling functions, we need to step back and briefly mention about the continuous version of the dyadic wavelet transform from which Mallat derived his discrete dyadic wavelet transform, which we have discussed on Section 3.2.3.

The continuous dyadic wavelet transform is defined by

$$W_{2^j} f(u) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{2^j}} \psi\left(\frac{t-u}{2^j}\right) dt = f * \underline{\psi_{2^j}}(u)$$

(4.3)

Mallat proved in Theorem 5.11 of [1] that in order for reconstruction to be possible from the dyadic wavelet transform, a wavelet function ψ must have a corresponding reconstruction wavelet function $\widetilde{\psi}$ such that

$$\sum_{j=-\infty}^{\infty} \overline{\psi^F}\left(2^j \omega\right) \widetilde{\psi}^F\left(2^j \omega\right) = 1 \quad \text{for} \quad \omega \in R - \{0\}$$

(4.4)

If (4.4) is satisfied, the original function f can be reconstructed from its dyadic wavelet transform:

$$f(t) = \sum_{j=-\infty}^{\infty} \frac{1}{2^j} W_{2^j} f * \widetilde{\psi}_{2^j}(t)$$

(4.5)

A scaling function φ and a reconstruction scaling function $\widetilde{\varphi}$ are defined such that a multiresolution ladder can be maintained. Mathematically, φ and $\widetilde{\varphi}$ are defined such that they have the following relationship with the wavelets ψ and $\widetilde{\psi}$:

$$\overline{\varphi^F}(\omega)\widetilde{\varphi}^F(\omega) + \overline{\psi^F}(\omega)\widetilde{\psi}^F(\omega) = \overline{\varphi^F}(\omega/2)\widetilde{\varphi}^F(\omega/2)$$

(4.6)

It can be proven (4.6) leads to a multiresolution relationship:

$$P_k f + Q_k f = P_{k-1} f$$

(4.7)

where $P_k f$ and $Q_k f$ are defined by

$$P_k f = \frac{1}{2^k} A_{2^k} f * \tilde{\varphi}_{2^k} \quad \text{and} \quad Q_k f = \frac{1}{2^k} W_{2^k} f * \tilde{\psi}_{2^j}$$

(4.8)

In (4.8), $A_{2^k} f(u) = \left\langle f(t), \frac{1}{\sqrt{2^k}} \varphi\left(\frac{t-u}{2^k}\right) \right\rangle$, whereas $W_{2^j} f$ is defined according to (4.3).

Suppose the filters $h, g, \tilde{h}$ and $\tilde{g}$ are defined by

$$\varphi^F(\omega) = \frac{1}{\sqrt{2}} h^f\left(\frac{\omega}{2}\right)\varphi^F\left(\frac{\omega}{2}\right), \quad \psi^F(\omega) = \frac{1}{\sqrt{2}} g^f\left(\frac{\omega}{2}\right)\varphi^F\left(\frac{\omega}{2}\right),$$

$$\tilde{\varphi}^F(\omega) = \frac{1}{\sqrt{2}} \tilde{h}^f\left(\frac{\omega}{2}\right)\tilde{\varphi}^F\left(\frac{\omega}{2}\right) \quad \text{and} \quad \tilde{\psi}^F(\omega) = \frac{1}{\sqrt{2}} \tilde{g}^f\left(\frac{\omega}{2}\right)\tilde{\varphi}^F\left(\frac{\omega}{2}\right).$$

(4.9)

It has been verified by Mallat [1] that (4.4) and (4.6) imply

$$\tilde{h}^f(\theta)\overline{h^f(\theta)} + \tilde{g}^f(\theta)\overline{g^f(\theta)} = 2 \quad \text{for all } \theta \in R$$

(4.10)

In the following, we will demonstrate how the spline dyadic wavelets are derived using (4.9) and (4.10).

4.2.2. The Spline Dyadic Wavelets

In constructing the spline dyadic wavelets, Mallat [1] started by defining the scaling function to be the box spline function:

$$\varphi^F(\omega) = e^{-j\frac{\varepsilon\omega}{2}}\left(sinc\left(\frac{\omega}{2\pi}\right)\right)^{m+1} \quad \text{with} \quad \varepsilon = \begin{cases} 1 & \text{if } m \text{ is even} \\ 0 & \text{if } m \text{ is odd} \end{cases}.$$

(4.11)

With the scaling function φ defined, the scaling filter *h* can be easily computed according to (4.9):

$$h^f(\omega) = \sqrt{2}\,\frac{\varphi^F(2\omega)}{\varphi^F(\omega)} = \sqrt{2}e^{-j\frac{\varepsilon\omega}{2}}\left(cos\frac{\omega}{2}\right)^{m+1}$$

(4.12)

Table 6 in Section A.2 of the Appendices gives the coefficients of *h* for order 0 to 7.

We then progress on finding the wavelet filter *g*. Apparently, it can be defined arbitrary as long as (4.10) is satisfied. However, it has been pointed out in Section 3.2.1 that we want to define the wavelet function ψ according to (3.1):

$$\psi(x) = -\frac{d\theta(x)}{dx}$$

(3.1)

where $\theta(x)$ is a smoothing function that converges to 0 at infinity and it has an integral that equals to a positive value.

Mallat proved [1] that a wavelet has *n* vanishing moments if and only if there exists a smoothing function $\theta(x)$ such that

$$\psi(x) = (-1)^n\frac{d^n\theta(x)}{dx^n}$$

(4.13)

Moreover, $\psi(x)$ has no more than *n* vanishing moments if and only if $\int_{-\infty}^{\infty}\theta(x)dx \neq 0$. Therefore, in order for (3.1) to be satisfied, $\psi(x)$ must have exactly one vanishing moment. It can be proven that [1] the number of vanishing moments of ψ is equal to the number of zeroes of $\psi^F(\omega)$ at $\omega = 0$. Since $\varphi^F(0) = 1$, $g^f(\omega)$ must have the same number of zeroes at $\omega = 0$ as $\psi^F(\omega)$ according to the second equation in (4.9). As a result, $g^f(\theta)$ must have exactly one zero at $\theta = 0$ in our case and it could be chosen to be

$$g^f(\theta) = -j\sqrt{2}\,e^{-j\frac{\theta}{2}}\,sin\left(\frac{\theta}{2}\right)$$

(4.14)

The Fourier transform of the resulting wavelet is

$$\psi^F(\omega) = \frac{1}{\sqrt{2}} g^f\left(\frac{\omega}{2}\right)\varphi^F\left(\frac{\omega}{2}\right) = -\frac{j\omega}{4} e^{-j\omega\left(\frac{1+\varepsilon}{4}\right)}\left(sinc\left(\frac{\omega}{4\pi}\right)\right)^{m+2}$$

(4.15)

The reconstruction filters $\widetilde{g}$ is obtained by first choosing $\widetilde{h} = h$ and then applying the reconstruction condition (4.10). However, for edge detection, we are only interested in finding the dyadic wavelet transform of the original image and we are not planning to reconstruct the original image from its dyadic wavelet transform. Therefore, in the derivations that will be presented in Section 4.2.3, we will omit the computation of $\widetilde{h}$ and $\widetilde{g}$.

In Section 4.1, we mentioned that the dyadic wavelet transform coefficients must be appropriately shifted before the modulus of the gradient vector is computed. Noticing the smoothing function θ is centred at $(\varepsilon+1)/4$ from (3.1) and (4.15), and repeating the analysis in Section 4.1, it can be found that, for the spline dyadic wavelets family, the modulus of the gradient vector at scale 2^j should be defined by

$$M_j f[m,n] = \sqrt{\left|d_j^1\left[m-2^{j-2}(\varepsilon+1), n-2^{j-2}\varepsilon\right]\right|^2 + \left|d_j^2\left[m-2^{j-2}\varepsilon, n-2^{j-2}(\varepsilon+1)\right]\right|^2}$$

(4.16)

Note that the problem of shifting 1/2 pixel still exists when $j=1$ and $\varepsilon=1$. In this case, the solution we introduced in Section 4.1 will be used.

4.2.3. Dyadic Wavelets based on Daubechies' Symlets and Coiflet

Section 4.2.2 shows how Mallat constructed the spline dyadic wavelets family. He first chose a scaling function φ, from which he can easily obtain the scaling filter h. For his multiscale edge detection algorithm, we pointed out that the wavelet ψ must have exactly one vanishing moment. The simplest wavelet filter g that leads to a wavelet function ψ having one vanishing moment is the one he chose in (4.14). The wavelet function ψ can be found after g is obtained as in (4.15). Finally, to determine how the coefficients of the dyadic wavelet transform should be shifted, he also need the smoothing function θ.

His path is followed when we attempt to develop another dyadic wavelets family. The first step is to choose a scaling function. Different families of scaling functions φ and the corresponding scaling filters h are readily available in MATLAB. To choose the scaling function that fits into the edge detection algorithm framework, the first thing to consider is the symmetry of the scaling functions. Symmetry is important because the two-

dimensional smoothing functions have a direct relationship with the scaling function (see
(3.29) and (3.30)) and the smoothing function is expected to be symmetric about the
centre of its support. Compact support would be another desired property of the scaling
function because, as discussed in Section 3.1.2, Part (c), the smoothing function should
be localized enough so that the position of the edge can be identified with reasonable
accuracy.

With these considerations in mind, we have chosen two families of scaling functions as
our starting point: The Symlets and Coiflets. Apart from having the symmetry and
compact support properties, Coiflets have an additional property that the scaling function
of order N Coiflet has $2N-1$ vanishing moments. It can be expected that, the dyadic
wavelet transform defined by this family of scaling function will produce a set of wavelet
coefficients that have strong contrast between smooth region and region with edge
features. However, this statement is not verified until Section 5.3, at which point we will
present our results.

As a demonstration on how we construct our new dyadic wavelets, we will start with the
scaling function corresponds to Symlet-4.

The scaling function φ of Symlet-4 is plotted in Figure 47. Although there is no analytic
expression for this function, we know that scaling filter h that leads to this scaling
function. The scaling filter coefficients of the 4th to 8th order Symlets are tabulated in
Table 7 of the Appendices, Section A.2.
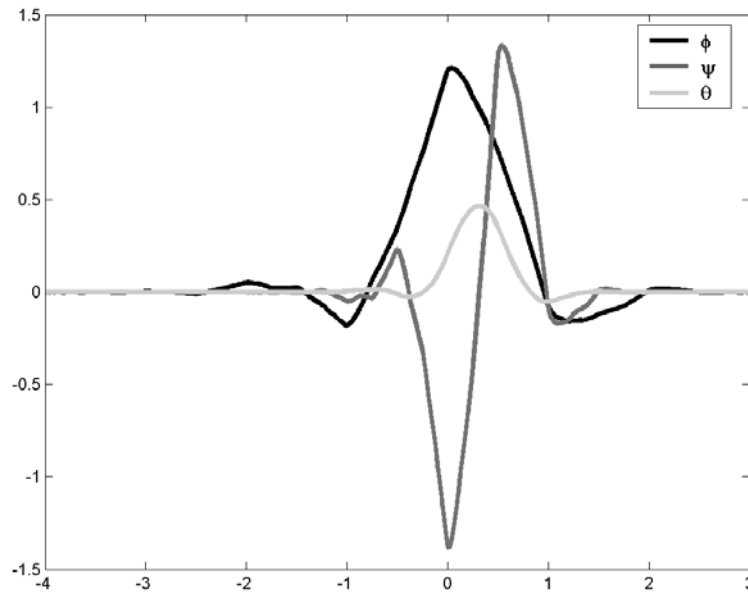
Figure 47 φ, ψ and θ of the dyadic wavelets constructed based on Symlet-4

The next step is to find the wavelet filter g. Recall that we need a wavelet filter that has exactly one zero at $\theta = 0$ and the simplest filter that have this property is the one shown in (4.14). We will use this filter as our wavelet filter to calculate our wavelet function ψ. Using the wavelet filter coefficients shown in Table 1 and the relation between the wavelet function ψ and the wavelet filter g in (3.18), we get

$$\psi(x) = \sum_{k \in Z} g[k]\sqrt{2}\varphi(2x - k) = -\varphi(2x) + \varphi(2x - 1),$$

(4.17)

which is plotted in Figure 47. The smoothing function θ can be obtained using (3.1) and it is also plotted in Figure 47.

Finally, we need to decide the amount of shift we should make on the coefficients resulting from the dyadic wavelet transform. Since φ can always be translated to the origin, the shifting due to θ is the only concern. From Figure 47, we know that θ is centred at 1/4. Repeating the analysis in Section 4.1, $\underline{\theta^1_{2^j}}(x, y)$ is centred at $\left(-2^j(1/4), 0\right)$ and $\underline{\theta^2_{2^j}}(x, y)$ is centred at $\left(0, -2^j(1/4)\right)$. Therefore, the modulus of the gradient vector at scale 2^j is defined by

$$M_j f[m, n] = \sqrt{\left| d_j^1\left[m - 2^{j-2}, n\right] \right|^2 + \left| d_j^2\left[m, n - 2^{j-2}\right] \right|^2}$$

(4.18)

Similar to the problem we faced in Section 4.1, we are required to shift d_j^1 and d_j^2 by 1/2 when $j = 1$. In this case, we can also choose not to shift and (4.18) is modified to be

$$M_j f[m, n] = \begin{cases} \sqrt{\left| d_1^1[m, n] \right|^2 + \left| d_1^2[m, n] \right|^2} & \text{for } j = 1 \\ \sqrt{\left| d_j^1\left[m - 2^{j-2}, n\right] \right|^2 + \left| d_j^2\left[m, n - 2^{j-2}\right] \right|^2} & \text{otherwise} \end{cases}$$

(4.19)

Although the amount of shifts for d_j^1 and d_j^2 are computed for Symlet-4, (4.19) is applicable for Symlets and Coiflets of all order. It is because all scaling functions can be translated so that it centres at the origin, and the definition of ψ in (4.17) and the fact that φ is approximately symmetric about the origin implies that θ, computed using (3.1), must be centred at 1/4. The construction of dyadic wavelets based on other Symlets or Coiflets is similar and not to be repeated here.

4.3.   Definition of an Energy Field based on Approximate Coefficients

This section introduces an energy field that will be used in the proposed algorithm. This energy field is defined based on the approximated coefficients of the dyadic wavelet transform.

In a typical prostate ultrasound image, the interior of the prostrate is relatively dark (i.e., the greyscale value is relatively low), while the exterior is relative bright (i.e., the greyscale value is relatively high). It can be expected that, in the smoothed image generated using dyadic wavelet transform, where the greyscale value of a pixel near the edge is computed by averaging the pixel value of the original image in its neighbourhood, the greyscale value near the edge is between the two extremes. Assuming the four initial points a user enters are reasonably close to the edge, the greyscale value of these four pixels on the smoothed image will be very useful in defining the boundary. Specifically, a set of pixels on the smoothed image that has greyscale values close to a weighted mean of these four values approximately define the boundary of the prostate. Therefore, for the DDC model, one can define an energy field that penalize the difference between this weight mean and the greyscale value of pixels on the smoothed image a_j, represented by the set of approximate coefficients generated using dyadic wavelet transform at the scale 2^j. For example, one can define this energy field, denoted by E_{im1}, as

$$E_{im1}[m,n] = -\left| a_j[m,n] - f\left(\left\{a_j[x_n, y_n] : n = 1,2,3,4\right\}\right) \right|$$

(4.20)

where $f(.)$ takes some kind of weighted mean of its argument and $[x_n, y_n]$ denotes the coordinates of the user-defined initial point n. The negative of the absolute difference is taken because the DDC model moves to the peak of the energy field.

To completely define the energy field E_{im1}, we are required to choose the scale 2^j, at which the smoothed image is obtained, and how the weighted mean is defined. Recall that the scale 2^j should be large enough so that the small-scaled variations, mostly caused by noise, are suppressed, and it should be small enough so that the outline of the prostate is not too coarse. By experiment, it is found that $j = 4$ is appropriate for our prostate boundary detection problem. Figure 48 shows an ultrasound image smoothed at scale 2^4. On the other hand, f could be defined so that it takes the arithmetic mean of its arguments. Figure 49 shows the plot of $-E_{im1}$ (i.e., the absolute difference defined in (4.20)).

Figure 49 shows that the energy field E_{im1} defines the prostate boundary quite well, except near the bottom of the prostate. This effect can be explained by the fact that we

choose the arithmetic mean of the greyscale value at the four initial points to define the boundary everywhere on the image. By observing Figure 48, we can observe that the grey-level of the smoothed image at the initial point at the bottom is much higher than that at other initial points. Therefore, it is unwise to choose a single greyscale value to define the boundary and the definition of the weighted mean, denoted by $f(.)$ previously, should be localized. A reasonable solution would be to define $f(.)$ so that at location $[x, y]$ of the smoothed image, we only take the weighted mean of the greyscale value at the two user-defined initial points nearest to $[x, y]$:

$$f_{[x,y]}\left(\left\{a_j[x_n, y_n] : n = 1,2,3,4\right\}\right) = w_1 a_j[x_{c1}, y_{c1}] + w_2 a_j[x_{c2}, y_{c2}]$$

(4.21)

where $[x_{c1}, y_{c1}]$ and $[x_{c2}, y_{c2}]$ are the two points closest to $[x, y]$ chosen from the four initial points $\{[x_n, y_n] : n = 1,2,3,4\}$. The weight w_1 and w_2 are defined according to the distance between $[x, y]$ and the corresponding initial points:

$$w_1 = 1 - \frac{dist([x_{c1}, y_{c1}], [x, y])}{dist([x_{c1}, y_{c1}], [x, y]) + dist([x_{c2}, y_{c2}], [x, y])},$$

$$w_2 = 1 - \frac{dist([x_{c2}, y_{c2}], [x, y])}{dist([x_{c1}, y_{c1}], [x, y]) + dist([x_{c2}, y_{c2}], [x, y])}$$

(4.22)

where $dist([x, y], [x', y'])$ defines the distance between the points $[x, y]$ and $[x', y']$.

