CHAPTER ONE
INTRODUCTION

1.1 Background of Study

The prostate gland is a part of the male reproductive system and is located just below the bladder. Prostate cancer is a very serious disease and accounts for greater number of deaths in men. According to the National Cancer Institute of Canada (NCIC), prostate adenocarcinoma is the most common disease in middle aged and elderly men all over the world (Canada Cancer Society [CCS], 2007). Among the malignant tumours, the prostate cancer occupies a prominent place worldwide. The incidence and mortality rates of the disease varies form one geographical location to another. The increase of incidence started from the 1980s. This rapid increase is due to the extensive spread of determination of prostatic specific antigen (PSA) (Gyorgy, 2005).

As a widespread but less aggressive cancer, the early diagnosis of prostate cancer is critically important for the treatment of this disease. Treatment is available, but it is most successful when the cancer is in early stage. Treatment of an advanced cancer especially if it has spread beyond the prostate can be very complex and in many cases incurable. For detection of cancer in early stages, it is important for men over the age of 50 to get an annual screening(Yu et al., 2014).

Presently, several different screening techniques are used to diagnose the presence of the prostate carcinoma, the most common of which are digital rectal examination (DRE), prostatic specific antigen (PSA) blood test, and trans-rectal ultrasound (TRUS)(Pathak et al., 2000). The DRE test is carried out by a skilled physician who manually feels for any abnormalities in the prostate gland through the rectum. The accuracy of this examination is not high. Nevertheless, DRE is inexpensive, easy to perform, and can detect most of the tumours of sufficient volume. PSA is a reliable test for the early detection of cancer. PSA is an enzyme which is secreted by the prostatic cells. High PSA values suggest problems with the prostate gland. Then TRUS is used to display images of the prostate gland for a radiologist to interpret. The sonography professional mentally segment these images to detect the edges ofthe prostate and different tissue textures within the prostate that might

represent prostatecancer.If any of the previous screening modalities suggest any abnormalities, biopsies should be conducted, and then examination by a pathologist (Awad, 2007).

On the other hand, modern imaging and computer techniques open the window for improving diagnostic accuracy of prostate cancer using computer-aided systems. Identification of biological features and the segmentation is done more accurately by applying artificial intelligence methods. Artificial intelligence (AI) encompasses methods that exploit machine learning, reasoning, and problem solving techniques to provide solution to human problems in some specific discipline. AI techniques usage in medicine is useful in storage, data retrieval and optimal use of information analysis for decision making in solving problems (Rasgarpour and Shanbehzadeh, 2011). Medical images play significant role in assisting health care providers to assess patients for diagnosis and treatment. Studying medical images depends mainly on the visual interpretation of the radiologists. However, this consumes time and usually subjective, depending on the experience of the radiologist.

Consequently, the use of computer-aided systems becomes very necessary to overcome these limitations. AI methods such as digital image processing and also its combinations with others like expert systems, machine learning, fuzzy logic and pattern recognition are so valuable in visualization and analysis of medical images (Rasgarpour and Shanbehzadeh, 2011).

Recent advances in the field of artificial intelligence have led to the emergence of expert systems for medical applications. Moreover, in the last few decades computational tools have been designed to improve the experiences and abilities of physicians for making decisions about their patients (Keles and Yavuz, 2011). An expert system (ES) is an intelligent computer system, which contains a well-organized body of knowledge that imitates experts' problem-solving skills to solve complex decision problems in a specific domain. An ES is not based on black-box formulation and it is easier for users to understandits structure (Yunusoglu and Selim, 2013). For example, expert systems in medical application are being used for diagnosis of disease. The patient's information and symptoms are embedded to the system, and the system gives back obtained results such as probable diagnosis, recommended treatments or drugs prescription (Keles and Yavuz, 2011).

Rule based expert systems are the most commonly known type of knowledge based systems. The knowledge is represented in the form of IF-THEN rules (Ciabattoni et al., 2013). Decision making in the field of medical diagnosis also depends upon experience, expertise and perception of the

physicians. The increase of complexity of the system make uncertainty and vagueness become more tangible (Sikchi and Ali, 2013). The uncertainty in the knowledge base is usually represented as linguistic variables or vague numeric values in the rule's antecedents, consequences, or both. Fuzzy logic has brought a drastic change in handling these uncertainties and vagueness in our systems (Leung et al., 2009). Fuzzy rulebased system is a mathematical tool for dealing with the uncertainty and the imprecision typical in medical field. Thereasoning is based on compositional rule of fuzzy inference and the knowledge of specialists is important to determine the parameters (Castanho et al., 2013). In these systems, information gathered from the domain experts must be transferred to knowledge and must be used at the right time. This knowledge can be incorporated in the form of rule based expert system in detection and diagnosis of prostate cancer.

Image analysis is a key tool that guides medical experts in diagnosis of diseases. All image techniques can be grouped under a general framework; Image Engineering (IE), which consists of three layers: image processing (lower layer), image analysis (middle layer), and image understanding (high layer), as shown in Figure 1.1. In recent years, image engineering has formed a new discipline and made great progress. Image segmentation is the first step and also one of the most critical tasks of image analysis, which has the objective of extracting information (represented by data) from an image via image segmentation, object representation, and feature measurement, as shown in Figure 1.1. It is evident that the result of segmentation will have considerable influence over the accuracy of feature measurement (Zhang, 2001). Computerized medical image segmentation plays an important role in medical imaging applications. It is widely used in different applications such as diagnosis, localization of pathology, study of anatomical structure, treatment planning, and computer-integrated surgery. However, medical image segmentation remains a hard problem due to the variability and the complexity of the anatomical structures in the human body (Wahba, 2008). Besides, there is a wide variety of medical imaging modalities. In order to detect prostate cancer, imaging modalities such as Magnetic Resonance Imaging (MRI) and Trans-Rectal UltraSound (TRUS) are relied on. MRI images are more comprehensible than TRUS images which are corrupted by noise such as speckles and shadowing. However, MRI screening is costly, often unavailable in many community hospitals, time consuming, and requires more patient preparation time.

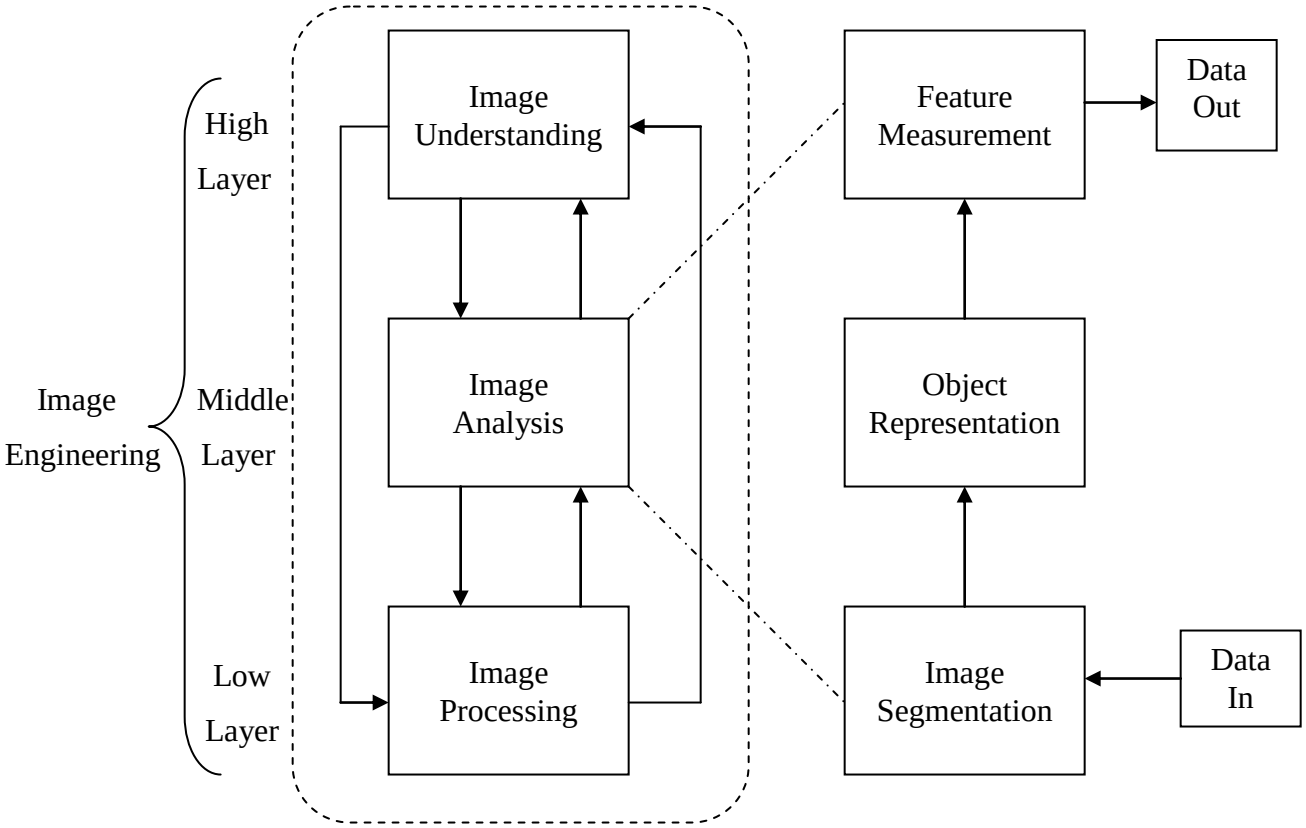


Figure 1.1: Image engineering and image segmentation (Zhang, 2001).

Therefore, TRUS is more popular for screening and biopsy guidance for prostate cancer. It is a safe means of imaging the human body because it emits no ionizing radiation, and it uses low-power sound waves. It is relatively cheaper and portable to handle. For these reasons TRUS images was chosen for this research.

In this dissertation, the interest is in prostate segmentation towards hyperechoic pixel (suspected cancerous cells) detection in ultrasound images. Prostate segmentation is an essential process because it detects prostate area, volumes and boundaries, which both play an important role in the diagnosis of prostate disease and treatment. Moreover, in case of prostate cancer, prostate volumes and boundaries are so important in follow up of the cancer, which may help in reducing death rate.

There are many techniques used for prostate segmentation in ultrasound images. The classical techniques do not depend on shape modelling and knowledge of ultrasound images, while recent techniques focus on the incorporation of prior knowledge about shape and speckle (Noble and Boukerroui, 2006).

1.2 Statement of Problem

There is need to provide a consistently reliable and proven tool for diagnosing prostate cancer since it has become a pandemic. The need also exists to develop a system that goes beyond the competences of human expertise in prostate disease diagnosis by using artificial intelligence. There is need to find a faster and automated means of segmenting medical 2D-images towards prostate cancer diagnosis.

1.3 Aim and Objectives of Research

The aim of this research work is to design and develop an expert system based prostate disease diagnostic tool using image segmentation with the following objectives:

i. To develop a novel algorithm for the segmentation of TRUS 2D-images of the prostate gland towards area estimation.

ii. To develop anovel algorithm for detecting suspected cancerous regions of the prostate gland from TRUS 2D-images and give results of diagnosis.

iii. To develop a novel algorithm for zoning the matrix of TRUS 2D-images of the prostate gland using ratio-based metrics towards localization of suspected cancerouspixels/tissues.

iv. To develope a novel algorithm for capturing expert marking of the segments of the prostate gland and suspected sections of the three zones in a TRUS 2D-image of the prostate.

v. To validate and evaluate the resultsof the algorithm for segmentation of TRUS 2D-images of the prostateand cancer detection using area-based metrics that involves expert's results.

1.4 Justification of Research

This research work is significant in that it has:

i. Prepared an automated, fast and simple diagnostic tool to be used for routine clinical practice.

ii. Proposed a new technique for interpretation of medical images using digital image processing in artificial intelligence.

iii. Provided a tool that facilitates the prevention of frequency of biopsies through accurate segmentation of the prostate gland boundaries to guide the biopsy needles to suspected cancerous sites during treatment procedures.

iv. Presented a platform for assessment and evaluation of the degree of accuracy of predicting cancer status using artificial intelligence techniques.

v. Provided a proven and robust diagnostic support to managers of prostate disease by facilitating the estimation of prostate area/volume to enable follow-up of patients with raised PSA level.

1.5 Scope of Research

This research work covers study of expert systems and image analysis. It is focused on developing a modified algorithm for segmenting 2D-images of the prostate and extracting features for detecting prostate cancer. Also developing an algorithm that enables an expert radiologist segment same image and mark suspected cancerous points on the gland zones.Digital image samples of the format,

jpg is collected and used to test the algorithm. The specification of the image samples used for this work is as follows: Horizontal resolution of 96dpi; Vertical resolution of 96 dpi. Dimensions of the images is within 435 x 308 pixels (4.53 x 3.21 inches) and 270 x 232 pixels (2.81 x 2.42 inches) (See section 2.1.2.4 for resolution determination). The TRUS 2D-images for the test is for prostate gland of patients with raised PSA level (suspected to have cancerous cells).

1.6 Block Diagram Overview of Project Stages

The stages outlining the proposed activities that shall be embarked upon towards the realisation of this project is as shown in Figure 1.2 below.

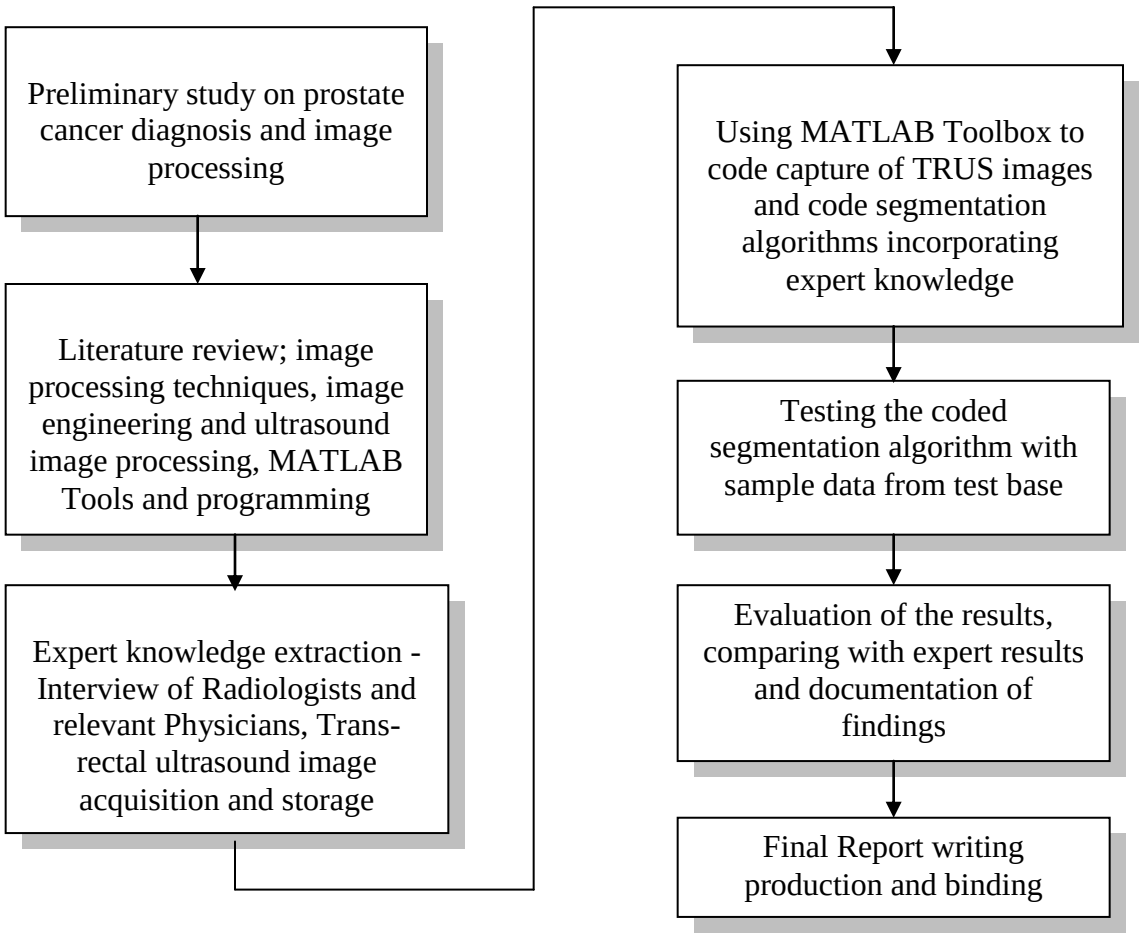


Figure 1.2: Block diagram overview of proposed project stages

1.7 Project Report Organisation

This dissertation is divided into five chapters. Following this chapter, chapter 2 gives a review of related works and literatures. First, an overview of relevant concepts like the prostate gland anatomy, prostate cancer detection/diagnosis, the mechanisms of ultrasound imaging and modalities with special reference to trans-rectal probe, and subsequently carried outa review of the related works wasundertaken. A review of various image segmentation techniques is presented.

Chapter 3 presents the methodology employed in the realisation of the goal of this thesis. The method for evaluating the result of the proposed algorithm is presented. Presented an analysis of the existing system where methods employed in gathering information from different sources and analysis of data gathered for the work was summarized. The limitations of the existing systems were summarized and then a block diagram overview of the proposed new system presented. A description of the proposed enhanced automated region growing technique highlighting the difference between the new and the conventional was also done.

System design is also presented. The design follows a structural approach with the modular structure of the expert system presented for the new system. It summarizes the specifications of the proposed system, defining input/output requirements. The rest of the chapter describes the modules of the new system that performs the functions of the major components of the expert system structure presented. The system graphic user interface design is discussed. Also the image data files and table structures and formats were presented. The image enhancement technique applied, that is use of expert knowledge and sticks filtering are presented. Finally the algorithms for the enhanced region growing segmentation technique as well as the procedure for obtaining the gold standard, to be use as benchmark for evaluation are presented.

Chapter 4 focuses on system implementation and testing. It presents the hardware environment for the implementation of the finished work. System testing using sample TRUS 2D-images was presented. The results were validated using area based metrics which compared the new system with the benchmark, gold standard, or prostate boundary marked by expert radiologist. Finally the systems performance was compared with previous works.

The summary of achievements, problems encountered, contributions to knowledge, suggestions for future work, conclusions, and recommendation are presented in chapter 5.

CHAPTER TWO
LITERATURE REVIEW

2.1 Overview of Relevant Concepts

In this section some of the primary concepts in this research shall be presentedin order to lay the foundation for the discussions that shall follow in the subsequent sections of this dissertation. Three major concepts shall be given attention as follows; prostate cancer diagnosis, medical image segmentation, and ultrasound imaging.

2.1.1 The Prostate Cancer Diagnosis

Prostate cancer (PCa) is the most prevalent adenocarcinoma and the second highest cause of cancer-related deaths in men. Early diagnosis is required to identify the development of PCa to reduce the risk of the disease metastasising to different regions of the body (Conor, 2010).Studies reveal that the probabilityof developing prostate cancer is about 14% of men, whereas the probability of death as aresult of prostate cancer is about 4%(Peng, 2010). Recently, the public has become much more awareof prostate cancer, leading to efforts to improve early detection.

2.1.1.1 Prostate Gland Anatomy

The prostate is an accessory gland of the male reproductive system and a part of a man’s urological system. It is a walnut shaped gland that surrounds the neck of the bladder and the beginning of the urethra.The major function of the prostate is to secrete and store seminal fluid and to regulate urination. The normal adult prostate typically weights 20-30 grams and is 4× 3× 2cm in size(Tewari, 1998).The term prostate is derived from the Greek prohistani which means "to stand in front of ".This expression was adopted by Herophilus of Alexandria in 335 B.C. to describe the organlocated in front of the urinary bladder. Figure 2.1 is an anatomical view of the humanbody, including the prostate, kidneys, bladder, lymph nodes and ureters. The prostate’sanatomy, physiology and

pathology have been described in detail only within the last sixdecades. The prostate gland location is illustrated in Figure 2.2.

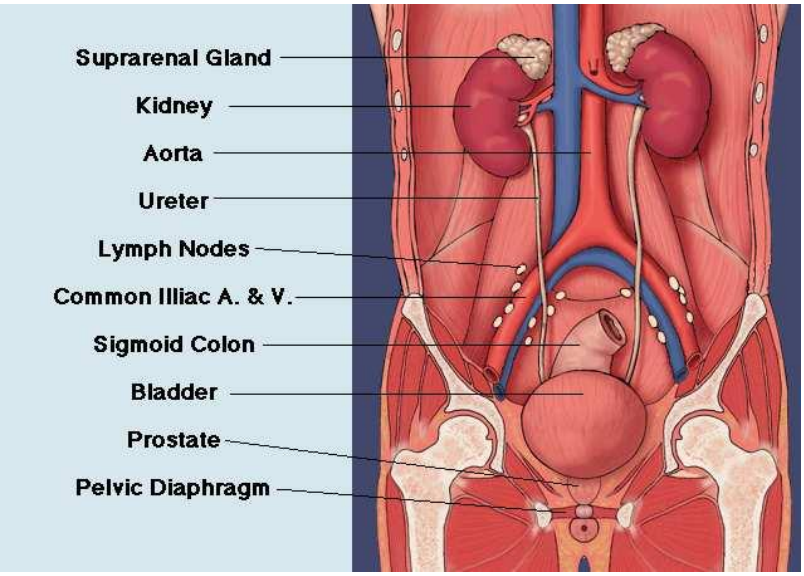


Figure 2.1: Anatomical section shows prostate, bladder, and kidneys (Peng, 2010)

The three major prostatediseases are Benign Prostatic Hyperplasia (BPH), carcinoma of prostate, and prostatitis. Amongst cancer are two other disorders of the prostate which include BPH and prostatitis. According to Kirkham et al. (2006), in MRI imagingthese can be easily confused and exhibit similar artefacts to cancer in the output image.

BPH can be identified in the prostate as early as age forty to fifty years of life. BPH is notonly due to an increase in the cell population, but is also to changes in the architectureof the ducts and acini. Nodular hyperplasia is a characteristic histological feature of BPHand occurs most commonly in the transition zone near the distal end of the bladder necksmooth muscle. Progressive transition zone enlargement results in a mean decline in the urinary flow rate. Nodular hyperplasia in the periurethral zone leads to a mass of dorsaltissue at the bladder neck(Garraway, 1995).

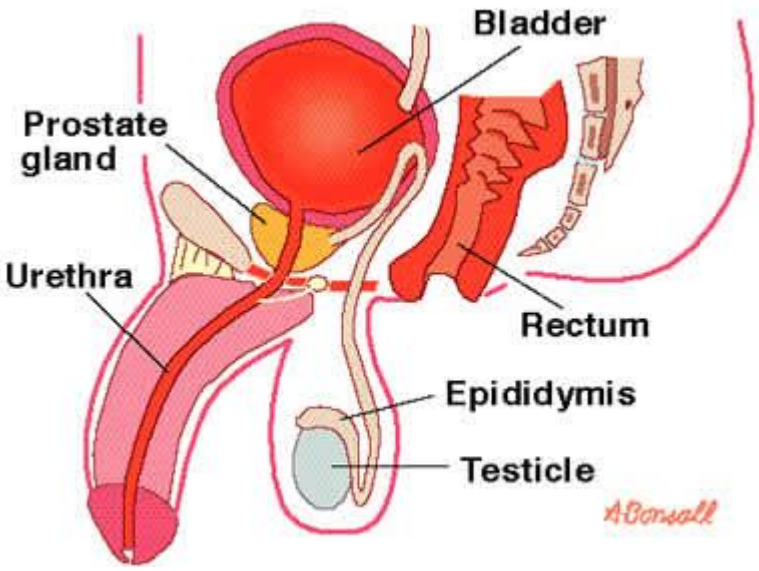


Figure 2.2: Location of the prostate gland (Southern Cross Medical Library [SCML], 2005).

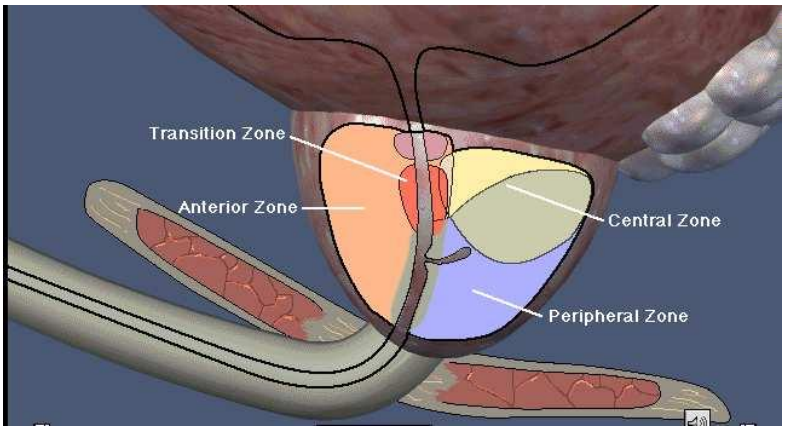


Figure 2.3: Anatomical zones of the prostate gland (Peng, 2010).

Prostatitis is defined as microscopic inflammation of the tissue of the prostate gland, which spansa broad range of clinical conditions. There are 4 types however the most common form in 95%of diagnosis is non-bacterial prostatitis also known as Chronic Pelvic Pain Syndrome (CPPS).It is characterized primarily by pain complaints in the absence of urinary tract infection. It issubdivided into inflammatory and no-inflammatory subtypes. Inflammatory CPPS is definedby the presence of

white cells in the semen, non-inflammatory CPPS is defined by the absenceof white cells. Approximately 8.2% of men have prostatitis at some point in their lives(Hedayati and Keegan, 2009).

McNeal(1968) has identified three anatomical zones: the peripheral zone, transition zone and central zone, in Figure 2.3. The peripheral zone represents approximately 65%of the prostatic volume. This zone extends around the postero-lateral peripheral aspectsof the gland from its apex to its base. The second largest part of the prostate is thecentral zone. It is a cone-shaped region that represents approximately 25% of the prostaticvolume. The third zone of the prostate is called the transition zone, which represents only 5% to 10% of a typical prostatic volume. The transition zone is separated from the other two zones by a narrow band of fibromuscularstroma. It extends, in an arc, from the posterior urethra in the mid-prostateto the most anterior aspect of the gland. The clinical significance of zonal anatomy isbest observed in BPH. Although it is clear that most nodular hyperplasia occurs in thetransition zone, malignancy can affect any or all of the three zones. More than 70% ofcases originate in the peripheral zone (Pathak et al., 2000). The percentage of malignancy in the transitionzone is about 10 - 20%, and in the central zone, about 5- 10%. The normal prostate sizeis 4 - 6 cm in length, 3 - 4 cm in width, and 2 - 3 cm in thickness. Malignancy can cause anincrease in the prostate size, and BPH can increase the prostate size 100 - 200% (Peng, 2010).

2.1.1.2 Prostate Cancer Detection/Diagnosis

The principal diagnostic tools for evaluating the risk for prostate cancer include acombination of Digital Rectal Examination (DRE) and Prostate Specific Antigen (PSA)prompting a decision to perform a Transrectal Ultrasound (TRUS) guided prostate biopsy. The early detection of prostate cancer is pivotal for the success of treatment. It is difficultto detect prostate cancer early, since it does not produce any symptoms in most patients.However, in advanced cases of prostate cancer, there are symptoms such as bladder outletobstruction, acute urinary retention, and neurological symptoms of cord compression or pathologicfractures secondary to bony metastases. The currently used screening tests are asfollow(Peng, 2010),(SCML, 2005):

Digital Rectal Examination (DRE): It is easy and inexpensive, and is the most commontest for prostate cancer detection. This exam can detect localized advanced cancers with sufficient volume. However, the accuracy of the DRE exam is relatively low andis operator-dependent. The percentage of DRE positive predictive value is from 21%to 39%. In addition, the sensitivity of DRE is quite low (i.e., high false negative),indicating that DRE alone can not be relied on to detect prostate cancer.

Serum PSA: PSA is considered as the best serum marker in the early detectionof prostate cancer. PSA is an enzyme, secreted by prostatic cells. High PSA values(>10 ng/ml) have a positive predictive value equal to 66%. PSA values in the rangeof 4 to 10 ng/ml have a positive predictive value ranging from 22% to 35%. There isan overlap between prostate cancer and benign hyperplasia.

PSA Density: It is defined as the PSA per unit prostate volume. It is useful todifferentiate between prostate cancer and benign hyperplasia. The prostate volumecan be found from TRUS screening.

PSA Velocity: It is defined as the rate of change of serum PSA with time. For anormal prostate, PSAV is about 0.04 ng/ml/yr. However, for prostate cancer, PSAV is high (0.7 ng/ml/yr).

Age Specific PSA: To increase the detection of prostate cancer in younger menand to reduce the biopsies of insignificant cancers in older patients, age should beconsidered in PSA cut-off values.

Free and Total PSA: Malignant prostate cells produce less free PSA than hyperplasiatissue. Therefore, it is a useful test to differentiate between prostate cancerand benign hyperplasia.

TransRectal UltraSound (TRUS) and TRUS Biopsy: TRUS is used to displaythe prostate, visualize the cancer, and guide the needles to obtain biopsies from theprostate. A TRUS image is illustrated in Figure2.4. The most common approachto biopsies is the sextant approach. Biopsies from the basal, mid, and apical zonesof the prostate are taken from both sides of the gland. TRUS can detect only theperipheral zone prostate cancer, which represents about 70% of prostate cancer. Theaccuracy of TRUS in detecting cancer is in the range of 57 to 76%.

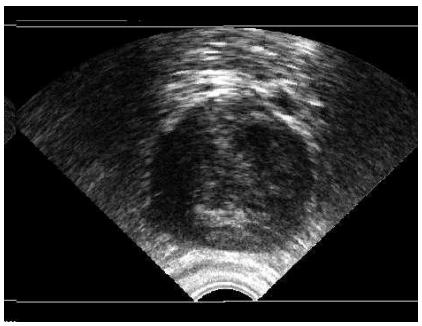


Figure 2.4: Transrectal ultrasound image of the prostate (Awad, 2007).

2.1.1.3 Summary

The early detection of prostate cancer plays a crucial role in the success ofthe cancer treatment. There are several screening tools used to detect prostate cancer, these includeTRUS and MRI imaging. MRI images are very clear, compared with TRUS images,which contain noise such as speckles, shadowing, and other image artifacts. However, MRIscreening which is more costly and not available in many community hospitals, is time consuming,and requires more patient preparation time. Therefore, TRUS is more commonly usedthan MRI for screening and biopsy guidance for prostate cancer. More details of the characteristicsof ultrasound are given in the next section. In this investigation TRUS images arechosen for this research. The challenge is to use ultrasound images and perform some noisereduction techniques so that an appropriate segmentation technique can be developed tosegment the prostate gland. The result of this segmentation shall serve as experimental data for the next stage in the process of prostate cancer diagnosis which is to find the cancerous regions inside the detected prostate boundary.

2.1.2 Ultrasound Imaging

Medical ultrasound is an imaging modality widely used for many clinical applications. Ultrasound images are obtained by transmitting high frequency sound waves into the part of the body to be imaged and then capturing and processing the reflected signals from the different tissues of the region of interest (Pavlin and Foster, 1995). Ultrasound imaging provides a cross-section of the soft-

tissue volume under investigation; however, it cannot be used to image bone and bodies of gas such as found in lungs. It is widely used clinically; it has the advantage of being a real-time modality that does not use any ionizing radiation and can provide quantitative measurement and imaging of blood flow (Youmaran, 2005). Today, the role of medical imaging is not limited to simple visualization andinspection of anatomic structures, but it goes beyond that to patient diagnosis,advanced surgical planning and simulation, radiotherapy planning, etc.Various medical imaging modalities such as Ultrasound, X-ray, Computer Tomography(CT), and Magnetic Resonance Imaging (MRI) are widely usedin routine clinical practice(Wahba, 2008).

2.1.2.1 Ultrasound Equipment Features

Ultrasound is one of the most common medicalimage modalities, and it is used in nearly all hospitals and clinics. Its distinctive feature is that it uses high frequency ultrasound to construct an imagerather than the traditional x-ray. It is an imaging technique in which deepstructures of the body are visualized by recording the reflections (echoes) ofultrasonic waves directed into the tissues. There are three steps to createan ultrasound image. First is producing a sound wave, then, receiving the echoes. Lastly is interpreting those echoes. The ultrasonic waves are producedby electrically stimulating a piezoelectric crystal called a transducer. As thebeam strikes an interface or boundary between tissues of varying acousticimpedance (e.g., muscle and blood) some of the sound waves are reflectedback to the transducer as echoes. The echoes are then converted into electricalpulses that travel to the ultrasonic scanner, where they are processedand transformed into a digital image displayed on the monitor, presentinga picture of the tissues under examination.

However, the interference ofthose echoes, having different phases, add together to give a resultant wavewhose amplitude and phase varies randomly resulting in the speckle, whichis common in ultrasound images. It should be noted that the speckle is moredependent on the measuring system than on the tissue itself. The speckle isdetrimental because it reduces both image contrast (the ability to see the desiredobject against the background) and the distinction of subtle gradationsand boundaries in the tissue structure. A conventional ultrasound imaging device is shown in Figure 2.5. Frequencies in the range of 2 to 18 MHz are usedin diagnostic ultrasonography.

Figure 2.5: Ultrasound machine (Awad, 2007)

The choice of frequency is a trade-off between spatial resolution of the image and imaging depth. The lower frequencies provide a greater depth of penetration but produce less resolution, while theupper ranges provide less penetration with high resolution. Ultrasound imagesare highly detailed and geometrically correct to the first order (Szabo, 2004). Thereare two basic equations used in ultrasonic imaging as presented in equations (2.1) and (2.2)(Jesen, 2006). The first equation is;

$$d = \frac{1}{2}\ tc \qquad (2.1)$$

where;

d : the one way distance of an object that cause the echo

t : time delay

c: the speed of sound in tissue (between 1450 and 1520 m/s)

The second equation is;

$$S(t) = T(t) * B(t) * A(t) * \eta(t) \quad (2.2)$$

where;
S(t): received signal strength
T(t): transmitted signal
B(t): transducer properties
A(t): attenuation of signal path to and from the scatterer
η(t): strength of the scatterer

2.1.2.2 **Ultrasound Imaging Modes**

Ultrasound imaging can be used in many modalities. They include A-mode, B-Mode, M-Mode, and Doppler mode. These are different formats for displaying ultrasound echo data.

A-mode is the simplest type of ultrasound, in which a transducer is usedto scan a line through the body. The returned echoes are plotted on anoscilloscope as a function of depth.An A-mode (or A-scan) is a one-dimensional ultrasonic display showing echoes along the ultrasonic beam as vertical spikes on a horizontal time axis indicating the depth of the reflectors (Figure 2.6). The plots showing two peaks (A and B) corresponding to two objects with different consistency and hardness (taken from www.frca.co.uk/ Images). The horizontal axis corresponds to depth while the y-axis reflects wave amplitude.The amplitudes of the spikes are directly proportional to the echo strengths after time gain compensation (TGC), and the horizontal position (left-right) ofthe spikes is determined by the time difference between the transmission of the ultrasound pulse and the arrival of the echo at the transducer (Anderson and Trahey, 2000). AnA-scan signal calculates the distance (or depth) into the tissues from the time taken by an ultrasound pulse using the following equation (2.3):

$$z = \frac{ct}{2} \quad (2.3)$$

where z is the distance measure, t is the time taken by the pulse to travel back to the transducer, and the factor 2 takes into account the fact that a pulse travels the same distance twice when it reflects on a boundary.

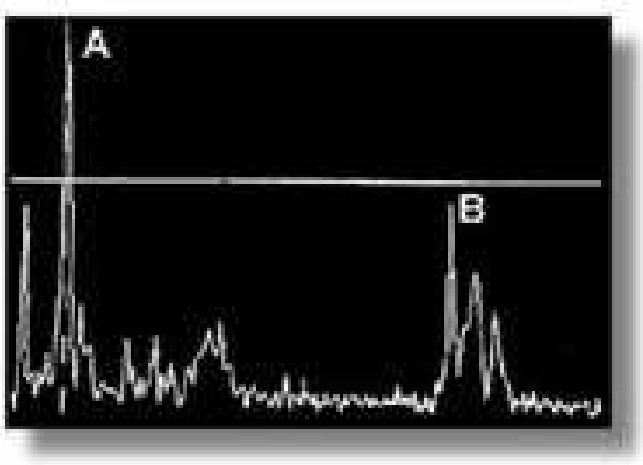


Figure 2.6: An example of an A-scan showing two peaks (A and B)(Anderson and Trahey, 2000).

In B-mode, a linear array of transducers is used to simultaneously scan aplane through the body, and the strength of the returned echoes is displayedas bright spots in their geometrically correct direction and distance. It isdisplayed as a 2D image.A two-dimensional image or B-scan images the reflectivity of a two-dimensional slice through a portion of the anatomy. It is composed of pixels representing the ultrasoundecho amplitude after TGC. The B-scan is formed by coding the A-scan line as brightness along one line of sight and then by linearly scanning the transducer at a uniform velocity. The B-scan needs some kind of mechanical or electronic technique to radially sweep the US beam into the tissue. One way of achieving a sector sweep is using the phased array method where a phase difference is introduced between adjacent transducing sources to flex the plane wave front, without having to mechanically move the transducer. Since a B-scan is made of many different beams, it has a much lower imaging rate than the A-Scan.

In M-mode, a rapid imagesequence of B-mode scans are displayed onscreen to enable doctors to see and measure the range of motion of an organ,whose boundaries produce motion reflections relative to a fixed probe.Doppler mode is based on the Doppler Effect. It can be defined as thechange in

frequency and wavelength of a wave for an object moving relativeto the source of the waves (Awad, 2007).

Ultrasound is extensively used for obstetric/gynaecologic, cardiac, vascular, prostate, breast,and general abdominal imaging. On the other hand, there are characteristicartifacts which make the segmentation process tough such as the inherentpresence of speckle noise, attenuation, shadows, and signal dropout due tothe orientation dependence of the acquisition which can cause missing boundaries.These artifacts can have negative effects on image interpretation anddiagnostic tasks performed by experts or by computerized automated featuredetection and extraction techniques. Additionally, these images usuallyhave low contrast and fuzzy boundaries between the organ or the region ofinterest, to be segmented, and the background. These difficulties make theultrasound images inherently hard to segment (Szabo, 2004). A lot of efforts have been done over the years for modeling and reducingthe speckle noise in ultrasound images using different filters and enhancingtechniques. For example, several techniques for suppressing speckle noisehave been developed for ultrasound images in (Eltoft, 2003), (Xiao et al., 2004).These efforts achievegood results but consume time, hence, eliminating the advantage of being afast imaging technique.

Most of ultrasound exams are non-invasive, performed by using a transduceron the skin; however, some of them is done inside the body (invasiveultrasound). In these exams, the transducer is attached to a probe. Thisprobe is placed into different places in the body such as transoesophageal echocardiogram, in which the transducer is inserted into the oesophagus toobtain images of the nearby heart. In transvaginal ultrasound, a transduceris inserted into a woman's vagina to view her uterus and ovaries. In transrectalultrasound (TRUS) imaging, a transducer is inserted into a man'srectum to view his prostate (Xiao et al., 2004).Since we are interested in prostate segmentation,we only consider TRUS in this project, and it is explained in detail inthe next section.

2.1.2.3 Trans-Rectal Ultrasound (TRUS) Imaging

Trans-rectal ultrasound (TRUS) imaging is a procedure, in which a probeis inserted into the rectum, as shown in Figure 2.7. It emits high energysound waves, which penetrate internal tissues or organs and produce echoes. These echoes create an image of organs in the pelvis. Trans-rectal

ultrasound(TRUS) images are widely used for the screening of the prostate gland. TRUSimages may reveal prostate cancer, benign prostatic hypertrophy, or prostatitis.TRUS images may also be used to help guide a biopsy of the prostate(National Cancer Institute [NCI], 2005).

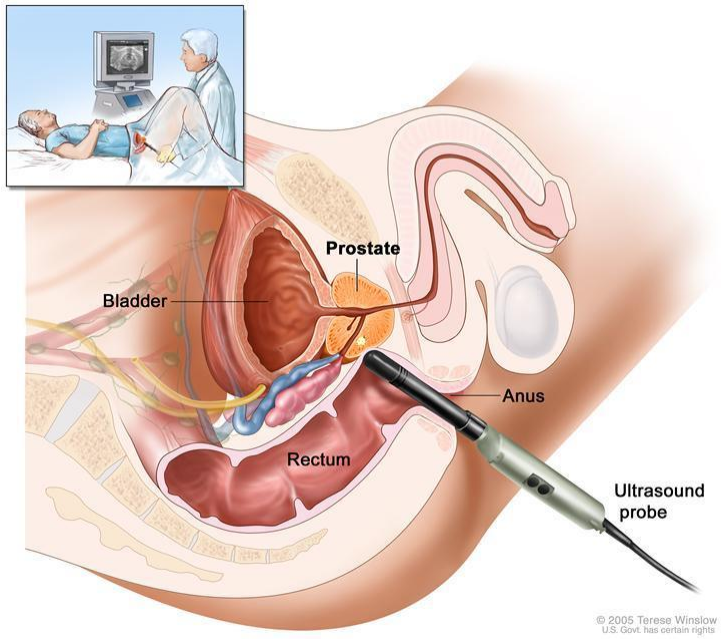


Figure 2.7: Trans-rectal ultrasound (TRUS) (NCI, 2005).

Table 2.1 lists the major applications in which prostate boundaries and volumes obtained from TRUS images play a key role in clinical decisions (Pathak et al., 2000). The gland volume may be derived from the boundaries by planimetric volumetry. However, prostate boundary detection from ultrasonographic images still remains a challenging task because of the poor contrast between the prostate and surrounding tissues, speckle noise, shadowing, and refraction artifacts.

Table 2.1. Major applications for which prostate boundaries and volumes play key roles in clinical decisions

Application	**Parameter of Interest**
Diagnosis	
Prostate-specific antigen density	Volume
Assessment of benign prostatic hyperplasia	Volume
Treatment	
Prostate brachytherapy	Boundary
High-intensity focused ultrasonography	Boundary
Cryotherapy	Boundary
Transurethral microwave therapy	Volume
Resection of benign tissue	Volume
Follow-up	
Hormonal treatment	Volume

Adapted with permission from Y. Kim, PhD (University of Washington, Seattle, WA) (Shao et al., 2003).

2.1.2.4 Theory of Digital Imaging

The digital image is made up of rectangular arrays of pixels of varying pixel intensity, $\rho(m,n)$. The pixel is the smallest addressable unit of a digital image. In a digital image, every pixel value $\rho(m,n)$, has two positive sample numbers m and n where m is the row sample numbers and n is the column sample numbers respectively. Each pixel value in a digital image is a data sample in the image. The brightness of an image depends on the intensity values of each pixel and the frequency of occurrence, $\Omega(m,n)$ of each pixel. The frequency of occurrence is the number of times a given pixel intensity, $\rho(m,n)$ repeats in an image. The relationship between noiseless image, $\bar{Y}(m,n)$ and the pixel intensity is an identity relationship (Posada-gómezand Sandoval-Gonzalez, 2001) and can be represented as shown in equation (2.4).

$$\bar{Y}(m,n) \equiv \rho(m,n) \tag{2.4}$$

The relationship in equation (2.4), shows that although $\bar{Y}(m,n)$ and $\rho(m, n)$ are not equal, they have equal value at any given location. This implies that they are equivalent but not equal. The

relationship in equation (2.4) can be represented in the form of a rectangular array of pixels as shown in equation (2.5).

$$\bar{Y}(m,n)=\begin{bmatrix} \rho(1,\ 1) & \rho(1,\ 2) & \dots & \rho(1,\ N-1) & \rho(1,\ N) \\ \rho(2,\ 1) & \rho(2,\ 2) & \dots & \rho(2,\ N-1) & \rho(2,\ N) \\ \rho(3,\ 1) & \rho(3,\ 2) & \dots & \rho(3,\ N-1) & \rho(3,\ N) \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \rho(M-1,1) & \rho(M-1,2) & \dots & \rho(M-1,N-1) & \rho(M-1,N) \\ \rho(M,1) & \rho(m,n) & \dots & \rho(M,\ N-1) & \rho(M,\ N) \end{bmatrix} \quad (2.5)$$

Every digital image is made up of a certain number of rows and columns of pixels. The digital image size (ψ) is determined by the product of row size (M) and column size (N) of the image. Mathematically, the image size (ψ) of a digital image is M x N.

A digital image is represented as a 2-dimensional lattice of r-dimensional pixels, where r is one colour per pixel in the grayscale image, three colours per pixel in colour (RGB) image and r is greater than three colours per pixel in multispectral image represented using RGB colours, Ultraviolent and near-infrared light. The space of the lattice is the spatial domain while the gray level, colour or spectral information is the range domain (Comaniciu et al., 2002).

(i) Types of Digital Images

Digital images are divided into three important types: colour image, grayscale image and multispectral image. However, for conventional use, such as biomedical imaging systems and digital cameras, digital images are divided into the grayscale and colour image. The multispectral image is reserved for unconventional use such as military radar system, satellite system.

(a) Grayscale Images

The gray scale (black and white) image is made up of pixels and each of the pixels holds a single colour corresponding to the gray level of the image at a particular location. In 8-bit gray level image, each pixel is encoded with 8-bits and the gray levels smoothly span from 00 (black) to 255 (white) which means 256 possible colours per pixel are used and requires 1byte per pixel memory capacity

for storage. Other forms of grayscale images are binary image and normalized image. An example of grayscale images is biomedical images captured using CT (Sachs, 1999).

(b) Normalized grayscale image

Most often, grayscale images are normalized to reduce the range of the pixel intensity required. For a normalized grayscale image, the pixel intensities are real numbers varying from 0.0 pixel value (black) to 1.0 pixel value (white). A grayscale image is normalized by dividing each of the pixel values by 255 (white colour value). The variation in the colours of a normalized grayscale image has three segments, Black, Threshold and White. The threshold is the point where there is an equal mixture of black and white colour. A threshold in grayscale image is defined as a pixel value such that when all pixel intensity ρ (m, n) > T are converted to 1.0 and pixel intensity $\rho \leq T$ are converted to 0.0, the original image will be accurately represented. At any point between threshold colour and black colour, the proportion of black colour is more than the proportion of white colour in the image. For any point between threshold colour and white colour, the proportion of white colour is more than that of black colour(Sachs, 1999).

(c) Binary image

A modified version of a grayscale image is a binary image. It is an image represented by 0 and 1 only. It is generated by converting any pixel intensity more than the threshold to 1.0 (HIGH) and any pixel intensity equal to or less than threshold to 0.0 (LOW). The threshold of a grayscale image depends on the proportion of each colour in the image. For diagram shown in Figure 2.1, the threshold is 0.5. The process of generating a binary image from grayscale image is called segmentation.

(d) Colour Image

On the other hand, the colour image is represented using three colours; red colour (R), green colour (G) and blue colour (B) and is also known as an RGB image. In an RGB image, each pixel is encoded using 24-bit corresponding to 8-bit of red (R), 8-bit of green (G) and 8-bit of blue (B) per image pixel. The RGB colour is formed by mixing primary colours additively. It has domain ranging from 0 (black) -16,777,215 (white) which corresponds to 224 (16,777,216) possible colours per

pixel and it therefore needs higher storage memory (3bytes per pixel) for storage compared to gray image (1byte per pixel) since a byte represents 8bits. An example of a colour image is an image captured using a digital camera(Sachs, 1999).

(e) Multispectral image

For non-conventional applications such as military and communication, multispectral images are used. The multispectral image is represented using RGB colour, Ultraviolent and near-infrared wave. The image contains 240 (1,099,511,628,000) colours and spectra and requires 5bytes per pixel for storage. An example of multispectral image is remotely sensed images captured using radars and satellites (Paul et al., 2004).

(ii) Generation of digital image

Digital image is generated from continuous-time image by 2-dimensional sampling at a frequency at least equal to the Nyquist rate. The Nyquist rate is the minimum frequency at which a continuous-time image must be sampled so that it can be reconstructed from the digital image without loss of information(Sachs, 1999). Sampling of the image below the Nyquist rate causes an aliasing effect. Aliasing effect is the distortion of the image due to under-sampling. The Nyquist rate for a digital image is defined as 2B where B is the number of bits per pixel in the image. For effective recovery of images, the Nyquist rate for grayscale image is 16-bit/pixel while the Nyquist rate required for colour image is 48-bit/pixel (Sachs, 1999).

(iii) Corruption of Digital Image

Digital images are corrupted by noise when some of the pixel intensities are modified by unwanted random signal. Noise is an unwanted signal that combines with desired image to reduce the quality of the image. There are many ways that noise can be introduced into a digital image, depending on how the image is acquired. Some of these ways are(Paul et al., 2004):

(a) Noise due to the film grain or the result of damage to the film present in a scanned picture. Film grain is the random visual texture of processed photographic film due to the presence of small particles of a metallic silver, dye clouds, or dust in the photographic film before processing.

(b) Noise due to error in detector such as Charge-Coupled detector (CCD) in image capturing devices. A CCD (Charge Coupled Device) is a silicon based multichannel array detector of UV, visible and near-infra light.

(c) Noise due to the electronic transmission of image.

When a noiseless digital image, $\bar{Y}(m,n)$interacts with noise, it combines additively or multiplicatively with the noise signal,א*(m,n)* to form corrupted digital image, $\chi(m, n)$(Sachs, 1999). The presence of noise in a digital image distorts the information carried by the image and erodes the quality of the image. The reduction of image quality due to noise has great impact on modern technology such as radar system, satellite system and biomedical imaging system where the accurate recognition of image features has played a great role. The image features are the properties of regions in the image that makes it possible for the image or regions in the image to be accurately recognized(Paul et al., 2004). A typical example of digital image from biomedical imaging system is TRUS prostate image (Sachs, 1999),(Paul et al., 2004).

2.1.2.5 Image Resolution

Digital images are made up of square-shaped dots (pixels). And, the more dots a digital imagehas the clearer the image, which is why high resolution (lots of dots) images are more desirable (Sarra, 2007).

(i) Pixel Dimensions

The total pixel dimensions of an image will detemine how many total pixels(dots) the image is made up of. For example, let’s say we have a digital image that is 1200x1800pixels (dots). That means our digital image is 1200 dots high by 1800 dots wide.

(ii) Digital Image Size

This is the physical size in inches of a digital image. The digital image size is shown as “Width” x “Height” inches.

(iii) DPI of Digital Image

DPI means **D**ots **P**er **I**nch. This number is calculated using a digitalimage’s pixel dimensions and digital image size. It represents the resolution of the image in pixels/inch. It is the ratio of the pixel dimension to the image size taken width to width and height to height respectively.

(iv) Pixel Dimensions and DPI

There are three different ways to describe a digital image’s resolution that essentially mean thesame thing: (1) total pixel dimensions, (2) DPI at a certain digital image size, and (3) DPI at acertain output size.

The total pixel dimensions of an image will determine how many total pixels (dots) the image ismade up of. For example, let’s say we have a digital image that is 1200x1800 pixels (dots).That means our digital image is 1200 dots high by 1800 dots wide. So, unless you resample(which means you artificially alter the pixel dimensions in a photo editing program) or crop theimage, your image will **always** be 1200x1800 pixels.People regularly discuss digital images in terms of DPI, which stands for **D**ots **P**er **I**nch. The DPIof a digital image is calculated by dividing the total number of dots wide by the total number ofinches wide OR by calculating the total number of dots high by the total number of inches high.For example, a digital image that is 1200x1800 pixels (dots) and 4x6 inches insize. That means the digital image is 1200 dots high by 1800 dots wide and 4 inches high by 6inches wide. The digital image has 300 DPI. It is derived by dividing the number of dotswide by the number of inches wide or dividing the number of dots high by the number of inches highas expressed in equations (2.6) and (2.7) (Sarra Compos, 2007):

$$\frac{1800\ dots\ wide}{6\ inches\ wide} = \frac{300\ dots}{1\ inch} = 300\ \boldsymbol{Dots\ Per\ Inch} \qquad (2.6)$$

$$\frac{1200\ dots\ wide}{4\ inches\ wide} = \frac{300\ dots}{1\ inch} = 300\ \boldsymbol{Dots\ Per\ Inch} \qquad (2.7)$$

2.1.2.6 Summary

Ultrasonographic imaging has several favourable properties compared with other medical imaging modalities(Lee et al., 1998). It does not require the special equipment that radiographic computed tomography and magnetic resonance imaging require, and portable ultrasonographic imaging instruments are available. Unlike other tomographic techniques, ultrasonography offers interactive visualization of the underlying anatomic structures in real time and has the ability to show dynamic structures within the body. Needles and catheters can also be deployed under ultrasonographic guidance. For this reason, TRUS is commonly used for diagnosis of prostatism, detection and staging of prostate cancer, and real-time image guidance of minimally invasive therapeutic procedures (Aarnink et al., 1998), (Lee et al., 1998).

2.1.3 Medical Image Segmentation

Image segmentation plays an important role in medical applications. Such segmentationhelps in delineating the anatomy of an organ and its volume. This is crucial in the diagnosisof prostate cancer, where the ratio of the PSA to the volume of the prostate is importantto detect cancer and aid in treatment planning (Hu et al., 2002). Medical images, in general, are difficultto segment, especially, ultrasound images which suffer from a lake of contrast and arecorrupted by speckle noise (Bovik, 1988).

2.1.3.1 Image Segmentation Definition

Generally, image segmentation can bedefined as the process of partitioning a digital image into multiple regions(set of pixels) (Shapiro and Stockman, 2001).Image segmentation divides an image into regions with no overlapping. Each region possesses different features including colour, intensity, texture or other statistical properties. The goal of segmentation is to simplify and/or

change therepresentation of the image into something that is more meaningful and easierto analyze. The result of image segmentation is either a set of regions thatcollectively cover the entire image or a set of contours extracted from the image(Shapiro and Stockman, 2001). According to this general definition, we can define the medical imagesegmentation, more specifically, as the process of delineating and separatingthe anatomical structure (region of interest) so that it can be viewed individually in order to achieve important objectives such as patient diagnosis,radiotherapy planning, and advanced surgical planning and research.

If the domain of the image is given by I, then, the segmentation problem is to determine the sets whose union is the entire image I. Thus, the sets that make up segmentation must satisfy equation (2.8):

$$I = \bigcup_{k=1}^{N} S_k \qquad (2.8)$$

Where*N* is the number of possible sets of regions in the image, $S_k \cap S_j = \emptyset \ \forall k \neq j$, and each S_k is a connected region(Wahba, 2008).The expression $S_k \cap S_j = \emptyset \ \forall k \neq j$, says that for any two sets $S_k$and$S_j$belonging to the image I, that satisfies eqtn 2.4, their intersect ($\cap$) is a null set (Ø) for all values of ($\forall$),*k* not equal to *j*. It implies that all the sets that make up the region of the image do have any intersection.Ideally, segmentation method defines those sets that belong to distinctive anatomical structures or regions of interest in the image(Wahba, 2008).

The direct wayto segment an image is by applying an edge detection technique, especially if the imageconsists of clear objects on a different intensity background. However, this technique canfail, if the image contains noise.Validation experiments are necessary to quantify the performance of a segmentation technique.They can be conducted by comparing an automatic segmentation with the manualsegmentation of an expert. Another common approach is the use of physical phantoms orcomputational phantoms (Pham et al., 2000).

2.1.3.2 Image Segmentation Concepts

There are two concepts of segmentation: hard segmentation and soft (fuzzy) segmentation. Hard segmentation is the segmentation that forces a decision of whether a pixel is inside or outside the object. Hard segmentation is based on the concept of a characteristic function, which is simply an indicator function of whether a pixel is inside or outside its corresponding set (Pham et al., 1998).

For a location,$j \in$ I, (j is an element ($\in$) inimage, I), the characteristic function X_k (j) of the set $S_k$is defined in equation (2.9) as(Pham et al., 1998):

$$X_k(j) = \begin{cases} 1 \text{ if } j \in S_k \\ 0 \text{ otherwise} \end{cases} \qquad (2.9)$$

Equation 2.9 says that the characteristic function,X_k (j), of the set,$S_k$has a value of 1 if the location j in the image I is an element($\in$)in the set,S_k., otherwise the value is zero (0).

Soft (fuzzy) segmentation, on the other hand, allows regions to overlap.Soft segmentation is very important in medical image segmentation because of partial volume effects, where multiple tissues contribute to a single pixelresulting in a blurring of intensity across boundaries.

Soft segmentation isconsidered the generalization of the characteristic function in equation (2.9). Thisgeneralization is defined as the membership function which does not have tobe binary valued. A membership function M_k(j) should satisfy the following conditions in equation (2.10) (Pham et al., 1998):

$$\begin{aligned} &0 \le M_k(j) \le 1 \qquad \forall k, j \\ &\sum_{k=1}^{N} M_k(j) = 1 \qquad \forall j \end{aligned} \qquad (2.10)$$

Equations 2.10 says that for membership function, $M_k(j)$,of the k^{th}set in an image,I of N sets,at a location$j$, withvalues lying between 0 and 1, for all values ($\forall$) of k*and* j, the sum of all membership functions for the entire image is 1 for all values ($\forall$) of j.

The general imaging artifacts such as speckle noise, motion, and partial volume effects can significantly affect the outcome of the segmentation process. Hence, in order to guarantee a good segmentation, pre-processing steps should be performed before applying any segmentation algorithm(Pham et al., 1998). The major problem faced in medical images is the time consumed in the pre-processing steps. To overcome these problems, different methods and filters such as mean filters and median filters are used for de-noising and enhancing the images (see section 2.1.4).

2.1.3.3 Summary of Medical Image Segmentation

Image segmentation is the first step andalso one of the most critical tasks of image analysis and processing becausethe segmentation results affect all the subsequent processes of image analysissuch as object representation and description, feature measurement, andeven the following higher level tasks such as object classification (Wahba, 2008). Hence, imagesegmentation is the most essential and crucial process for facilitating thedelineation, characterization, and visualization of regions of interest.

Manualsegmentation is not only a tedious and time consuming process, but also itis inaccurate especially with the increasing medical imaging modalities andunmanageable quantity of medical images that need to be examined. Segmentationby experts has shown to be variable up to 20%(Wahba, 2008). It is thereforedesirable to use automated algorithms that are accurate and require as littleuser interaction as possible. In the segmentation process, the anatomicalstructure or the region of interest needs to be delineated and extracted outso that it can be viewed individually.

2.1.4 Noise Types in Images

Digital images are prone to a variety of types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene (The Mathworks Inc., 2015b).Image noise is the random variation of brightness or colorinformation in images produced by the sensor and circuitry of ascanner or digital camera. Image noise can also originate in filmgrain and in the unavoidable shot noise of an ideal photondetector (Boncelet, 2005). Image noise is generally regarded as an undesirableby-product of image capture. Although these unwantedfluctuations became known as "noise" by analogy withunwanted sound they are inaudible and such as dithering (Patidar et al., 2010). The magnitude of image noise can range from almost imperceptible specks on a digital photograph taken in good light, tooptical and radio astronomical images that are almost entirely noise, from which a small amount of information can bederived by sophisticated processing (a noise level that would be totally unacceptable in a photograph since it would beimpossible to determine even what the subject was).Thetypes of Noise are as follows (Patidar et al., 2010), (Farooque and Rohankar, 2013) :-

- Speckle noise;
- Gaussian noise (Amplifier noise);
- Quantization noise (uniform noise);
- Anisotropic noise
- Salt-and-pepper noise
- Poisson noise (Shot noise)
- Film grain noise

2.1.4.1 Speckle Noise

Speckle noise is a granular noise that inherently exists in and degrades the quality of the active radar andsynthetic aperture radar (SAR) images. Speckle noise in conventional radar results from random fluctuations in thereturn signal from an object that is no bigger than a single image-processing element. It increases the mean grey level of alocal area. Speckle noise in SAR is generally more serious, causing difficulties for image interpretation. It is caused bycoherent processing of backscattered signals from multiple distributed targets. In SAR oceanography (Sedef et al., 2004), for example,speckle noise is caused by signals from elementary scatters, the gravity-capillary ripples, and manifests as a pedestalimage, beneath the image of the sea waves (Farooque and Rohankar, 2013).