With this definition of the weight mean, the energy field E_{im1} can be computed according to (4.21). Figure 50 shows the plot of $-E_{im1}$. The abrupt change appears in this energy field comes from the fact that a different set of two points is selected to calculate the weight mean at different locations. It can be observed that the prostate boundary is much better defined by this modified energy field.
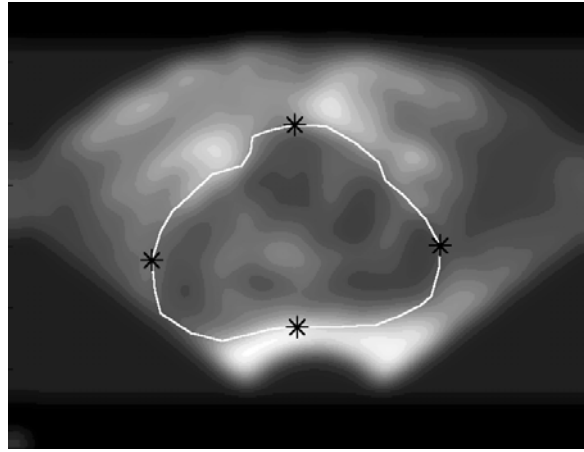
Figure 48   An ultrasound prostate image smoothed at a scale of 2^4. The four user-defined initial points is marked with the asterisk (∗). The contour on the background is the manual outline of the prostate.
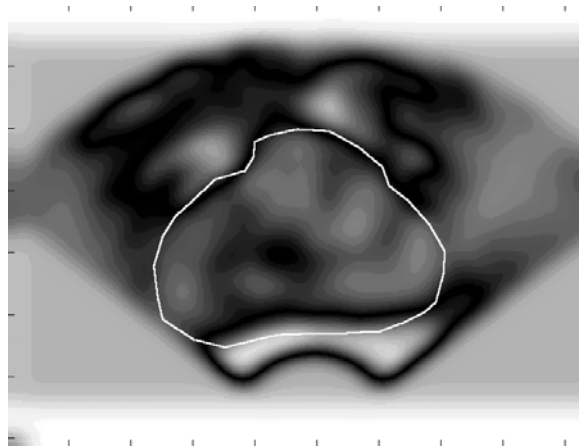
Figure 49   The energy field -E_{im1} is plotted along with the prostate boundary drawn by an expert observer.
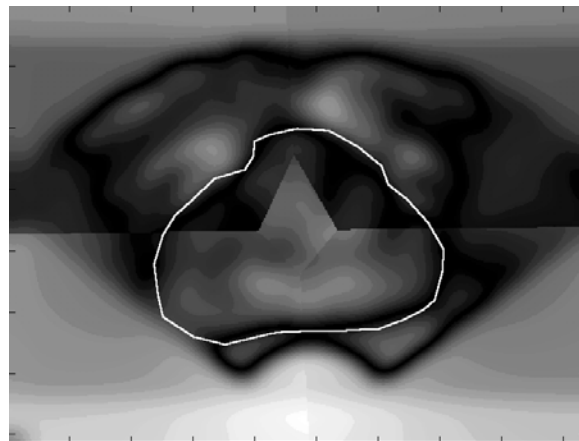
Figure 50   The modified energy field -E_{im1} and the prostate boundary drawn by an expert observer are plotted.

4.4.   An Automatic Initialization Method

Up to this point, we focus on introducing the building blocks of a semi-automatic segmentation algorithm that requires an expert observer's supervision in the initialization process. However, as mentioned in Chapter 1, we are only one step away from a fully-automatic segmentation algorithm. This step is to find a fully-automatic initialization algorithm. In this section, such an algorithm is introduced.

Figure 51 shows the block diagram of the proposed initialization method. In our semi-automatic segmentation algorithm, the user is required to enter four points – two of which lie on the vertical line that passes through the centre of the prostate and the other two lie on the horizontal line that passes through the centre. In this algorithm, the vertical line (or the centre column) is first identified using the procedure that will be introduced in Section 4.4.2. After that, two initial points on the centre column will be identified according to smoothed image and the gradient modulus obtained previously using the dyadic wavelet transform. This process is described in Section 4.4.1. At the same time, strong edges on the image will be identified by locating points where the gradient modulus is greater than a certain threshold. Since the border of the ultrasound image is known a priori and the boundary of the prostate is not likely to be very close to the border of the ultrasound image according to our observation, strong edge points close to the border of the image is taken out. The location of the strong edges and the two initial points identified are the essential information in the global and local initialization process, which will be described in detail in Section 4.4.4 and 4.4.5. Finally, the available initial points are interpolated according to the spline function. This process is similar to that described in Section 3.3.1.
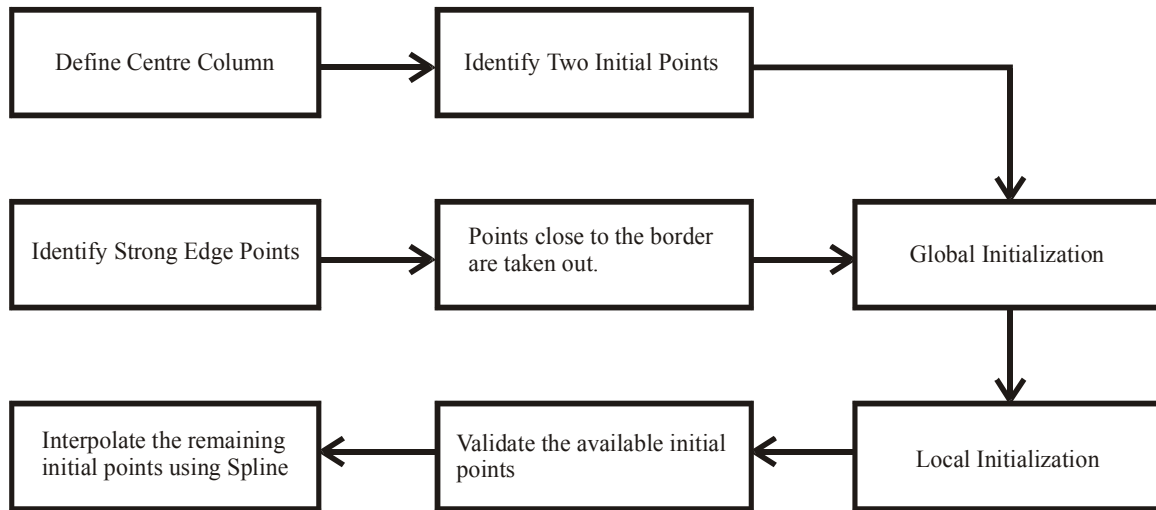
Figure 51   The block diagram of the proposed initialization method

4.4.1. Identification of Two Initial Points

The reason we choose to discuss the method used to define the two initial points first, rather than explaining our algorithm sequentially, is that this method will play a major role in the definition of the centre column.

At this point, the centre column is assumed to be given. Figure 52(a) shows a prostate ultrasound image and the centre column, whereas Figure 52(b) shows the intensity profile of the centre column in the smoothed image. Usually, the centre of the prostate is located where the intensity profile of the centre column attains its local minimum. In our algorithm, we identify the smallest local minimum and take this point to be the estimate of the centre of the prostate, which is represented by the asterisk (*) in Figure 52(b).

After identifying the centre, we locate two "significant" local maximum points closest to the centre. The reason behind this procedure is that the prostate boundary is normally characterized by a sharp increase in the light intensity (see Figure 52(a)). We define a "significant" point as a point that has value greater than a certain threshold. In our algorithm, this threshold is chosen to be the 90% percentile of the greyscale values in the centre column. We must choose one point higher and the other point lower than the centre. If there is no point that is greater than the threshold either on the upper side or the lower side of the centre, the greatest local maximum point is chosen. In Figure 52(b), the threshold is represented by the vertical line and the two local maximum points are represented by the crosses (×).

(a)   (b)
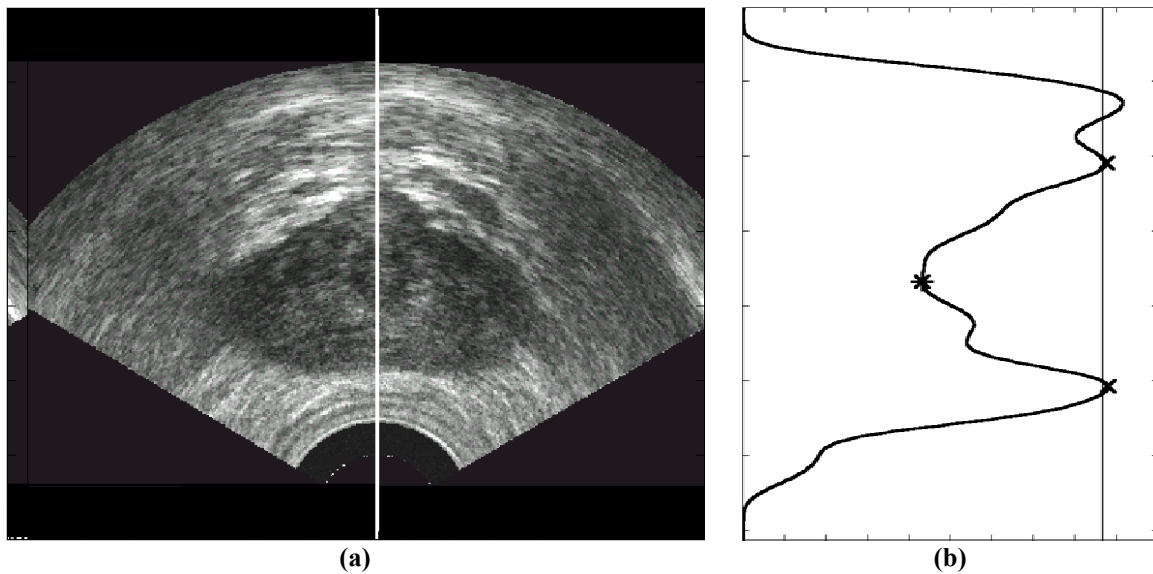
**Figure 52   (a) A prostate ultrasound image with the centre column represented by the line; (b) The
intensity profile of the centre column in the smoothed image**
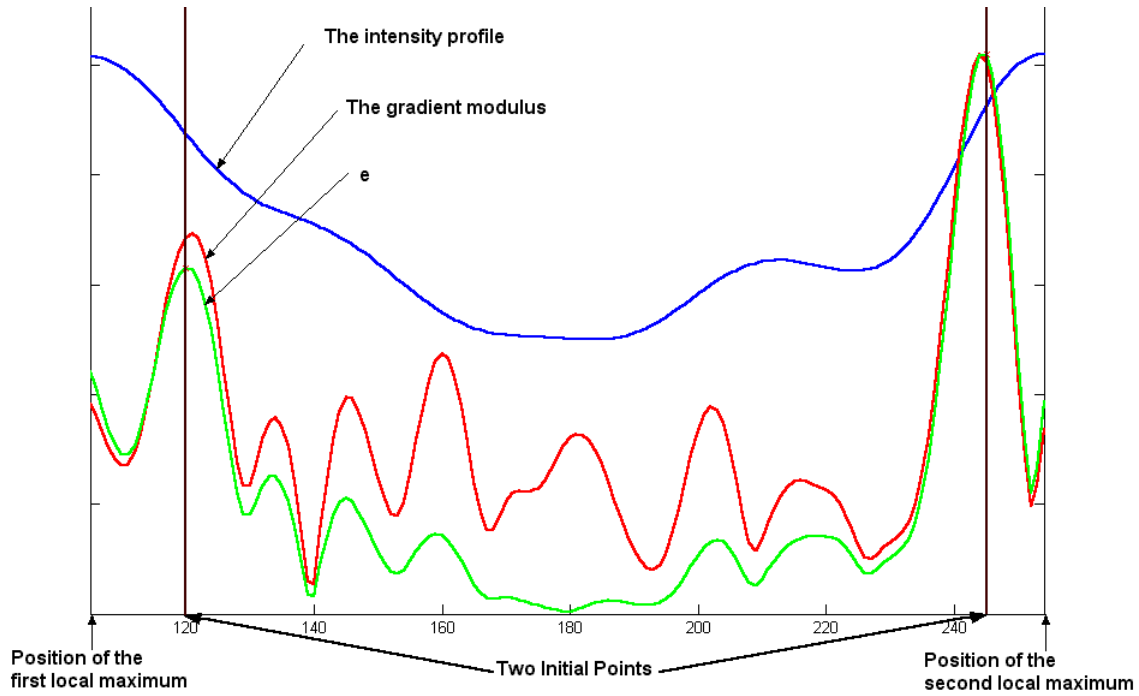
**Figure 53 The intensity profile, the gradient modulus and the function *e* from the position of the first
to the second local maximum of the intensity profile (Note: The plot is scaled so that all
three functions have the same maximum.)**

After identifying the two local maximum points, we restrict our attention to the interval
from the first to the second local maximum points. We try to find two points that are
reasonably close to the two local maximum points and have a high value of gradient
modulus. To achieve this, we first define a function *e*:

$$e(x) = G(x)|x - c|$$

(4.23)

where G is the gradient modulus and c is the midpoint between two local maximum
points. This function is plotted in Figure 53.

The two initial points are determined using the following rule:

$$x_1 = arg\left[\max_{x \in [a, \lfloor c \rfloor]}\{e(x)\}\right],$$
$$x_2 = arg\left[\max_{x \in [\lfloor c \rfloor + 1, b]}\{e(x)\}\right]$$

(4.24)

where the first and the second initial points are denoted by *a* and *b* respectively. These
two initial points are plotted in Figure 54.
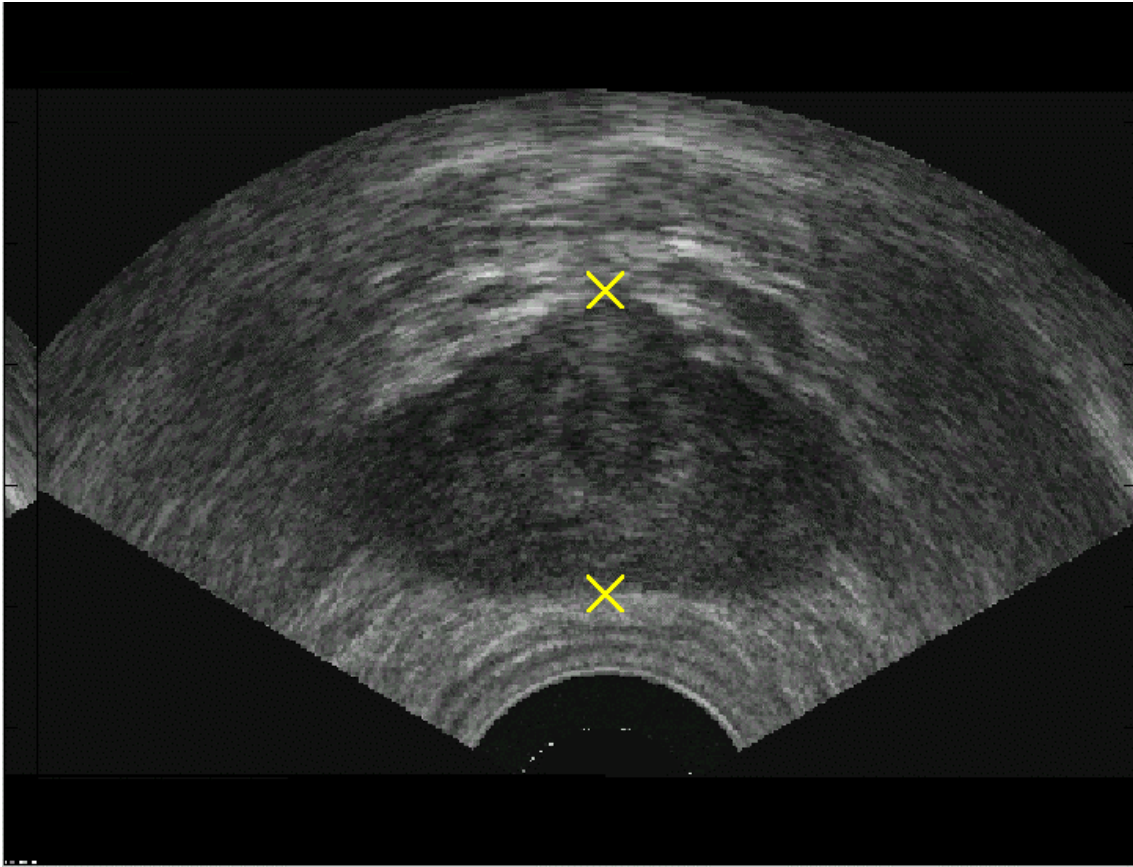
Figure 54   The two automatically identified initial points

4.4.2.   Definition of the Centre Column

As far as defining the centre column is concerned, images can be classified in two
categories. The centre column is easily identified in one class and more difficult in the
other.

Case 1: Easy

Figure 54 shows an example of this case. One can observe that the light intensity is very
high around the centre of the prostate. We can take advantage of this property and define
the centre column using the following algorithm:

Step 1) Obtain the smoothed image using the dyadic wavelet transform described in
Section 3.2.3, Part B. (see Figure 55(a))

Step 2) Take the average greyscale value for each column. (see Figure 55(b)).

Smoothed Image

(a)   (b)   (c)

Midpoint

132   249   366
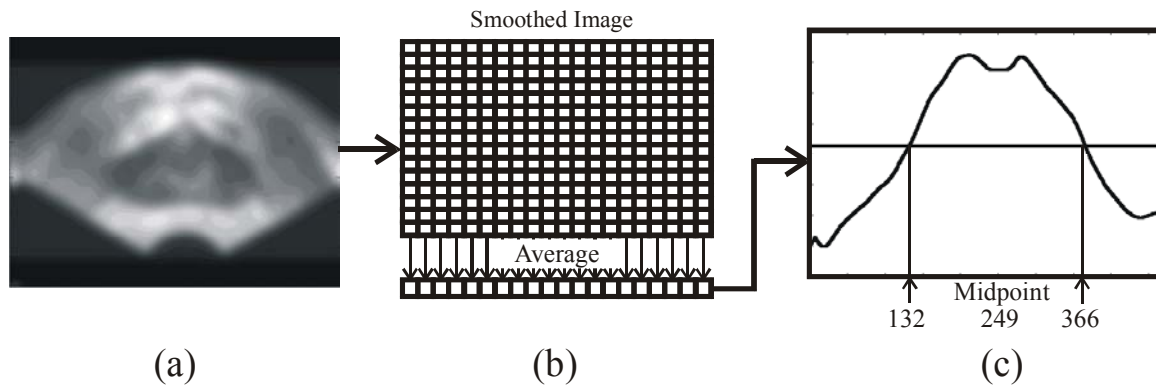
Figure 55   (a) A smoothed ultrasound image; (b) The average greyscale value is computed for each column; (c) The average intensity vs. column profile. In Case 1, there are only one low-to-high crossing and one high-to-low crossing of the threshold, which is represented by the horizontal line.

Step 3) After obtaining the average intensity vs. column profile (see Figure 55(c)), define a threshold – represented by the line in Figure 55(c) – and locate the crossings of the threshold. In the case shown in Figure 55, the crossings are at column 132 and 366.

Step 4) In Case 1, there are only one low-to-high crossing and one high-to-low crossing. Take the midpoint of the two crossings to be the centre column. In the case shown in Figure 55, column 249 is taken to be the centre column.

The example image used in demonstrating how the two initial points are found in Section 4.4.1 can be classified into Case 1. The centre column identified is shown in Figure 52(a).

Case 2: Difficult

Centre column is difficult to be identified in this case because the average greyscale value vs. column profile has multiple peaks – defined to be the interval from a low-to-high to a high-to-low crossing of the threshold. Figure 56 shows a typical image of this class. Since the light intensity on the top of the prostate is normally high, rather than taking the average of the whole column as we did in Case 1, we can take the average only for pixel near the top of the prostate of each column. This idea is applied in the development of the following algorithm:

First, the two edge points for all possible centre columns are identified. For the image shown in Figure 56(a), there are two possible centre columns, each associated to a peak shown in Figure 56(b). These two columns are represented by the vertical lines in Figure 57. For each column, two edge points are identified using the method introduced in Section 4.4.1. The edge points are represented by the crosses (×) in Figure 57.