2.1.4.2 Salt-and-Pepper Noise

An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions (Boncelet, 2005). This typeof noise can be caused by dead pixels, analog-to-digitalconverter errors, bit errors in transmission, etc.This can beeliminated in large part by using dark frame subtraction and byinterpolating around dark/bright pixels.

2.1.4.3 Gaussian Noise (Amplifier Noise)

The standard model of amplifier noise is additive, Gaussian,independent at each pixel and independent of the signalintensity.In color cameras where more amplification is used inthe blue color channel than in the green or red channel, there canbe more noise in the blue channel. Amplifier noise is a majorpart of the "read noise" of an image sensor, that is, of theconstant noise level in dark areas of the image (Boncelet, 2005), (Farooque and Rohankar, 2013).

2.1.4.4 Poisson Noise (Shot Noise)

Poisson noise or shot noise is a type of electronic noise thatoccurs when the finite number of particles that carry energy,such as electrons in an electronic circuit or photons in an opticaldevice, is small enough to give rise to detectable statisticalfluctuations in a measurement (Boncelet, 2005), (Patidar et al., 2010).

2.1.4.5 Quantization Noise (Uniform Noise)

The noise caused by quantizing the pixels of a sensed image to a number of discrete levels is known as quantization noise.It has an approximately uniform distribution. Though it can be signal dependent, it will be signal independent if othernoise sources are big enough to cause dithering, or if dithering is explicitly applied (Farooque and Rohankar, 2013).

2.1.4.6 Film Grain Noise

The grain of photographic film is a signal-dependent noise, with similar statistical distribution to shot noise. If filmgrains are uniformly distributed (equal number per area), and if each grain has an equal and independent probability ofdeveloping to a dark silver grain after absorbing photons, then the number of such dark grains in an area will be randomwith a binomial distribution. In areas where the probability is low, this distribution will be close to the classic Poissondistribution of shot noise. A simple Gaussian distribution is often used as an adequately accurate model.Film grain is usually regarded as a nearly isotropic (non-oriented) noise source. Its effect is made worse by thedistribution of silver halide grains in the film also being random (Farooque and Rohankar, 2013).

2.1.4.7 Anisotropic Noise

Some noise sources show up with a significant orientation in images. For example, image sensors are sometimes subjectto row noise or column noise(Farooque and Rohankar, 2013).

2.1.5 Image Enhancement

Image enhancement results in an image which either looks better to an observer, usually subjective, or which performs better in a subsequent computer-aided analysis and diagnosis operations (Dougherty, 2009). Image enhancement techniques bring out the detail in an image that is obscured or highlight certain features of interest in an image. Enhancement techniques include contrast adjustment, filtering, morphological filtering, and deblurring. Image enhancement operations typically return a modified version of the original image and are frequently used as a preprocessing step to improve the results of image analysis techniques (The Mathworks Inc., 2015). Enhancement might involve adjusting the *brightness* of the image, if it were too dark or too bright, or its *contrast*, if for example it comprised of only a few shades of gray, giving it a washed-out appearance. Alternatively, it might involve *smoothing* an image that contains a lot of noise or speckle, or *sharpening* an image so that edges within it are more easily seen (Dougherty, 2009).

Image enhancement approaches fall into two broad categories: spatial domain methods and frequency domain methods. The term spatial domain refers to the aggregate of pixels comprising an image or image plane itself, and approaches in this category are based on direct manipulation of pixels in an image. These techniques include *point operations*, where the output pixel value depends only on its corresponding input value, and *localor neighborhood operations*, where the eventual pixel value depends on the neighborhood of input pixels values. These later operations include convolution, which uses appropriate masks or kernel to produce smoothing or sharpening of an image (Gonzalez and Woods, 2002).

On the other hand, frequency domain processing techniques are based on modifying the Fourier transform of an image. *Low frequencies* in the Fourier transform are responsible for the general gray-level appearance of an image over smooth areas, while *High frequencies* are responsible for detail, such as edges and noise (Gonzalez and Woods, 2002). The low-pass filtered images have less sharp details than the original because the high frequencies have been attenuated. Hence smoothing (blurring) is achieved in the frequency domain by attenuating a specified range of high frequency components in the transform of a given image. Similarly, a high-pass filtered image would have less gray level variations in smooth areas and emphasized transitional (e.g. edge) gray-level detail. Image sharpening can be achieved in the frequency domain by a high-pass filtering process, which attenuates the low-frequency components without disturbing high-frequency information. Such an image will appear sharper. This property is useful for boundary detection in segmentation of images (Dougherty, 2009), (Gonzalez andWoods, 2002).

2.1.5.1 Image Enhancement – Operations

Property based preprocessing operation are based on the transformation properties, the local preprocessing operation can be classified into the following methods: Linear filtering operation and nonlinear filtering operation. The linear filtering and nonlinear filtering operation give virtually the same value of the Mean Square Error (MSE) when the Peak Signal to Noise Ratio (PSNR) tends to one (low noise level). However, when the PSNR tends to zero (high noise level), the nonlinear filtering operation gives the best result.

(i) Linear Filtering Operation

In its simplest form, the linear filtering operation involves taking the average of all the pixels over a local neighbourhood. Linear operations computes the output pixels values, ρ(m,n) in the output image, ϒ(m,n) as a linear combination of pixel values in a local neighbourhood of the pixel within the input image, *X*(m,n) (Pitas and Venetsanopoulos, 1999). The contribution of the pixels in the neighbourhood is weighted by filter coefficients, using the convolution operation. Linear filtering is a very important tool in signal and image filtering for Gaussian noise (noise having a probability density function (PDF) equal to that of normal Gaussian distribution) due to its mathematical and design simplicity and it also gives a good performance in certain applications such as computer vision and image recognition systems. However, the use of linear filters fails when it is applied to image corrupted by non-Gaussian noise (noise with a probability density function (PDF) that has no relationship to that of normal Gaussian distribution) or signal dependent (impulsive) noise which occurs when an image is transmitted through non-linear channels. Linear filters blur the image and destroy edges, lines and other image details. ***Good examples of linear filters are the averaging filter and Gaussian filter*** (Pitas and Venetsanopoulos, 1999).

(ii) Nonlinear Filtering Operation

A nonlinear filtering operation provides effective solution to problems where linear filtering operation is not adequate. It involves using nonlinear filters in image filtering. Some examples of nonlinear filters are: ***median, rank-order, range, α-trimmed mean and median mean filters***(Gasteratos et al., 1998).Nonlinear filters have excellent and robust properties such as suppression of high frequency, removal of impulse noise and avoid extreme blurring of the image. They have good edges and image detail preservation properties and therefore have the dual function of edge enhancement and noise reduction unlike the linear filters which has only noise reduction property. Most nonlinear filters meet various performance criteria, like robustness, adaptively to noise probability distributions, preservation of edge information, preservation of image details.

(iii) Convolution Operation

Convolution operation is a linear mathematical operation between an input function, *X(m,n)* and filter,*h(m,n)* to produce an output function, ϒ(m,n) that is modified version of the input function. The convolution between input function and system response is called linear filtering operation. Mathematically, the convolution operation is given as shown in equation (2.11).

$$ϒ(m,n) = h(m,n) * X(m,n) \quad (2.11)$$

The convolution operation obeys some linearity law of signal processing such as commutative and associativelaws. A convolution operation for a non-causal system according to (Mcandrew, 2004), (Brodic and Milivojevic, 2010) is shown in equation (2.12). A non-Causal system is a system that has values at every sample m and n such that m has the range- $\infty \leq m \leq \infty (m \epsilon R)$and n has the range $-\infty \leq n \leq \infty (n \epsilon C)$. In equation (2.12),the operation is a discrete convolution with the h (m, n) as the filter kernel or a mask and *X*(m, n) as the corrupted image function of size RxC.

$$ϒ(m,n) = \sum_{j=-Rf}^{Rf} . \sum_{k=-Cf}^{Cf} (X(m-j, n-k)h(j,k)) \quad (2.12)$$

Where $C_f = \frac{Mf-1}{2}$ *and* $R_f = \frac{Nf-1}{2}$ *;* Mf and Nf are the row size and column size of the filter respectively, and therefore the filter has size of $M_f \times N_f$. In practice, M_f and N_f are odd numbers and they are equal in length ($M_f = N_f$). The convolution operation is an important operation in image preprocessing, and is applied in filtering when the noise is linear.

(iv) Homomorphic Filtering

A homomorphic filtering technique is an edge preserving image filtering technique (Padmavathi et al., 2010) for removing multiplicative noise. The technique involves transforming image corrupted by multiplicative noise to image corrupted by additive noise and then applying linear filters and

convolution operation to the transformed image. An expression representing an image corrupted by multiplicative noise is shown in equation (2.13).

$$X(m,n) = \Upsilon(m,n)N(m,n) \quad (2.13)$$

The homomorphic filtering involves performing the logarithmic transformation of equation (2.13) according to (Mathur et al., 2011),(Sudha et al., 2009)as shown in equation (2.14).

$$log_e X(m,n) = log_e \Upsilon(m,n) + log_e N(m,n) \quad (2.14)$$

The convolution operation can be applied to equation (2.14) to remove noise from the image. After the convolution operation, the filtered image can be transformed back by finding the exponential of the output as shown in equation (2.15).

$$\Upsilon(m,n) = exp(\sum_{j=-Rf}^{Rf}. \sum_{k=-Cf}^{Cf} log e\, X(m-j,n-k)(h(j,k))) \quad (2.15)$$

The application of linear filtering operation to signal corrupted by multiplicative noise without first applying homomorphic transformation makes the linear filter used in the filtering useless.

(v) Suitability of Preprocessing Filters

Preprocessing filters can be used to sharpen images, obtain edges. In most cases, pre-processing filters are used to remove noise which distorts the image and therefore, negatively affects the human interpretability and machine perception of the image. The type of filter used to enhance noisy image depends on the type of noise that corrupts the image. The averaging and Gaussian filters are found useful for removing film grain (additive noise) from a photographic film while median filter which enhances images with minimal blurring of the edge is used to remove impulse (salt and pepper) noise (Miljkovic, 2009). The salt and pepper noise consists of random pixels being set to black or white. On the other hand, Weiner filter (an adaptive filter) is used for removing additive noise (constant intensity noise) for best result (Balakrishnan et al., 2012)but it also removes multiplicative noise. To effectively remove a multiplicative (speckle) noise which is found mostly in radar and

satellite images, an anisotropic diffusion filter is preferred since it blurs the image without compromising the image quality (Murali and Dinesh, 2012). When image with orientation properties like fingerprint image is to be filtered, Gabor filter is employed because it has a frequency-selective and orientation-selective properties (Rajinikannan et al., 2010*)*. For effective results, multi-stage bilateral filters give better result (Goyal et al., 2013).

2.1.5.2 Image Enhancement Filtering Types

There are a number of filter types in the spatial domain techniques both from anisotropic and isotropic filters. **Anisotropic filter** is a filter that is orientation sensitive. A filter is orientation sensitive if the direction of the maximum performance of the filter can be changed depending on the filter orientation when handling directional image. An anisotropic filtering effectively preserves image features and removes noise from the image when a reliable orientation is provided. When a reliable orientation is selected for the filter, the anisotropic filtering produces better results than isotropic filtering. An example of anisotropic filter is Gabor filter. **An isotropic filter** is a filter that is not orientation sensitive. Examples of isotropic filters are: Gaussian filter, Auto enhancement algorithm, Median filter, Averaging filter, and Wiener filter. Some of these are briefly described here.

(i) Gaussian Filter

The Gaussian filters find widespread application in signal processing because Gaussian masks have non-negative values ($h(x) \geq 0$) for the range $-\infty < x < \infty$, and therefore do not invert the filtered signal. It is one of the best signals filtering method based on the Gaussian distribution invented by a German mathematician, Carl Friedric Gauβ. Gaussian filter is a non-causal filter which has found applications in low-level computer vision (Van Den Boomgaard and Van Der Weij, 2001) and pattern recognition (Evangelista et al., 2007) which includes; image clustering, anomaly detection, classification, regression and principal component analysis. It exists in two major forms: the standard Gaussian filter, recursive Gaussian filter.

(a) Standard Gaussian filter

The standard Gaussian filter involves the use of one-stage Gaussian filter in signal filtering. It is always used in simple applications because it is fast and easy to implement. In general, an A-dimensional continuous-time standard Gaussian filter, **h(x)** with the mean at the origin is represented in equation (2.16) where **x** is a vector representing the spatial coordinates and **σ** represents the standard deviation of the filter (Benoudjit et al., 2002).

$$h(x) = \frac{1}{\left(\sqrt{2\pi\sigma}\right)^{A}} e^{-\frac{\sum_{i=1}^{A} x_i^2}{2\sigma^2}} \quad (2.16)$$

When the filter mean is not at the origin, the A-dimensional Gaussian filter in equation (2.16) is rewritten as shown in equation (2.17) with **μ** as the mean where the parameter **|x- μ |** is called the Euclidean distance.

$$h(x) = \frac{1}{\left(\sqrt{2\pi\sigma}\right)^{A}} e^{-\frac{\sum_{i=1}^{A} \left(x_i-\mu\right)^2}{2\sigma^2}} \quad (2.17)$$

(b) 1-dimensional Gaussian filter

A 1-dimensional Gaussian filter is a Gaussian filter that has value in one spatial coordinate only. The 1-dimensional equivalent of equation (2.16) is shown in equation (2.18) (Van Den Boomgaard and Van Der Weij, 2001). The 1-dimensional Gaussian kernel is applied in filtering of 1-dimensional signals (where A=1).

$$h(x) = \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)} e^{-\frac{x^2}{2\sigma^2}} \quad (2.18)$$

The Gaussian filter is non-zero function that theoretically requires an infinite window length (x Ɛ [-∞, ∞]) for precise filter implementation. The geometric distribution of the mapped time coordinates in kernel space of a Gaussian distribution depends on the standard deviation (σ).

The size of a Gaussian filter determines the standard deviation needed for effective filtering. The σ needed is directly proportional to the filter size. The selection of an appropriate value of standard deviation (σ) for different Gaussian filter sizes, is known as ***tuning***(Evangelista et al., 2007). For optimum tuning, the value of σ must not be too big or too small, but must minimize intra-class variation and maximize inter-class variation and at the same time reduces the execution time (Wang et al., 2009). The standard deviations for different filter sizes can be obtained from equation (2.19) where M_f and N_f are the filter row size and column size respectively

$$\sigma = \frac{N_f}{6} = \frac{M_f}{6} \tag{2.19}$$

(c) Recursive Gaussian filter

For a better filtering result, a recursive Gaussian filter is used in signal filtering. The recursive Gaussian filter is a modified form of standard Gaussian filter (Goyal et al., 2013). It applies multiple stages of standard Gaussian filter in image processing. The filtering operation of the recursive Gaussian filters involves repeatedly convoluting the signal with a series of standard Gaussian filters until the required level of enhancement is achieved. The filtering operation of the recursive Gaussian filter is given in equation (2.20)(Goyal et al., 2013).

$$\Upsilon(m) = X(m) * h_1(m) * h_2(m) * h_3(m) \ldots \tag{2.20}$$

Since convolution is a linear process, and therefore obeys the law of associativity, the filtering operation involves convolving the signal with the equivalent impulse response of all the filter stages. This operation can be shown in equation (2.21)(Goyal et al., 2013).

$$\Upsilon(m) = X(m) * h_{eg}(m) \tag{2.21}$$

where $h_{eg}(m) = h_1(m) * h_2(m) * h_3(m) \ldots$

(ii) **Gabor Filter**

The Gabor filter is an excellent linear band-pass filter used for image preprocessing and edge detection invented by Dennis Gabor (Rajinikannan et al., 2010),(Chaudhary and Singh, 2012). It is a double bilateral filter (Goyal et al., 2013) formed when Gaussian function (envelope) is modulated by a sinusoidal plane wave (carrier). The Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function (Al-tarawneh, 2012). It is used in edge detection because it simultaneously reduces noise and preserves the edge of an image and it is sensitive to the orientation of the image coordinates. A Gabor filter set with a given orientation gives a strong response for locations of the target images that have structures in that direction. The orientation-space representation is treated as a continuous variable to which the Gabor filter can be tuned. According to tuning of the Gabor filter involves selecting the suitable bandwidth for the filter; the smaller the bandwidth, the more orientation sensitive of the filter. Frequency and orientation representations of Gabor filters are similar to those of the human visual system and the simplest cells in the visual cortex of mammalian brains can be modeled by Gabor functions since it has frequency and orientation selection properties (Daugman, 1985),(Liu and Dai, 2006).

The Gabor filter is uniquely used in fingerprint preprocessing because of its frequency-selective and orientation-selective properties (Rajinikannan et al., 2010). A 2-dimensional Gabor function, according to Prasad and Domke (2005), is shown in equation (2.22).

$$h(x,y,\theta,\lambda,\gamma) = e^{\frac{x_\theta^2+\gamma^2 y_\theta^2}{2\sigma^2}} * \left(cos\left(2\pi \frac{1}{\lambda} x_\theta\right) + jsin\left(2\pi \frac{1}{\lambda} x_\theta + \varphi\right)\right) \quad (2.22)$$

where $x_\theta = xcos\theta + ysin\theta,$ $y_\theta = -xsin\theta + ycos\theta,$

The parameter,$\boldsymbol{\theta}$ **is the orientation of the filter** and has values of 0°, 45° or 90° (maximum performance at =90°), **γ is the aspect ratio** and has values of 0.5(elliptical shape) and 1.0 (circular shape), **ʎ is the wavelength** such that $\lambda < \frac{M}{5}$, **φ is the phase (-** $\boldsymbol{\pi} \leq \varphi \leq \boldsymbol{\pi}$**)** with a best result at 0°, and **σ is the standard deviation given by σ = 0.6ʎ**.

The Gabor filter is made up of the real part and the imaginary part. In most applications, the real part of Gabor filter is used (Amayeh and Tavakkoli, 2009) and therefore, in practice, Gabor filter is represented as shown in equation (2.23).

$$h(x,y,\theta,\lambda,\gamma) = e^{\frac{x_\theta^2+\gamma^2 y_\theta^2}{2\sigma^2}} * cos\left(2\pi\frac{1}{\lambda}x_\theta + \varphi\right) \quad (2.23)$$

The limitations of Gabor filter are that the maximum bandwidth is approximately one octave and Gabor filters are not optimal if one is seeking broad spectral information with maximal spatial localization.

(iii) Median Filter

The median filter is a nonlinear digital filtering technique used to remove impulse noise from a digital signal. It is of two types: adaptive-length (recursive) and fixed-length (standard) median filter. The use of adaptive-length median filter gives better result compared to fixed-length median filter (Goyal et al., 2013). It gives very good results when used to filter digital image corrupted by impulse noise which is generated from sensors or communication channels because it preserves edges and at the same time removes noise(Gomez-Moreno, 2001), (Chan, Hu & Nikolova, 2004). Noise removal, Edge and information preservation is essential in image filtering owing to the nature of visual perception of image (Bo and Wei, 2007). The median filter is spatial filter that replaces every noisy pixel value in the local image neighbourhood with the median of all the pixel values in the neighbourhood when arranged in order of pixel intensity.

(a) Standard Median Filter

The simplest form of a median filter, standard median filter is a simple nonlinear smoothening filter that suppresses noise while retaining all the sharp edges in the signal (Stork, 2003). It is used to filter image corrupted by fixed-valued impulse (salt and pepper) noise. However, in practice, images are corrupted by random-valued impulse noise (Chan et al., 2004) which standard median filter cannot remove due to blurring and edge jittering of the image. For image filtered using a 2-dimensional standard median filter of row size,$M_f$and column size, N_f. The output of the system is given by equation (2.24).

$$\acute{\Upsilon}(m) = \text{med}\{X(m+R_f,n+C_f),\ \ldots\ X(m-1,n-1),X(m,n),\ \ldots X(m-R_f,n-C_f)\} \quad (2.24)$$

(b) Recursive Median Filter

To achieve a better filtering, a recursive median filter (RMF) is used to filter image corrupted by random-valued impulse noise. The recursive median filter combines multiple stages of the standard median filter of varying filter sizes (filter sizes increases from the first stage to the last stage) to remove only noisy pixels. Some recursive median filters combine linear and nonlinear filters of varying filter sizes.

$$\acute{\Upsilon}(m) = \text{med}\{\acute{\Upsilon}(m+R_f,n+C_f),\ldots\acute{\Upsilon}(m-1,n-1),X(m,n),\ \ldots X(m-R_f,n-C_f)\} \quad (2.25)$$

$$\text{where } R_f = \frac{Mf\ -1}{2}\ ,\ C_f = \frac{Nf\ -1}{2}$$

In real life, 2-stage recursive median filters combine linear filter (first stage) to reduce noise and nonlinear (second stage) to enhance the edge of the image. It uses the concept of threshold comparison to detect noisy pixels and effectively filter image corrupted by high impulse noise unlike standard median filter that removes high impulse noise at the expense of increased distortion (Gomez-Moreno, 2001),(Stork, 2003). For a recursive median filter of row size (M_f) and column size (N_f) . The output is given by equation (2.25). Some popular forms of the recursive median filters are: center-weighed median filter, adaptive center-weighed median filter, the adaptive median filter, the median filter based on homogeneity information and switching median filter. The switching median filter uses the concept of threshold to detect the impulse noise level and therefore select filter bank for effective filtering. These filters are very good at locating the noise, even in image with high noise level. The filtering of random-valued impulse-noise is much more difficult compared to the fixed-valued impulse noise.

(iv) **Averaging filter**

The averaging filter is the simplest low-pass filter in digital signal processing with the ability to reduce noise while retaining sharp edges. It replaces each pixel value in the local image neighbourhood with the average of all the pixel values in the neighbourhood. The operation has the effect of eliminating pixel values which did not represent their surrounding pixel value while keeping the sharp edges. It gives optimal performance when applied in the removal of white (Gaussian) noise. The averaging filter is the worst filter for frequency domain encoded signal since it has little ability to separate the different bands and it exists in two forms: standard averaging filter (SAF) and recursive averaging filter (RAF)(Goyal et al., 2013).

(a) Standard Averaging Filter

Standard averaging filter (SAF) is the most popularly used averaging filter. For a 1-dimensional signal processing, a 1-dimensional averaging filter is used to remove noise and an example of 1×3 standard averaging filter is as shown in equation (2.26). The convolution algorithm for 1-dimensional signal filtering using non-causal 1-dimensional averaging filter is shown in equation (2.27) where M_f is the number of data points and ***m*** sample number(Goyal et al., 2013).

$$h(m) = \frac{1}{3}(111) \tag{2.26}$$

$$\Upsilon(\mathrm{m}) = \frac{1}{M_f}\sum_{j=-R_f}^{R_f} X(m-j) \tag{2.27}$$

In image filtering, a two-dimensional averaging filter is used to remove noise from the image. To ensure that unique center element exists in the filter kernel, an odd filter with square filter size is used in image processing. The two-dimensional averaging filter can be of sizes: 3×3, 5×5, 7×7 and so on. A 3×3 averaging filter is shown in equation (2.28)(Goyal et al., 2013).

$$h(m,n) = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \tag{2.28}$$

The image filtering using an averaging filter is implemented by finding the convolution between the images and averaging filter kernel. A non-causal 2-dimensional averaging filtering convolution is shown in equation (2.29) ***where M_f and N_f are the filter row size and column size respectively***, *m and n are the row and column sample numbers respectively* and j and k are row and column sample intervals respectively(Goyal et al., 2013).

$$\Upsilon(\mathrm{m}) = \frac{1}{M_f N_f} \sum_{j=-R_f}^{R_f} \sum_{k=-C_f}^{C_f} \chi(m-j, n-k) \qquad (2.29)$$

where $R_f = \frac{Mf\ -1}{2}$, $C_f = \frac{Nf\ -1}{2}$

In practice, the standard averaging filters have the following problems:

- A single pixel with a very unrepresentative value can significantly affect the mean value of all the pixels in its neighbourhood.

- When the filter neighbourhood straddles an edge, the filter will interpolate new values for pixels on the edge and so will blur that edge. This may be a problem if sharp edges are required in the output.

(b) Recursive Averaging Filter

When a more accurate average filtering is needed, a recursive averaging filter is applied. A recursive averaging filter is a modified version of standard averaging filter that applies multi-stage standard averaging filter with varying filter sizes.

$$\Upsilon(\mathrm{m}) = X(m) * h_1(m) * h_2(m) * h_3(m) \ldots \qquad (2.30)$$

It involves repeatedly convoluting the signal with a series of standard averaging filter until the required level of enhancement is achieved. The convolution operation of the recursive averaging

filter is given in equation (2.30). Since convolution is a linear process, and therefore obeys the law of associativity, the filtering operation involves convolving the signal with the equivalent convolution result of all the filter stages. This operation can be shown in equation (2.31)(Goyal et al., 2013).

$$\Upsilon(m) = X(m) * h_{eg}(m) \tag{2.31}$$

where $h_{eg}(m) = h_1(m) * h_2(m) * h_3(m) \ldots$

The algorithm for recursive averaging filters is faster than other digital filters since it involves only addition and subtraction as mathematical operations and carries out two computations per point. It uses only integer representation which works better than floating point which can produce unexpected result due to round-off error.

(v) Wiener filter

The wiener (minimum mean square error) filter is adaptive linear filters applied to an image locally by taking into account the local image mean (μ), local image variance (σ^2) and estimated noise variance (δ). The Wiener filter gives light local smoothening when the variance in an image is large, but gives an improved local smoothening when the variance in the image is small (Balakrishnan et al., 2012) . The expression for wiener filtering according to (Barney et al., 2010) is shown in equation (2.32).

$$\Upsilon(m) = \mu + \frac{(\sigma^2 - \delta)(\chi(m,n) - \mu)}{\sigma^2} \tag{2.32}$$

Where μ and σ^2 are local mean and the local variance of the noisy image and δ is the estimated noise variance. The Wiener filter is an adaptive filter that uses the estimation of noise present in the image to determine the extent of filtering.

2.1.5.3 Image Enhancement Using Stick Filtering

Most image enhancement techniques blur the edge information in order to filter out the noise in the images. Further study of the performance of low-pass filters and adaptive lowpassfilters has shown that while these filters successfully removemuch of the speckle, they can cause a loss of detail in low-contrastborder regions. An alternative method of filtering speckle using *sticks*,as proposed in (Czerwinski et al.,1998) and utilized in (Pathak et al.,2000) as a smoothing filter was implementedwith success.The sticks filtering algorithm takes on the challenge of filteringspeckle in ultrasound images, without losing edge detail, by determiningwhether a linear feature passes through pixel (x,y) and then calculatingthe filtered pixel intensity $g(x,y)$, which is the arithmetic mean of neighboringpixels in the direction of the *stick*—the most likely direction ofthe linear feature passing through (x,y). Stick filter is a formally defined as follows(Czerwinski et al.,1998):

Given original image f, stick length n, andstick thickness k, the set of all sticks is$S = \{s_{\theta i} | i = 1 \ldots 2n-2\}$, where$s_{\theta i}$is a stick of length n, thickness k, and orientation θ_i. Assuming n isthe length in pixels, there are $2n-2$ possible orientations the stick canbe uniquely arranged in. Mathematically, a stick can be described as aspatial filtering mask as expressed in equation (2.33):

$$s_{\theta i}(s,t) = \begin{cases} 1/n & \text{,if } (s,t) \text{ is along angle } \theta_i.; \\ 0 & \text{otherwise,} \end{cases} \quad \text{for all } (s,t) \in D[s_{\theta i}] \tag{2.33}$$

From the above definition of a stick, it is important to see that **convolving**f $with s_{\theta i}$ smooths the speckle while ***highlighting the linear features*** in direction θ_i.

However, pixels with lines passingthroughthem in directionθ_j, where $j \neq i$, will be assigned an undesired graylevel. To correctly filter f, it is important to define the set of all images,each highlighting a different direction θ_i. as expressed in equation (2.34):

$$H = \{h_i | i = 1 \ldots 2n-2\}, \text{ where } h_i = f * s_{\theta i} \tag{2.34}$$

Each pixel (*x,y*) will then have a gray level *g*(*x,y*), as expressed in equation (2.35),

$$g(x,y) = \mathbf{max}\{h_i(x,y)\}, \text{ for } i \ldots 2n-2 \qquad (2.35)$$

By implementing the concept of a *stick* passing through each pixel(*x,y*) and using that stick as a basis for the interpolated intensity *g*(*x,y*),the final filtered image *g* will be smooth and contain contrast enhanced region borders. An example of a set of sticks *S*, of length fivepixels and one pixel thickness is shown in Figure 2.8, where the whiteand black pixels have a value of *0* and *1/5*, respectively (Czerwinski et al.,1998),(Stefan, 2007).

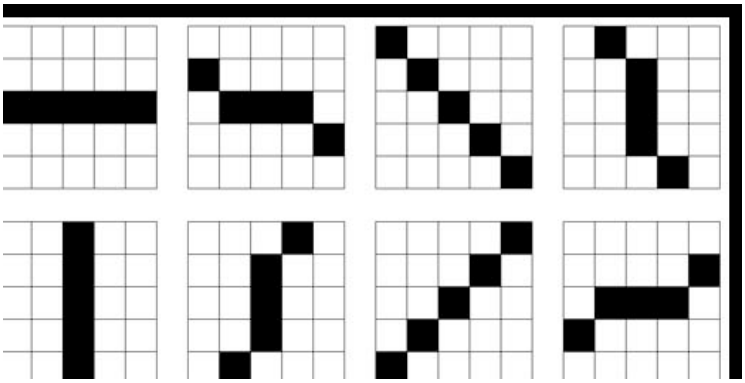


Figure 2.8: Set of stick with n=5 and k=1

Various stick lengths and thicknesses have different filtering effects.Increasing stick length leads to a more smoothly filtered image, at theexpense of weakly highlighting tightly bound curves - a result of the stickbeing longer than some of the boundary edges. Similarly, thicker stickssuppress more noise, at the expense of making thin boundaries less visible.A thick stick can be used to smooth noise, similar to an arithmeticmean filter, with the addition of highlighting broad region differences(Stefan, 2007).

Studies by Czerwinski et al. (1999) showed that increasing stick length remove more speckle noise, while blurring the details of the tightly curving edges. Also they have found that increasing stick thickness enhances the broad edges, while blurring the thin edges.Awad (2007) used the stick technique with a length of 15 pixels to enhance ultrasound images. It is found that applying

sequential sticks with lengths varying from 3 pixels to 17 pixels, an increment of two improves the details of the tightly curving edges and decreases the speckle noise much better than the fixed sticks technique. The intensity level of the image is readjusted to the original level.

2.1.5.4 Summary of Image Enhancement

Image enhancement produces an image which gives better result either for manual or automated examination. This achieved through smoothing or sharpening the image by operating on its pixel properties or frequency components of the image. Convolution or Homomorphic operations are used in most cases to generate the filtering function. Sometimes the function is performed in a recursive many to improve filering performance.

2.1.6 Expert Systems Concepts

An expert is a person who has the special knowledge, judgment, experience, and methods to give advice and solve problems, along with the ability to apply these talents (Turban et al., 2005). Expert systems embody knowledge about a specialized field of human endeavour, such as medicine, engineering, or business. Expert systems have been defined by various scholars. The following definition is given in (Nilsson, 1998):

> AI programs that achieve expert-level competence in solving problems by bringing to bear a body of knowledge are called knowledge-based systems or expert systems. Often the term expert system is reserved for programs whose knowledge base contains the knowledge used by human experts, in contrast to knowledge gathered from textbooks or non-experts. More often than not, the two terms – expert system and knowledge-based systemare used synonymously.

The major components of an expert system are as follows: knowledge base, inference engine, and user interface. In general, though, an expert system that interacts with the user can contain some

additional components like: knowledge acquisition subsystem and Explanation subsystem. Figure 2.9 shows the general structure of an expert system(Nilsson, 1998).

2.1.6.1 User Interface

This might consist of some kind of natural language processing system that allows the user to interact with the system in a limited form of natural language. Graphical user interfaces with menus and electronic forms are used sometimes.

2.1.6.2 Explanation Subsystem

The explanation subsystem analyzes the structure of reasoning performed by the system and explains it to the user (Nilsson, 1998). It explains the expert system behaviour by interactively answering questions like: Why was a certain question asked by the expert system? How was a certain conclusion reached? Why was a certain alternative rejected? What is the plan to reach the solution? (Turban et al., 2005).

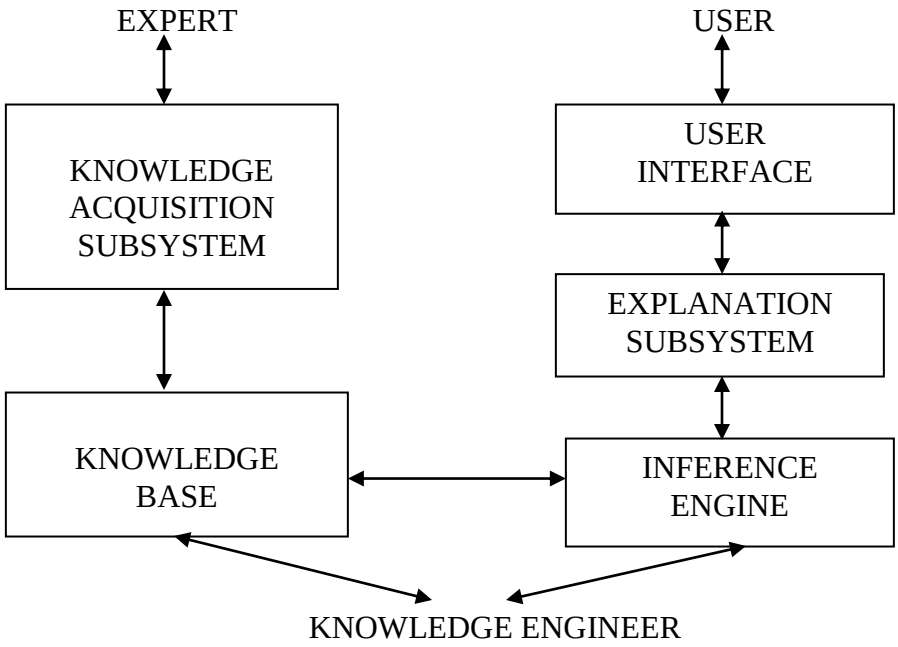


Figure 2.9: Basic structure of an expert system

2.1.6.3 Knowledge Acquisition Subsystem

This component performs the function of accumulating, transferring and transforming the expertise for solving the specific problem from the expert or documented knowledge source to a computer program for constructing or expanding the knowledge base. The knowledge to be stored can be sourced from human experts, textbooks, multimedia documents, databases, special research reports, and information available on the web. Knowledge acquisition involves the knowledge engineer interacting with one or more experts to build a rich and reliable knowledge base (Turban et al., 2005), (Nilsson, 1998).

2.1.6.4 Knowledge Base

This is the foundation of an expert system. It houses the required knowledge necessary for understanding, formulating and solving problems. A typical knowledge base may include two basic elements (i) facts such as the problem situation and the theory of the problem area; and (ii) special heuristics or rules that direct the use of knowledge to solve specific problems in a particular domain. (In addition, the inference engine can include general purpose problem-solving and decision-making rules). Knowledge, not mere facts, is the primary raw material of expert system (Turban et al., 2005),(Nilsson, 1998).

Expert knowledge must be represented in a computer-understandable format and organized properly in the knowledge base. There are many different ways of representing human knowledge, including production rules, semantic networks, and logic statements. In rule-based systems, knowledge in the knowledge base is represented in IF - THEN rules that combine the CONDITION and the CONCLUSION for handling a specific situation. The IF part carries the CONDITION for the rule to be activated, and the THEN part contains the ACTION or CONCLUSION if all IF conditions are satisfied. An example of knowledge-based rules generated for an expert system for selecting a notebook computer (Turban et al., 2005):

Rule 1:
IF the primary task = word processing

AND primary usage = travel
THEN weight requirement = light

Rule 2:
IF the primary task = word processing
AND primary usage = office
THEN weight requirement = don't care

Rule 3:
IF budget <= 2,000
AND budget > 1,000
AND weight requirement = light
THEN NB model = Dell Latitude X1

Rule 4:
IF budget < 1,000
AND weight requirement = don't care
THEN NB model = Toshiba Satellite A100

The advantage of using production rules is that this method is easy to understand, and new rules can easily be added to the knowledge base without affecting existing rules. Uncertainty associated with each rule can be added to enhance its accuracy. A major task of expert system development is to acquire knowledge from human expert and then convert them to knowledge-based rules (production rules) that an inference engine can handle. The inference engine chooses applicable rules from the knowledge base integrates them, and then reasons to find the conclusion (Turban et al., 2005).

2.1.6.5 Uncertainty Handling in Expert Systems

Information can be incomplete, inconsistent, uncertain, or all three. In other words, information is often unsuitable for solving a problem. Uncertainty is defined as the lack of the exact knowledge that would enable us to reach a perfectly reliable conclusion (Negnevitsky, 2002), (Dubey et

al.,2014). It is lack of information to formulate a decision. Classical logic permits only exact reasoning. It assumes that perfect knowledge always exists and the *law of the excluded middle* can always be applied:

IF	A is true	IF	A is false
THEN	A is not false	THEN	A is not true

(i) Types of Uncertainty

a) **Uncertainty**- It is not possible to determine whether anassertion in the model is true or false. For example, theremight be uncertainty about the fact "the height of plant is38."

b) **Imprecision**- The information available in the model isnot as specific as it should be. For example, when adistinct value is required, the information availablemight be a range (e.g., "the height of plant is between 37and 43"**),** disjunctive (e.g., "the height of plant is either37 or 43"), negative (e.g., " height of plant is not 37"), oreven unknown (often referred to as incompleteness).

c) **Vagueness**- The model includes elements (e.g.,predicates or quantifiers) that are inherently vague; forexample, "Plant is in early middle age." A particularformalization of vagueness is based on the concept offuzziness.

d) **Inconsistency**-- The model contains two or moreassertions that cannot be true at the same time; forexample, “height of plant is between 37 and 43" and "theheight of plant is 35."

e) **Ambiguity**--some elements of the model lack completesemantics, leading to several possible interpretations.For example, it may not be clear whether statedtemperature is in Fahrenheit or Celsius.

(ii) Sources of Uncertain Knowledge

(a) **Weak Impplications:**Domain experts and knowledge engineers have the painful task of establishing concrete correlations between IF (condition) and THEN (action) parts of the rules. Therefore, expert systems need to have the ability to handle vague associations, for

example by accepting the degree of correlations as numerical certainty factors. Information can be unreliable.

(b) **Imprecise Language**: Our nautral language is ambiguous and imprecise. We describe facts with such terms as *often* and *sometimes*, *frequently* and *hardlyever*. As a result, it can be difficult to express knowledge in the precise IF-THEN form of production rules. However, if the meaning of the facts is quantified, it can be used in expert systems.

(c) **Unknown Data**: When the data is incomplete or missing, the only solution is to accept the value "unknown" and proceed to an approximate reasoning with this value. Incomplete information for drawing inferences.

(d) **Combining the Views of Different Experts**: Large expert systems usually combine the knowledge and expertise of a number of experts. Unfortunately, experts often have contracdictory opinions and produce conflicting rules. To resolve the conflict, the knowledge engineer has to attach a weight to each expert and then calculate the composite conclusion. But no systematic method exists to obtain these weights (Negnevitsky, 2002), (Dubey et al.,2014).

2.1.6.6 Strategies for Dealing with Uncertainty in Expert Knowledge

Dealing with uncertainty requires reasoning under uncertainty along with possessing a lot of common sense. Different strategies or theories have been proposed for handling uncertainty in expert systems. The basis for selection depends on the nature of uncertainty. Some of the theories include; Bayesian Probability, Certainty Factor Theory, Hartley Theory, Shannon Theory, Dempster-Shafer Theory, Markov Models and Zadeh's Fuzzy Theory(Dubey et al., 2014). Bayesian Probability Reasoning and Certainty Factors Theory shall be discussed briefly.

(a) Bayesian Reasoning

Bayes's theorem is a mechanism for combining new andexistent evidence, usually given as subjective probabilities. Itis used to revise existing prior probabilities based on newinformation. The Bayesian approach is based on subjectiveprobabilities (i.e., probabilities estimated by a

managerwithout the benefit of a formal model); a subjectiveprobability is provided for each proposition.

Suppose all rules in the knowledge base are represented in the following form in equation (2.36):

IF H is true
THEN E is true {with probability p} (2.36)

This rule implies that if event H occurs, the the probability that event E will occur is p. In expert systems, H usually represents a hypothesis and E denotes evidence to support this hypothesis. The Bayesian rule expressed in terms of Hypotheses and evidence looks like equation (2.37):

$$p(\mathrm{H}|\mathrm{E}) = \frac{p(\mathrm{E}|\mathrm{H}) * p(\mathrm{H})}{p(\mathrm{E}|\mathrm{H}) * p(\mathrm{H}) + p(\mathrm{E}|\neg\mathrm{H}) * p(\neg\mathrm{H})} \quad (2.37)$$

where$p$(H) is the prior probability of hypothesis H being true;
p(E|H) is the probability that hypothesis H being true will result in evidence E;
p(¬H) is the prior probability of hypothesis H being false;
p(E|¬H) is the probability of finding evidence E even when hypothesis H is false.

In expert systems, the probabilities required to solve a problem are provided by experts. An expert determines the ***prior probabilities*** for possible hypotheses p(H) and p(¬H) , and also the ***conditional probabilities*** for observing evidence E if hypothesis H is true, ***p*(E|H),** and if hypothesis H is false, p(E|¬H).

Users provide provide information about the evidence observed and the expert system computes p(H|E) for hypothesis H in the light of the user-supplied evidence E. Probability ***p*(H|E)** is called the ***posterior probability*** of hypothesis h upon observing evidence E (Negnevitsky, 2002), (Dubey et al.,2014).

The bias of the Bayesian method is that the framework for Bayesian reasoning requires probability values as primary inputs. The assessment of these values usually involves human judgment. However, psychological research shows that humans either cannot elicit probability values consistent with the Bayesian rules. This suggests that the conditional probabilities may be

inconsistent with the prior probabilities given by the expert. Domain experts do not deal with conditional probabilities and often deny the very existence of the hidden implicit probability (Negnevitsky, 2002).

(b) Certainty Factors Theory and Evidential Reasoning

Certainty theory is a framework for representing andworking with degrees of belief of true and false inknowledge-based systems. In certainty theory as in fuzzylogic, uncertainty is represented as a degree of belief. Thereare two steps in using every nonprobabilistic method ofuncertainty. First, it is necessary to be able to express thedegree of belief. Second, it is necessary to manipulate (e.g.,combine) degrees of belief when using knowledge-basedsystems. Certainty theory relies on the use of certaintyfactors. Certainty factors (CF) express belief in an event (or afact or a hypothesis) based on evidence (or on the expert'sassessment). There are several methods of using certaintyfactors to handle uncertainty in knowledge-based systems.One way is to use 1.0 or 100 for absolute truth (i.e., completeconfidence) and 0 for certain falsehood. Certainty factors arenot probabilities. For example, when we say there is a 90percent chance of rain, there is either rain (90 percent) or norain (10 percent)(Dubey et al., 2014). In a non probabilistic approach, we can saythat a certainty factor of 90 for rain means that it is very likelyto rain. It does not necessarily mean that we express anyopinion about our argument of no rain (which is notnecessarily 10).Thus, certainty factors do not have to sum upto 100. Certainty theory introduces the concepts of belief anddisbelief (i.e., the degree of belief that something is not goingto happen).These concepts are independent of each other andso cannot be combined in the same way as probabilities(Negnevitsky, 2002), (Dubey et al.,2014).

Another assumption of certainty theory is that the knowledge content of rules is much more important than the algebra ofconfidences that holds the system together. Confidence measures correspond to the information evaluations that human experts attach to their conclusions (e.g., "It is probably true" or "It is highly unlikely"). Certainty factors represent information about how certain the conclusion in a rule may be. Certainty factors can be attached both to the conditions in an if-then rule and to its conclusion. They aread hoc values, given by the experts based on experience or bythe

users when providing initial data. Certainty factors arenot probabilities, they represent beliefs about how strong given evidence is and to what degree the evidence supports ahypothesis. Certainty factors are measured using variousscales bot numeric (0 – 100, 0 – 10, 0 – 1, -1 to 1) andlinguistics ones (certain, fairly certain, likely, unlikely,highly unlikely, definitely not), see Table 2.2. It is used to judge uncertain evidence or conclusion. It deals with evidence in terms of their belief ordisbelief of each hypothesis (Dubey et al., 2014).

Table 2.2 Uncertain Terms and their Interpretation
(Negnevitsky, 2005)

Term	**Certainty Factor**
Definitely not	-1.0
Almost certainly not	-0.8
Probably not	-0.6
Maybe not	-0.4
Unknown	-0.2 to +0.2
Maybe	+0.4
Probably	+0.6
Almost certainly	+0.8
Definitely	+1.0

- Highercertainty factors indicate strong confidence in a hypothesis.
- Certainty factors that approach -1 indicate confidence against a hypothesis.
- Certainty factors around 0 mean that we don’t have information either for or against a hypothesis.

As the certainty factor (CF) approaches 1, the evidence isstronger for a hypothesis; as CF approaches - 1, theconfidence against the hypothesis gets stronger; and a CFaround 0 indicates that either little evidence in the rule'sreliability. Certainty measures may be adjusted to tune thesystem's performance, although slight variations in theconfidence measure tend to have little effect on the overall ofthe system. This second role of certainty measures confirmsthe belief that "the knowledge gives the power", than is, theintegrity of the knowledge itself best supports the productionof correct diagnoses (Isram andSimri, 2013).

In expert systems with certaintyfactors, the knowledge base consists of a set of rules that havethe following syntax in equation (2.38):

IF <evidence>
THEN <hypothesis> {cf} (2.38)

where *cf* represents *belief* in *hypothesis*, H given that *evidence*, E has occurred.

The certainty factors theory is based on twofunctions: measure of belief *MB(H,E)* and measure ofdisbelief *MD(H,E).*The values of *MB(H,E)* and *MD(H,E)* range between 0 and 1 in equations (2.39 and 2.40).

$$MB(H,E) = \begin{cases} 1 & \text{if } p(H) = 1 \\ \dfrac{\max\,[p(H|E),p(H)] - p(H)}{\max\,[1,0] - p(H)} & \text{otherwise} \end{cases} \quad (2.39)$$

$$MD(H,E) = \begin{cases} 1 & \text{if } p(H) = 0 \\ \dfrac{\min\,[p(H|E),p(H)] - p(H)}{\min\,[1,0] - p(H)} & \text{otherwise} \end{cases} \quad (2.40)$$

Where $p(H)$ is the prior probability of hypothesis H being true;
$p(H|E)$ is the probability that hypothesis H is true given evidence E;

The strength of belief or disbelief in hypothesis, H depends on the kind of evidence, E observed. Some facts may increase the strength of belief, but some increase the strength of disbelief. The total strength of belief or disbelief in a hypothesis is as expressed in equation (2.41) (Negnevitsky, 2002):

$$cf = \frac{\text{MB}\,(H,E) - \text{MD}\,(H,E)}{1 - \min\,[\text{MB}\,(H,E),\text{MD}\,(H,E)]} \quad (2.41)$$

The certainty factor assigned by a rule is propagated through the reasoning chain. This involves establishing the net certainty of the rule consequent when the evidence in the rule antecedent is uncertain given by equation (2.42):

$$cf(H,E) = cf(E) * cf \quad (2.42)$$

For example,

IF sky is clear
THEN the forecast is sunny {cf 0.8}

and the current certainty factor of "*sky is clear"* is 0.5, then cf(H,E) = 0.5 * 0.8 = 0.4 . This result can be interpreted as "It may be sunny".

The certainty factor for **conjunctive rules** such as equation (2.43):

IF <evudence, E_1>
.
.
.
AND < evidence, E_n>
THEN <hypothesis, H> {cf} (2.43)

the certainty of hypothesis H, is established as follows in equation (2.44):

$$cf(H,E_1 \cap E_2 \cap \ldots \cap E_n) = \min [cf(E_1), cf(E_2), \ldots, cf(E_n)] * cf \quad (2.44)$$

For example,

IF *sky is clear*
AND the *forecast is sunny*
THEN the action is '*wear sunglasses'* {cf 0.8}

and the current certainty factor of "*sky is clear"* is 0.9 and the certainty of the *forecast of sunny* is 0.7, then

$cf(H,E_1 \cap E_2) = \min[0.9, 0.7] * 0.8$
$= 0.7 * 0.8 = 0.56$

The certainty factor for **disjunctive rules** such as equation (2.45):

IF <evudence, E_1>
.
.
.
OR < evidence, E_n>
THEN <hypothesis, H> {cf} (2.45)

the certainty of hypothesis H, is established as follows in equation (2.46):

$$cf(H,E_1UE_2U \dots UE_n) = \max [cf(E_1), cf(E_2), \dots, cf(E_n)] * cf \quad (2.46)$$

For example,

IF sky is overcast

OR the forecast is rain

THEN the action is 'take an umbrella' {cf 0.9}

and the current certainty factor of "*sky is clear"* is 0.6 and the certainty of the *forecast of rain* is 0.8, then

$cf(H,E_1UE_2) = \max[0.6, 0.8]* 0.9$

$= 0.8 * 0.9 = 0.72$

Certainty factors are used if theprobabilities are not known or cannot be easily obtained.Certainty theory can manage incrementally acquired evidence, the conjunction and disjunction of hypotheses, aswell as evidences with different degrees of belief. Althoughthe certainty factors approach lacks the mathematicalcorrectness of the probability theory, it outperformssubjective Bayesian reasoning in such areas as diagnostics.Certainty factors are used in cases where the probabilities arenot known or are too difficult or expensive to obtain. Theevidential reasoning mechanism can manage incrementallyacquired evidence, the conjunction and disjunction ofhypotheses, as well as evidences with different degrees ofbelief. The certainty factors approach also provides betterexplanations of the control flow through a rule-based expertsystem(Negnevitsky, 2002).

This alternative is chosen, because the uncertainty in the proposed expert system design is not tangible, a certainty factor of unity (1.0) is assumed for majority of the hypothesis (conditions) and evidences (actions). Moreover the proposed expert system is not an interactive one; it is automated as soon as an image is selected or acquired. No other user inputs parameters are required in the process of decision making by the system. The question of how certain the expert knowledge is has been settled before rules are designed. The logic in the rules designed is only the path where the expert is certain the rule will follow action given the conditions, otherwise the alternative action will be taken.

The increase of complexity of the system makes uncertainty and vagueness become more tangible (Sikchi and Ali, 2013). It therefore implies that expert systems that are less complex in decision making parameters involve intangible uncertainty issues. It is common in analytical systems whose decision making is based on outcome of experimental tests (such as image analysis or test specimen analysis). Also in expert systems whose decision requires little or no patient related symptom input response.

2.1.6.7 Inference Engine

Inference is the process of chaining multiple rules together based on available data (Turban et al., 2005). The inference engine consists of all the processes that manipulate the knowledge base to deduce information requested by the user. It is also known as the control structure or the rule interpreter (in rule-absed expert system). Basically, it is a computer program that provides a methodology for reasoning about information in the knowledge base and for formulating conclusions (Nilsson, 1998). The inference engine provides directions about how to use the system's knowledge by developing the agenda that organizes and controls the steps taken to solve problemswhenever consultation takes place. Two common inference approaches are forward chaining and backward chaining. Forward chaining starts with the IF part of the rule, and ensure the CONDITIONS for the CONCLUSION are found. Backward chaining is the reverse of forward chaining. It starts from the CONCLUSION and hypothesizes that the CONCLUSION is true. The inference engine then identifies the IF CONDITIONS necessary for making the CONCLUSION true

and locates facts to test whether the IF CONDITIONS are true (Turban et al., 2005), (Nilsson, 1998).

2.1.6.8 Summary of Expert Systems Concepts

Expert systems are computer-based information systems that use expert knowledge to attain high-level decision performance in a narrow problem domain (Turban et al., 2005), (Nilsson, 1998). The basic concepts of expert systems include how to determine who experts are, the definition of expertise, how expertise can be transferred from a person to a computer, and how the system works (Turban et al., 2005), (Nilsson, 1998).

2.2 Review of Related Works on Prostate Segmentation

Over the last few years of intellectual discourse and research, the problem of prostate segmentation, delineating boundaries and volumes, has received considerable attention(Pham et al., 1998),(Xin et al., 2009),(Chiu, 2003). Different segmentation techniques have been applied by different researchers. Among these techniques include edge-detection, graph-searching, deformable models, geodesic active contours, thresholding, classification, clustering, region growing, split and merge, atlas-guided, artificial neural network, and watershed. There are many segmentation techniques. However, there is no standard segmentation technique that works well for all images. Many of these techniques have been used for segmenting medical images by different research groups with varying degrees of success. With the huge number of the developed techniques,a classification of image segmentation techniques becomes an essential task.Different classifications have been proposed. For example, in Withey and Kole(2007), segmentationtechniques are classified into three generations according to the level ofalgorithmic complexity which is added by each of them. In this thesis, theclassification of the medical image segmentation techniques is performed sothat the algorithms in the same class have some common properties. Additionally,the algorithms in different classes show certain distinguishable properties.Hence, all classes together can include all algorithms.

The review shall be undertaken according to three classes of segmentation techniques proposed by Lakare and Kaufman (2000), namely structural, stochastic and hybrid. This review shall concentrate on some relevant studies on segmenting the prostate gland.

2.2.1 Structural Segmentation Techniques

These techniques use some of the information on the structural features of the region to be segmented to implement segmentation. Some of the algorithms include edge-detection, graph searching, isosurfaces and level sets, and deformable models.

Aarnink et al. (1994) introduced a practical method for automated prostate contour detection based on an edge detection pre-processing algorithm (minimum/maximum filtering). In this algorithm, the second derivative in gradient direction is implemented with local minimum and maximum filters, combined with a gray value to assess the local gradient in the direction from the minimum to the maximum. With this algorithm, all edges in the images are detected. Then knowledge based features, such as expected shape (kidney-like) and ultrasonographic appearance of the prostate (looking from within the gland, the edges to be detected are transitions from dark to light), are used to select the correct edges. With adaptive interpolation, the prostate boundary is finally produced. A major disadvantage of their method is that artifacts such as cysts, calcification, and shadowing could lead to erroneous edges.

In order to improve the edge detection and localization, Aarnink et al. (1998) further introduced a technique of adjusting the edge detection parameter to signal information. First they investigated the influence of parameter settings associated with the filtering (the size of the smoothing filter and the size of minimum/maximum filter). Then the local SD (standard deviation) of the gray value is used to identify more or fewer homogeneous regions that are filtered with coarse resolution, whereas areas with greater deviation indicate that gray level transitions occur, which should be preserved with the use of smaller filter sizes to improve edge localization. With the improved result, more interpolation to find a closed contour is needed, and the definition of the prior knowledge becomes important.

Pathak et al.(2000) used an edge-based technique for outlining the prostate boundary. Their algorithm comprises three stages. First an algorithm called Sticks is used to enhance contrast and at the same time reduce speckle in the TRUS (Czerwinski et al., 1993). Then the sticks-enhanced image is further smoothed by an algorithm called weak membrane fitting, which is similar to the anisotropic diffusion filter(Perona and Malik, 1990). Finally, some basic prior knowledge of the prostate, such as shape and ultrasonographic appearance, as addressed previously, is used to detect the most probable edges describing the prostate (Aarnink et al., 1994). This detected prostate boundary is presented as a visual guide to the observer, followed by manual editing. The detected edges are overlaid on top of the image and then presented as a visual guide to the observers for manually delineating the prostate boundary.

Pathak et al. (1998)in a further work presented an algorithm based on snakes to detect the prostate boundary from TRUS. First the sticks algorithm is used to selectively enhance the contrast along the edges, and afterward an active contour model or snakes are applied. This integrated algorithm requires an initial curve input by the user for each ultrasonographic image to initiate the boundary detection process, and the results show their sensitivity to the curve initialization. When an initial contour is reasonably close to the prostate boundaries, the algorithm successfully delineates the prostate boundaries in an image.

Ladak et al. (2000) developed a method based on a deformable contour model, named the discrete dynamic contour (DDC)(Lobregt and Viergever, 1995). In this method, initialization requires the user to select only four (4) points fromwhich the outline of the prostate is estimated by cubic interpolation functions and shape information. In order to improve the model's performance, gradient direction information is used during deformation to push the model toward the boundaries. However, the success of their approach is dependent on the careful initialization of the contour, which requires the user to select points on the prostate boundary. In order to overcome this drawback, the authors added a tool to edit the detected boundary and then re-deform it.

2.2.2 Stochastic Segmentation Techniques

These techniques are applied on discrete pixels without considering any structural information of the region to be segmented (Awad, 2007). They are based only on statistical analysis. Among these are clustering, thresholding, classifiers (pattern recognition), and markov random field models.

Wahba (2008) implemented Fuzzy C-Means (FCM) clustering algorithm where segmentation is obtained via fuzzy pixel classification. FCM allows pixels to belong to multiple classes with varying degrees of membership. It employs an objective function that is intuitive. The technique clusters data by iteratively computing a fuzzy membership function and a mean value estimates for each cluster. The FCM objective function is minimized when high membership values are assigned to pixels whose intensities are close to the centroid for its particular cluster, and low membership values are assigned when the pixel data is far from the centroid. This technique was applied on a set of biomedical TRUS images without any pre-processing steps to enhance the images. The author observed that FCM segments the prostate inaccurately due to the speckled noise that is inherently present in ultrasound image. FCM could not handle this noise because it appends all pixels with membership values satisfying the above criteria to the prostate cluster. Hence, they conclude that FCM gives accurate segmentationresults only for data sets composed of well-separated clusters, which isnot always the case in medical images. In other words, the images to be segmentedusing FCM, must undergo a series of enhancing techniques to removethe noise, so that FCM can give accurate results. They also noted that FCM consumes time which may not be acceptable in critical medical applications.

Ohm and Ma (1997)proposed a feature based cluster segmentation method for image sequences. The algorithm analyses specific features from the image sequence and checks their reliabilityand evidence locally for images in order to build segments that are probably part of one object. The segmentation procedure is basically a clustering procedure, which takes into account different features such as colour and motion. The approach is similar to vector quantisation.Different weights are applied to different features according to their reliability. The pixel-feature vector is then compared to a set of cluster-feature vectors and hence classified to a feature class to generate clusters. The set of cluster feature vectors is updated for each image inthe sequence, which is also used for the segmentation of the next image. The labels associatedwith different clusters remain identical for different images of a same scene. After a scene change, the whole process is started

with a completely new set of feature vectors, which arecomputed, from the first frame after the change and hence the whole sequence is segmented.The algorithm was performed on a sequence of flower garden images. In all of the frames the segmentation and tracking was done automatically. The authors concluded that the algorithm isa hybrid combination of a block-recursive and a pixel-recursive technique and produces a densevector field. The algorithm also has the capability to set flexible weights for different feature components automatically.

Li et al.(1997) proposed that the use of two dimensional histograms of an image is more useful forfinding thresholds for segmentation rather than just using grey level information in onedimension. In 2D histograms, the information on point pixels as well as the local grey level average of their neighbourhood is used. The authors show that the application of Fisher lineardiscriminant to the histogram results in an optimal projection where the data clusters are betterdefined and hence easier to separate by choosing appropriate thresholds. The experimental results show that the proposed technique has the same computational demands as of onedimensional techniques but gives better segmentation results.

2.2.3 Hybrid Segmentation Techniques

Hybrid approaches are techniques which apply a combination of structural and stochastic features in segmentation of region of interest. They include region growing, Split and merge, atlas-guided, artificial neural network and watershed.

Prater and Richard (1992)described a method for segmenting TRUS images of the prostate using feed-forward artificial neural networks. They presented three (3) neural network architectures for this purpose. Each of these networks was trained with the use of a small portion of a training image segmented by an expert radiologist. This method could provide a good result of segmentation; however, it requires extensive training sets, so it greatly complicates the detection procedure. In addition, incorporating user-specified boundary information into the neural networks is also a difficult task.