(a)

(b)

Average of each column in the smoothed image
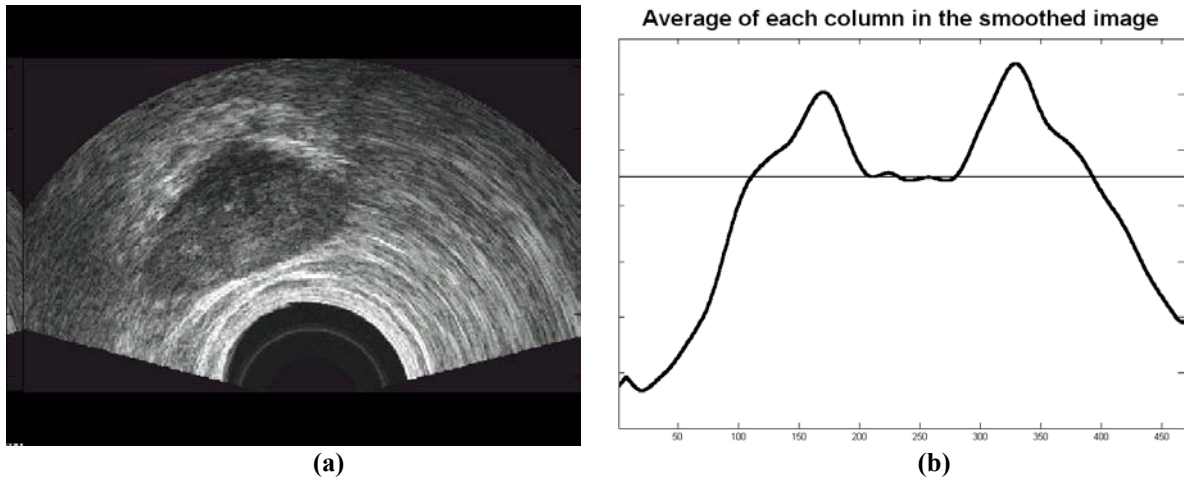
Figure 56   (a) A typical ultrasound image of Case 2; (b) The average intensity vs. column profile. There are two peaks in this profile.
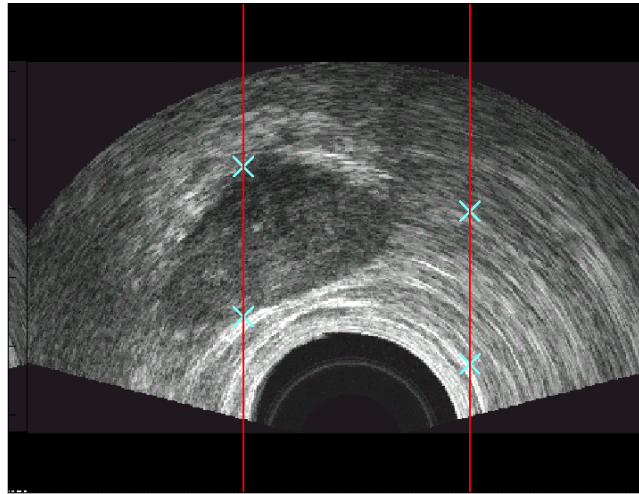
Figure 57   There are two possible centre column, represented by the vertical line. The crosses represent the edge points associated to each possible centre column.

(a)   **(b)**   **(c)**
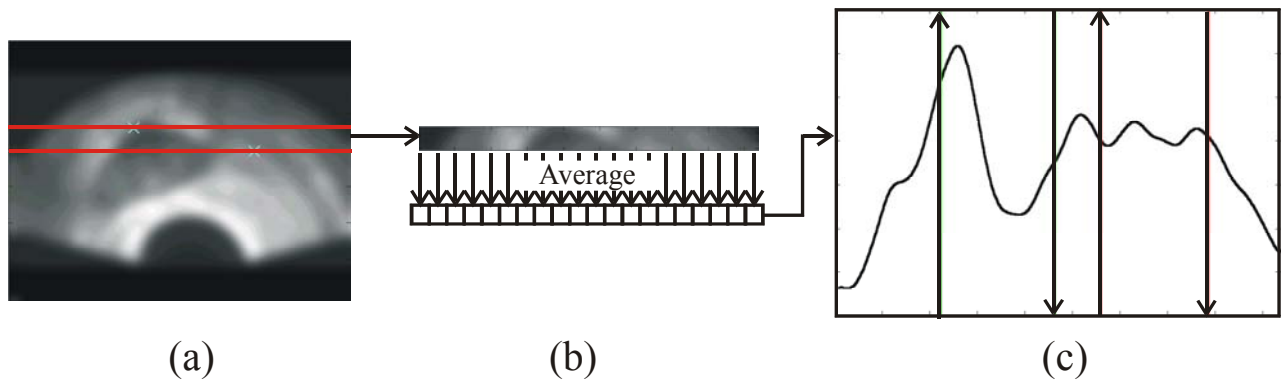
Figure 58   (a) The smoothed ultrasound image; (b) The average of each column in the extracted image is computed; (c) The average intensity vs. column profile. The crossings associated to each possible column are indicated by arrows.

The upper edge points associated to all possible centre columns are recorded.

The image bounded by the minimum and the maximum of the row indices of all the edge points collected in Step 2 is extracted. (see Figure 58(a))

For each column in the extracted image, the average is taken. (see Figure 58(b))

Recall that there are two crossings associated to each possible centre column. (see Figure 56(b)) In Figure 58(c), the low-to-high crossings are represented by arrows pointing upwards and the high-to-low crossings are represented by arrows pointing downwards. For each possible centre column, the maximum of the interval bounded by the two associated crossings is found.

Compare the maxima associated to each possible centre column. If the maximum associated to one centre column is significantly higher than that associated to other centre column, that centre column will be chosen. For our example image shown in Figure 57, the highlighted column on the left is chosen to be the centre column. Otherwise, as in the example shown Figure 59, we assume the two possible centre columns identified previously correspond to the sides of the prostate. Therefore, in this case, the centre column is estimated to be in the middle of the two possible centre columns. The centre column identified for the example image shown in Figure 59 is plotted in Figure 60.

4.4.3. Identification of Strong Edge Points

As mentioned at the beginning of this section, strong edges on the image will be identified by locating points where the gradient modulus is greater than a certain threshold. In our algorithm, the threshold is taken to be the 90% percentile of the gradient modulus. Figure 62(a) shows the positions of pixels where the gradient modulus is higher than the threshold. Figure 62(b) shows the remaining points after the points close to border are taken away.
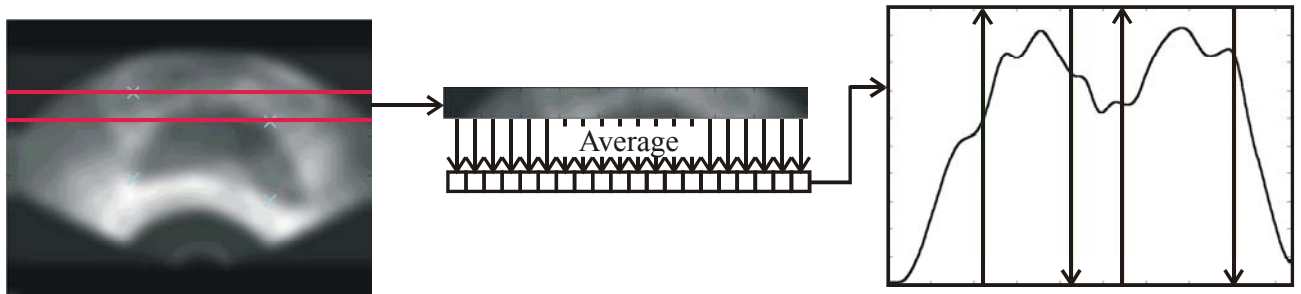
Figure 59   An example where, in the average intensity vs. column profile, the maximum value associated to a possible centre column is close to that associated to another possible centre column.
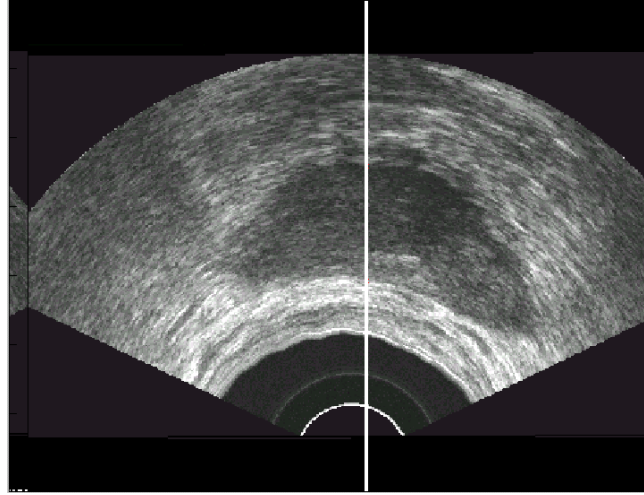
Figure 60   The centre column identified for the example shown in Figure 59
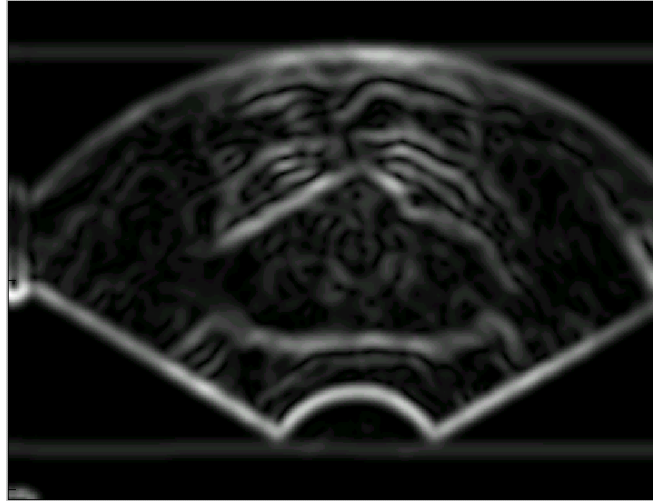
Figure 61   The gradient modulus of an ultrasound image
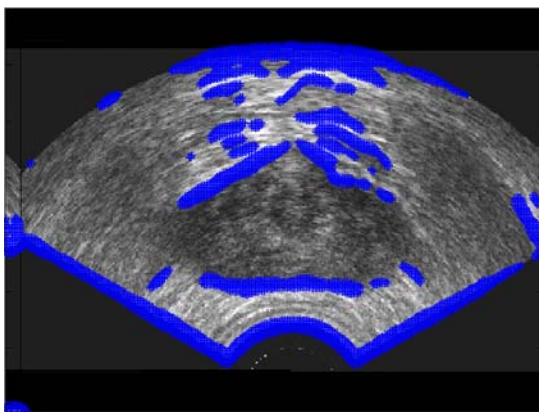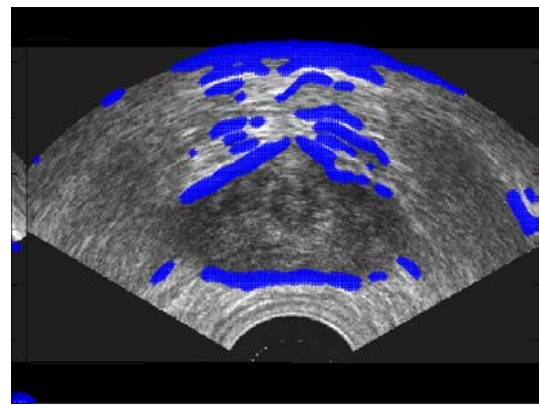
(a)	(b)

Figure 62   (a) The pixels with the gradient modulus higher than the threshold are shaded; (b) The remaining points after the points close to the border are filtered out

4.4.4. Global Initialization

Recall that two initial points are identified in Section 4.4.1 and the "qualified" edge
points are identified in Section 4.4.3. Now, other initial points on the contour should be
defined. This is done by a process we call "global initialization" followed by the "local
initialization". The "global initialization" process is defined by the following algorithm:

Step 1) First, the distance between the two initial points is measured and denoted by d.
Then two circles, one with diameter less than d and the other with diameter
greater than d, are drawn. In our algorithm, the smaller circle has a diameter of
0.95d and the diameter of the bigger one is *1.2d*. We restrict the algorithm to pick
initial points that are outside the smaller circle and inside the bigger one. (see
Figure 63)

Step 2) The region outside the smaller circle and inside the bigger circle is divided into N
segments, where N is chosen to be *32* in our algorithm. From each segment, a
point closest to the centre is chosen to be an initial point. In Figure 63, the initial
points are represented by the crosses (×).
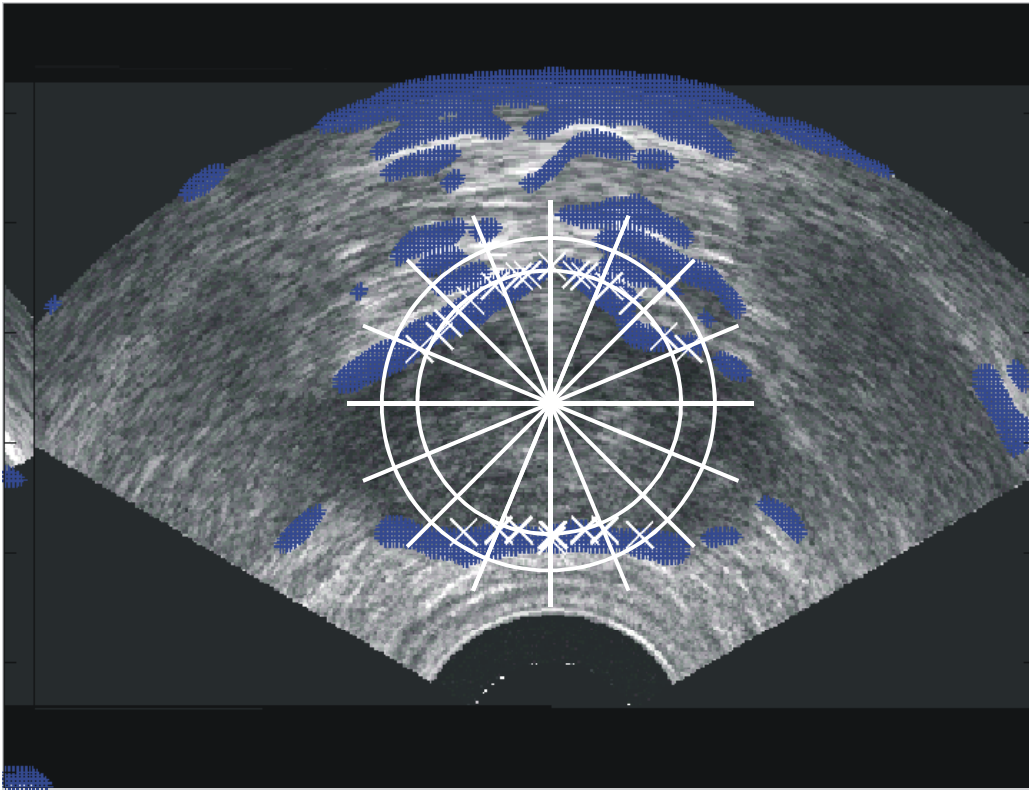
Figure 63   Initial points identified by the global initialization process

4.4.5. Local Initialization

At this point, a number of initial points have been identified using the process of "global initialization" described in Section 4.4.4. Obviously, as observed in Figure 63, the initial points are concentrated at the top and the bottom of the prostate and no initial points can be identified on both the left and the right side of the prostate. More initial points are required in order to initialize a contour that is reasonably close to the actual boundary. We use a process we call "local initialization" to locate initial points that are close to the existing initial points. The following algorithm defines the "local initialization" process:

Step 1
First, the algorithm requires three user-defined parameters, tabulated below:

initial_contour	A vector containing the initial points identified using the "global initialization" process.
resolution	A user entered parameter indicating the desired separation between adjacent vertices of the contour.
qualified_points	The set of points that satisfies the criteria introduced in Section 4.4.3. Figure 62(b) shows an example of this set of points.

Step 2
The length of each segment between adjacent initial points is calculated. In the following pseudocode, the number of points in initial_contour is denoted by *num* and the function $dist(\cdot,\cdot)$ returns the distances between two arguments.

for *k* = 1:*num*
        segment_length(k) = dist(*initial_contour(k)*, *initial_contour*((*k+1*) mod [1,*num*]));
end

Step 3
Starting from *k = 1*, the length of the kth segment is compared to a certain threshold, which is set to be *1.5* times resolution in our algorithm. If the length of the kth segment is greater than the threshold, Step 4 will be carried out. If not, *k* is incremented and this step is repeated until the length of all the segments of the contour have been compared to the threshold, at which stage the algorithm is completed.

Step 4
This is a step in which more initial points are added. In the following, the function $length(\cdot)$ returns the total number of elements contained in the argument:

p_{k-1} = *initial_contour*((*k-1*) mod [1,*num*]);
p_k = *initial_contour(k)*;
p_{k+1} = *initial_contour*((*k+1*) mod [1,*num*]);

dist_to_next_point = dist(p_k, p_{k+1});
$d_{i,global}$ = p_{k+1} - p_k;

point_in_question = p_k;
point_added_temp = [];
while (*point_in_question* is not empty) & (*dist_to_next_point* > 1.5 × *resolution*)
    if length(*point_added_temp*) >= 2;
        last_index = length(*point_added_temp*);
        d_{i-1} = *point_added_temp*(*last_index*) - *point_added_temp*(*last_index-1*);
    elseif length(*point_added_temp*) = 1
        d_{i-1} = *point_added_temp*(1) - p_k;
    else; d_{i-1} = p_k - p_{k-1};
    end

Calculate the vector from "*point_in_question*" to each point in the set "*qualified points*" defined in Step 1. (Denote this vector as d_i hereafter. Note that each point has an associated d_i.)

qualified_points = a set of points that satisfies the following four conditions:
          a) points must be in the set "*qualified_points*" defined in Step 1;
          b) $d_i \bullet d_{i-1} > 0$;
          c) $d_i \bullet d_{i,global} > 0$;
          d) $|d_i| < 1.5 \times resolution$.

$$(4.25)$$

idx_{chosen_point}= $\arg\min_{i}$ {dist(*qualified_points*(i), *point_in_question*) - *resolution*};

$$(4.26)$$

point_found = *qualified_point*(idx_{chosen_point});

point_added_temp = [*point_added_temp* *point_found*];
point_in_question = *point_found*;

if *point_in_question* is not empty
    dist_to_next_point = dist(*point_in_question*, p_{k+1});
    end
end

An example will be used to demonstrate the above procedure. In the example illustrated in Figure 64, Segment 1 to 14 is shorter than the threshold, and therefore, no action is taken. The 15th segment is longer than the threshold and the algorithm shown above is executed. Before locating the first additional point, the values of the variables

point_in_question, d_{i-1} and $d_{i,global}$ are assigned, which are illustrated in Figure 64(a).
Then d_i, the vector from "*point_in_question*" to each point in the set "*qualified points*" –
all elements of which are shaded in blue in Figure 64(a) – is calculated. After d_i
associated to each point in the set "*qualified_points*" is calculated, the set
"*qualified_points*" will be redefined, choosing points that satisfy the four conditions
stated in (4.25). Conditions *b* and *c* are to ensure the algorithm selects points in a
consistent direction – the counter-clockwise direction in this case. Condition *d* prevents
the algorithm from selecting an initial point that is very far away from the last initial
point. The implementation of this restriction is based on the rationale that if the last initial
point lie on the desired boundary, a point located far away from this point is very likely to
be part of a false edge. From the redefined set "*qualified_points*", one point will be
chosen be the next initial points according to (4.26). This criterion chooses the next initial
point such that the distance between this point and the last initial point identified (labelled
as "*point_in_question*" above) is as close to the user-defined variable "*resolution*" as
possible. The first additional point chosen is plotted in Figure 64(a). The second points
chosen, shown in Figure 64(b), is defined in the same way, except the variables d_{i-1} and
point_in_question are changed. The variables associated to the selection of the second
point are shown in Figure 64(b). After this point has been found, the "while" loop in the
above algorithm run for a third time. However, on this run, no point satisfies the four
conditions stated in (4.25) and the algorithm leaves Step 4. At this point, *k* is incremented
and the algorithm returns to Step 3.

Step 5

Step 3 and Step 4 are repeated in the clockwise direction.

This concludes the discussion on the local initialization process. In Figure 65 shows all
the points identified by the global and the local initialization process.

(a)

(b)
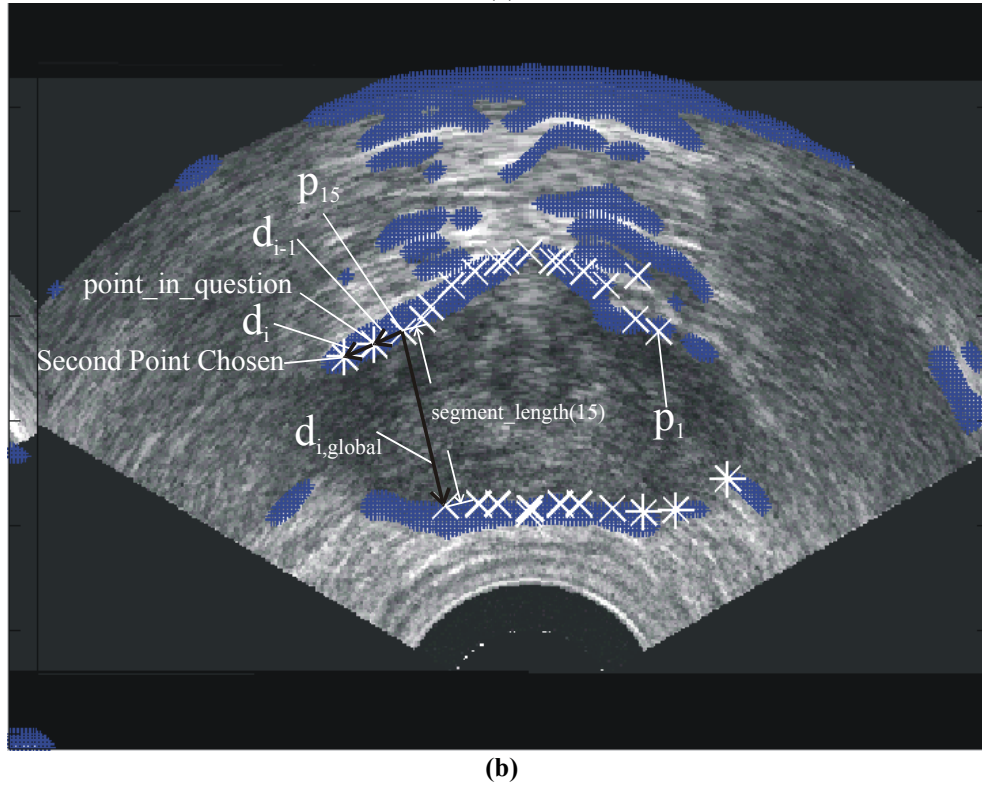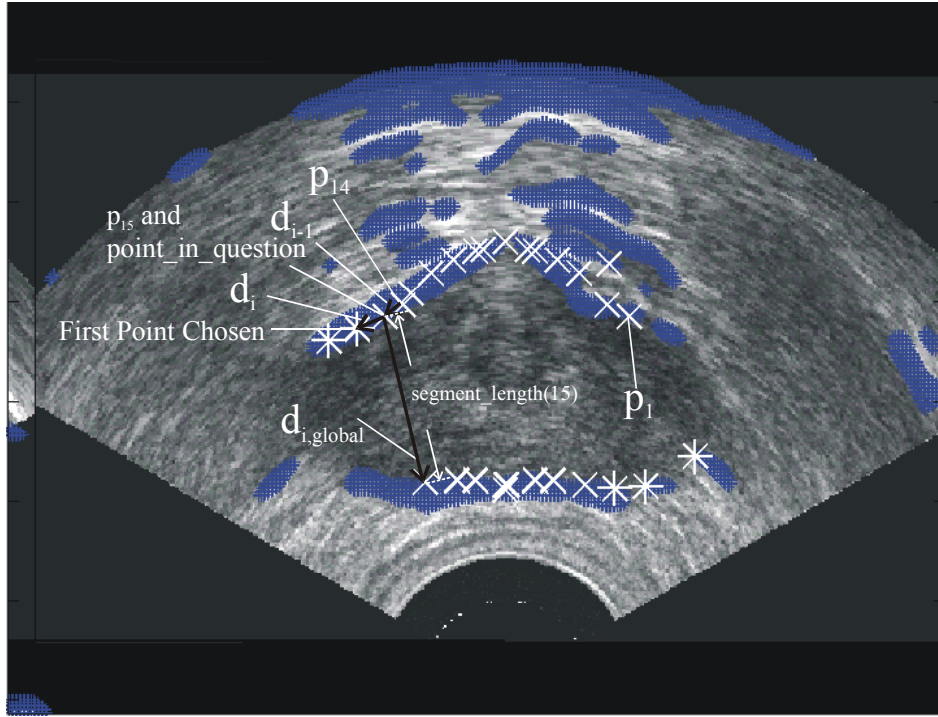
Figure 64   An example illustrating the procedure in Step 3 of Section 4.4.5
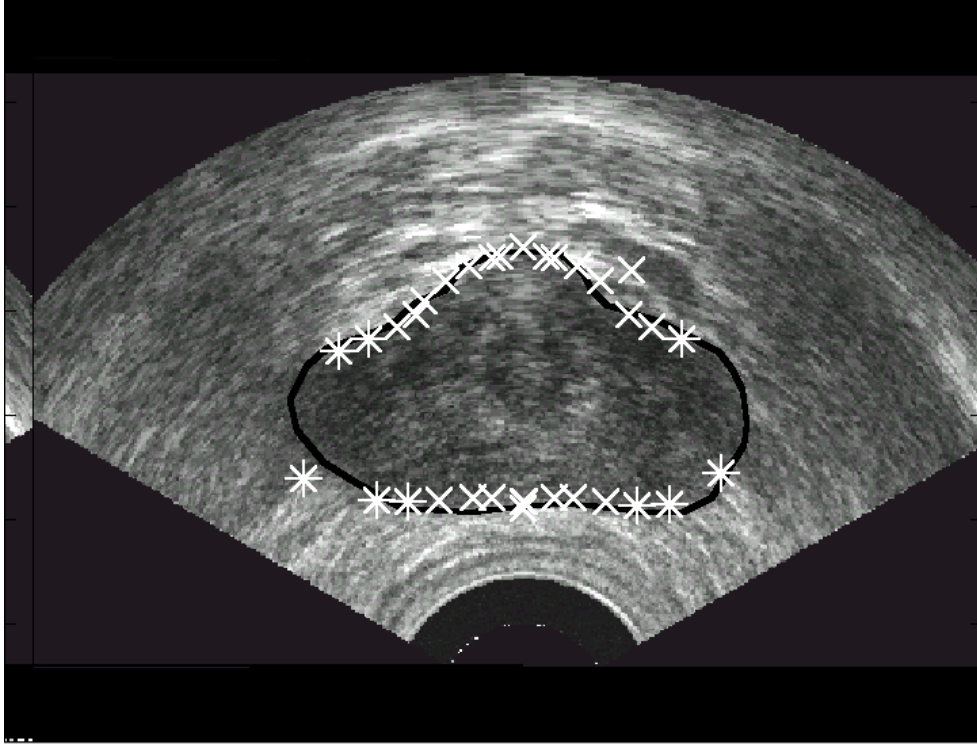
Figure 65   Initial points identified in the global initialization process are represented by the crosses (✕), and those identified in the local initialization process are represented by asterisks (✱). The manually outline contour is also plotted (in black) for comparison.

4.4.6. Validation of the Available Initial Points

Now, we have a number of initial points, identified by the global and local initialization process. Unfortunately, a small portion of initial points may lie on false edges as shown in Figure 66. The presence of such points will distort the initial contour, and therefore, it is important to design a mechanism to differentiate accurate and inaccurate initial points.

In the following, we propose a method that differentiates accurate and inaccurate points according to the distances between these points and the centre, which is defined to be the midpoint of the segment joining the two initial points identified in Section 4.4.1:

Step 1

The process begins by defining two variables based on the variable *initial_contour*, which is a vector containing all the initial points identified in the global and local initialization process:

init_pt_angle	The kth element of this vector is the angle associated with the vector from the centre to *initial_contour(k)*. The vector *initial_contour* has been sorted so that *init_pt_angle* is an ascending sequence and the values of its elements are in the interval $[0,2\pi)$.
dist_centre_profile	The kth element of this vector is the distance between the point *initial_contour(k)* and the centre. The length of this vector is equal to that of *initial_contour*.
dist_centre_profile_ratio	The kth element of this vector is the ratio between *dist_centre_profile(k+1)* and *dist_centre_profile(k)*. The total number of elements in this vector is less than that of *dist_centre_profile* by *1*.

Step 2

The definition of the vector *dist_centre_profile_ratio* implies all of its elements should be close to 1, because two adjacent elements of *dist_centre_profile* should have a similar value. If this is not the case, there are two possibilities: (1) The point corresponds to one of the two elements in question does not lie on the boundary. For example, two adjacent points are represented by Point A and Point B in Figure 66(a), and Point B does not lie on the boundary; (2) The two points correspond to the adjacent elements in question both lie on the boundary. However, because of the geometry of the prostate, the two adjacent elements of *dist_centre_profile* in question are significantly different. An example of this case is shown in Figure 66(b). Note that the distance between the centre and Point B is almost three times as large as that between the centre and Point A, but both points lie on the boundary.

To differentiate between the two cases mentioned above, the following algorithm is used:

while *k <= length(dist_centre_profile_ratio)-1*

if *dist_centre_profile_ratio(k) > 2 | dist_centre_profile_ratio(k) < 0.5*
    n = the index *n* such that

$$init\_pt\_angle(n) = \max_{m \in [1, length(init\_pt\_angle)]} \left\{ \begin{array}{l} init\_pt\_angle(m): \\ init\_pt\_angle(m) \leq (init\_pt\_angle(k+1) + \pi/2) \end{array} \right\};$$

    d = dist_centre_profile(k+1);

    for *u = k+1:n*
      if *(0.65 × d < dist_centre_profile(u) < 1.25 × d)*
      qualified_point = [qualified_point initial_point(u)]
    else
      not_qualified_point = [not_qualified_point initial_point(u)]

```
        end

        p = length(qualified_point)/(n-k);
        if p >= 0.5
           Delete all elements in not_qualified_point from initial_point
        else
           Delete all elements in qualified_point from initial_point
        end
        k = n + 1;
else
        k = k + 1;
end
end
```

To demonstrate how the algorithm works, we use the examples shown in Figure 66. In both examples, the element of *dist_centre_profile_ratio* that equals to the ratio of the distance between Point B and the centre and that between Point A and the centre is found to be greater than 2. This makes us suspicious of whether Point B lies on the boundary. The propose algorithm first collects points that are inside the $\pi/2$ segment starting from Point B. Then, it classifies this set of points into two categories: the one with distance from the centre appromately equal to *d*, the distance between the centre and Point B, and those with distance from the centre that is not close to *d*. In Figure 66, points that are classified into the first group are those inside the shaded segment and the remaining points inside the $\pi/2$ segment are classified into the second group. If more than half of the points inside the $\pi/2$ segment are classified into one of the two groups, points belong to that groups are retained and points belong to the other group are eliminated. In Figure 66(a), the points belong to the first group are eliminated. It can be observed that this decision is correct; the points classified into the first group do not lie on the boundary. In Figure 66(b), the points belong to the second group are eliminated. In this case, some points close to the boundary are eliminated. However, it is only a minor problem because the segment represented by the deleted points can be closely approximated by a straight-line segment joining two retained vertices, and therefore, the deletion of these initial points does not have a significant impact on the final contour generated by the DDC algorithm. On the other hand, it is important to keep points that are in the first group, which is essential for defining the corner segment of the prostate accurately. Therefore, it can be said that the algorithm makes a good decision in this case.

4.4.7. Interpolation of the Remaining Initial Points Using Spline

Now, we have a number of initial points, identified by the global and local initialization process. The final task remains to be done in our initialization algorithm is to connect the available points to form an initial contour. Of course, we could easily connect the points with straight lines. However, since the boundary of the prostate is usually smooth, the

initial contour would be more accurate if the available points are joined using the spline interpolation method introduced in Section 3.3.1.

The procedure of interpolating the available initial points are almost exactly the same as that used to interpolate the four user-defined initial points in Section 3.3.1, except the knot sequence is defined differently. In Section 3.3.1, the four initial points are approximately equally-spaced, and therefore, we can use a uniform knot sequence $\{0,1,2,3,4\}$ to define the data points. However, it is obvious that we cannot claim the distance between adjacent initial points are equal here, and therefore the knot sequence cannot be uniform. In the proposed algorithm, the knot sequence is defined according to the length of the straight-line segment joining adjacent initial points – the first knot is 0 and the $k+1^{th}$ knot is the sum of the length of the first k segments. After defining the knot sequence, the cubic polynomials defining the x-coordinate and the y-coordinate of the interpolated contour, denoted by $x(u)$ and $y(u)$ respectively in Section 3.3.1, can be readily obtained using the Spline toolbox in MATLAB.
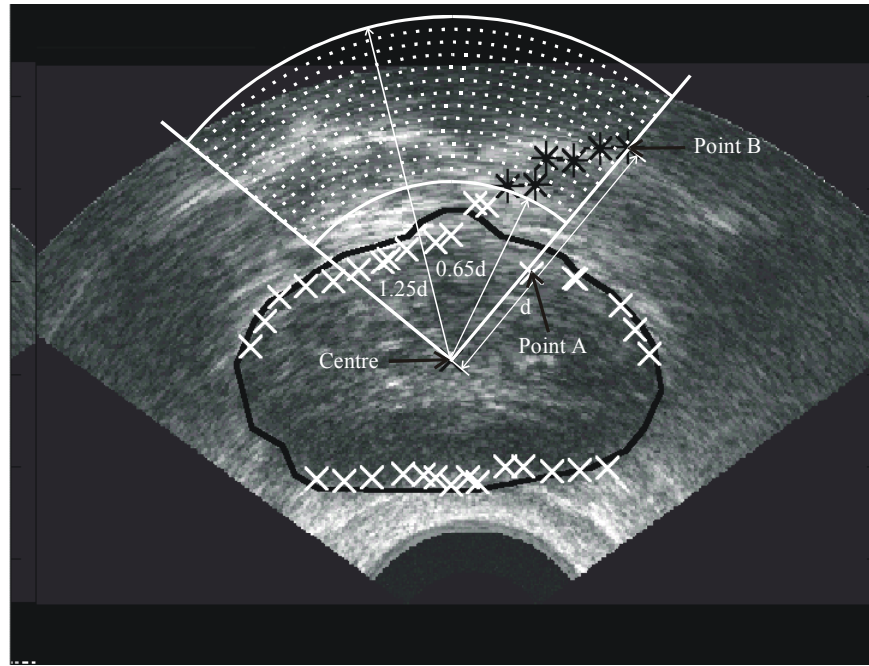
(a)
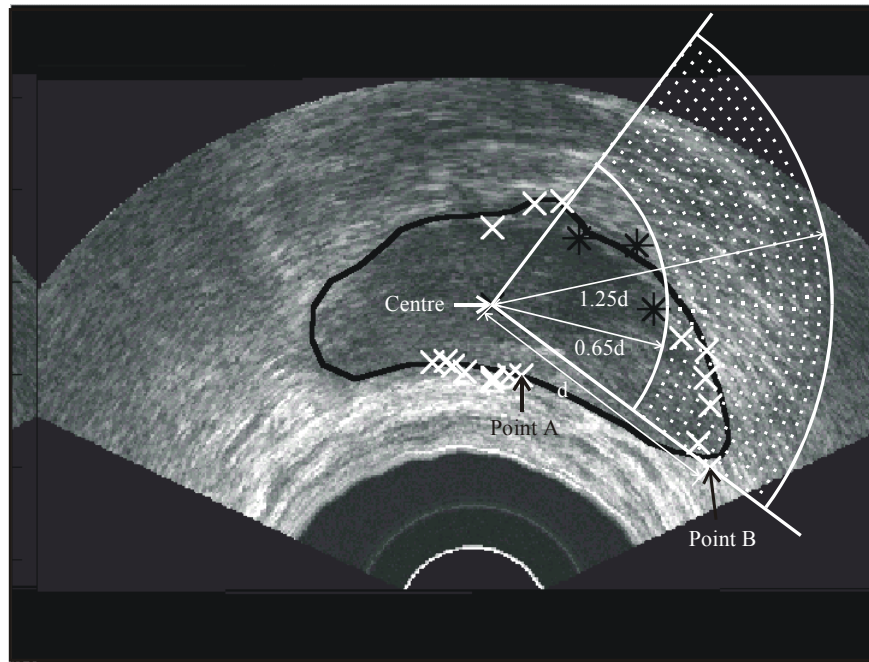
(b)

Figure 66   Two examples showing the effects of the validation process (Initial points identified after the global and local initialization process are marked either in white crosses (✗) or black asterisks (✱). The validation process eliminates points that are represented by black asterisks and keeps those that are marked in white.)