Some researchers have proposed automatic algorithms (Shen et al., 2003), (Gonzalez and Woods, 2002)for prostate segmentation in 2D ultrasound images. In order to avoid manual initialization,

manual segmented images are used as a training set to produce the initial shape of the prostate (Shen et al., 2003), (Betrouni et al., 2005). However, this technique has a severe drawback. Since the prostate glands have diverse shapes, the initialization does not work for many images. Moreover, the use of manual segmented images in the training can lead to biased results.

Shen et al. (2003) used the previously mentioned initialization, but acknowledged that such initializations are not possible with different prostate shapes. The authors used Gabor filter features to segment the prostate gland. The evaluation of this algorithm calculated for only eight images, which is not good enough to prove the stability of the algorithm.

Wahba(2008) proposed an automation of the conventional region growing technique for the segmentation of TRUS images. She considered the speckle not as noise, but as informative signals. The algorithm overcomes the manual interaction problem associated with region growing technique by selecting the seed point automatically by using the information from speckle noise to determine the markers. Her approach also relies on pixel intensity and spatial Euclidean distance as stopping criteria and for overcoming the effect of the speckle noise. Her experimental result achieved an average similarity of 96.4% with gold standard.

Eskandari et al. (2012) proposed a method which is based on the features extracted from the intensity of the TRUS images, specifically the gradient of it. Six 2D TRUS images acquired from one patient were used to evaluate the performance of the proposed algorithm. Quantitative assessment of the method was done by comparing the automatic segmentation results with the corresponding gold standard obtained from manual segmentation of the target organ. The resulted accuracy, sensitivity and specificity were 98.69±0.27%, 96.40±1.26% and 99.33±0.45%, respectively.

Wu et al. (2015) exploited theintrinsic properties induced by speckles to facilitate segmentation,based on the observations that sizes and orientations of specklesprovide salient clues to determine the prostate boundary. Sincethe speckle orientation changes in accordance with a statisticalprior rule, rotation-invariant texture feature is extracted alongthe orientations revealed by the rule. In order to address the problem offeature changes due to different speckle sizes, TRUS images aresplit into several arc-like strips. In each strip, every individualfeature vector is sparsely represented, and representation residualsare obtained. The residuals, along with the spatial

coherenceinherited from biological tissues, are combined to segment theprostate preliminarily via graph cuts. After that, the segmentationis fine-tuned by a novel level sets model, which integrates 1) theprostate shape prior, 2) dark-to-light intensity transition near theprostate boundary, and 3) the texture feature just obtained. Theproposed method is validated on two 2-D image datasets obtainedfrom two different sonographic imaging systems, with the meanabsolute distance on the mid gland images only 1.06 ±0.53 mmand1.25 ± 0.77 mm, respectively. The method is also extendedto segment apex and base images, producing competitive resultsover the state of the art.

2.2.4 Summary of Review of Related Works on Prostate Segmentation

Most of the above reviewed techniques depended on statisticalestimationforinitializationand someof these initializationmethodswerenotaccurateenough(Shen et al., 2003).Someothermethodsdepend onchoosingthe rightseed point,otherwisethe algorithmwillnot convergetotherightboundary.Whileall theabovemethodsdependonsolvingoptimizationproblemsthatare parametersensitiveandtimeconsuming.Generally, prostate segmentation methods have limitationswhenthe imagecontainsshadowswithsimilargraylevelandtextureattachedtothe prostate.Inthesecasesthesegmentationerrormayincreaseconsiderably.Anotherproblemmay be the lackof sufficientnumberoftrainingimagesif a learningtechniqueis used.Algorithmsbasedonactive contourshavebeenquitesuccessfullyimplementedwiththemajordrawbackthatthey dependonuserinteractiontodeterminetheseedpointsortheinitialsnake.Based on the previous literature review of the existingmethods, a new approach should ideally be:

- Independent on user interaction as user interaction (e.g.defining seed points or initial contours) has drawbackssuch as time consumption, human error etc.
- Independent on training images where training imagesis typically difficult to obtain, especially if the samplesshould be prepared by an expert. Hence, sample-basedlearning should be avoided.
- Independent on noise level where the approach must berobust and deal with the presence of noise and shadow.

A good ultrasound image segmentation technique should make use ofall prior knowledge such as image features (e.g., intensity) and shape. Allmethods that proved to be successful are based either implicitly or explicitlyon this concept.Our motivation is to achieve the aforementioned requirements for goodimage segmentation, in general, and for good ultrasound image segmentation,in particular, by applying region growing segmentation technique onultrasound imaging modality.

2.3 Review of Image Segmentation Methods

There are many segmentation techniques. However, there is no standard segmentationtechnique that works well for all images. This section highlights three classes (Lakare and Kaufman, 2000) of segmentation techniques. The most common image segmentation techniques among the classes are examined. Thefocus in this section is on the region growing algorithm which is the preferred option in this work for medicalimage segmentation.

2.3.1 Structural Segmentation Methods

These methods utilize information about the structural features of the image to implement segmentation of the target image. Some of the common structural methods are discussed under this subsection.

2.3.1.1 Edge-Detection Methods

Edge detection is a problem of fundamental importance in image analysis.In typical images, edges characterize object boundaries; therefore, they areuseful for segmentation. Edge detection techniques are commonly used fordetecting edges in an image to perform segmentation. The classical edge detectiontechniques usually use the edge detection operators, which are basedmainly on the gradation such as Sobel, Robert, and Prewitt edge detectors(Lobregt and Viergever, 1995). Edges are formed at the intersection of two regions with different intensities. Canny method finds edges by looking local maxima of the gradient of the image. The gradient is calculated using the

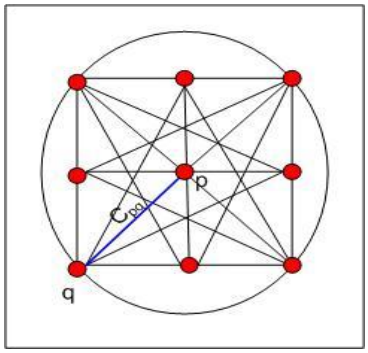
derivative of a Guassian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges (The Mathworks Inc., 2015c).

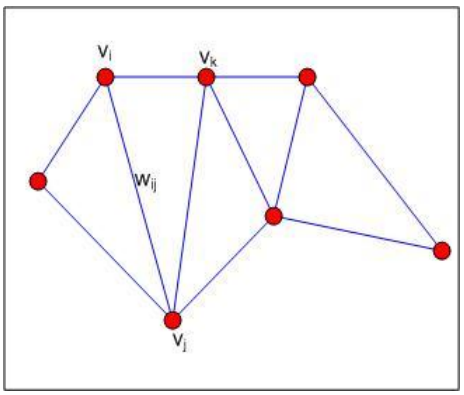
The advantage of these techniques is that they work very well only on images with goodcontrast between different regions. Their disadvantages include; they detect all the edges; hence, it is very difficult to find the relation between the edges and the region of interest. In addition,those algorithms are sensitive to noise.

2.3.1.2 Graph-Searching Algorithms

Graphs are an abstract representation consisting of a set of vertices V = {v_1 , · · · , v_n }, and a set of edges E = {..., {v_i , v_j } , ...}, where e k = {v_i , v_j } indicates that there is an edge (arc) from vertex i to vertex j. Each edge has an associated weight representing the cost of transition from v_i , v_j asshown in Figure2.10(b). In these algorithms, edges and surfaces are representedas graphs, and the algorithm tries to find the lowest-cost path, (C_{pq}) indicatedin Figure2.10(a), between two nodes of the graph using any search algorithm.

Those algorithms have the advantage that they can perform well even if the separationsbetween regions are broken.Their disadvantage is that of requiring surfaces to be represented asgraphs, which could be tricky.





(a) Fully connected graph (b) Graph representation

Figure 2.10: Graph Search Algorithm (Wahba, 2008)

2.3.1.3Deformable Models

Deformable models are physically motivated, model-based techniques for delineatingregion boundaries using closed parametric curves or surfaces that deform under the influence of internal and external forces. To delineate an object boundary in an image, a closed curve or surface must first be placed near the desired boundary and then allowedto undergo an iterative relaxation process. Internal forces are computed from withinthe curve or surface to keep it smooth throughout the deformation. External forces areusually derived from the image to drive the curve or surface towards the desired featureof interest.

Mathematically, a deformable model moves according to its dynamic equations and seeks the minimum of a given energy function (Terzopoulos and Szeliski, 1992). The process of deformation isperformed by minimizing the energy function E(x),eqtn (2.47), that is designed in suchaway that its local minimum is obtained at the boundary of the object(Hamarneh, 2001).An active contour, or snake, is a curve, $\boldsymbol{x}(s)$, with internal (tensionand rigidity) and external (linear features and boundaries) forces, thatmoves through the spatial domain of an image, minimizing an energyfunction, *E(x)*, as expressed in equation (2.47):

$$\mathrm{E(x)} = \int_0^1 \frac{1}{2}[\alpha|x'(s)|^2 + \beta|x''(s)|^2 + E_{ext}\,x(s)]ds, \qquad (2.47)$$

and αis the adjustable weight of the snake's tension, βis the adjustableweight of the snake's rigidity, and $\boldsymbol{x}$' and $\boldsymbol{x}$" are the first and second derivativesof $\boldsymbol{x}$, with respect to *s*. A normalized GVF (gradient vector flow) field is used as the externalforce to minimize *E(x)*and iteratively solve the GVF snake according to equation (2.48):

$$x_t(s,t) = \alpha x''t(s,t) - \beta x''''_t(s,t) + v,$$

where t is time and v is the GVF field (2.48)

The above equations and further details regarding the solution to theGVF snake are explained in detail in Xuand Prince(1998a) andXuand Prince(1998b).

The main advantages of deformable models are their ability to directly generate closed parametric curves or surfaces from images and their incorporation of a smoothness constraint that provides robustness to noise and spurious edges. Moreover,they offer sub-pixel accuracy for the boundary representation, which may beimportant to medical applications.

An important drawback of the snakes is that they requiremanual interaction to place an initial model or to choose appropriateinitial parameters. Besides, they cannot be used in non-interactive applications,unless they are initialized close to the structure to be segmentedbecause they are designed as interactive models. Furthermore, the internalenergy constraints of snakes can limit their geometric flexibility, and preventa snake from representing long tube-like shapes or shapes with significantprotrusions or bifurcations. Additionally, since classical deformable contourmodels are parametric and are incapable of adapting topological changeswithout additional machinery, the topology of the structure of interest mustbe known in advance. Finally, they are non intrinsic because the energy isA disadvantage is that they require manual interaction to place an initial model and choose appropriate parameters.

2.3.2 Stochastic Segmentation Methods

Stochastic methods are based on statistical analysis. These methods operate on discrete pixels of the image without consideration to the structural information of the region of interest.

2.3.2.1 Thresholding Methods

The thresholding technique is the simplest image segmentation technique. It is based on theassumption that the objects and the background in the image have a bimodal distribution.Typically, this assumption is not valid for most images, especially medical ones. The keypoint in this segmentation technique is to determine an intensity value, called the threshold,which separates the desired classes. Thresholding segmentation transforms input image,I, to a binary image, g, by grouping the pixels with intensities, higher than the threshold intoone class, and the other pixels into another class. For any 2D digital image,equations (2.49) and (2.50) applies (Awad, 2007).

$$I(i, j) \quad \text{for } i = 1, 2, \ldots, M \quad \text{and} \quad j = 1, 2, \ldots.., N \tag{2.49}$$

the thresholding technique is defined as:

$$g(i, j) = \begin{cases} 1 & \text{for} \quad I(i, j) \geq T \\ 0 & \text{for} \quad I(i, j) < T \end{cases} \tag{2.50}$$

Some images can be segmented by using more than one thresholding point, which is called multi-thresholding, as depicted in Figure 2.11, where T1and T2 are the two thresholds segmenting the image. The output image, resulting from multi-thresholding, is no longer binary, but consistsof a limited number of grey levels as expressed in equation (2.51):

$$g(i, j) = \begin{cases} 0 & \text{for} \quad I(i, j) < T_1 \\ 1 & \text{for} \quad T_1 * I(i, j) < T_2 \\ . & \\ . & \\ . & \end{cases} \tag{2.51}$$

$$n \quad \text{for} \quad Tn * I(i, j) < T_{Ng-1}$$

where N_g = the number of grey levels in the image (256)

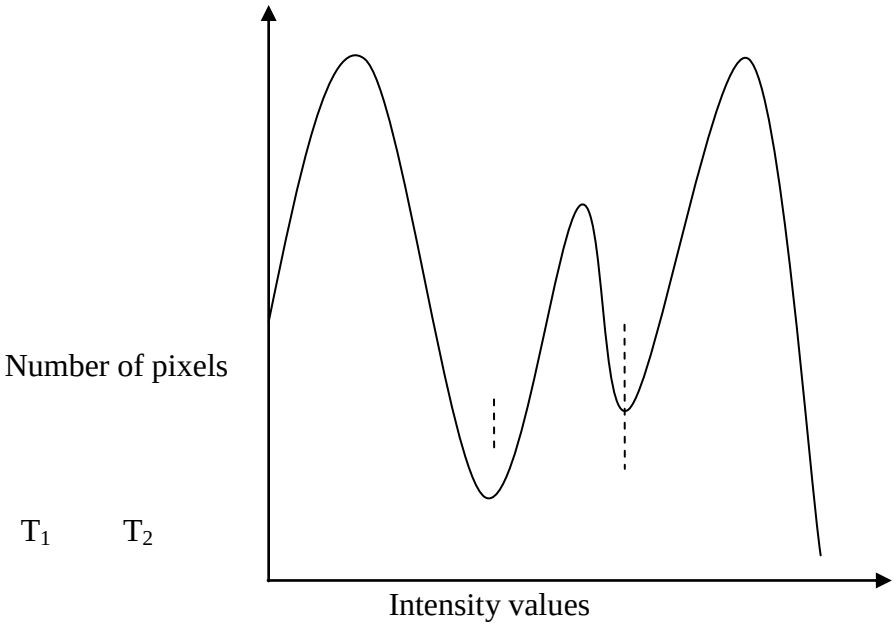


Figure 2.11: Histogram showing three apparent classes, using two thresholds T_1 and T_2

There are two main methods of thresholding segmentation. The simplest and fastestmethod is called global thresholding, where one threshold value is used for the entire image.The second method is called local thresholding, where a different threshold value is assignedto each sub-image. Although the second method is more accurate than the previous one,the second one is more computationally complex. To determine the threshold value, thehistogram of the image is calculated and the minimum value of the histogram is used asthe threshold value. The histogram can be mathematically described as in (Pham et al., 2000),(Sahoo et al., 1992).

Thresholding is used to detectthe pubic arch in TRUS during transperineal prostate brachytherapy (Pathak et al., 1998). The techniqueis also used at the initial stage in multi-stage segmentation techniques. It yields coarseestimation of the objects in the image, which can be modified (Potocnik

et al., 1998). One limitation ofthe global thresholding technique is that it does not include the spatial information of theimage. This causes it to be sensitiveto noise and intensity inhomogeneities. Both these artifacts essentially corrupt the histogram of the image, making separation more difficult. One solution is the connectivity-based thresholding algorithm, proposed in(Lee et al., 1998).However, the thresholding technique is not powerful enough for a complicatedproblem such as prostate segmentation in TRUS.

2.3.2.2 Classifier Methods

Classifier methods are pattern recognition techniques that seek to partition a featurespace derived from the image using data with known labels (Schalkoff, 1992). A feature space isthe range space of any function of the image, with the most common feature space beingthe image intensities themselves. A histogram, as shown in Figure 2.10, is an exampleof a 1-D feature space. Figure 2.12 shows an example of a partitioned 2-D and 3D feature spacewith two apparent classes. All pixels with their associated features would be grouped into one class. Although the features used can be related totexture or other properties, we assume for simplicity that the features are simply intensityvalues.

Classifiers are known as supervised methods since they require training data that aremanually segmented and then used as references for automatically segmenting new data.There are a number of ways in which training data can be applied in classifier methods.

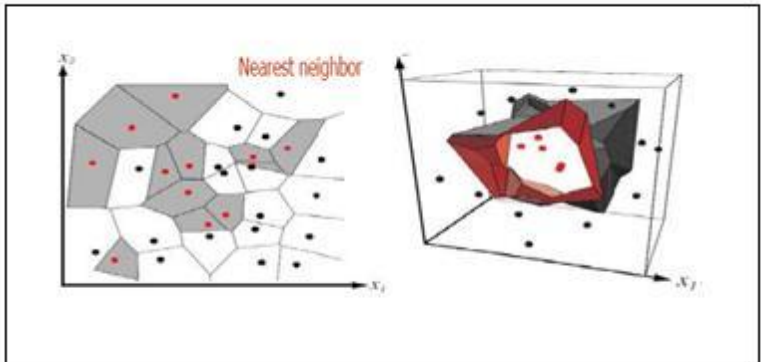


Figure 2.12: Classifier for 2D and 3D (Wahba, 2008)

- ❖ Non Parameteric Classifiers

Non parametric classifiers are not based on any assumptions about the statisticaldata structure. Examples of non parametric classifiers include:

- KNN (K-nearest neighbour) classifier: Each pixel is classified in thesame class as the training datum that has the nearest intensity.
- Parzen window classifier: The classification is made according to themajority of the pixels confined within a predefined window of the feature

(for example, intensity) centered at the unlabeled pixel intensity.

- ❖ Parametric Classifiers

Parametric classifiers assume that pixels intensities are independent samplesfrom a mixture of probability distributions, usually Gaussian. Bayes classifieris a parametric classifier. It is based on a probabilistic model specification(Cheng et al., 2003).

Classifiers are non-iterative, they are relatively computationally efficient. Unlike thresholding methods, they can be applied to multi-channel images. A disadvantage of classifiers is that they generally do not perform any spatial modeling. Another disadvantage is the requirement of manual interaction for obtaining training data. Training sets can beacquired for each image that requires segmenting, but this can be time consuming and laborious. On the other hand, use of the same training set for a large number of scans can lead to biased results which do not take into account anatomical and physiological variability between different subjects(Pham et al., 1998).

2.3.2.3Clustering Methods

Clustering algorithms essentially perform the same function as classifier methods without the use of training data. Thus, they are termed unsupervised methods. In order to compensate for the lack of

training data, clustering methods iterate between segmenting the image and characterizing the properties of the each class. In a sense, clustering methods train themselves using the available data.Three commonly used clustering algorithms are the K-means or ISODATA algorithm, the fuzzy c-means algorithm, and the expectation-maximization (EM) algorithm (Pham et al., 1998).K-means starts by partitioning the input pixels into k initial clusters. It thencalculates the mean pixel, or centroid, of each cluster. It constructs a newpartition by associating each point with the closest centroid. Then, the centroidsare recalculated for the new clusters, and the algorithm is repeated byalternate application of these two steps until convergence, which is obtainedwhen the pixels no longer switch clusters (or alternatively centroids are nolonger changed), is reached.

In FCM, each pixel has a degree of belonging toa number of clusters, as in the fuzzy logic, rather than belonging completelyto just one cluster, which is the case in K-means. Thus, pixels on the edge of a cluster, may be in the cluster to a lesser degree than pixels in the centerof cluster. In other words, FCM allows labels to be ”fuzzy”, that is, a pixelcan be partially in one cluster and partially in others. FCM can also provides information on how well a pixel ”fits” a cluster. Since in clustering techniquesthere is no need for training data, they are termed as unsupervised techniques (Xu and Wunsch, 2005).The EM algorithm applies the same clustering principles with the underlying assumption that the data follows a Gaussian mixture model. It iterates between computing the posterior probabilities and computing maximum likelihood estimates of the means, covariances, and mixing coefficients of the mixture model. Although clustering algorithms do not require training data, they do require an initialsegmentation (or equivalently, initial parameters). The EM algorithm has demonstratedgreater sensitivity to initialization than the K-means or fuzzy c-means algorithms (Pham et al., 1998). Like classifier methods, clustering algorithms do not directly incorporate spatial modeling and can therefore be sensitive to noise and intensity inhomogeneities. This lack ofspatial modeling, however, can provide significant advantages for fast computation.Robustness to noise can beincorporated using Markov random field modeling(Hebert, 1997).

2.3.3 Hybrid Segmentation Methods

2.3.3.1 Artificial Neural Network Methods

Artificial neural networks (ANNs) are massively parallel networks of processing elementsor nodes that simulate biological learning (see Figure 2.13). Each node in an ANN is capable ofperforming elementary computations. Learning is achieved through the adaptation ofweights assigned to the connections between nodes. ANNs represent a paradigm for machine learning and can be used in a variety ofways for image segmentation. The most widely applied use in medical imaging is as a classifier (Hall et al., 1992), (Gelenbe et al., 1996) where the weights are determined using training data, and the ANNis then used to segment new data. ANNs can also be used in an unsupervised fashion asa clustering method (Bezdek et al., 1993), (Reddick et al., 1997) as well as for deformable models (Vilarino et al., 1998). Although ANNs are inherentlyparallel, their processing is usually simulated on a standard serialcomputer, thusreducing this potentialcomputational advantage.

ANNs partially overcomes the drawbacks of the conventional segmentationalgorithms based on structural knowledge, which often requireconsiderable user expertise. In addition, spatial information can be easilyincorporated into its classification procedures due to the large number ofhighly interconnected and tied processing elements.

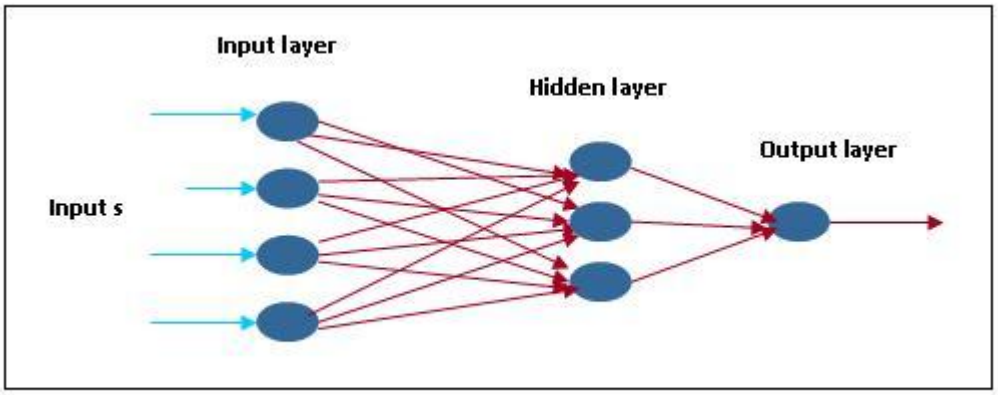


Figure 2.13: Artificial neural network (Karray & Silva, 2004)

2.3.3.2Atlas-Guided Methods

Atlas-guided approaches are a powerful tool for medical image segmentation when astandard atlas or template is available. The atlas is generated by compiling informationon the anatomy that requires segmenting. This atlas is then used as a reference frame forsegmenting new images.

Conceptually, atlas-guided approaches are similar to classifiersexcept that they are implemented in the spatial domain of the image rather than in a featurespace.The standard atlas-guided approach treats segmentation as a registration problem (Maintz and Viergever, 1998).It first finds a one-to-one transformationthat maps a pre-segmented atlas image to the target image that requires segmenting.This process is often referred to as atlas warping. The warping can be performedusing linear (Andreasen et al., 1996), (Talairach and Tournoux, 1998) transformations but because of anatomical variability, a sequentialapplication of linear and non-linear (Sandor and Leahy, 1997), (Davatzikos, 1996)transformations is oftenused. This process results in a correspondence field, which maps each pixelin the atlas space to one in the patient coordinate system. The spatial priorsare then applied to correspondence field resulting in patient specific spatialpriors, which guide the segmentation as indicated in Figure 2.14.

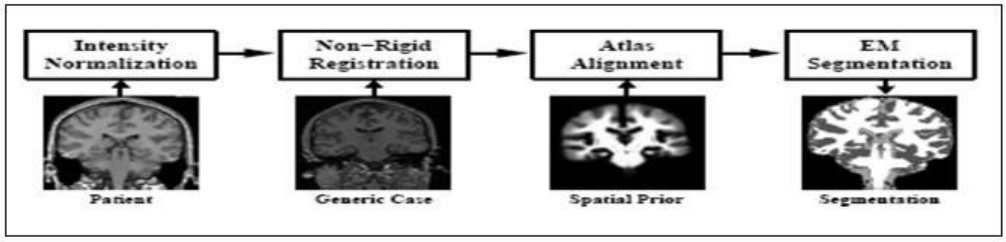


Figure 2.14: Segmentation using atlas-guided approach (Wahba, 2008)

Atlas-guided approaches have been applied mainly in MR brain imaging. An advantage of atlas-guided approaches is that labels are transferred as well as the segmentation.They also provide a standard system for studying morphometric properties (Davatzikos et al., 1996), (Thompson and Toga, 1997). Their disadvantage is that even with non-linear registration methods accurate segmentations of complex structures is difficult due to anatomical variability. Thus, atlas-guided approaches are generally better-suited for segmentation of structures that are stable overthe population of study. One method that helps model anatomical variability is the use ofprobabilistic atlases (Thompson and Toga, 1997) but these require additional time and interaction to accumulatedata.

2.3.3.3Region Growing Methods

Region growing approaches exploit an important fact that the pixels close together have similar gray values. The main idea of this approach consists in the observation that the pixels belonging to one element of the object can possess similar properties, for example, the gray value. Therefore if the considered pixel has gray value that is near the common gray value of the region, this pixel can be associated into this region. In its simplest form, region growing requires a seedpoint that is manually selected by an operator.

This approach generally consists of the following three steps:

- Initially, a new region consists of only one pixel which is called the seed pixel is selected. This pixel is determined by user, and is used as a-priori information. The given initial pixel determines the 8-neighbour where all pixels are sequentially checked. If a pixel does not belong to any other regions it can be associated into the given region if the difference between the gray value of the pixel and the averaged gray value of the whole region is less than a predefined threshold, T (Garcia-Consuegra et al., 2000).

- Second, when the pixel is connected to the region the averaged gray value of this region is re-calculated. After all the neighbouring pixels are reviewed, new neighbourhood for the enlarged region will be determined and the first step is repeated. The averaged gray value of the whole region is being modified therefore the pixels, being declined on previous steps, may be re-associated into the bigger region on the next steps.

- Finally, the region has to be closed if there are no more neighbour pixels that can be associated into this region. After the region is closed the averaged gray value is re-calculated and all the pixels belonging to the region have to be set to this averaged gray value. The pixels outer to the closed region are candidates of new regions and the process is continued until all the pixels of an image are associated into regions.

Sometimes the goal is to determine only one closed region. In this case the algorithm is stopped if the given region is closed. Sometimes in order to improve the recognition accuracy, we may need to combine different algorithms so that results of one algorithm can deliver a-priori information for the other algorithm (Pan, 2004).

Region growing is a bottom-up procedure that starts with a set of seed pixelsas in the algorithm below. The aim is to grow a uniform, connected region from each seed. A pixel is added to a region if and only if: It has not been assigned to another region; It is a neighbor of that region; The new region created by addition of the new pixel is still uniform. Let *f*be an image, and R_1 , R_2 ,..., R_n be a set of regions each consisting of a single seed pixel. Δ**,** is the Characteristic function for determining elements segmented into the set of regions.

Repeat
    for i=1...n
        for each centroid pixel,*p* at the border of *Ri*
            for all neighbors of *p*
                Let *x*, *y* be the neighbor's coordinates
                Let μi be the mean gray level of pixels in *Ri*
                if neighbor unassigned and absolute{f(x,y) -μi}<=Δ
                Add neighbor to *Ri*, updateμi
Until no more pixels are being assigned to regions

The criteria to choose the seed in practice will depend on the nature of the problem. If targets need to be detected using infrared images for example, choose the brightest pixel. Or if without a-priori knowledge, compute the histogram and choose the gray-level values corresponding to the strongest peaks. And how do we choose the similarity criteria? The homogeneity predicate can be based on any characteristic of the regions in the image such as average intensity, variance, color, texture, motion, shape, size and so on (Pan, 2004).

Region growing is a useful algorithm allowing an extraction of different elements contained in an image. Its primary disadvantage is that it requires manual interactionto obtain the seed point. Thus,

for each region that needs to be extracted, aseed must be planted. Region growing can also be sensitive to noise, causingextracted regions to have holes or even become disconnected. Conversely, partial volumeeffects can cause separate regions to become connected(Pham et al., 1998).

2.3.4 Summary of Review of Image Segmentation Methods

The goal of this section was to examine the most common image segmentation techniques to find the most suitable one to use in our research. Several segmentation techniques have been examined, including thresholding, edge-detection, region growing, classifiers, graph-searching, clustering, artificial neural networks, deformable models, and atlas-guided.The concept of eachtechnique has been briefly discussed. Also, the advantages and disadvantages ofeach technique are explored.

Thresholding is straightforward and yields good results for simple images; however it isnot capable of handling ultrasound images. The segmentation problem is also consideredas a classification problem, but this requires training data. Training data is not always available in the medical field. Also, anatomical and physiological variability in the training datadecreases the accuracy of the results. Moreover, it is difficult to create a large training-datasetin the medical field, especially, by an expert. For these reasons, classification techniqueis not suitable for the research described in this thesis. Clustering is also consideredfor image segmentation. Clustering technique does not require training data, a definiteadvantage over the classification approach. However, the accuracy of clustering techniquesis usually less than that of the classification techniques, in particular, with high noisyimages such as ultrasound images. Artificial neural networks(ANNs) are also used in medical image segmentation, but requires a large training-data setwhich is not available for the research point. Moreover, finding the "right" structure of aneural network is not easy.Deformable models technique is applied for image segmentation, but the selection of the appropriateinitial parametersand initialize contour close to the structure to be segmented is challenging. Atlas-guided techniquerequires a template which is not available in many situations.

It is then concludedthat region growing method is amenable to automation and variety of image structures.Region growing technique has been used for segmenting medical images, but is slow because of random search for elements belonging to regions and requires distinct/dynamic seed

pixel for each region of interest. This method shall be adopted with some enhancements to the algorithm by using knowledge-based rule to automate the implementation of:

- **Radial/Axial search/scanning to grow region of interest.**
- **Dominant pixel intensity for characteristic function evaluation.**
- **Static/common Centroid seed point/pixel.**

Also sequential sticks technique shall be used to deal with the speckle noise which usually affects the robustness of region growing segmentation results.

2.3.5 Existing Research Gaps Identified from Review

(i) Most algorithms are semiautomatic requiring human interaction and therefore produce operator-dependent results. *An algorithm that avoid operator-dependent result is needed*.

(ii) Some semiautomatic algorithms require manual editing during the segmentation process which implies more effort, more time, and operator-dependent results.*An algorithm that is fast and requires no further human input after result is generated is the next level of automation required in segmentation algorithm.*

(iii) For most automatic algorithms manually segmented images are adopted as a training data - not suitable for prostate segmentation (*Hard to find a single mean shape for variable prostate shapes.; Hard to create large training data set in the medical field by an expert).*

(iv) Region growing segmentation by random scanning using dynamic seed point is time consuming. *Random scanning and dynamic seed point for segmentation of region of interest in region growing segmentation consumes the time saved by automating it. Hence an different scanning technique that saves time is necessary. Also working with a common/static seed point will also increase the speed of segmentation.*

2.3.6 Proposed Enhanced Region Growing Image Segmentation Algorithm

As stated above in the summary of the review, this work shall use radial/axial search/scanning and static/common centroid seed point and dominant pixel intensity, all facilitated by knowledge-based rule. The algorithm for conventional region growingreveals the random nature of growing regions. This is evidenced by the Repeat … Until construct. This will lead to searching the whole image space each time a new set belongingto the region of interest is to be grown. The loop 'For each centroid pixel, p, at the border of Ri' also shows where a distinct centre pixel is to be fetched for a new set, Ri, that will be part of the region of interest. The characteristic function, Δ, used for the hard segmentation implemented, is a mean gray level of the neighbors of the pixel to be considered for addition to the set. The proposed enhancement shall use a characheristic function based on dominant intensity parameters built from knowledge-based rule.

For the image f, with pixel coordinates as x and y, there are,for the proposed algorithm, two characteristic functions as expressed in equations (2.52) and (2.53);

Fr = (IF f(x,y) =< RegMaxDomIntensity AND f(x,y) >= RegMinDomIntensity) (2.52)

; for determining pixels that belong to the region of the prostate gland.

Fm = (IF f(x,y) =< MaligMaxDomIntensity AND f(x,y) >= MaligMinDomIntensity) (2.53)

; for determining the pixels that belong to the suspected
cancerous region of the prostate gland.

RegMaxDomIntensity and RegMinDomIntensity are respectively, the maximum and minimum dominant intensity for growing prostate gland region. MaligMaxDomIntensity and MaligMinDomIntensity are respectively, the maximum and minimum dominant intensity for detecting the suspected cancerous pixels within the grown region of the prostate gland. The algorithm given below shows the proposed enhanced region growing algorithm.

*{cf_m}*is the certainty factor that the maligmaxdomintensity and maligmindominensity values are true is given as 0.9. *{cf_s}*is the certainty factor that the regmaxdomintensity and regmindominensity values are true is also given as 0.9. The values of the evidences (***RegMaxDomIntensity, RegMinDomIntensity,MaligMaxDomIntensity and MaligMinDomIntensity)*** in the rules were obtained through experimentation with several images, hence the high certainty values assigned. It is therefore almost certainly true (see Table 2.2) that the rule produces results that are almost certainly true, since *{cf_m} x {cf_s} = 0.9 * 0.9 = 0.81*.

For axis, Ax = 1 to 4; to enable scan/searchin 4 axial directions, CEA, AFD, DGB, BHC

For all pixels bound by the axis rowwise and coolumnwise from the common seed point,

IF f(x,y) =< RegMaxDomIntensity AND f(x,y) >= RegMinDomIntensity
THEN Add pixel to R_1 with {cf_s} ; set of Prostate gland region

IF f(x,y) =< MaligMaxDomIntensity AND f(x,y) >= MaligMinDomIntensity
THEN Add pixel to Ro with {cf_m}; set of Malignat reg. Prostate gland
END

END

Next pixel in axis

Next axis

From the definition of segmentation in section 2.1.3.1 and 2.1.3.2, we see that the result of segmdentation will be two sets of regions, namely R_1, region of the prostate gland and R_2 region of tissues surrounding the prostate gland. By definition, we have equation (2.54);

$$f = \bigcup_{k=1}^{2} R_k \qquad (2.54)$$

The expression implies that the image, f is the union of two sets of regions segmented, R_1 and R_2. It is such that ($R_1 \cap R_2$.) = ϕ. That is the two regions do not intersect, producing a null set, ϕ. The third set, Ro is the set of Malignat regions within the prostate gland. Ro is a subset of R_1,(Ro c R_1).

The region of the prostate gland, R_1shall be segmented further, using a zone lookup table to be developed (see Figure 4.18 and the corresponding algorithm in section 4.4), into three sets of regions. They are R_{11}, R_{12}, and R_{13} comprising the zones of the gland, namely; Peripheral/Anterior, Central and Transition zones respectively. Consequently, R_1 is the union of three sets of regions, R_{11}, R_{12}, and R_{13}. Therefore we have the expression in equation (2.55);

$$R_1 = \bigcup_{i=1}^{3} R_{1i} \qquad (2.55)$$

Also, it is such that ($R_{11} \cap R_{12}$.) = ϕ, ($R_{12} \cap R_{13}$.) = ϕ, and ($R_{11} \cap R_{13}$.) = ϕ. That implies that none of the three sets of regions intersect, hence, the intersection of any two is a null set, ϕ.

CHAPTER THREE
METHODOLOGY

3.1 Method of Analysis and Design

There are different methods that are used to analyse and design systems by software engineers (Jawadekar, 2004), (Pressman, 2005). In this work,structured systems analysis and design (SSAD) was adopted. With this method, the system was viewed from different angles and decomposed into modules and components, as a meaningful representation of the system to form the basis for building the software requirement specification(Jawadekar, 2004). SSAD considers data and the processes that transform the data as separate entities. Data objects are modeled in a way that defines their attributes and relationships Processes that manipulate data objects are modeled in a manner that shows how they transform data as data objects flow through the system(Pressman, 2005).

SSAD method of design was chosen because this work deals with expert system, in which data was modeled as knowledge base and the processes that transform data was modeled as inference engine. The first step in structured systems analysiswas to gather information through interviews, questionnaire, existing system manuals, documents, and reports, and so forth. This was followed by analysis of the system environment and finally design and development of the new system arising from the analysis.

3.2 Analysis of Existing System

The existing system's functions, features, information and data are gathered and analysed to bring to light the areas of need for improvement to bring in efficiency and effectiveness in functionality and features.

3.2.1 Information Gathering

The information gathered for the purpose of this research was done using two methods. First, study of physical documents, literatures as well as on-line materials from the internet on the concepts and technologies related to this work was done. Secondly, understudy of the existing processes/methods

of prostate disease diagnosis employed by the experts and consulting with experts associated with management of the disease of the prostate.

Documents and literatures on concepts and technologies related to this work were studied with a view to finding the current practice and determining the direction of contribution and proffering solution as well as improving on existing technologies. The following materials were studied:

i. Samples of TRUS 2D images of the prostate and Ultrasound Machines
ii Samples diagnosis report of TRUS images by radiologists (see section 4.1.3)
iii Literature on the Ultrasound sound machine (see section 2.1.2)
iii. Literature on the Prostate gland, anatomy, histology, and pathology (see section 2.1.1)
iv. Literature on Artificial intelligence problem solving techniques especially image processing and machine learning (see section 2.1.3 to 2.3).
v. Literature on segmentation of ultrasound images especially TRUS images (see section 2.3)
vi. Literature on MATLAB image processing and programming (see section 5.1)

3.2.1.1 Consultation with Experts

The most commonly used form of knowledge acquisition is face-to-face consultation analysis. In this work the walk-through method was applied. The experts were presented with samples of previously handled cases of TRUS prostate image segmentation. Several sessions of consultations were conducted with experts from the University of Nigeria Teaching Hospital, Ituku/Ozalla, Nigeria and University of Ilorin Teaching Hospital, Ilorin, Nigeria involved in the diagnosis and treatment of prostate disease like radiologists, physicians and surgeons at different stages of this work. Consultations were held both to gather initial data, to confirm findings before documentations and during testing for confirmation of test results.

The information elicited from the experts showed that to detect the prostate regions or boundaries from the TRUS image, the following knowledge rules should be applied:

i. TRUS 2D images consist of three different regions: the prostate, the tissues around the prostate, and the background.
ii. The background of the image is black.

iii. The grey level of the prostate is low in respect of the tissues surrounding it.
iv. The prostate is not in the periphery of the image, not necessarily in the middle.
v. The prostate has a smooth curvature shape.
vi. The labels on the image are on the axis and have highest intensity, white.
vii. The suspected malignant tissues within the prostate glands have much lower pixel intensity.

From consultations with experts and the examination of expert radiologist's diagnosis report from TRUS 2D images of the prostate, it was identified that the following parameters are used to indicate the physiological state of the prostate gland which guides physicians in determining the level of disease condition on the prostate gland:

a. The size of the prostate is estimated in area or volume
b. The presence of abnormal tissue textures (hypoechoic regions) is indicated and extent of the zones or lobes where they occur are marked or labelled.
c. The hypoechoic regions are more likely to exhibit cancer.
d. The peripheral zone is more likely to show cancer.
e. The central zone is unlikely to have cancer.
f. It is less likely to have many scattered small regions of cancer that can be recognized radiologically.

3.2.1.2 Data Analysis

The analysis of data gathered on TRUS 2D images of the prostate revealed the following:

i. The prostate images are affected by noise.
ii. The prostate is detected by delineating the boundaries from the surrounding tissues.
iii. The background of TRUS 2D image is black as designed by Ultrasound designers.
iv. The shape of the prostate image shown by TRUS varies from patient to patient.

3.2.2 Limitations of Expert Manual System of Segmentation/Detection

i. The use of only human agent and expertise in the interpretation of diagnostic finding using ultrasound is subject to the degree of expertise of the radiologist on duty. Proper diagnosis is a function of the professional competence of the expert.

ii. The radiologist's opinion about the 2D-images segmentation and detection of malignant zones or lobes is not consistently reliable because of fatigue.

iii. Manual segmentation and detection of 2D-images of the prostate for diagnosis result is time consuming.

3.2.3 Limitations of Existing Segmentation Algorithms

From the analysis of the existing algorithms on 2D-image segmentation surveyed in section 2.1, the following limitations are noted:

i. Most algorithms for prostate segmentation are semiautomatic and require human interaction and produce operator-dependent results (Pathaket al., 2000). Semiautomatic algorithms are sensitive to manual initialization.

ii. Some semiautomatic algorithms require manual editing during the segmentation process (Pathak et al., 2000), (Ladak et al., 2000), which implies more effort, more time, and operator-dependent results.

iii. For most automatic algorithms manual segmented images are adopted as a training data (Shen et al., 2003), which is not suitable for prostate segmentation for some reasons:

- Due to the variability of prostate gland shapes, it is hard to find a single mean shape which represents all the variability of the prostate shapes. As a result, if the mean shape is not very close to the prostate boundary of the test image, the contour is attracted to false edges due to the noise.
- The training set leads to operator-dependent results.
- Usually it is hard to create a large training data set in the medical field, especially by an expert.

iv. Using most of the data set as a training data makes the results of the algorithm biased to the training data (Betrouni et al., 2005).

These limitations in the existing algorithms reviewed are to be overcome through the enhanced algorithm presented in this project. The prostate segmentation system block diagram overview is shown in Section 3.3. The algorithm is based on region growing for the following reasons:

1. Not much work has been done to segment the prostate gland using this technique
2. The previous works done with this technique were not automated or semi-automated
3. Some previous work treated noise as information (Wahba, 2008), (Wu et al., 2015),but this work shall remove speckle noise using the sequential sticks technique before segmentation (Czerwinski et al., 1993).
4. Conventional region growing uses random scanning of neighboring pixels to grow region which consumes much time, but this work shall use sequential radial scanning (Awad, 2007).

3.3 Block Diagram Overview of System

Figure 3.1 shows the block diagram overview of the system. It's comprised of five stages. The enhanced automated region growing segmentation methodimplements experts' knowledge. It is automated, robust,relatively fast, and reliable.In order to automatically segmentthe prostate gland and extract the regions exhibiting cancer properties in the image, a multi-stage segmentation algorithm involving region growing method is presented. The first stage of the new segmentation algorithm is preparing the image using the knowledge obtained in section 3.2.1 to remove the labels from the background of original ultrasound image. In the second stage, the image is enhanced using sequential sticks algorithm. This stage deals with the speckle noise associated with TRUS images which usually affect the result of conventional region growing algorithm.

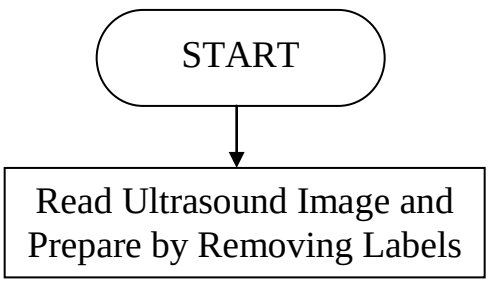
In the third stage another improvement was introduced to the conventional region growing algorithm by predetermining the dominant tissue intensity within the prostate gland through analysis of histogram of image with some a-priori knowledge. In the fourth stage, a point is automatically located, using a-priori knowledge, within the prostate close to the center from where region growing starts, usually taken as seed pixel in conventional region growing.

The final stage deals on detecting the prostate gland region and section of malignant tissue/cells within the prostate gland. Here this work introduces two major improvements:

1. Conventional region growing uses dynamic centroid pixel intensity (initially the seed point pixel intensity and later the average pixel intensity of grown region), but this work shall use static centriod pixel intensity (obtained from analysis of histogram of image using a-priori knowledge of pixel intensity, where there is a maxima at lower pixel intensity but not lowest, giving the dominant pixel intensity).

2. Starting at a seed point automatically located, the region of the prostate gland as well as the detected clusters of pixels with cancer properties (hyperechoic pixels) were grown from this point by radial/axial scanning. The enhanced region growing algorithm described in section 2.3.5 is introduced.

1. Image Preparation



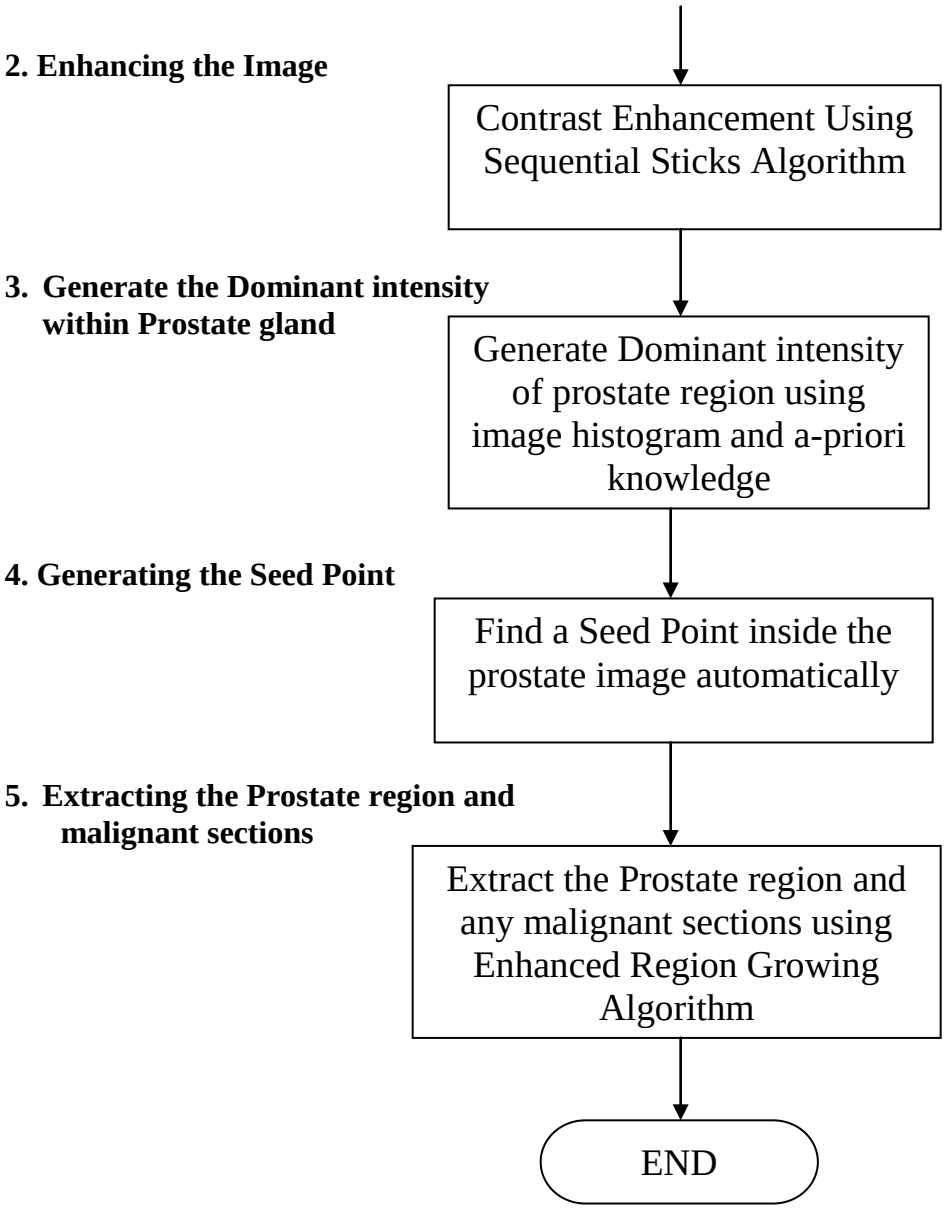


Figure 3.1 Block diagram/flowchart of system.

3.4 Method of Validating the Result of System

Different methods have been used to validate the result of segmentation algorithms. In this work the use of control experiment by subjecting the same image samples to segmentation by a human expert and then computing the accuracy and sensitivity of the segmentation algorithm was chosen. The method is refered to as area-based metrics.

3.4.1 Area Based Metrics

The validation of the system uses area-based metrics. The area of an image or a region in the image is given by the total number of pixels comprising the image or the bound region of interest in the image. The area can be refined by normalizing it with respect to pixel size with respect to image scaling parameters for the purposes of determining size of a segment or region of interest (Gonzalez and Woods, 2002).

The area-based metrics procedure is as follows (Ladak et al., 2000):

1. Determine the area inside the expert manual contour, A_m. (To be determined by number of pixels inside the prostate gland contour as marked by the expert).

2. Determine the area inside the proposed algorithm contour, A_s.(To be determined by number of pixels inside the prostate gland contour by the proposed algorithm)

3. Calculate the true positive area TP (see equation 3.1) which is defined by the common area between both the expert manual and the proposed algorithm contours, as displayed in Figure 3.2.

$$TP = A_s.(\text{EX-NOR}).\ A_m;\ \text{or}\ TP = (A_s \cap A_m) \qquad 3.1$$

4. Compute the false positive (false acceptance) area FP (see equation 3.2) which is equal to the area inside the proposed algorithm contour but outside the expert manual contour, as indicated in Figure 3.2.

$$FP = A_s - TP\ ;\ \text{or}\ \ FP = (A_s \cap A_m') \qquad 3.2$$

5. Evaluate the false negative (false reject) area FN (see equation 3.3) which is represented by the area inside the expert manual contour but outside the proposed algorithm contour, as indicated in Figure 3.2.

$$FN = A_m - TP;\ \text{or}\ \ FN = (A_s' \cap A_m) \qquad 3.3$$

6. Compute the Sensitivity, Cs(see equation 3.4) which measures the accuracy of a marking method to identify all marked pixels/regions, from the following:

Sensitivity $Cs = TP/A_m*100$ 3.4

7. Then Compute the Accuracy, Ca (see equation 3.5) which measures the ratio betweenthe pixels/regions which are correctly identified to the total number of pixels/regions, from the formula:

Accuracy $C_a = \{1- (FP+FN)/A_m\}*100$ 3.5

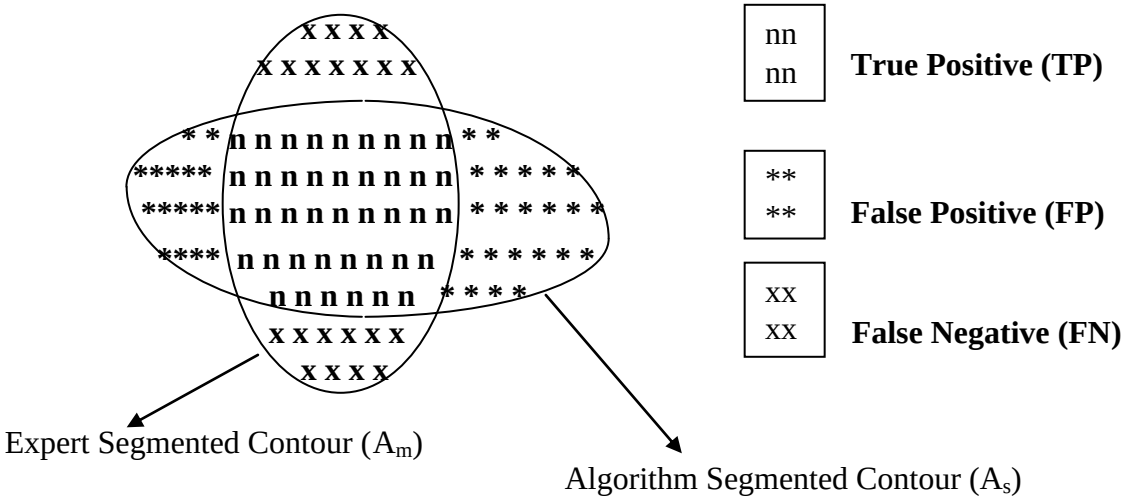


Figure 3.2 Area-based metrics

When Accuracy and Sensitivity are high it shows the system's performance and result was good.

3.5 Method of Implementing the System

The implemtation strategy adopted for the designed algorithm was MATLAB programming language/scripts. This was used to develop corresponding and commensurate codes to actualize the objective of the research.

Matlab programming language was chosen because of its robust capacity/tools for image processing by converting the image files into (M x N) matrices that corresponds in dimension to the resolution parameters of the digital image supplied as input data according to the specified scope in section 1.5. It is also amenable to modular design methodoly which was adopted. This was accomplished by the function call construct and functionality.

The internal text editing functionality was used to code all the commands arising from the various modules designed. The main user interface module which shall use a menu-driven approach was coded as the main function which was then call other modules as subfunctions into the main program codes for execution.

The data generated from analysis of the matrix of each image was stored in variables for display on the graphic user interface associated with the system main menu as output with explanations/descriptions and subsequently accumulated for storage intables for final reporting and representation in tables and graphs. Such data like the number of pixels in the segmented gland area, total number/percentage of pixels in the gland suspected to be cancerous, the number/percentage of pixels in each zone suspected to be cancerous, accuracy and sensitivity for both segmentation and cancer detection. Also the sample image were displayed side by side with segmented image as well as the image marked with sections suspected to be cancerous.

3.6 Design of System

This section presents the design of the various modules of the system. It includes specifications,description of procedures, system flowcharts/algorithms and pseudo-codes of the modules and sub-modules. The input/output (image file formats, storage files and transaction tables) specifications are presented.

The design of the system follows the structure of an expert system that implements the expert knowledge base gathered during consultation with experts. The structure of expert system described in section 2.1.5 is used to model this system modules as shown in Figure 3.4.

3.6.1 System Input Specifications

This includes the original image and the gold standard or expert segmented image which serve as the input. The format and size of the input image is here specified according to the scope in section 1.5. The digital graphics image is captured in jpg format. This shall be brought into the MATLAB program working memory using a read command as MxN array matrix. A typical image is in gray level. Maximum sample value of 255and minimum sample value of 0 is common. Copies from two sections of a screen display of .jpg image file of size 469x356 on MATLAB Command Window are shown in Figure3.3A and 3.3B.

```
255  254  244  228  232  146  140  132  127  125  126  127  127
 246  255  255  231  255  225  205  173  142  122  114  115  118
 177  164  144  139  230  198  206  200  168  123   94   93  103
  87   71   66   89  119  107   99  117  140  145  146  133  103
 126  111   94   88   94   98   84   87   91   87  101  119  115
 129  119   97   72   58   67   65   79   81   64   70   89   91
 119  133  138  132  123   51   54   73   76   56   53   63   59
 170  156  139  119  103  141  128  129  127  115  122  137  134
 177  178  174  162  149  122  106  108  112  107  115  122  109
 169  169  167  152  135  129  113  113  114  104  108  113  100
 176  179  186  177  161  176  143  117   93   74   89  118  124
 225  229  217  198  184  116  145  148  124  126  146  130   88
 255  255  241  224  217  210  182  149  130  126  134  147  158
 255  255  255  248  248  242  218  194  182  175  170  169  173
```

Figure 3.3A: Sample of the content of grey level image matrix with high gray values.

```
1   1   1   1   1   1   1   1   1   1   1   1   1
  0   0   0   0   0   0   0   0   0   0   0   0   0
  3   3   3   3   3   3   3   3   3   3   3   3   3
  0   0   0   0   0   0   0   0   0   0   0   0   0
 223 223 223 223 223 223 223 223 223 223 223 223 223
 223 223 222 220 219 217 216 215 223 223 223 223 223
  0   0   0   0   0   0   0   0   0   0   0   0   0
```

0	0	1	1	2	3	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	3	2	2	2	2	2	1	1	1	1	1
0	0	0	0	0	1	2	2	2	2	2	2	2
6	5	4	1	0	0	0	0	0	0	0	0	0

Figure 3.3B: Sample of the content of grey level image matrix with more low gray values.

In Figures.3.3A and 3.3B, each of the numbers displayed in the sub-matrices (two separate section of the matrix of type in equation 2.5) of a sample image, shows the gray value or the intensity of the image at the point in the image represented by that matrix element. A maximum value of 255 represnts points of highest intensity, white or very bright and a value of 0 represents point of lowest intensity, black or very dark.

3.6.2 System Output Image and Data FilesSpecifications

The format and size of the segmented, output, image is an MxN matrix. This was stored as digital graphics image of jpg format using MATLAB write command. A number of other output tables was generated and stored as MATLAB binary (Mat) data files. They include:

(i) Image Sample Intensity Frequency Distribution File: Freqdistfl.mat

This file with format shown in Table 3.1 is used to store the frequency distribution data for the image file being analyzed to determine the pixel intensity that has the highest occurrence or dorminant. It has two fields, pixelintsval, which stores he values of the pixel intensity whose occurrence is to be accumulated in the second field, freqintsval, frequency of the intensity value.

Table 3.1 Image sample intensity frequency distribution file format

VariableName	*Class*
Pixelintsval	double
Freqintsval	double

(ii) Area of Segmented Image Samples File: Areasegfl.mat

The file contains the area in no of pixels for the proposed algorithm segmented image stored in the field, Areainpixelsys as well as that of the expert segmented image stored in Areainpixelexp. The image file name is stored in the field, imagfilenam. The file format is shown in Table 3.2

Table 3.2 Area of segmented image sample file format

VariableName	*Class*
Imagfilenam	char
Areainpixelsys	double
Areainpixelexp	double

(iii) System Image Malignant Location/Zones Identification File: SysMalgzonfl.mat

File keeps record of all malignant (hypoechoic) pixels coordinates identified during image segmentation using axial scanning region growing algorithm. The pixel location coordinates are stored for each file, imagefilenam, in the fields, pixelcolsys and pixelrowsys.This file with format shown in Table 3.3, stores the names of the zones for each location of the prostate gland identified as malignant (hypoechoic regions) by the proposed algorithm in the field, zoneidsys. The corresponding pixel coordinates are stored in pixelcolsys and pixelrowsys. The zoneidsys updated from lookup table by processing.

Table 3.3 System Image malignant location/zones identification file format

VariableName	*Class*
Imagfilenam	char

Pixelcolsys	double
Pixelrowsys	double
Zoneidsys	char

(iv) Expert Image Malignant Location/Zones Identification File: ExpMalgzonfl.mat

This file with format shown in Table 3.4 stores the names of the zones of the prostate gland identified as malignant by the expert manual method in the field, zoneidexp. The corresponding pixel coordinates are stored in pixelcolexp and pixelrowexp. This file is generated by processing the image file marked by expert. The zoneidexp is updated from lookup table, **Gldcodzonlkuptb.mat.**

Table 3.4 Expert image malignant location/zones identification file format

VariableName	*Class*
Imagfilenam	char
Pixelcolexp	double
Pixelrowexp	double
Zoneidexp	char

(v) Gland Coordinate Zone Lookup Table File: Gldcodzonlkuptb.mat

This file whose format is shown in Table 3.5 stores the zone identity generated during processing for all coordinates within the gland region for a given image. It will serve as a lookup table for converting the coordinates of the malignant pixels to zones they fall in and the zoneidsys or zoneidexp filled in the malignant file.

Table 3.5 Gland coordinate zone lookup table file format

VariableName	*Class*
Imagfilenam	char
Pixelcolimg	double

Pixelrowimgl	double
Zoneid	char

(vi) Image Samples Edge Coordinates File: Sysedgcoodfl.mat

This file contains the coordinates of the pixels of the image at the edges or borders of the segmented image file. The format is shown in Table3.6. It is made up of three fields, imagefilenam to store the name of the image file, pixelcolbnd and pixelrowbnd to store the column and row coordinates of the pixels at the edges. The coordinates can be used to trace a curve along the edges of the image.

Table 3.6 Image samples edge coordinates file format

VariableName	*Class*
Imagfilenam	char
Pixelcolbnd	double
Pixelrowbnd	double

(vii) System Segmented Image Coordinates File: Sysegcoodfl.mat

File contains the pixel coordinates of the pixels bound by the proposed algorithm segmented prostate gland for each image file. It is made up of three fields, imagefilenam to store the name of the image file, pixelcolalg and pixelrowalg to store the column and row coordinates of the pixels at the edges. This file is generated during segmentation by the proposed algorithm. The count of these coordinates gives the area of segmented prostate gland in pixels. The format is shown in Table 3.7.