Since the initial contour consists of only a finite amount of vertices, we need a mechanism to define the vertices lying on the initial contour. In the proposed algorithm, initial points are added to the set of available initial points only if the segment joining adjacent vertices is too long. In other words, if the difference between two adjacent knots is larger than a certain threshold, say the parameter "resolution" defined in Step 1 of 4.4.5, $x(u)$ and $y(u)$ will be evaluated at some u between these two adjacent knots. Suppose e is the sequence of u at which $x(u)$ and $y(u)$ will be evaluated. Then, e is defined by the following algorithm:

e = original knot sequence = $\begin{bmatrix} u_o & u_1 & \cdots & u_{N-1} & u_N \end{bmatrix}$;
for $k = 1{:}N$
     if $u_k - u_{k-1} > resolution$

  $$numpo\,int = \left\lfloor \frac{u_k - u_{k-1}}{resolution} \right\rfloor;$$

  $u_{add} = \begin{bmatrix} u_k + resolution & u_k + 2 \times resolution & \cdots & u_k + numpo\,int \times resolution \end{bmatrix}$;
       Insert u_{add} between u_{k-1} and u_k in the sequence e;
     end
end

This completes the description of the proposed automatic initialization method. The resulting initial contour for a sample ultrasound image is illustrated in Figure 67.
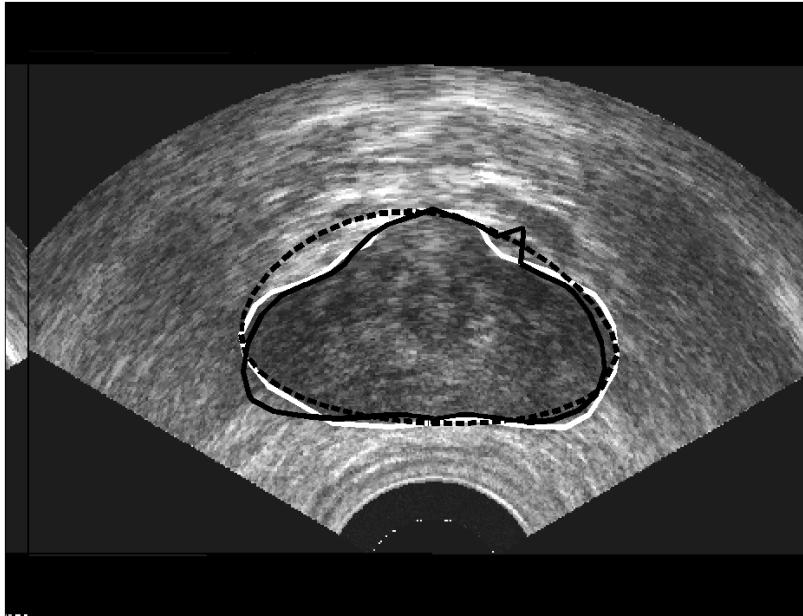
Figure 67   The initial contour produced by the proposed initialization algorithm (drawn in solid black line) is compared to the professionally outlined contour (drawn in solid white line) and the initial contour defined in Section 3.3.1. (drawn in dashed line)

4.5.   Summary and Remarks

In this chapter, some modifications to Mallat's dyadic wavelet transform are introduced. First of all, it is pointed out that, in his book [1], Mallat shifted the wavelet coefficients inappropriately. It is demonstrated in Section 4.1 that the shifting error in the modulus $M_{2^j} f$ is very significant if j is large. This error causes the segment identified by his algorithm to be shifted with respect to the actual border. Also in this section, a detailed explanation is given on why one must shift the wavelet coefficients generated by the dyadic wavelet transform, which was not explained clearly in [1]. This interpretation did not only allow us to determine the accurate shift in Mallat's algorithm, it also helps us to find out how much shift on the wavelet coefficients is required when we use other dyadic wavelets, such as those we constructed in Section 4.2 based on the scaling functions of Symlets and Coiflets, rather than the quadratic spline wavelet.

In Chapter 3, it is argued that Mallat's dyadic wavelet transform method is more flexible than taking the numerical gradient of the Gaussian-smoothed image. As an effort to realize the potential of this method, we made use of the scaling functions of the Symlets and Coiflets families as a starting point, and followed Mallat's method in constructing the spline dyadic wavelet, in an attempt to construct new dyadic wavelets. Until we present our results in Section 5.3, we do not know how much or whether the segmentation results will be better when these wavelets are used. However, it is believed that the dyadic wavelets constructed based on Coiflets will give good results because the scaling functions have a number of vanishing moments.

In Section 4.3, a new energy field is defined based on the approximate coefficients of the dyadic wavelet transform. It is observed that, in an ultrasound image, the greyscale value of pixels near the border of the prostate tends to lie on a certain range, which can be estimated using the greyscale value of the pixels at the four points user enter as initial points of the DDC model. The new energy field was defined in a way such that, if it is used to drive the DDC model, the contour will converge to location where pixels have greyscale value in this range. This new energy field gives information about the border of the prostate that is not available when gradient-based method is used. This energy field serves a very important role in the proposed semi-automatic algorithm, which will be introduced in Section 5.1.

After introducing all the building blocks of the semi-automatic algorithm, we try to extend existing ideas in developing a fully-automated segmentation algorithm. The benefits associated with a fully-automated segmentation tool are very significant as mentioned in Chapter 1. Because only the initialization stage of the semi-automatic algorithm needs human supervision, the proposed semi-automatic algorithm would become a fully-automatic one if the semi-automatic initialization process is replaced by a fully-automatic one. In Section 4.4, a fully-automatic initialization process is introduced. The whole segmentation algorithm will be presented in Section 5.1 and the performance of this algorithm will be evaluated in Section 5.3.

Chapter 5

The Proposed Algorithm

5.1. The Proposed Algorithm for Prostate Boundary Detection

Figure 68 shows the structure of the proposed segmentation algorithm. There are four "functional blocks" in the diagram. A functional block is used to represent operation that generates one or more output variables, when given the required input variables. In Figure 68, the input and the output variables are listed on the left and the right side of the functional blocks respectively. The operations of the "fast dyadic wavelet transform" block and the "fully-automatic initialization" block have been introduced in Section 3.2.3 and 4.4 respectively. In this section, the remaining two blocks – the "semi-automatic initialization" block and the "final contour definition" block – will be defined.

5.1.1. The "Semi-Automatic Initialization" Block

It can be observed in Figure 38 that the initial contour defined using the spline interpolation technique cannot capture the bulge on the top of the prostate. Since there are many false edges near the top of the prostate, the DDC model driven by an energy field proportional to the gradient modulus is easily attracted to these edges, and therefore, cannot capture the shape of the bulge accurately. To improve the accuracy of the segmented boundary, a more accurate initial contour is needed. The purpose of the "semi-automatic initialization" block is to reinitialize the contour so that it better approximates the actual boundary.

An energy field that can be used to define the edge is introduced in Section 4.3, and, as explained in that section, this energy field should be able to drive the DDC model towards the bulgy shape because of the high light contrast near the top of the prostate.

In the "semi-automatic initialization" block, this energy field is used to drive a modified DDC model. This DDC model only takes into account of the external forces that drive the contour towards the centre of the prostate, or more specifically, the direction of the radial vectors defined in (3.42), and ignores the external forces opposite to this direction. This constraint is implemented because, in most cases, the bulgy shape is inside the initial contour defined by the spline interpolation introduced in Section 3.3.1. Figure 69 shows the structure of the "semi-automatic initialization" block. Figure 70 compares the output contour with the manually outlined boundary of the prostate in an ultrasound image.
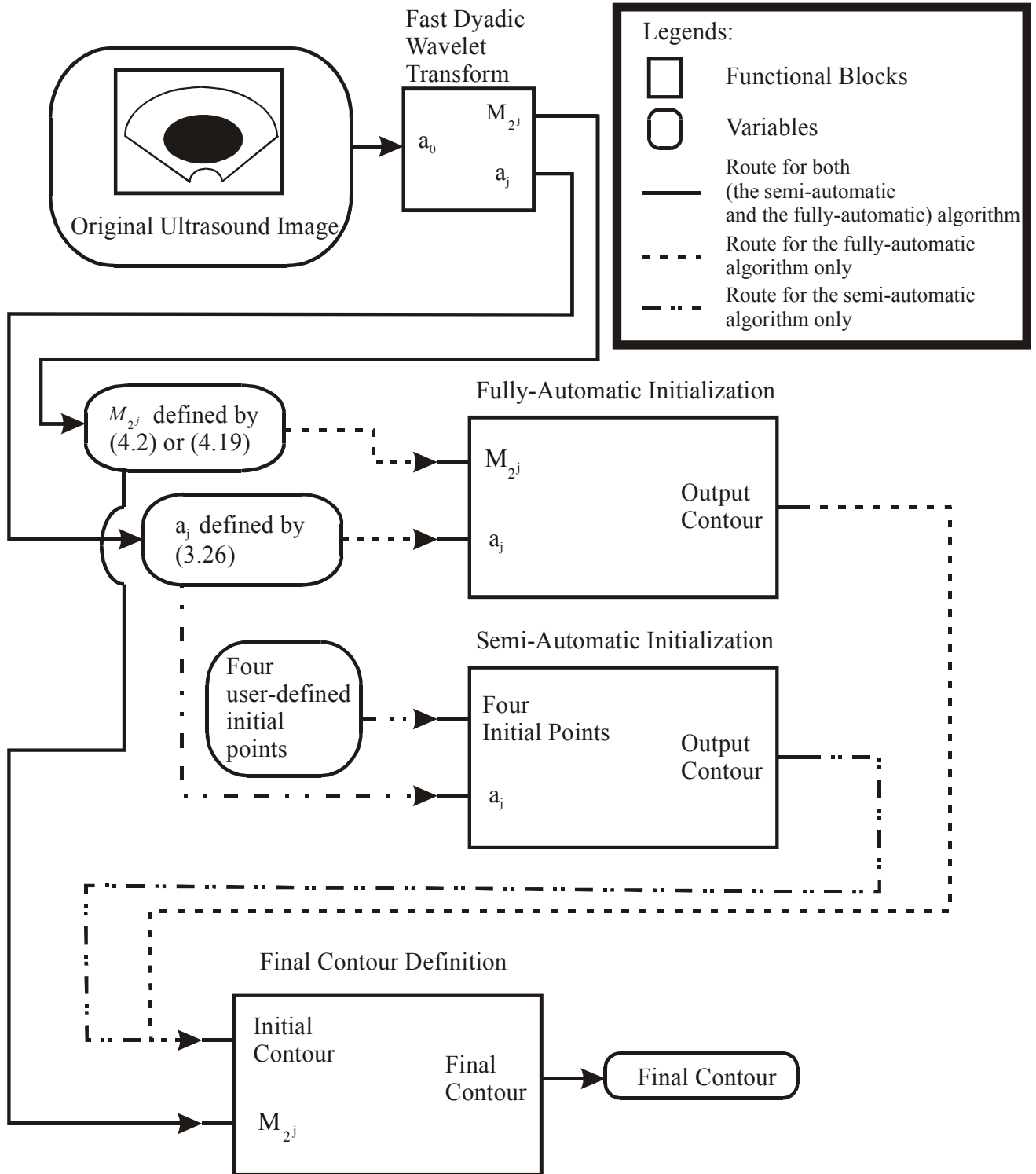
Figure 68 Proposed algorithm for prostate boundary detection in 2D ultrasound image

Semi-Automatic Initialization

| Four Initial Points | Initialization Algorithm introduced in Section 3.3.1 | A Modified DDC Model | Output Contour |

Four Initial Points → Four Initial Points | Initial Contour → Initial Contour | Resulting Contour → Output Contour

a_j → a_j | E_{im1} defined by (4.20) → Energy Field | Resulting Contour
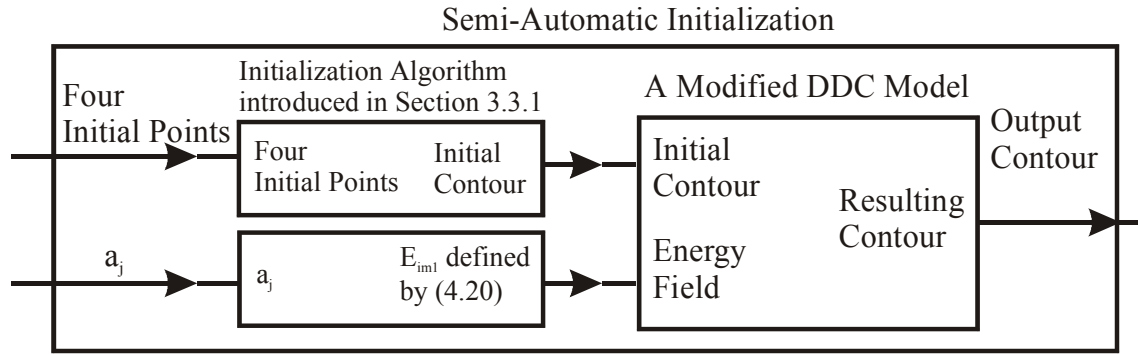
Figure 69 The "Semi-Automatic Initialization" block
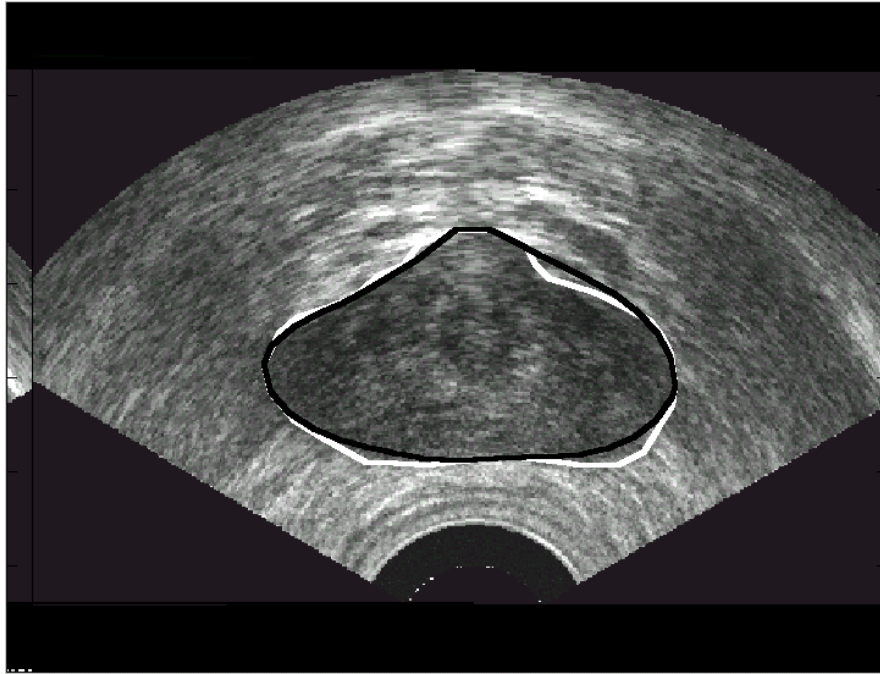
Figure 70   The output contour of the semi-automatic initialization block (in black) is compared to the manually outlined contour (in white).

5.1.2. The "Final Contour Definition" Block

The purpose of this block is to define the prostate boundary accurately, given the initial contour, which have been evaluated either in the "semi-automatic initialization" block or the "fully-automatic initialization" one, and the gradient modulus obtained from the dyadic wavelet transform.

Final Contour Definition
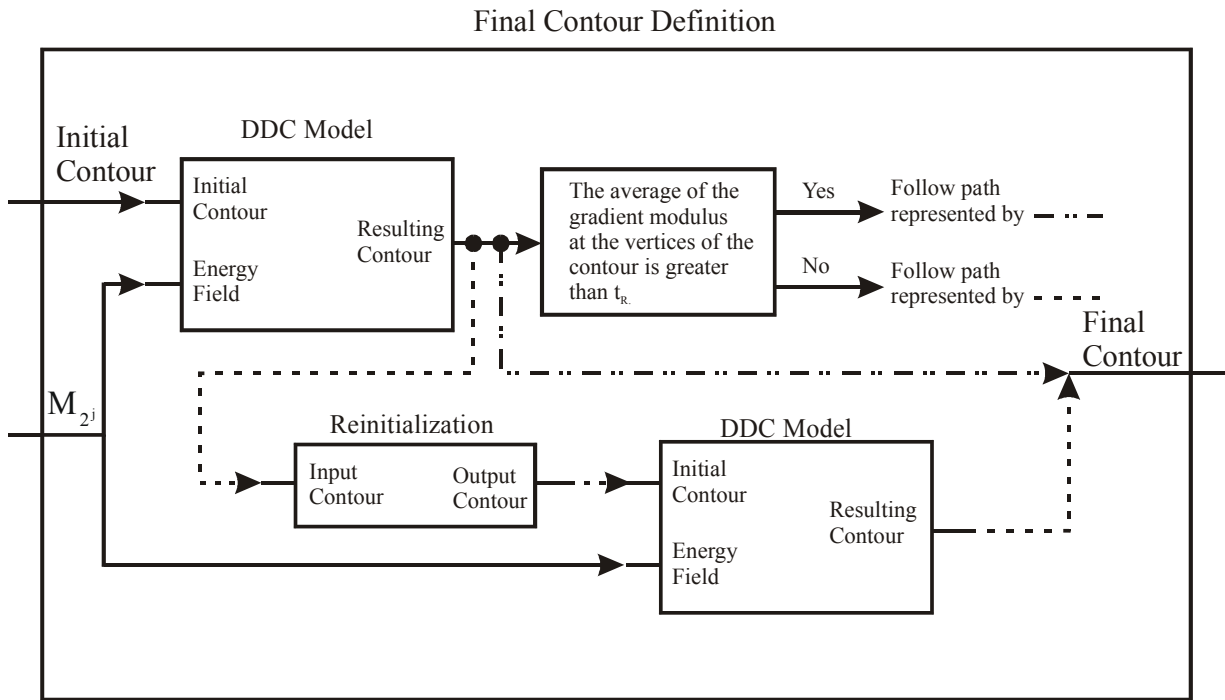
Figure 71   The "Final Contour Definition" block
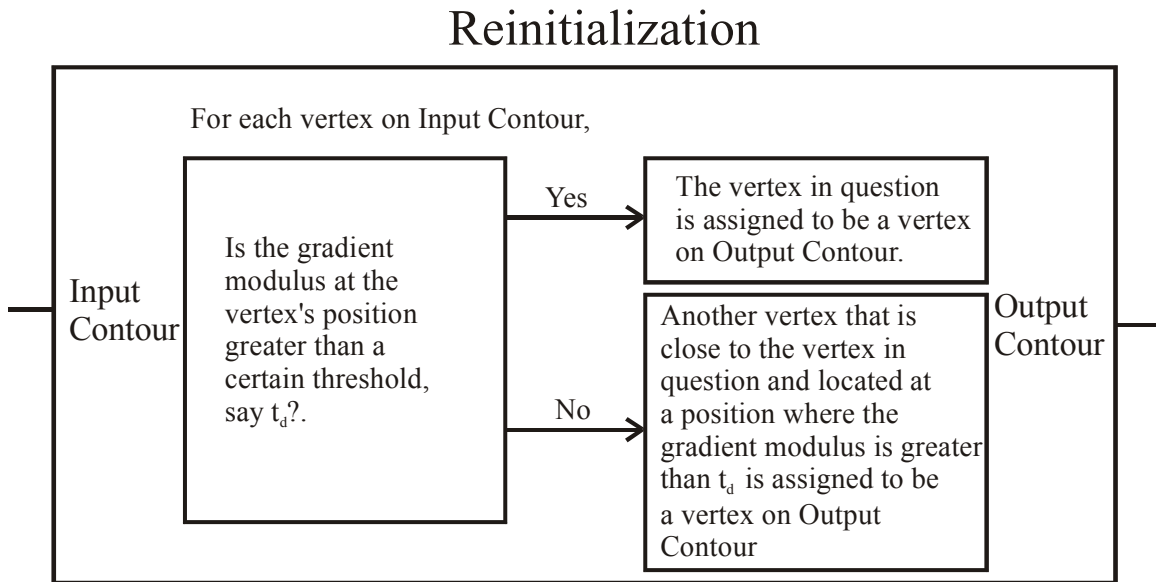
Reinitialization

Figure 72   The "Reinitialization" block – a sub-block of the "Final Contour Definition" block

To allow fair comparison on the performance of different wavelets, the image should be smoothed by smoothing functions with approximately equal effective support before the gradient modulus is taken. For this reason, we have chosen to assign the energy field to be M_{2^6} if the zero order spline dyadic wavelet transform is used, M_{2^5} for first order spline transform, M_{2^4} for higher order spline transform and M_{2^5} for all orders of Symlet and Coiflet transform.