Table 3.7 System segmented image coordinates file format

VariableName	*Class*
Imagfilenam	char
Pixelcolalg	double

Pixelrowalg	double

(viii) Expert Segmented Image Coordinates File: Expsegcoodfl.mat

File contains the pixel coordinates of the pixels bound by the expert segmented prostate gland for each image file. It is made up of three fields, imagefilenam to store the name of the image file, pixelcolexp and pixelrowexp to store the column and row coordinates of the pixels at the edges. This file is produced by processing the expert segmented images. The file format is shown in Table 3.8.

Table 3.8 Expert segmented image coordinates file format

VariableName	*Class*
Imagfilenam	char
Pixelcolexp	double
Pixelrowexp	double

3.6.3 Knowledge Acquisition and Knowledge Base Design

The knowledge elicited from the expert during interviews is used to draw up the production rules for the various operational modules in the segmentation algorithm. These production rules constitute the knowledge base by applying acquired expert knowledge by the knowledge engineer as shown in Figure 3.4.

The production rules presented according to the operational modules of the system include:

i. Image Preparation Production Rule

Rule 1:

IF pixel intensity value=highest value

THEN pixel intensity value=lowest value with {cf_1}

This rule ensures that all the labels (with highest intensity) in the original image are removed by turning their intensity value to black (lowest intensity), same as the background. This will ensure they do not interfere with segmentation processing of image. The certainty factor is definitely certain, therefore **cf_1= 1.0.** It is definitely certain that the pixels with high intensity at the borders around the background region of the image are labels and so should be set to black.

ii. Seed Point Location Production Rules

Rule 2:

IF pixel intensity value=lowest value

THEN pixel intensity value=highest valuewith {cf_2}

This rule ensures that the background that is black (lowest intensity) is turned to white (highest intensity) before locating the seed point, where we have lower intensity than the surrounding tissues to be inside the prostate gland. It is performed for all image matrix elements.The certainty factor here is also definitely certain, therefore **cf_2 = 1.0.** It is definitely certain that the background region of the image is black.

Rule 3:

IF operation=seed point sub matrix selection

THEN [Rowsbmax =(ROW – RND (1/3*ROW)) +1

Rowsbmin = RND (1/3*ROW)

Colsbmax =(COL – RND (1/4*COL)) +1

Colsbmin = RND (1/4*COL)] with {cf_3}

This rule ensures that the 1st and last 3rd of the rows and 1st and last quarter of the columns of the working image matrix is considered for seed point location. This is drawn from the knowledge that the prostate gland is neither on the periphery nor at the middle of the image. This

assumption ensures that the seed point located must be within the gland near the centre. Where ROW & COL are row and column values of image matrix, Rowsb & Colsb are the row and column values of the sub matrix produced by applying the expert knowledge. The certainty factor here is almost certainly, therefore **cf_3 = 0.8.** It is almost certain that the prostate gland is within the region specified by the action to be peformed on satisfaction of the condition.

iii. Dominant Intensity Location Production Rule

Rule 4:

IF operation = dominant intensity selection

THEN IF pixel intensity value=> 0 AND pixel intensity value<= 95

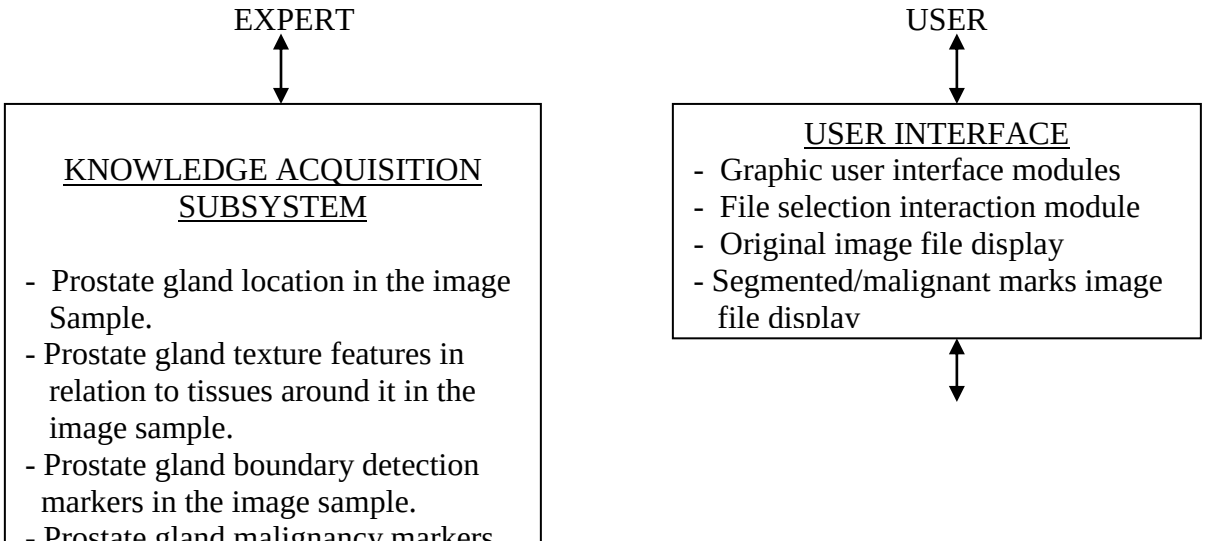
THEN add to count of Dark range image with {cf_4}

ELSE

IF pixel intensity value=> 96 AND pixel intensity value<= 255

THEN add to count of Light range image with {cf_5}

This rule ensures that the dominant parameters are chosen to suit image intensity dominant range. The knowledge that the grey level (intensity value) of the prostate gland is lower than the surrounding tissues but not as low as the black background which carries the lowest intensity value is taken into account in drawing this rule. The certainty factor here is also definitely certain, therefore **cf_4 = 1.0** and**cf_5 = 1.0.** It is definitely certain that the pixel values within the ranges specified lie in the dark and light range respectively. This was established through series of tests with 30 image samples.



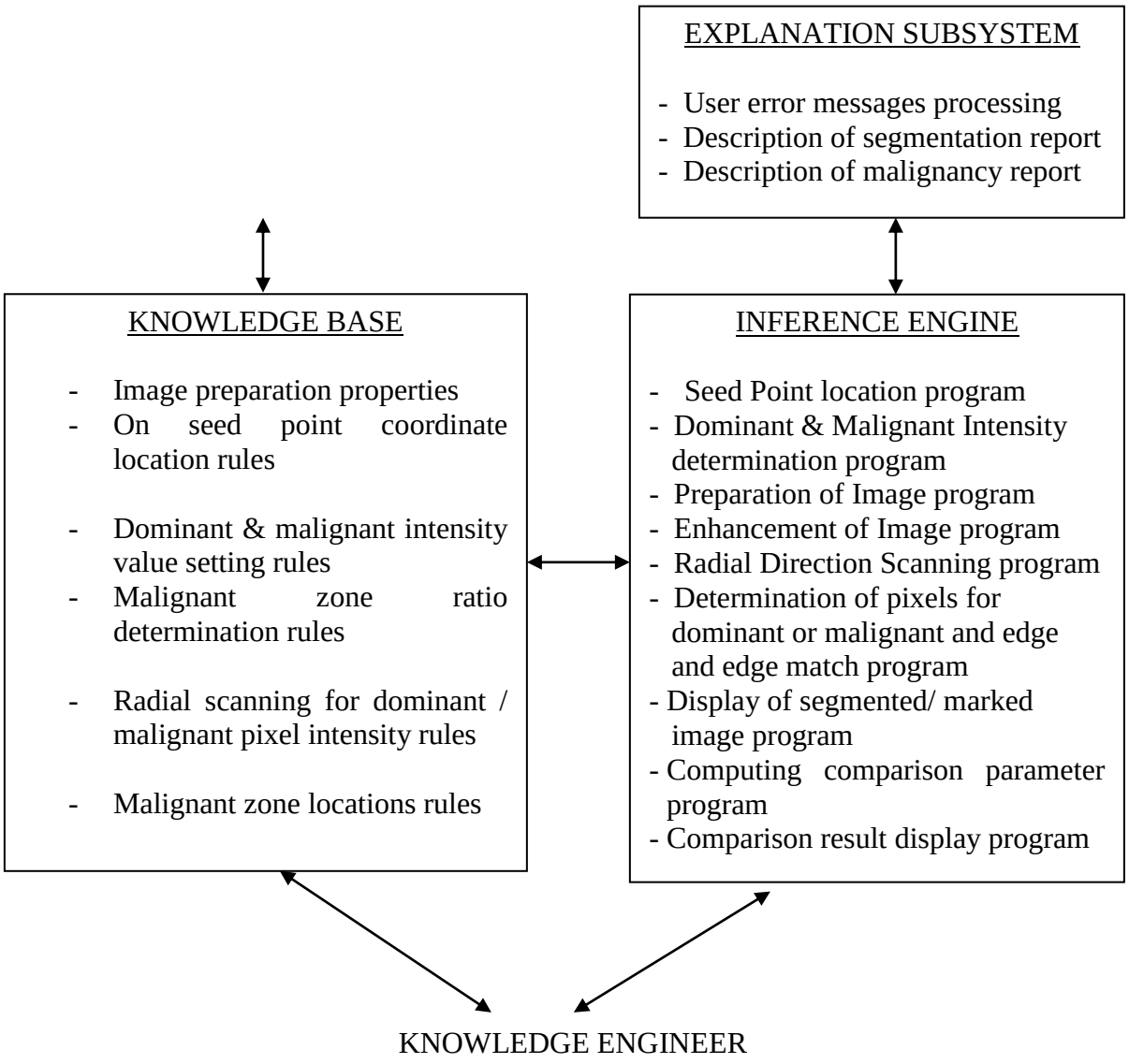


Figure 3.4: Expert system structure for the design of system.

iv. Malignant Intensity Location Production Rule

Rule 5:

IF operation = malignant intensity selection

THEN IF Light range count/Dark range count >= 2

Set dominant parameters for Light imagewith {cf_6}

ELSE THEN

Set dominant parameters for Dark imagewith {cf_7}

This rule ensures that the malignant parameters are set based on the ratio of the sum of the dominant intensity frequencies. The knowledge that malignant tissues are the hypo-echoic tissues which has lower intensity than the grey level of the gland is also implemented by this rule. The certainty factor here is also definitely certain, therefore **cf_6 = 1.0** and**cf_7 = 1.0.** It is definitely certain that the ratio of light pixel count to dark pixel count of greater or equal to two (2) indicate light image, else indicates dark image. This was also established through series of tests with 30 image samples.

v. Characteristic Function - Segmentation Rule

Rule 6:

IF f(x,y) =< RegMaxDomIntensity AND f(x,y) >= RegMinDomIntensity

THEN Add pixel to $Region_1$ with {cf_8} ; set of Prostate gland region

This rule ensures that the intensity of the pixel at x, y location in the image f lies within the estimated max. and min. values of region of the prostate gland. The certainty factor here is almost certain, therefore **cf_8= 0.9.** It is almost certain that the intensity within the region of the prostate gland is between 70 and 180 for light image, and between 30 and 150 for dark image. These values were also established through series of tests with 30 image samples.

vi. Characteristic Function - Malignancy Detection Rule

Rule 7:

IF f(x,y) =< MaligMaxDomIntensity AND f(x,y) >= MaligMinDomIntensity

THEN Add pixel to $Region_o$ with {cf_9} ; set of Malignat reg. gland

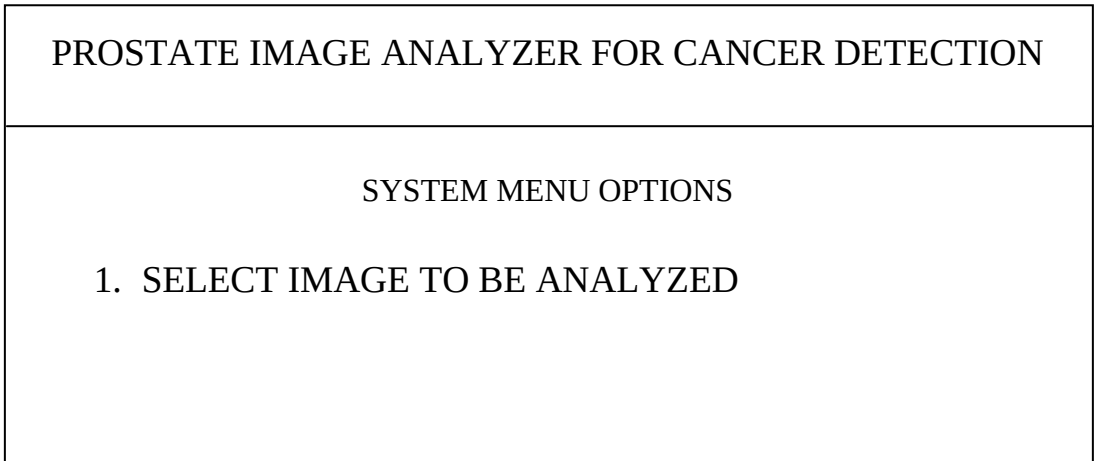
This rule ensures that the intensity of the pixel at x, y location in the image, f lies within the estimated max. and min. values of intensities within the region of the prostate gland suspected to be malignant. The knowledge that malignant tissues are the hypo-echoic tissues which has lower intensity than the grey level of the gland is also implemented by this rule. The certainty factor here is also almost certain, therefore **cf_9= 0.9.** It is almost certain that the intensity within the region of the prostate gland suspected to be malignant is between 0 and 50 for light image, and between 0 and 55 for dark image. These values were also established through series of tests with 30 image samples.

3.6.4 User Interface/Explanation Subsystem Design

This module covers the design of the graphic user interface and the associated system main menu and the subsystem that enables user to select/acquire the image file to be analyzed. The various error messages, operation prompts, and interpretation of report on analysis of images are components of the explanation subsystem and they also appear in the display of segmented image files.

3.6.4.1 System Main Menu Design

The control program for this system shall be menu-driven. The options available in the main menu are shown in Figure 3.5.The flowchart that represents the algorithm for implementing the automation of the system menu is shown in Figure 3.6.



2. ANALYZE IMAGE AND DISPLAY RESULTS

3. COMPARE EXPERT AND SYSTEM RESULTS

4. QUIT

Select Operation?

Figure 3.5: System main menu options.

The corresponding pseudo-code for the above algorithm is as follows:

1. Display the menu layout and menu options and heading prostate image analyzer for cancer detection

   Display message to prompt user to select option and receive input

   IF choice is invalid,

   Display error message and return to step 1

   Else

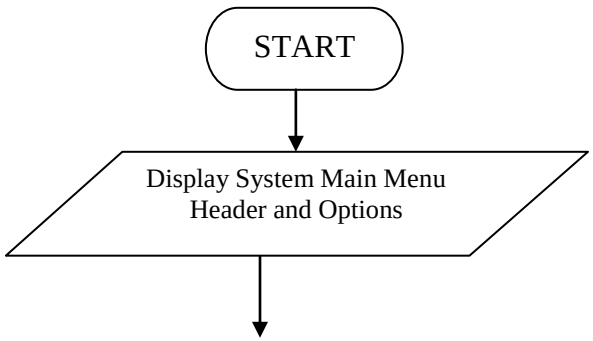
   IF choice is option 1

   Then call routine to Select image to be analyzed and return to step 1

   IF choice is option 2

   Then call routine to Analyze image with proposed segmentation algorithm and

   display resulting image and return to step 1



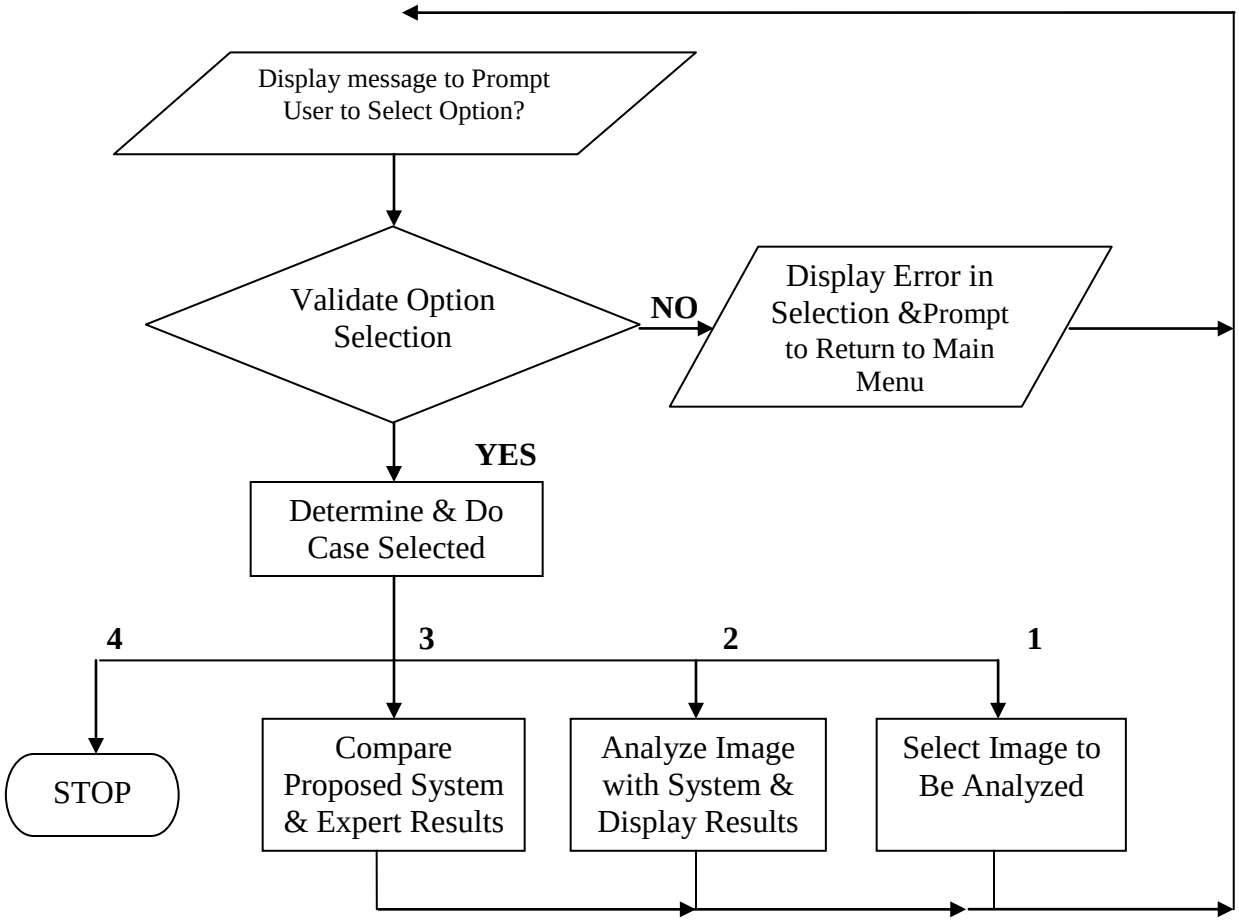


Figure 3.6:System menu flowcharts.

IF choice is option 3

Then call routine to Compare the segmentation and detection result of the proposed algorithm with expert's result and display results return to step 1

IF choice is option 4

Then Close menu and end.

Stop.

3.6.4.2 Image Acquisition Design

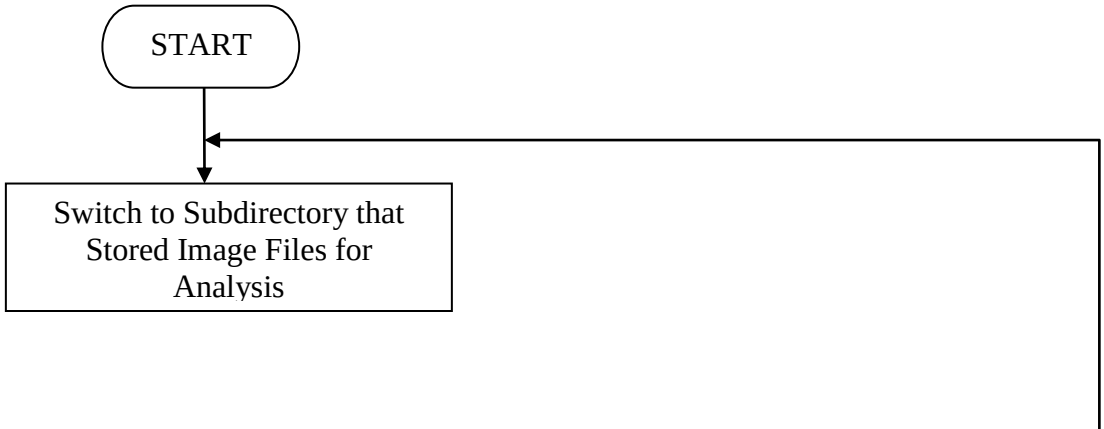
The2D TRUS images to be analysed were acquired in graphic digital form from an online source in JPEG format. These image files were copied into the MATLAB working subdirectory for this work.

3.6.4.3 Image Selection Design

This section automates the selection of images from a list of images acquired and stored in the MATLAB working subdirectory, which was named PROSTAIMAGES. This algorithm allows the user to select any one of the files displayed and stores the selected file for use in image analysis after validating its type. The design of the flowchart is shown in Figure 3.7.

The corresponding pseudo-code is as follows:

1. Change to the subdirectory that stores images files captured for analysis
   IF directory is empty or does not contain right image files
      Then Display error-empty or incorrect image files
2. Prompt user to copy right image files into named subdirectory and okay
      IF okay is pressed
         Then go to step 1.
      Else go to step 2
   Else Automate a Display of the image files captured for analysis
3. Prompt user to select an image file from list displayed
      IF the file selected is not a right image file format
         Then Display error – Wrong image and/or file format selected
            Go to step 3
      Else Store file selected as Target image file
   Stop.



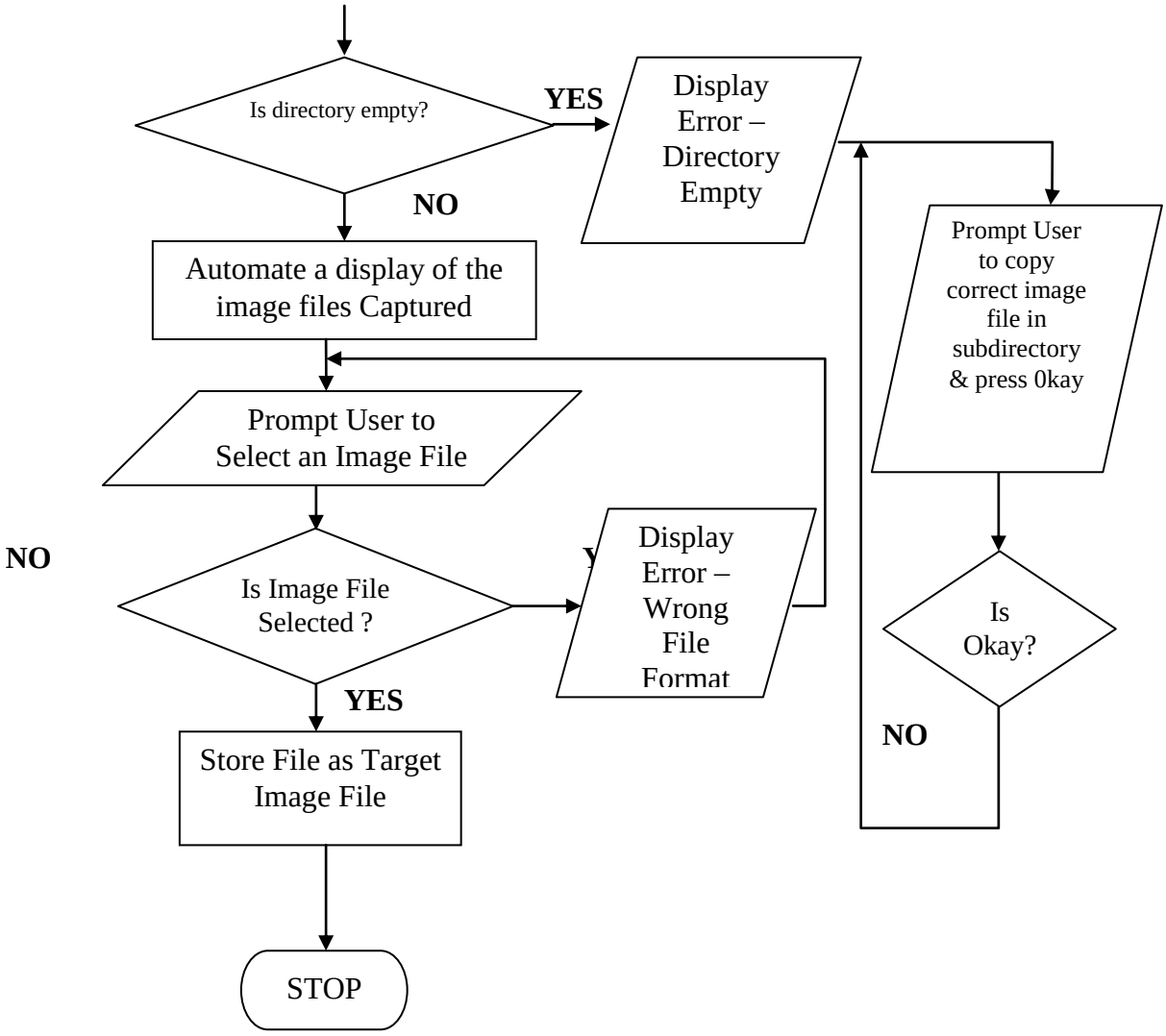


Figure 3.7:Image selection option flowchart.

3.6.5 Inference Engine Design

This subsystem deign covers the sub-modules involved in the realisation of research objectives i., ii., iii., and v. These sub-modules are listed in the inference engine in Figure 3.4. The production rules

1-5 are applied in the sub-modules to implement the knowledge of the experts in analyzing images to detect prostate cancer as requested by user input.

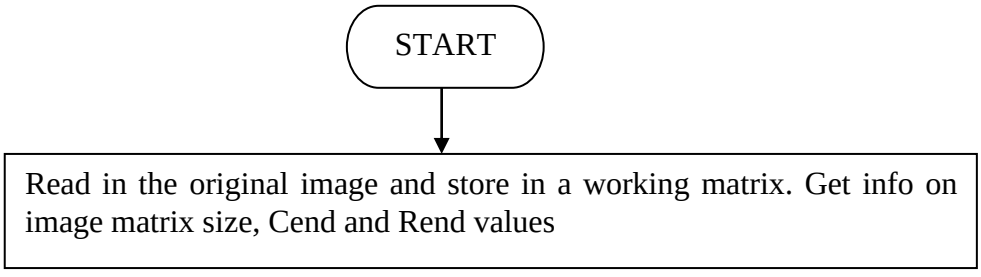
This section is also showing the sequence of operations to be performed when option 2 is selected from the system main menu shown in Figure 3.5.

3.6.5.1 Image Preparation Module Design

This module ensures that every image to be analysed was pre-processed to remove the usual Ultrasound image labelling that enable manual analysis to be performed by expert radiologist. Figure 3.8 shows the procedure for removing the undesirable labels that original ultrasound images carry. They are identified by their being the highest intensity in the borders of the image where the background is of lowest intensity. These labels are also located along the borders of the image matrix. The algorithm starts at the first cell of the original image matrix to search for these pixels with unwanted intensity and replaces them with acceptable background intensity. The process is performed for 20 pixels from each of the four borders of the image matrix.

The pseudo-code for the image preparation operation is as follows:

Read the Target image file selected and store in a working matrix
Get info on the image matrix size, Cend and Rend
Search image file for lowest intensity pixel value and color map and store for ref.
Check the top border of image in the ranges – (R1 to R20) and (C1 to Cend)
    Call routine to check for high intensity pixel & replace with the
        Stored lowest intensity ref. value if found.



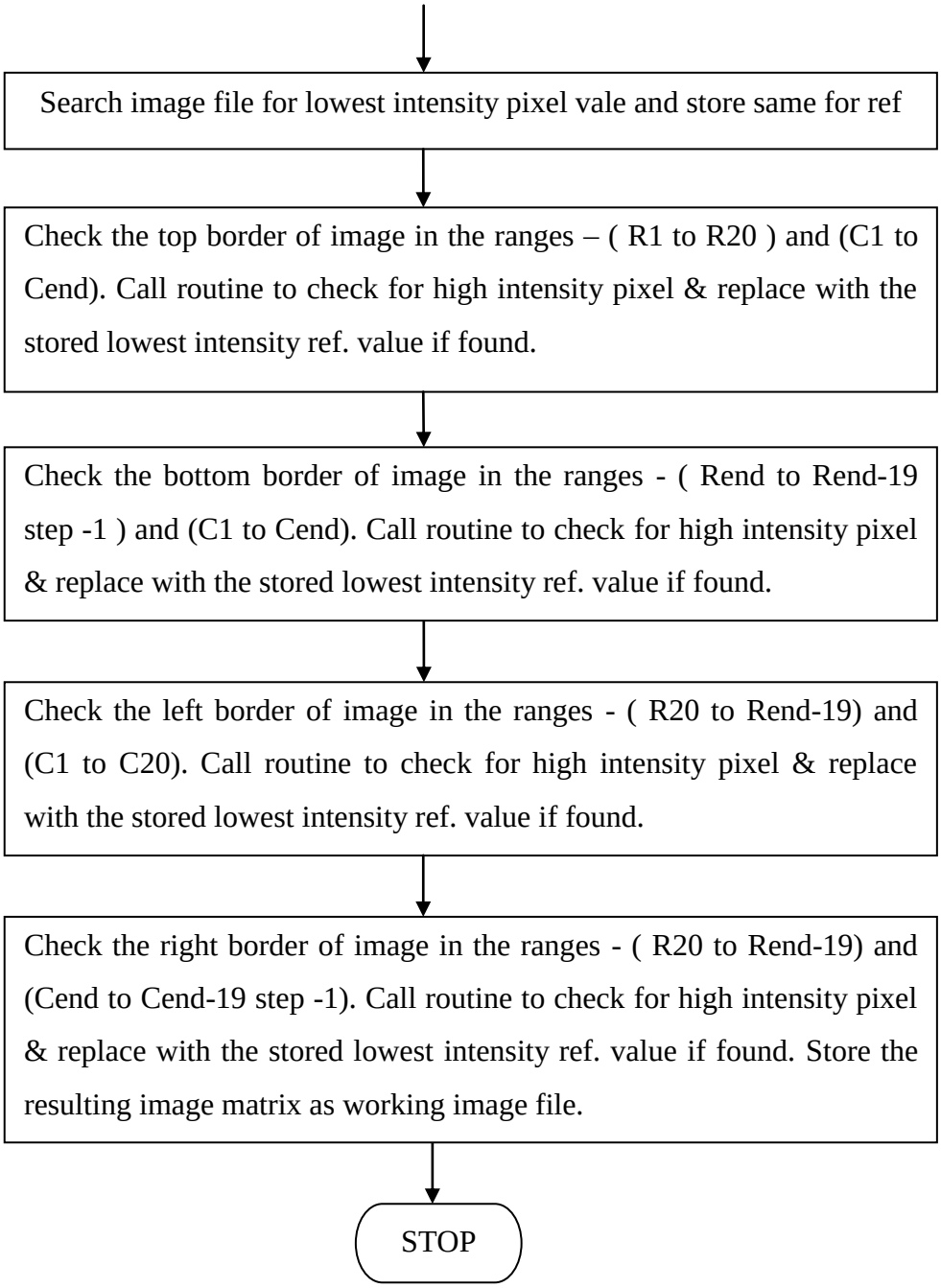


Figure 3.8: Image preparation operation flowchart.