Edges that are part of the desired prostate boundary are likely to be strong edges. Therefore, if a segment of the resulting contour generated by the first DDC run does not lie on strong edge, it is likely that the segment in question does not lie on the prostate boundary. In the proposed algorithm, a feedback mechanism is included to replace the segment on the resulting contour that is likely to be inaccurate with another segment that is more likely to accurately represent the prostate boundary. First, this mechanism should be able to detect whether the contour generated by the first DDC run is likely to be accurate – or in other words, whether it lies on strong edges. This task is done by comparing the average of the gradient modulus at the positions of all vertices on the contour to a threshold t_d. If the average is greater than t_d, then the contour generated by the first DDC run is declared to be the boundary detected by the proposed algorithm. Otherwise, the contour needs to be "reinitialized". Figure 72 shows the block diagram of the reinitialization process used in the proposed algorithm. The function of this process is to replace a vertex that lie on weaker edge – defined to be edge features where the gradient modulus is lower than t_d – by another vertex that lies on a stronger edge in the neighbourhood of the original vertex. The reinitialized contour will serve as the initial contour for the second DDC run.

Figure 74 shows the importance of the reinitialization mechanism. Figure 74 shows that the contour generated by the first DDC run is very inaccurate on the left half of the prostate. The reason for this inaccuracy is that the initial contour is very far off from the actual boundary, which, in turn, can be explained by the fact that no initial points can be identified around there using the proposed fully-automatic initialization method because the edge feature defining the prostate boundary is relatively weak there. (see Figure 73) By applying the initialization procedure, the inaccurate segment of the contour can be attracted to a strong edge point. If the reinitialized contour is used as the initial contour in the second DDC run, the contour obtained after the second DDC run is much more accurate than that obtained after the first DDC run. (see Figure 74)

This concludes the introduction of the proposed segmentation algorithm. The accuracy of the contour generated by the semi-automatic and the fully-automatic part of the algorithm will be evaluated in Section 5.3.
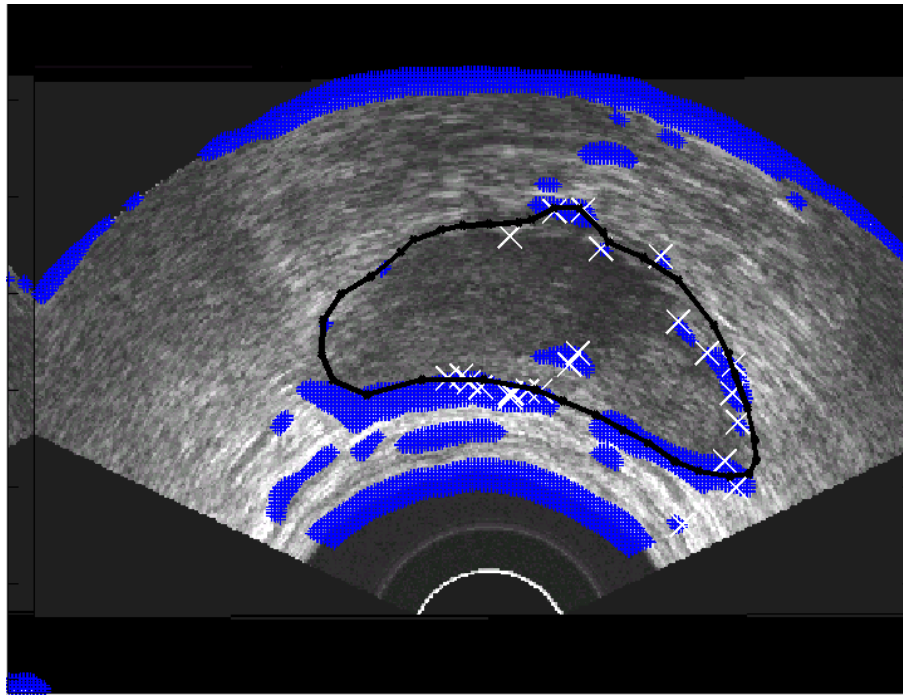
Figure 73   The automatically identified initial points (represented with the white crosses (✕)) and the manually outlined contour. Note that around the weak edge segment on the left hand side, there is no initial point identified.
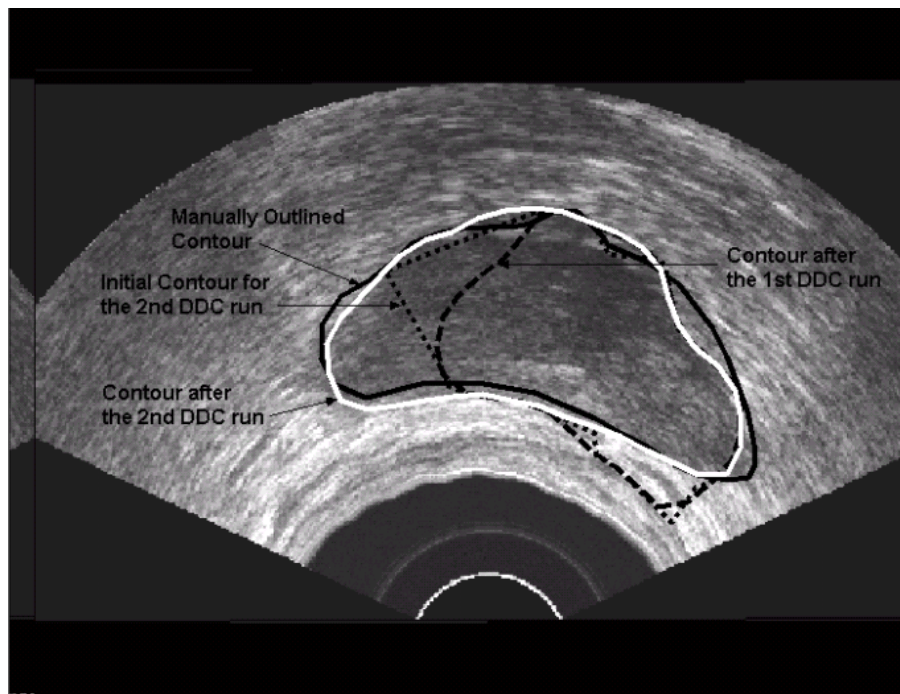
Figure 74   This figure shows the importance of the reinitialization process. Note that the boundary obtained after the second DDC run is much more accurate than that obtained after the first DDC run.

5.2.   Evaluation Criteria for the Segmented Boundaries

In evaluating the accuracy of the semi-automatically outlined boundary defined by their algorithm, Ladak [8] defined a distance-based metric. Before calculating various parameters in this metric, they chose the centre of the prostate. (see Figure 75) At a certain angle θ_i, the distance, from the boundary generated by their algorithm and from the manually outlined contour, to the centre are measured. The difference between these two distances is computed for N uniformly sampled angles in the interval $[0, 2\pi)$. This set of values, denoted by $\{\, d(\theta_1), d(\theta_2), \cdots, d(\theta_N)\,\}$, is used computed three quantities: (a) the mean difference (MD), (b) the mean absolute difference (MAD), (c) the maximum difference (MAXD). MD, MAD and MAXD are defined by

$$MD = \sum_{i=1}^{N} d(\theta_i) / N$$

(5.1)

$$MAD = \sum_{i=1}^{N} |d(\theta_i)| / N$$

(5.2)

$$MAXD = \max_{i \in [1,N]} \{ |d(\theta_i)| \}$$

(5.3)

5.2.1.   Problems Associated with Ladak's Distance-Based Metric

The first problem of Ladak's [8] distance-based metric is on the use of the mean difference as one of the parameters to evaluate the accuracy of the boundary. Since the positive errors and the negative errors cancel, this parameter actually gives no information about the accuracy of the segmented boundary. For this reason, the mean difference parameter is abandoned in the evaluation of the boundary segmented by the proposed algorithm.

The second problem is best demonstrated using an example. The automatically outlined contour (drawn in white) is compared to the manually outlined contour (drawn in black) in Figure 76. The centre is represented by the white cross (×). In a certain angle θ_i with respect to the centre, we find the distance $|d(\theta_i)|$, represented by a in Figure 76. However, it is obvious that the deviation is badly exaggerated, basically because the implicit assumption associated with this metric – the line along which the deviation is measured is approximately perpendicular to both boundaries – is violated. This unnecessary assumption occasionally makes their distance-based metric poorly reflects the degree of the contour's accuracy.
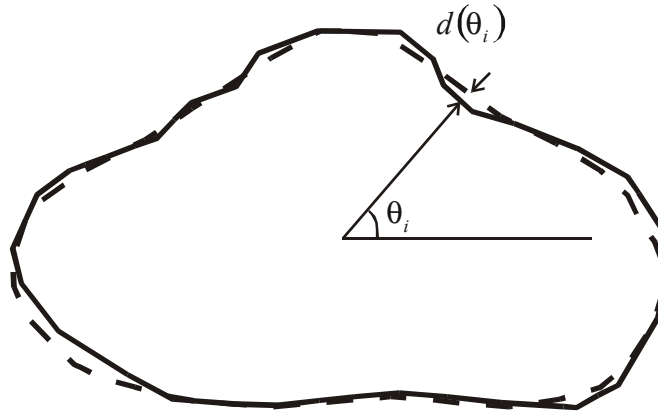
Figure 75   Illustration of d(θ$_i$), the difference between the manually outlined boundary and the boundary defined by a computerized algorithm at an angle θ$_i$
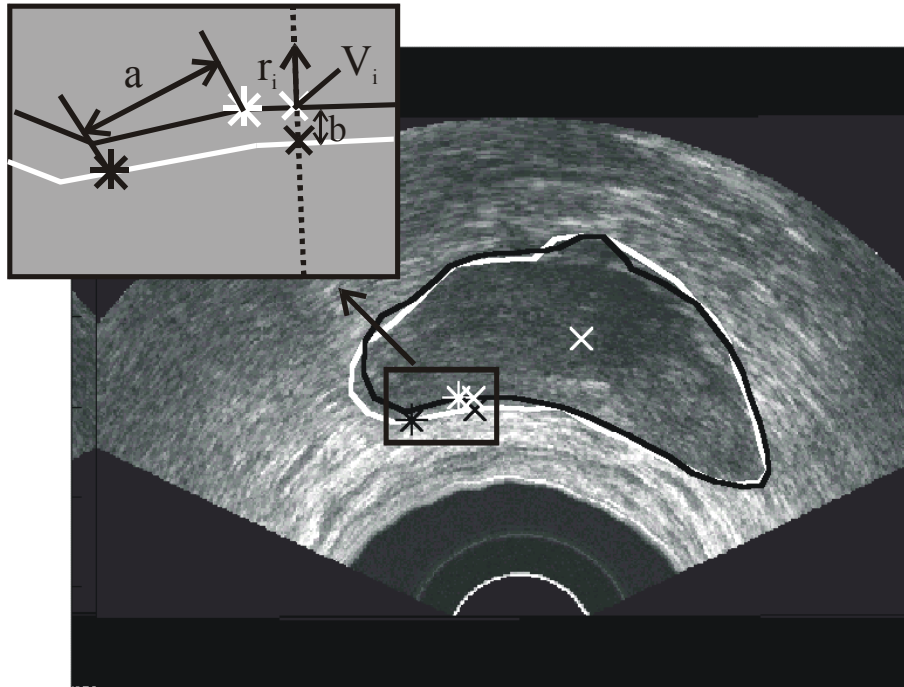
Figure 76   Comparison between the deviation measured by Ladak's distance-based metric, represented by a, and that measured by the proposed distance-based metric, represented by b

5.2.2.   The New Distance-Based Metrics

To solve the second problem of Ladak's [8] distance-based metric mentioned in Section 5.2.1, we must find a way so that we can measure the deviation along a direction that is approximately perpendicular to the boundaries at the location where the measurement is made. Recall that at a vertex V_i of the DDC model, the direction that is

perpendicular to the contour is defined by the radial vector $\hat{r}_i$, which is defined by (3.42). Therefore, one could completely abandon the rather arbitrary parameter – "the centre of the prostate" – and measure the difference of the boundaries along the direction of $\hat{r}_i$. In Figure 76, the distance between the two boundaries along the direction of $\hat{r}_i$ is represented by *b*. It is obvious that the distance measured in this way better reflects the degree of the contour's accuracy than the method introduced in Section 5.2.1.

In the revised distance-based metrics, the distance between the two boundaries is measured at some vertex V_i of a contour along the direction of the corresponding $\hat{r}_i$. Since the vertices of the automatically segmented contour are equally spaced, one would expect to obtain uniform samples of the distance between the two boundaries when choosing to measure the distance at all the vertices of the automatically segmented contour. However, there are two problems associated to this scheme: First, if a vertex on the automatically segmented contour is outside the manually segmented contour, there is no guarantee that we could find a point on the manually segmented contour along the direction of $\hat{r}_i$. The distance could not be defined in this situation. (see Figure 77(a)) Second, if a bulgy or circular shape is mistakenly detected by the segmentation algorithm, the lines along the direction of the radial vectors $\hat{r}_i$ associated to the vertices on the circular segment may intersect the manually defined contour in approximately the same location. This effect causes the oversampling of the distance between the two boundaries around the circular shape, which results in the unfair assignment of the mean deviation. (see Figure 77(b))

The two problems described occur because one or several neighbouring vertices of the automatically defined contour are outside the manually defined contour. These problems could be solved if the distance between two boundaries is always measured from the vertices on the inner contour to the outer contour along the direction of the vertices' radial vectors. (see Figure 77(b) and Figure 78(b))

The set of distances measured is represented by $\left\{ \left| d(V_1) \right|, \left| d(V_2) \right|, \cdots, \left| d(V_{num}) \right| \right\}$, where *num* denotes the total number of vertices chosen for distance measurement. The MAD and MAXD are defined the same way as in (5.2) and (5.3), except θ_i is replaced by V_i and *N* is replaced by *num*. An additional parameter we would like to evaluate is the percentage of values in the set $\left\{ \left| d(V_1) \right|, \left| d(V_2) \right|, \cdots, \left| d(V_{num}) \right| \right\}$ that is smaller than 4 pixels. This can be viewed as the percentage of points that are very close to the manually outlined contour.

In Section 5.3.1, we will show the computation of these quantities for the contour associated to the second order spline wavelet of an ultrasound image.

(a) **(b)**
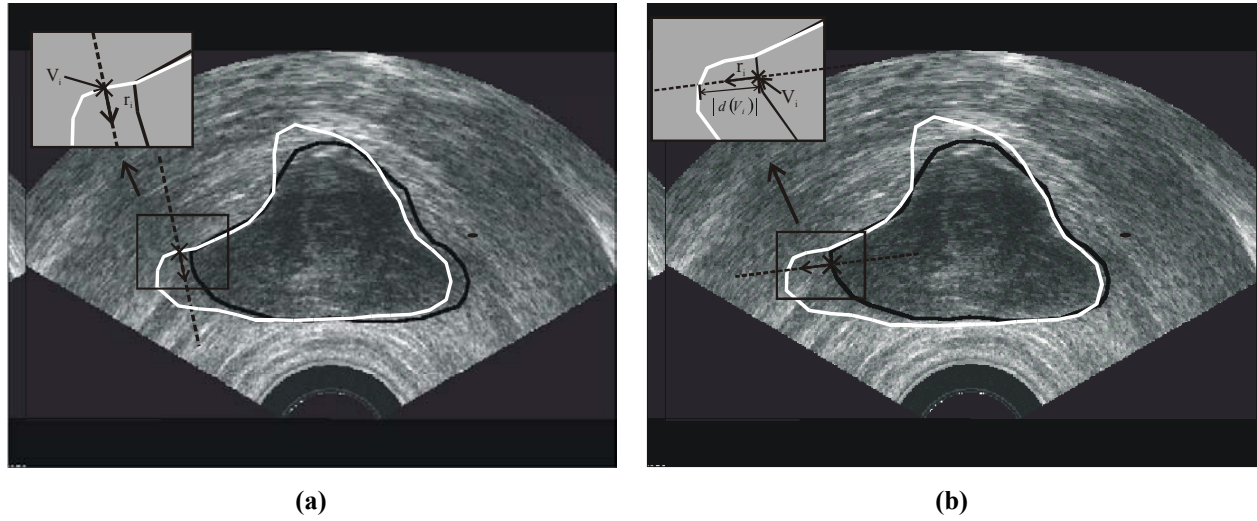
Figure 77 **(a) The distance from a vertex on the automatically defined contour (in white) to the manually defined boundary (in black) along the direction of $\hat{r}_i$ associated to the vertex is undefined; (b) The solution is to sample the distance of the two boundaries by choosing a vertex from the manually defined contour, which is inside the automatically defined boundary around that region, rather than a vertex from the automatic boundary, and measuring the distance from the chosen vertex to the automatic boundary along the direction of $\hat{r}_i$ associated to the chosen vertex.**

(a) **(b)**
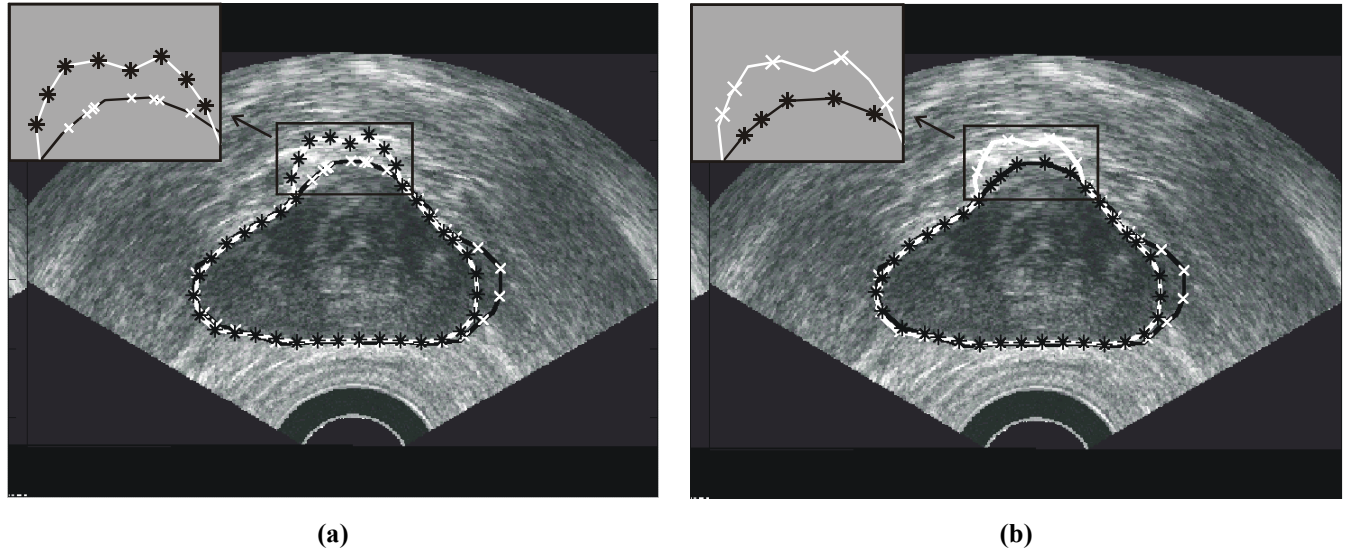
Figure 78 **(a) Distance between the two boundaries is measured from a vertex (marked by asterisks) on the automatically defined contour (in white) to the manually segmented boundary along the direction of $\hat{r}_i$ associated to the vertex. The white crosses represent the point where the line along the direction of $\hat{r}_i$ of a vertex intersects the manually segmented contour. If a bulgy shape outside the manual outline is mistakenly detected, the distance between the two boundaries is oversampled around the bulgy shape; (b) This problem is solved if the distance is measured from vertices on the manual outline, which are inside the automatic contour, to the automatic contour.**

5.3.   Results

In the proposed algorithm introduced in Section 5.1, segmented boundary can be obtained either automatically or semi-automatically. In this section, the boundaries obtained under these two conditions are evaluated.

The presentation of the results is organized in the following way: In Section 5.3.1, we will show the accuracy of the contours generated for a single image by the proposed semi-automatic and the automatic segmentation algorithm. The energy fields used in generating these two sets of results are based on the 2nd order spline wavelet. In Section 5.3.2, Part A, we will compare the performance of the proposed semi-automatic algorithm with the UWO algorithm in 8 randomly chosen test images. In this part, the 2nd order wavelet is used in generating the required energy fields for the proposed algorithm. In Part B of this section, we would like to know whether there is an improvement in the accuracy of the segmented contour if the input energy fields of the segmentation algorithm is generated based on different wavelets, and if so, how much the accuracy is improved. In this part, we use 14 different wavelets in generating the approximate and detailed coefficients for each image: spline wavelets of order 1 to 7, Symlets of order 4, 6 and 8, and Coiflets of order 1 to 4. The three quantities introduced in Section 5.2.2 will be calculated for all 14 contours associated with the 8 test images. The average performance parameters correspond to each wavelet will be tabulated. In Section 5.3.3, the accuracy of the boundary outlined by the proposed automatic algorithm in the 8 test images will be tabulated. The energy fields used in this section are based on the 2nd order spline wavelet.

5.3.1.   The Accuracy of the Contour for One Image

A) Semi-Automatic Segmentation Algorithm

Figure 79 shows the semi-automatically outlined boundary defined by the proposed algorithm. This boundary is compared to the one generated by the UWO algorithm introduced in Section 3.3, also a semi-automatic segmentation algorithm. The three accuracy indicators introduced in Section 5.2.2 − (a) MAD, (b) MAXD and (c) % of points where $|d(V_i)| < 4$ pixels − are computed for both contour, and the results are tabulated below:

	MAD (in pixels)	MAXD (in pixels)	% of points where $|d(V_i)| < 4$ pixels
The proposed semi-automatic algorithm	1.6357	5.4226	90.9091
UWO algorithm	4.1913	20.2452	74.3590

It can be observed that the improvement of performance in all three aspects is very significant. It can be explained by the fact that the initial contour generated by the semi-automatic initialization process introduced in Section 5.1.1 is much more accurate than that generated using the initialization process described in Section 3.3.1.

B) Fully-Automatic Segmentation Algorithm

Figure 80 compares the fully-automatically generated boundary with the manually outlined boundary. The three performance indicators associated to the fully-automatically outlined contour are tabulated in the following:

MAD (in pixels)	MAXD (in pixels)	% of points where $|d(V_i)| < 4$ pixels
1.8247	6.1485	90.9091

The result here is remarkable. For this image, the accuracy of the contour generated automatically is comparable to that generated semi-automatically. In fact, for this image, the accuracy of the automatically generated contour is better than that of the semi-automatically defined contour generated by the UWO algorithm. However, we should test the algorithm in a wider variety of images before a general conclusion is made.

Figure 79   The contour determined by the proposed semi-automatic algorithm (in black solid line) is much more accurate than that generated using the UWO algorithm (in black dotted line). The contour in white is the manual outline.

Figure 80   The automatically defined contour (in black) is compared with the manually outlined contour (in white).

5.3.2.   The Accuracy of the Semi-Automatically Outlined Contour for the Entire Set of Images

A) The Accuracy of the Contours Generated Based on the 2nd Order Spline Wavelet

The following tables show the accuracy indicators of the contour generated by the proposed semi-automatic algorithm and the UWO algorithm for 8 randomly chosen test images:

Image No.	MAD (in pixel)	MAXD (in pixel)	% of points where $\left|d(V_i)\right| < 4$ pixels
1	4.0170	16.1173	60.5263
2	2.3327	8.1926	74.4186
3	1.8704	5.7542	84.2105
4	3.2844	9.3350	64.2857
5	2.9317	7.9685	69.4444
6	1.7035	5.5453	94.4444
7	1.6357	5.4226	90.9091
8	1.7610	6.3826	90.3226
Mean	2.4421	8.0898	78.5702

Table 2   The accuracy parameters associated to the proposed semi-automatic algorithm

Image No.	MAD (in pixel)	MAXD (in pixel)	% of points where $\left| d(V_i) \right| < 4$ pixels
1	3.3765	16.0556	69.2308
2	4.2506	19.1801	68.4211
3	3.2786	18.3091	76.3158
4	2.7108	7.6951	64.2857
5	2.6840	9.0332	75.0000
6	1.9802	5.1042	88.8889
7	4.1913	20.2452	74.3590
8	2.7688	13.1185	86.2069
Mean	3.1551	13.5926	75.3385

Table 3   The accuracy parameters associated to the UWO semi-automatic algorithm

The boundary generated by the proposed semi-automatic segmentation algorithm is more accurate that that generated by the UWO algorithm in most of the test images – Image 2, 3, 6, 7 and 8 – whereas the accuracy of the contours generated using these two algorithms are effectively the same in the remaining images. It is because the prostates in Image 1, 4 and 5 have an elliptical shape that is closely approximated by the spline interpolation process introduced in Section 3.3.1, and therefore, performing the semi-automatic initialization process described in Section 5.1.1 does not offer much improvement, whereas the prostate boundaries in Image 2, 3, 6, 7 and 8 have some irregular features, such as a bulge, and the initial contour obtained using the proposed semi-automatic initialization process is significantly more accurate than that generated using the spline interpolation process. It is reasonable to expect that the shape of the prostate in most of the ultrasound image is irregular because it is likely that something abnormal is felt by the patient before the ultrasound images are taken for his prostate, and the abnormality tends to have an irregular shape. Therefore, the proposed semi-automatic segmentation algorithm should perform better in most ultrasound prostate images.

B) The Accuracy of the Contours Generated Based on Different Wavelets

Types of Wavelet	MAD (in pixel)	MAXD (in pixel)	% of points where $|d(V_i)| < 4$ pixels
Spline1	2.6331	8.2135	76.9839
Spline2	2.4421	8.0898	78.5702
Spline3	2.3282	7.7859	77.9449
Spline4	2.3536	7.7187	79.6802
Spline5	2.3117	7.1934	78.8496
Spline6	2.4609	8.3381	76.9000
Spline7	2.5197	8.8974	77.6454
Symlet4	2.6959	8.3915	71.5284
Symlet6	2.5861	8.4587	76.4143
Symlet8	2.7259	9.0345	73.3490
Coiflet1	2.5223	8.9134	78.3918
Coiflet2	2.4723	8.1749	76.5851
Coiflet3	2.5659	8.3783	75.0271
Coiflet4	2.5863	8.4295	73.5920

Table 4   The average accuracy parameters associated to the Spline-based, Symlet-based and Coiflet-based semi-automatic algorithm

The way the data are obtained in Table 4 was described in the introduction of this section. In this table, one can observe that, in the framework of the proposed semi-automatic algorithm, the resulting contour is more accurate than the one segmented by the UWO algorithm regardless of what wavelet is used. However, we also observe that the difference in the accuracy of the contour generated based on different wavelets is not very significant. Therefore, in testing the fully-automatic algorithm, we reduce the size of the experiment by using only the 2nd order spline wavelet, the one used by Mallat [5] originally.

5.3.3. The Accuracy of the Automatically Outlined Contour for the Entire Set of Images

The same 8 randomly chosen images used in testing the semi-automatic algorithm are used in testing the automatic algorithm. The accuracy indicators are tabulated below:

Image No.	MAD (in pixel)	MAXD (in pixel)	% of points where $\left|d(V_i)\right| < 4$ pixels
1	3.1779	14.3904	73.1707
2	2.4829	7.9437	74.4186
3	5.0793	15.0730	59.5238
4	4.9061	10.8015	38.4615
5	4.7708	13.3472	39.4737
6	2.4828	5.4513	94.1176
7	2.0700	7.7537	86.4865
8	4.5772	20.6382	67.7419
Mean	3.6934	11.9249	66.6743

Table 5   The accuracy parameters associated to the fully-automatic algorithm

If compared with the semi-automatic algorithms (either the UWO or the proposed one), the above data show that the automatic algorithm defines the contour less accurately. However, when the benefits associated to a fully-automatic segmentation algorithm and the accuracy shown in Table 5 are considered together, the results presented are very remarkable.

5.4.   Summary and Remarks

The purposes of this chapter are to construct a segmentation algorithm using the building blocks introduced in Chapter 3 and Chapter 4, and to evaluate the accuracy of the contour defined by this algorithm.

At the beginning of this thesis, the focus is on finding a semi-automatic segmentation algorithm that performs better than the UWO algorithm described in Section 3.3. The result of this effort is the semi-automatic part of the proposed algorithm presented in Section 5.1. Since this semi-automatic algorithm requires human supervision only in the initialization stage, it will become a fully-automatic algorithm if the initialization mechanism is made automatic. With this idea and the significant benefits of an automatic initialization method in mind, an effort is made in developing a fully-automatic initialization procedure. The end-product of this effort is the fully-automatic part of the proposed algorithm.

To evaluate the performance of the proposed algorithm, a quantitative metric should be used. Ladak [8] defined an evaluative metric that measures the distance between the automatically defined contour and the manually defined contour. However, as discussed

in Section 5.2, this distance-based metric poorly reflects the degree of the contour's accuracy under some conditions. To solve this problem, a new distance-based metric was developed. This metric is described in Section 5.2.2 and used as the metric to evaluate the accuracy of the contour defined by the proposed algorithm in Section 5.3.

The results presented in Section 5.3 suggest two important achievements of the proposed algorithm: First, the proposed semi-automatic algorithm performs better than the UWO algorithm, regardless of the type of wavelet used in computing the energy fields. Second, the accuracy of the automatically defined contour is comparable to the semi-automatically defined one. The benefits an automatic algorithm has over a semi-automatic algorithm and the accuracy of the automatically defined contour make the proposed automatic segmentation algorithm very attractive in the medical industry.

Chapter 6

Conclusion and Future Work

The major challenge of segmenting the prostate boundary in an ultrasound image is that it is difficult to differentiate between edge feature that is part of the boundary and the false edges – edges that come from different kinds of noise or imaging artifacts.

In this work, we first looked at the approach proposed by Ladak [8], which is implemented in a program we called "UWO program". In this program, the differentiation between the "actual" and the "false" edges is made semi-automatically. This program requires an expert observer to enter four initial points that are very close to the actual prostate boundary. The spline interpolation method introduced in Section 3.3.1 is used to form an initial contour from the four initial points. By running the DDC algorithm, the initial contour is attracted to the strong edge features in its neighbourhood. This segmentation algorithm relies very strongly on the initial contour (as the algorithm basically defines the prostate boundary to be the set of edge features that is closest to the initial contour), but the initial contour defined is not likely to be accurate. This inaccuracy is caused by an extrinsic and an intrinsic factor. The extrinsic factor is the inaccuracy of the initial points the user enters. As mentioned in Chapter 1, a certain level of expertise is required to outline the prostate boundary in an ultrasound image. Therefore, it is not likely for a general user to pinpoint the four initial points accurately. However, even if the initial points are identified by a professional radiologist, the spline interpolated contour does not necessarily define the boundary of the prostate with reasonable accuracy. Basically, the spline interpolation method just defines an ellipse (or a circle) by connecting the initial points. Large deviation occurs if there are irregular patterns, such as a tumour, in a prostate.

The proposed semi-automatic algorithm approaches the problem more intelligently. First of all, it can be observed that the deviation of the initial contour usually occurs near the top of the prostate. Also, it is observed that the intensity contrast between the interior and the exterior is often very high around the top of the prostate. As explained in Section 4.3, an energy field based on the intensity of the smoothed image can be used to drive the DDC model. It is discovered that by using the spline interpolated contour as the initial contour of a DDC run that is driven by the new energy field, it is possible to generate a contour that is closer to the manually defined contour. It is expected that if this new contour is used as the initial contour of the gradient-modulus-based DDC run, a more accurate boundary can be defined. This is the rationale behind the design of the proposed semi-automatic algorithm. By comparing the results obtained in Table 2 and Table 3, it can be observed that the proposed semi-automatic algorithm performs better than the UWO algorithm in terms of the three accuracy indicators introduced in Section 5.2.

Rather than requiring an expert observer to pick four initial points, if we could design a mechanism that initializes the contour automatically, the proposed semi-automatic algorithm will become a completely automatic one. Designing an automatic initialization algorithm is the most difficult part in the segmentation problem. There are two reasons

for this: First, the initial points selected by the algorithm should be likely to lie on the prostate boundary, rather than on the false edges. Second, enough initial points should be chosen so that the initial contour is accurate everywhere on the prostate boundary. The design of an automatic initialization algorithm was described in Section 4.4. One deficiency in the proposed algorithm is that it cannot identify many initial points on a relatively weak edge segment of the prostate boundary. This problem is partially solved by the reinitialization algorithm, which defines the initial contour of the second DDC run by first taking the final contour of the first DDC run, and then replacing its vertices that lie on weak edge by that lie on strong edge in the neighbourhood. However, this technique does not solve the whole problem because it is possible for the contour to be attracted to false – but strong – edges. A more effective approach would be to find out classifier for differentiating the "actual" and the "false" edges based on the geometry of the edges.

The boundary defined by the proposed automatic segmentation algorithm is reasonably accurate, as demonstrated by the fact that the mean absolute deviation (MAD) is less than 4 pixels and more than 65% of the points on the automatically defined contour are deviated from the professionally outlined contour by less than 4 pixels when tested with a set of 8 ultrasound images. An automatic segmentation tool that achieves this level of accuracy is very attractive in the medical industry because of the following three reasons: First, the efficiency of prostate segmentation will be improved; Second, the segmentation process will be less dependent on the expert observer; Third, the tool can be used to train non-professional health care practitioner, and therefore, the shortage of the skill required for prostate segmentation can be alleviated.

The future extension of this thesis can go in two directions: (1) To simplify the automatic initialization algorithm and to improve the accuracy of the initial contour; (2) to develop a program that can compute the volume of the prostate based on the 2D segmented boundaries.

The automatic initialization algorithm shown in Section 4.4 is rather complicated. It is the case because the algorithm is designed in a way so that it avoids detecting points that lie on false edges. Recall that a set of point is qualified to be possible initial points in Section 4.4.3. If one can design an algorithm that differentiates the "actual" and the "false" edges according to their shape and disqualify point that lie on the false edge – in the same way as the points near the border are disqualified, it is not necessary for him to be restricted to choose points with a certain distance from the centre as we are now in the global initialization process, and it is likely for him to find enough points to define the contour without going through the local initialization process. Also, he can make his choice of possible initial points more aggressively. In Section 4.4.3, a point is qualified to be a possible initial point if the gradient modulus at the location of the point is higher than a certain threshold. If this threshold is set lower, more points are qualified and, in our case, the probability of qualifying points that lie on false edge is higher. This is not a problem in his case because his algorithm disqualifies points lying on the false edge anyways. Therefore, by setting the threshold lower, initial points can be identified on the

relatively weak edge segment on the prostate boundary, and therefore, the accuracy of the initial contour can be improved.

In this work, we have designed a rather accurate tool in segmenting the prostate boundary in a 2D ultrasound image. This algorithm can be easily applied to segment the boundary in 100 2D ultrasound images taken for a patient. By interpolating these 100 boundaries, the information about the shape and the volume of the patient's prostate can be obtained. A program that can perform this task is valuable to the medical industry, as the doctor can get important information about the prostate automatically and quickly.

A. Appendices

A.1. Derivation of the Fast Dyadic Wavelet Transform for 2-D Signals

The 2-D fast dyadic wavelet transform algorithm is stated without proof in Section 3.2.3, Part B). In this section, we will prove the algorithm stated in (3.31) to (3.33):

$$a_{j+1} = a_j * \left(\underline{h}_j, \underline{h}_j \right),$$

(3.31)

$$d_{j+1}^1 = a_j * \left(\underline{g}_j, \delta \right),$$

(3.32)

$$d_{j+1}^2 = a_j * \left(\delta, \underline{g}_j \right).$$

(3.33)

According to (3.16), a_{j+1} is defined to be

$$a_{j+1}[m,n] = < f(x,y), \phi_{2^{j+1}}(x-m, y-n) >$$

Equivalently, a_{j+1} can be expressed as the result of a 2-D convolution:

$$a_{j+1}[m,n] = \left\{ f * \underline{\phi}_{2^{j+1}} \right\}[m,n]$$

(A.1)

where $\underline{\phi}_{2^j}(x,y) = \phi_{2^j}(-x,-y)$.

Since we have the following Fourier transform pair

$$\left\{ f * \underline{\phi}_{2^{j+1}} \right\}(x,y) \overset{F}{\Leftrightarrow} f^F(\omega_x, \omega_y) \overline{\phi_{2^{j+1}}^F}(\omega_x, \omega_y),$$

using the 2-D version of the sampling theorem yields

$$\left\{ f * \underline{\phi}_{2^{j+1}} \right\}[m,n] \overset{f}{\Leftrightarrow} \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f^F\left(\theta_x + 2\pi k, \theta_y + 2\pi l \right) \overline{\phi_{2^{j+1}}^F}\left(\theta_x + 2\pi k, \theta_y + 2\pi l \right)$$

(A.2)

This is equivalent to claim that the Fourier transform of the 2-D discrete signal a_{j+1} is

$$a_{j+1}^f(\theta_x,\theta_y) = \sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} f^F(\theta_x + 2\pi k,\theta_y + 2\pi l)\overline{\phi_{2^{j+1}}^F}(\theta_x + 2\pi k,\theta_y + 2\pi l)$$

(A.3)

Recall that the scaling sequence h and the wavelet sequence g have been defined in (3.18). Taking the Fourier transform of this equation yields

$$\varphi^F(\omega) = \frac{1}{\sqrt{2}}\varphi^F\left(\frac{\omega}{2}\right)\sum_{k\in Z} h[k]e^{-j\left(\frac{\omega}{2}\right)k} = \frac{1}{\sqrt{2}}\varphi^F\left(\frac{\omega}{2}\right)h^f\left(\frac{\omega}{2}\right)$$

(A.4)

$$\psi^F(\omega) = \frac{1}{\sqrt{2}}\varphi^F\left(\frac{\omega}{2}\right)\sum_{k\in Z} g[k]e^{-j\left(\frac{\omega}{2}\right)k} = \frac{1}{\sqrt{2}}\varphi^F\left(\frac{\omega}{2}\right)g^f\left(\frac{\omega}{2}\right)$$

(A.5)

By noting the fact that $\phi(x,y)$ can be written as separable products of one-dimensional scaling function φ according to (3.28) and using the relation that have just been stated in (A.4), one should be able to express $\phi_{2^{j+1}}^F(\omega_x,\omega_y)$ in terms of $\phi_{2^j}^F(\omega_x,\omega_y)$ in the following way:

$$\begin{aligned}\phi_{2^{j+1}}^F(\omega_x,\omega_y) &= \varphi_{2^{j+1}}^F(\omega_x)\varphi_{2^{j+1}}^F(\omega_y) = \sqrt{2^{j+1}}\varphi^F(2^{j+1}\omega_x)\sqrt{2^{j+1}}\varphi^F(2^{j+1}\omega_y)\\ &= 2^j h^f(2^j\omega_x)h^f(2^j\omega_y)\varphi^F(2^j\omega_x)\varphi^F(2^j\omega_y)\\ &= h^f(2^j\omega_x)h^f(2^j\omega_y)\phi_{2^j}^F(\omega_x,\omega_y)\end{aligned}$$

(A.6)

where we have make use of the fact $\phi_{2^j}^F(\omega_x,\omega_y) = 2^j\varphi^F(2^j\omega_x)\varphi^F(2^j\omega_y)$ in the last equality.

By observing (A.3), one easily sees that $a_{j+1}^f(\theta_x,\theta_y)$ equals to the function $f^F(\theta_x,\theta_y)\overline{\phi_{2^{j+1}}^F}(\theta_x,\theta_y)$ and its 2π-periodic extension, both in the θ_x and θ_y directions. One of the periodic extensions of this function would be

$$f^F(\theta_x + 2\pi k,\theta_y + 2\pi l)\overline{\phi_{2^{j+1}}^F}(\theta_x + 2\pi k,\theta_y + 2\pi l)$$

(A.7)

Using the result stated in (A.6), (A.7) can be expressed as

$$f^F\left(\theta_x + 2\pi k, \theta_y + 2\pi l\right)\overline{h^f}\left(2^j\left(\theta_x + 2\pi k\right)\right)\overline{h^f}\left(2^j\left(\theta_y + 2\pi l\right)\right)\overline{\phi^F_{2^j}}\left(\theta_x + 2\pi k, \theta_y + 2\pi l\right)$$

(A.8)

Since h^f is 2π-periodic and $j \geq 0$ (i.e., $2^j \geq 1$), (A.8) can be written as

$$\overline{h^f}\left(2^j\,\theta_x\right)\overline{h^f}\left(2^j\,\theta_y\right)f^F\left(\theta_x + 2\pi k, \theta_y + 2\pi l\right)\overline{\phi^F_{2^j}}\left(\theta_x + 2\pi k, \theta_y + 2\pi l\right)$$

(A.9)

Using this result, (A.3) is written as

$$
\begin{aligned}
a^f_{j+1}\left(\theta_x, \theta_y\right) &= \overline{h^f}\left(2^j\,\theta_x\right)\overline{h^f}\left(2^j\,\theta_y\right)\sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}f^F\left(\theta_x + 2\pi k, \theta_y + 2\pi l\right)\overline{\phi^F_{2^j}}\left(\theta_x + 2\pi k, \theta_y + 2\pi l\right) \\
&= \overline{h^f}\left(2^j\,\theta_x\right)\overline{h^f}\left(2^j\,\theta_y\right)a^f_j\left(\theta_x, \theta_y\right)
\end{aligned}
$$

(A.10)

This is to say that in order to get a_{j+1} from a_j, we need a 2-D filter $H[m,n]$ that has a transfer function $\overline{h^f}\left(2^j\,\theta_x\right)\overline{h^f}\left(2^j\,\theta_y\right)$. It is clear that $H[m,n]$ is equal to $\overline{h_j}[m]\overline{h_j}[n]$. This completes the proof of (3.31). (3.32) and (3.33) are proven almost exactly in the same way and their proofs are not shown here.

A.2. Scaling Filter Coefficients of Dyadic Wavelets Derived in Section 4.2

n	$h_0[n]/\sqrt{2}$	$h_1[n]/\sqrt{2}$	$h_2[n]/\sqrt{2}$	$h_3[n]/\sqrt{2}$	$h_4[n]/\sqrt{2}$	$h_5[n]/\sqrt{2}$	$h_6[n]/\sqrt{2}$	$h_7[n]/\sqrt{2}$
-4								0.00390625
-3						0.015625	0.0078125	0.03125
-2				0.0625	0.03125	0.09375	0.0546875	0.109375
-1		0.25	0.125	0.25	0.15625	0.234375	0.1640625	0.21875
0	0.5	0.5	0.375	0.375	0.3125	0.3125	0.2734375	0.2734375
1	0.5	0.25	0.375	0.25	0.3125	0.234375	0.2734375	0.21875
2			0.125	0.0625	0.15625	0.09375	0.1640625	0.109375
3					0.03125	0.015625	0.0546875	0.03125
4							0.0078125	0.00390625

Table 6   Scaling Filter Coefficients of the Spline Dyadic Wavelets of order 0 to 7

n	$h_4[n]$	$h_5[n]$	$h_6[n]$	$h_7[n]$	$h_8[n]$
-8					0.0019
-7					-0.0003
-6			-0.0078		-0.015
-5		0.0195	0.0018	0.0103	0.0038
-4	0.0322	-0.0211	0.0447	0.004	0.0491
-3	-0.0126	-0.1753	-0.0211	-0.1078	-0.0272
-2	-0.0992	0.0166	-0.0726	-0.14	-0.0519
-1	0.2979	0.634	0.3379	0.2886	0.3644
0	0.8037	0.7234	0.7876	0.7678	0.7772
1	0.4976	0.1994	0.4911	0.5361	0.4814
2	-0.0296	-0.0391	-0.0483	0.0174	-0.0613
3	-0.0758	0.0295	-0.118	-0.0496	-0.1433
4		0.0273	0.0035	0.0679	0.0076
5			0.0154	0.0305	0.0317
6				-0.0126	-0.0005
7				-0.001	-0.0034
8				0.0027	

Table 7   Scaling Filter Coefficients of the Symlets of order 4 to 7

n	$h_1[n]$	$h_2[n]$	$h_3[n]$	$h_4[n]$
-8				0.000892314
-7				-0.001629492
-6			-0.003793513	-0.007346166
-5			0.007782596	0.016068944
-4		0.016387336	0.023452696	0.0266823
-3		-0.041464937	-0.065771911	-0.0812667
-2	-0.07273262	-0.067372555	-0.06112339	-0.056077313
-1	0.337897662	0.386110067	0.405176902	0.415308407
0	0.85257202	0.812723635	0.793777223	0.782238931
1	0.384864847	0.417005184	0.428483476	0.434386056
2	-0.07273262	-0.076488599	-0.071799822	-0.066627474
3	-0.015655728	-0.059434419	-0.082301927	-0.096220442
4		0.023680172	0.034555028	0.039334427
5		0.005611435	0.015880545	0.025082262
6		-0.001823209	-0.009007976	-0.015211732
7		-0.000720549	-0.002574518	-0.005658287
8			0.001117519	0.003751436
9			0.000466217	0.001266562
10			-7.09833E-05	-0.000589021
11			-3.45998E-05	-0.000259975
12				6.2339E-05
13				3.12299E-05
14				-3.25968E-06
15				-1.78499E-06

Table 8   Scaling Filter Coefficients of the Coiflets of order 1 to 4

A.3.   Test Images Used in Section 5.3

(a)

(b)

(c)

(d)

(e)

(f)

(g) **(h)**
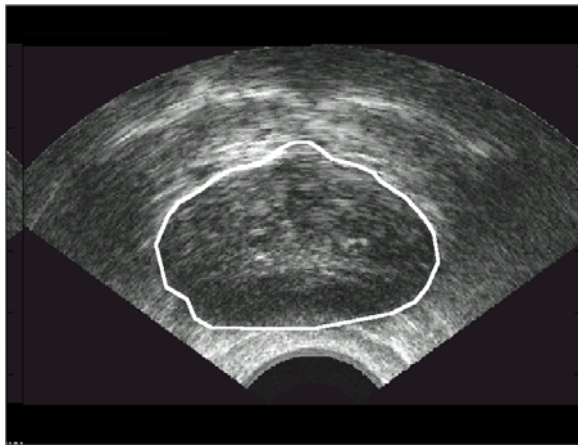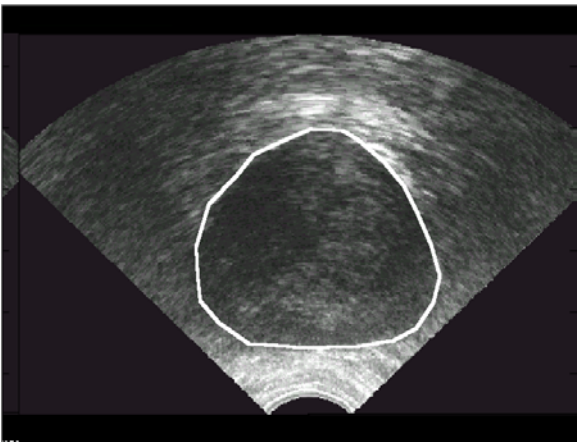
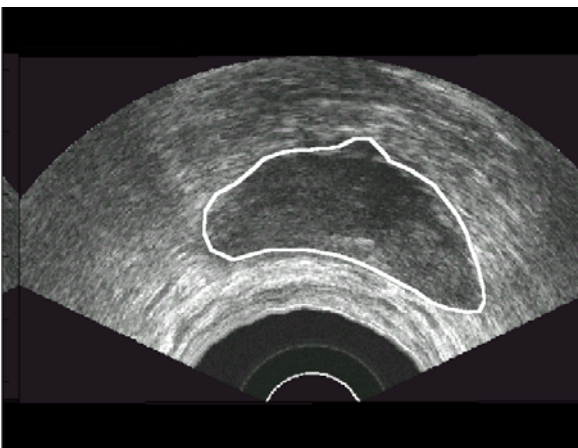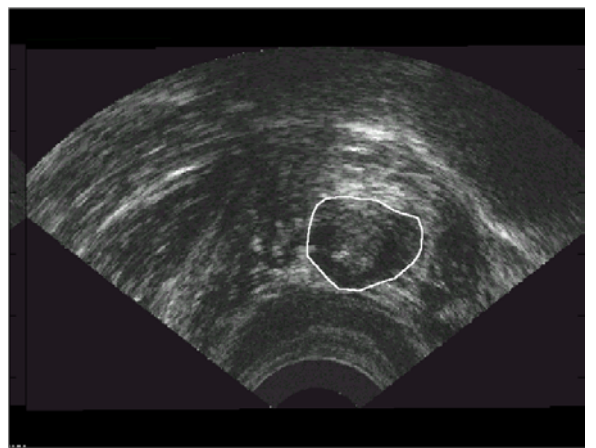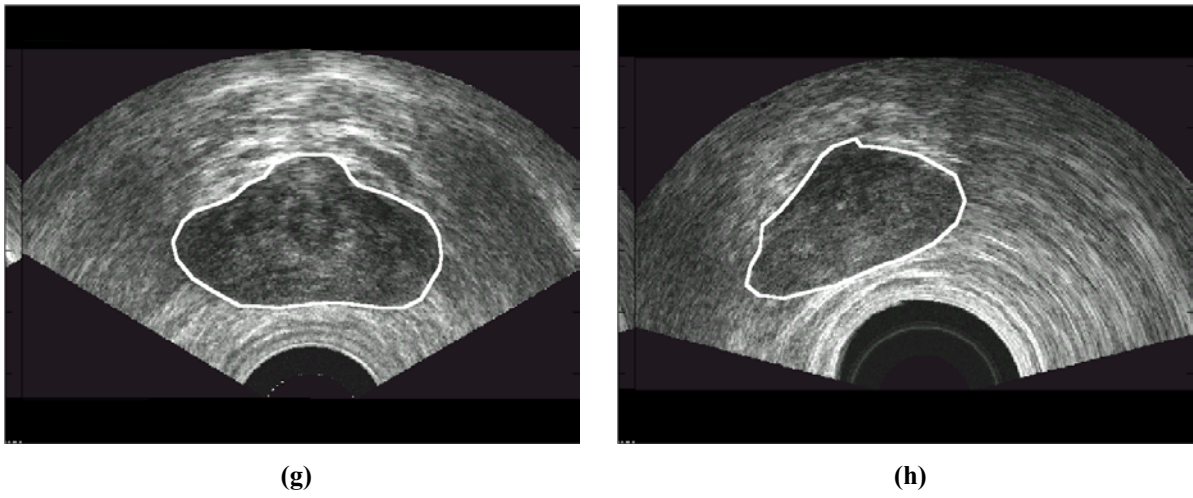**Figure 81** **Ultrasound prostate images (a) to (g) shown here are the test images No. 1- 8 used in Section 5.3 respectively.**

References

[1] S. Mallat, *A wavelet tour of signal processing*, New York: Academic Press, 1998.

[2] S. Mallat, *A wavelet tour of signal processing*. New York: Academic Press, 1999, pp. 198-199.

[3] M. Frazier, *An Introduction to wavelets through linear algebra*, New York: Springer-Verlag, 1999.

[4] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674-693, July 1989.

[5] S. Mallat and S. Zhong, "Characterization of Signals from Multiscale Edges," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 710-732, July 1992.

[6] R. Bartels, J. Beatty and B. Barsky, *An introduction to the use of splines in computer graphics*, 1983. Chapter 3.

[7] C. de Boor, *A Practical Guide to Splines*, Applied Mathematical Sciences vol. 27, Revised Edition, New York: Springer-Verlag, , 2001.

[8] S. Lobregt and M. Viergever, "A discrete dynamic contour model," *IEEE Trans. on Medical Imaging*, vol. 14, no. 1, pp.12-24, Mar. 1995.

[9] H. Ladak, F. Mao, Y. Wang, et al., "Prostate boundary segmentation from 2D ultrasound images," *Med. Phys*., vol. 27, pp. 1777-1788, 2000.

[10] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-197, Nov. 1986.

[11] D. Hykes, W. Hedrick and D. Starchman, *Ultrasound Physics and Instrumentation*, Churchill New York: Livingstone Inc., 1985.

[12] American Cancer Society: Cancer Facts and Figures-2002. Atlanta, Ga: American Cancer Society, 2002.

[13] WA Sakr, GP Haas, BF Cassin, et al: "The frequency of carcinoma and intraepithelial neoplasia of the prostate in young male patients". *Journal of Urology 150(2 pt 1),* pp. 379-385, 1993.

[14] FH Schroder, P van der Maas, P Beemsterboer, et al.: "Evaluation of the digital rectal examination as a screening test for prostate cancer". *Journal of the National Cancer Institute 90(23)*, pp. 1817-1823, 1998.

[15] SJ Jacobsen, EJ Bergstralh, SK Katusic, et al.: "Screening digital rectal examination and prostate cancer mortality: a population-based case-control study". *Urology 52(2),* pp. 173-179, 1998.

[16] KE Richert-Boe, LL Humphrey, AG Glass, et al.: "Screening digital rectal examination and prostate cancer mortality: a case-control study". *Journal of Medical Screening 5(2),* pp. 99-103, 1998.

[17] M. Benson, I. Whang, A. Pantuck, et al.: "Prostate specific antigen density: a means of distinguishing benign prostatic hypertrophy and prostate cancer". *Journal of Urology 147,* pp. 815-816; 1992.

[18] J. Hornak, "The Basics of MRI," December 2002. http://www.cis.rit.edu/htbooks/mri.

[19] National Cancer Institute, "Prostate Cancer: Screening," December 2002. http://www.nci.nih.gov/cancerinfo/pdq/screening/prostate/healthprofessional.

[20] National Cancer Institute, "Prostate Cancer: Prevention," December 2002. http://www.nci.nih.gov/cancerinfo/pdq/prevention/prostate/healthprofessional.

[21] M. Rifkin, E. Zerhouni, C. Gatsonis, et al., "Comparison of magnetic resonance imaging and ultrasonography in staging prostate cancer," *The New England Journal of Medicine,* vol. 323, pp. 621-626, Sept. 1990.

[22] J. Hsieh, M. Ko, H. Liao, et al., "A new wavelet-based edge detector via constrained optimization," *Image and Vision Computing*, vol. 15, pp.511-527, 1997.

[23] L. Zhang and P. Bao, "Edge detection by scale multiplication in wavelet domain," *Pattern Recognition Letters*, vol. 23, pp. 1771-1784, 2002.

[24] S. Burgiss, R. Whitaker, and M. Abidi, "Range image segmentation through Pattern analysis of the multiscale wavelet transform," *Digital Signal Processing*, vol. 8, pp. 267-276, 1998.

[25] P. Angel and C. Morris, "Analyzing the Mallat wavelet transform to delineate contour and textural features," *Computer Vision and Image Understanding*, vol. 80, pp, 267-288, 2000.

[26] B. Keserci and H. Yoshida, "Computerized detection of pulmonary nodules in chest radiographs based on morphological features and wavelet snake model," *Medical Image Analysis*, vol. 6, pp. 431-447, 2002.

[27] P. Carre and C. Fernandez-Maloigne, "Use of the angle information in the wavelet transform maxima for image de-noising," *Image and Vision Computing*, vol. 18, pp. 1055-1065, 2000.

[28] C. Knoll, M. Alcaniz, V. Grau, et al., "Outlining of the prostate using snakes with shape restriction based on the wavelet transform," *The Journal of the Pattern Recognition Society*, vol. 32, pp. 1767-1781, Oct. 1999.

[29] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annu. Rev. Biomed. Eng.*, Vol. 2, pp. 315-337, 2000.