Check the bottom border of image in the ranges - (Rend to Rend-19 step -1) and (C1 to Cend)
Call routine to check for high intensity pixel & replace with the
Stored lowest intensity ref. value if found.
Check the left border of image in the ranges - (R20 to Rend-19) and (C1 to C20)

Call routine to check for high intensity pixel & replace with the
Stored lowest intensity ref. value if found.
Check the right border of image in the ranges - (R20 to Rend-19) and (Cend to Cend-19 step -1)
Call routine to check for high intensity pixel & replace with the
Stored lowest intensity ref. value if found.
Store the working image as working image file
Stop.

3.6.5.2 Image Enhancement Module Design

The image selected for analysis at this stage is filtered to reduce noise and achieve significantly sharp edged images using sequential stick algorithm described in section 2.2.1.1(Awad, 2007). The sticks technique is adopted. The algorithm for a fixed stick technique is as follows:

1. Read image matrix, I
   Generate a set of filter bank as templates based on specified stick length
   While image pixels are not ended do the following
   Apply the filter bank, finding the largest output at each pixel by convolution
   For the NxN neighbors of each pixel in the image,
   Get 2n-2 short lines passing through the central pixel with N pixels in length
   Compute the sum of the pixel values along each segment
   Determine the maximum sum of the segments & put same in the centre pixel of theNxN sub-matrix in the image
   Next
   Endwhile
   Readjust the intensity level of the image to the original level
2. Output enhanced image matrix, J.

The above algorithm can be made a sequential routine by repeating steps 1 to 2 with varying stick length and using the output image as the next input image. Usually odd values of stick length have been found to yield better results. Iterations in sequence of 3, 5, 7, 9, 11, 13, and 15 are acceptable(Awad, 2007). An iteration of six sequences was used starting with stick length of 5 to 15.

The modified algorithm becomes:

Read image matrix, I

For stick-length =5 to 15 ; set up a sequence

1. Generate a set of filter bank as templates based on specified stick length

   While image pixels are not ended do the following

   Apply the filter bank, finding the largest output at each pixel by convolution

   For the NxN neighbors of each pixel in the image,

   Get 2n-2 short lines passing through the central pixel with N pixels in length

   Compute the sum of the pixel values along each segment

   Determine the maximum sum of the segments & put same in the centre pixel of the NxN sub-matrix in the image

   Next

   Endwhile

   Readjust the intensity level of the image to the original level
2. Output enhanced image matrix, J.

   Assign output as input I=J

   Next stick-length
3. Stop

3.6.5.3 Dominant Intensity Parameters Determination Design

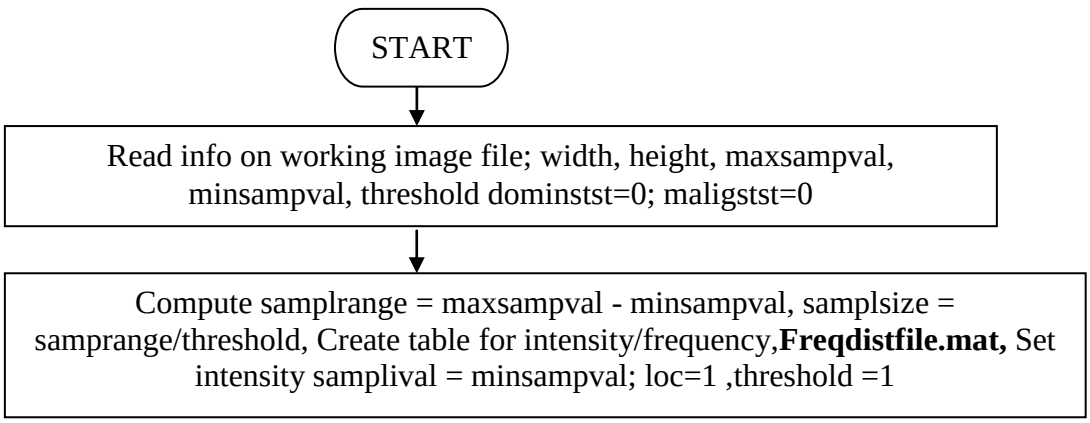
The knowledge-based rules are used here. The prostate image shows slightly lower intensity (darker pixels) than the surrounding tissue and with appreciable high occurrence. This pixel intensity is dominant within the prostate gland and can be determined using intensity frequency analysis of the image after changing the background (darkest pixels as designed by ultrasound manufacturers) to white to ensure darkest region with highest frequency is within the prostate gland. Malignant tissues within the prostate usually have even darker pixels, hyperechoic (very high echo density). Determination of intensity ranges that fall within the hyperechoic range (low intensity or dark grey rang) was done by conducting experiment with several images. Intensity from 0 – 95 was classified

as dark range while intensities from 96 to 255 as light range. This was used to sort the image intensity frequency distribution table into light and dark intensity group. The ratio of light to dark intensity was used to set the dominant intensity parameters for prostate region cells (malignant and normal) for the image. With several image samples tested the dark range and light rang images were found to give best result with minimum intensity threshold within the prostate gland of 30 and 70 respectively and maximum intensity threshold within the prostate gland of 150 and 180 respectively.Hyperechoic pixels used for detection of suspected cancerous cells have maximum values of 50 and 55 for light and dark images respectively.

The algorithms for determining the dominant and malignant tissue intensity to be used in segmenting the prostate gland and identifying malignant sections is described in the flowcharts shown in Figures 3.9A and 3.9B. Figure 3.9A shows the section of the algorithm that produces a frequency distribution of the image intensity and stores these intensities and corresponding intensity in a table for use in the next sub-module in Figure 3.9B.

The pseudo-code for this section of the algorithm is as follow:

1. Read Target image file into a working matrix
   Read following info of the image file: width, height, maximum and minimum intensity sample value, threshold = 1
   Determine; samplrange=maxsampval – minsampval, samplsize=samplrange/threshold
   Create table Freqdistfl.mat for pixel intensity and correpondg frequencies
   Set sampl intensity value, samplival=minsampval, table location counter, loc=1



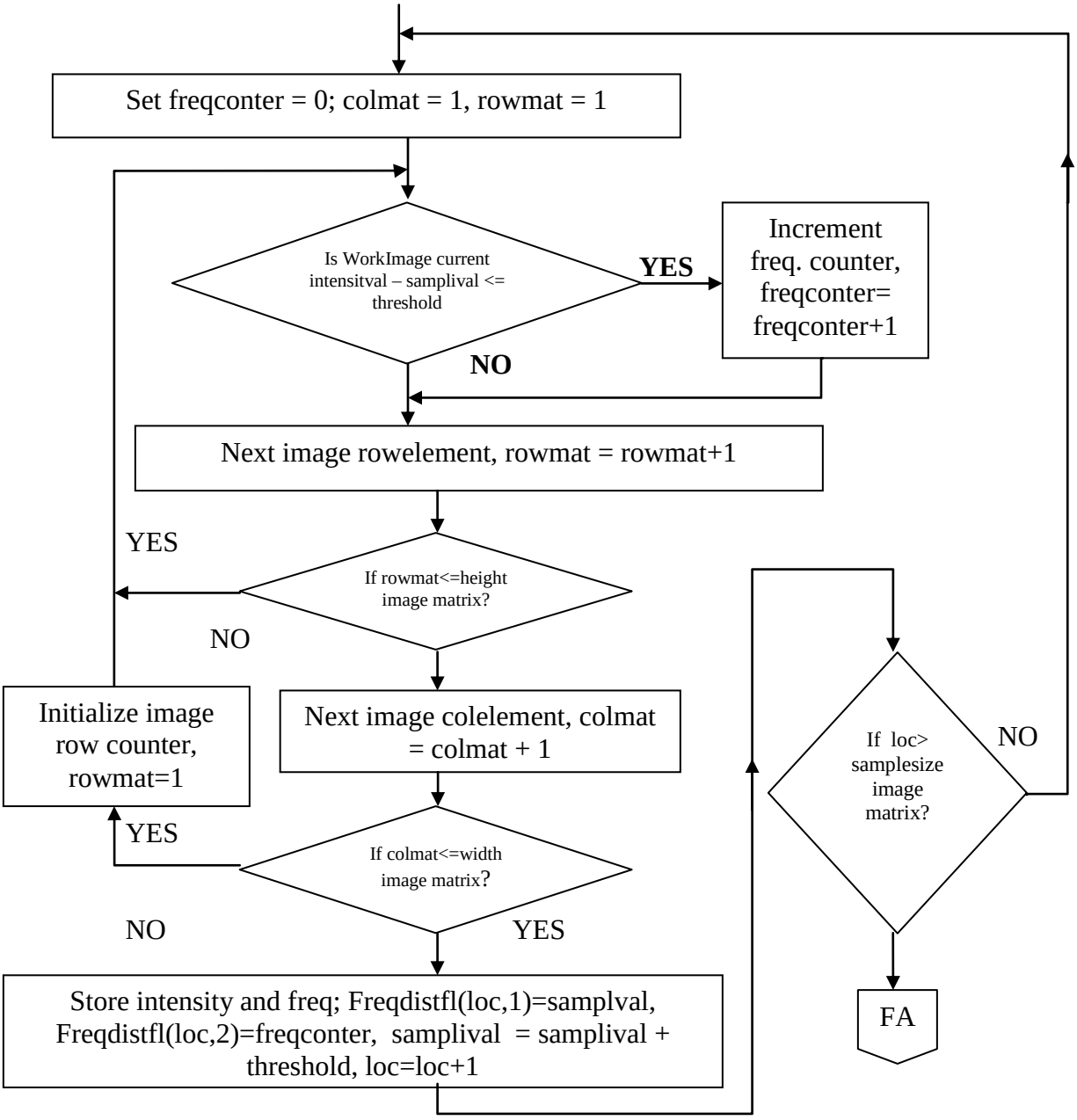
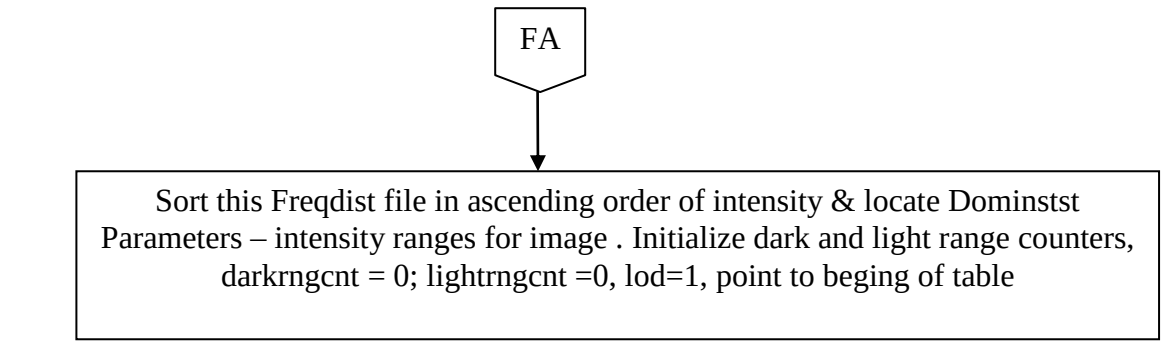


Figure 3.9A Flowchart for Initializing parameters and generating frequency distribution table of the image intensity.



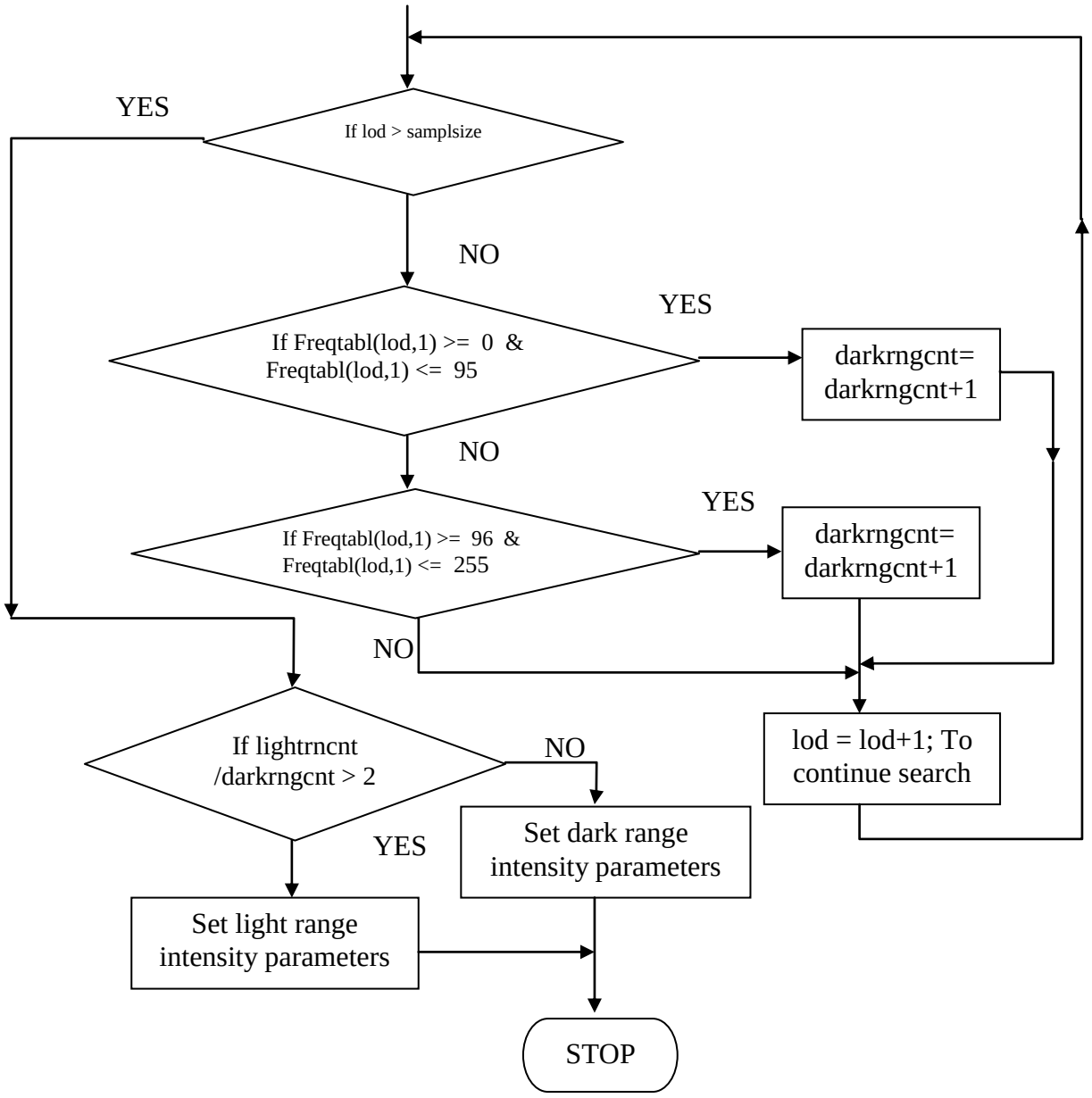


Figure 3.9B Flowchart for determining and setting dominantintensity parameters of image.

2. While not end of sample size

   Set freqcounter=0, columnmatrx=1, rowmatrx=1

For all columns of image matrix, columin to columax
For all rows of image matrix , rowmin to rowmax
IF imageMatrx(rowmatrx, colmatrx) – samplival = 0
Then increment, (freqcounter = freqcounter+1) to count this pixel intensity
Endif
Next row
Next column
Store intensity and its frequency computed in table
Freqdistfl (loc,1) = sampival, Freqdistfl (loc, 2) = freqcounter

Get next intensity value and next table location for entry,
samplival = samplival + threshold, loc = loc + 1
Endwhile, Then go to 2 if not end of samplsize
End intensity frequency distribution generation and Branch to sub-module that sorts and searches for Dominant and malignant intensities in the freq. distribution table.

The flowchart in Figure 3.9B is the section of the entire algorithm for determining and setting dominant intensity parameters of image. Thesubroutine implements Rules 4 & 5 to check the conditions set for dominant parameters setting.

The pseudo-code for this section of the algorithm is as follow:

1. Sort the frequency distribution file in descending order of frequency
   Declare global variables, and initialize darkrngcnt and lightrngcnt to zero
   Set frequency table counter, lod=1
2. IF lod > samplsize

   Then Go to 3

   IF Freqdistfl(lod,1) >= 0 &Freqdistfl(lod,1) <= 95 , …….. RULE 4
   Then Increment darkrngcnter
   IF Freqdistfl(lod,1)>= 96 &Freqdistfl(lod,1)<=255 , …….. RULE 4
   Then Increment darkrngcnter
   Endif

Endif

Lod = lod + 1, to go on searching
Go to 2

3. IF(lightrngcnt / darkrngcnt > 2) , …….RULE 5
   Set light image dominant parameters
   Else
   Set light image dominant parameters
   Endif
   Stop.

3.6.5.4 Seed Point Localization Design

This section describes the location of a seed point within the prostate gland in the image from where to commence prostate gland segmentation. It implements knowledge-based rules, Rule 3. It was noted that the prostate gland is not in the periphery of the image, thus it was assumed from observation that the gland is within the first and the last third of the rows, and the first and the last quarter of the columns of the image. Consequently, pixels outside this section can be excluded in the determination of the seed point. It is possible that the excluded sections of the image may include part of the prostate gland, but this operation guarantees stability of seed point determination. It concentrates search on the portions of the image containing the prostate gland and avoiding the shadowing.

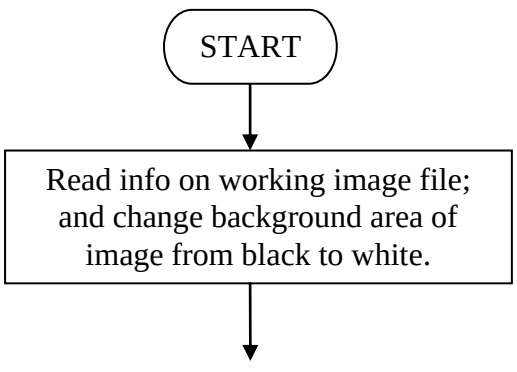
The algorithm shown in Figure 3.10 sorts the selected (first and the last third of the rows, and the first and the last quarter of the columns of the image) portion of the image in ascending order of pixel intensity. The number of low intensity pixels (that represents intensities within the prostate) that will be used in determining the seed point is derived from a ratio, P (Awad, 2007), where P is the ratio between the number of low intensity pixels that is considered for seed point localization, and the total number of pixels in the portion of the image selected for seed point localization.

This algorithm uses the same ratio, P of pixels to calculate the median coordinates of these pixels and assign this median to the seed point. It is found (Awad, 2007) that the value of P should be in the range of (0.2 to 0.8). I adopted P = 0.5 for determining the number of low intensity pixels to be

considered in determining the median low intensity pixel whose coordinates is taken as the seed point (row and column in the image matrix). The algorithm returns the row and column value as the seed point from where region growing of prostate gland can commence.

The Pseudo-code for seed point location is as follows:

```
Read Target image file info into working matrix
Change background of image from black to white to ensure that lowest intensities are
    not the background
While image matrix elements are not finished continue processing
        IF pixel intensity of element = lowest intensity, black
                Then replace intensity of element with highest intensity, white
        Else
                Skip the element and get next
        Endif
Wend
Mask out image elements portions not to be considered for seed point using RULE 3to
   propose a boundary. The result is stored in a table Seedpointbl(loc,3) to store intensity, row, col
        Rowmax = (ROW- RND(1/3*ROW))  + 1,  Rowmax is submatrix maximum
                                        row selected from ROW of image matrix of ROW
        Rowmin = RND(1/3*ROW), Rowmin is submatrix minimum row selected from
                                        ROW of image matrix of ROW
        Colmax= (COL-RND(1/4*COL))  + 1, Colmax is submatrix maximum
                                        Col selected from COL of image matrix of COL
        Colmin =RND(1/4*COL), Colmin is submatrix maximum   col selected from COL of image
                                         matrix of COL
```



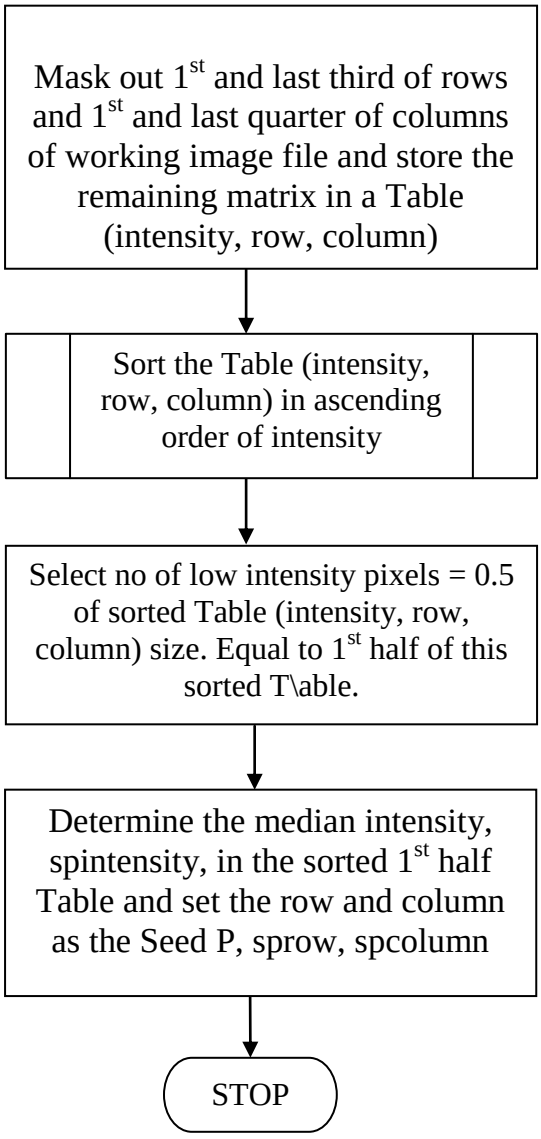


Figure 3.10: Flowchart for locating the seed point.

Set Seedptblsz = Rowmax * Colmax

Initialize Seedpointbl (Seedptblsz, 3) = 0, loc =1

Read content image matrix in the range selected into the table

FOR R = Rowmin to Rowmax

FOR C = Colmin to Colmax

```
                Seedpointbl(loc,1) = Imagematr(R,C)
                Seedpointbl(loc,2) = R, Seedpointbl(loc,3) = C
                Loc = loc+ 1
        Next C
Next R
Sort Seedpointbl (loc,3) in ascending order of the intensity so that lower intensities
        appear in loc 1 of table
Select no of low intensity pixels to participate in seed point localization, given by expression;
(No of pixels = 0.5* total pixels participating in seed point selection)
Nolowintpxl = (0.5 * Seedptblsz)

Determine the median intensity in the sorted table of size Nolowintpxl form beginning of
Seedpointbl
IF Nolowintpxl/2 = RND (Nolowintpxl/2)
        Then medintloc = Nolowintpxl/2
Else
         medintloc = RND (Nolowintpxl/2) + 1
Endif
Read out the seed point intensity and coordinates using the medintloc
        Sedintlocstr=Seedpointbl(medintloc)
        Seedptint = Seedpointbl(medintloc,1) ; contains intensity
        Seedptrow = Seedpointbl(medintloc,2) ;  contains row
        Seedptcol = Seedpointbl(medintloc,3) ; contains column
Stop
```

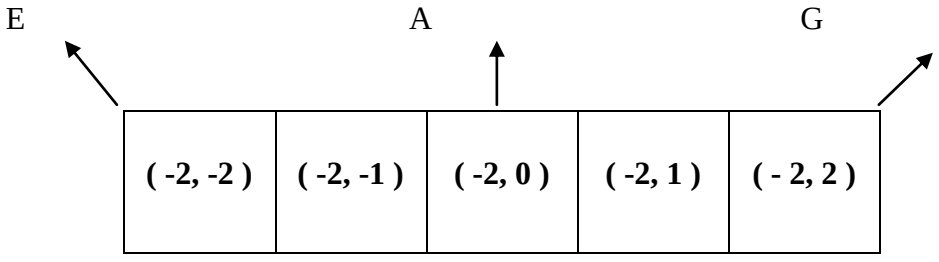
3.6.5.5 Segmentation by Enhanced Region Growing For Boundary and Malignant Tissues Detection Design

The algorithm designed here realises the first two research objectives. It takes as input an enhanced image whose dominant and malignant intensities as well as seed point and produces as output a

segmented image, a file containing coordinates of the edges of the segmented prostate gland, a file containing coordinates of points within the segmented portion suspected to be malignant tissue, and a file containing the name of image file segmented and the number of pixels in the segmented portion, representing the area of segmented prostate in pixels.

The algorithm executes for all pixels in the image, starting at the seed point automatically located, the region of the prostate gland is grown from this point by radial/axial scanning using the directions shown in Figure 3.11.

The flowcharts describing and presenting the details of the algorithm for the segmentation and detection of malignant pixels proposed are shown in modules (see Figures 3.12A, 3.12B, 3.12C, 3.12D, and 3.12E. Figure .12A shows the section of the algorithm that initializes tables and files to accumulate segmentation parameters, and start growing prostate gland region by radial scanning in CEA – radial direction from seed point coordinate in image matrix (see Figure 3.11). The parameters for the direction are set for row-wise and column-wise scanning. Subroutines for row-wise and column-wise scanning are respectively called along with a routine for removing false edges, trimming and fitting edges. Routines to accumulate detected edges and detect malignant pixels within the prostate boundary detected are called.



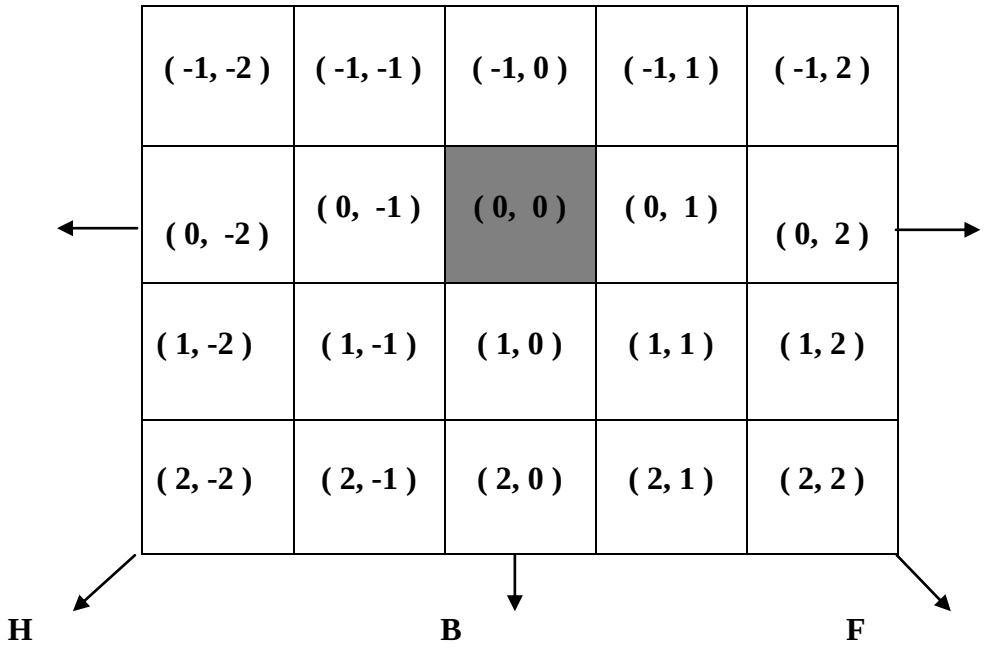


Figure 3.11:Possible scanning radial/axial directions with 24 – neighboursof target pixel atcentre (0, 0).

This figure shows an extract (sub-matrix)from the matrx location of a typical image matrix. The arrows pointing outward from the edges of the sub-matrix indicates the possible directions of scanning in radial/axial directions. Any given pixel location being considered for addition into the region can be considered to be in the location shaded, (0, 0). The directions shown will produce 8, 16, 24 – neighbours mask to pixel located at point (0, 0). To avoid redundancy in the scanning, four axes CEA, AGD, DFB, and BHC were chosen for movement to detect edges of the prostate gland (see proposed algorithm in Figure 2.15), starting at point (0, 0), which is called the seed point.

The Pseudo-code for segmentation for detection of prostate gland boundary and malignant tissues – initializations and CEA – Radial direction scanning is as follows:

1. Read info on enhanced image file into matrix, Imagmatrx(M,N)
   Initialize Segmatrx (M,N) to same size;
   M= Size Imagmatrx(1)
   N = Size Imagmatrx(2); Imagzs = M * N
   Initialize the following tables; , SegImgcood (imagsz,2) = 0; Segimgedg(imagsz,2) = 0
   Maligpxcood (Imagsz, 2) = 0; Initialize counters; Segcont = 0; locm = 0; loce = 0
   Assign seed point coordinates, Rs = Seedptrow, Cs = Seedptcol

   Start Radial/Axial scanning to grow prostate gland region from the seed point, CEA – axial/radial direction about the seed point Rs, Cs.

   Set parameter limits for image matrix row, column and tables for Row-wise scanning in this direction.

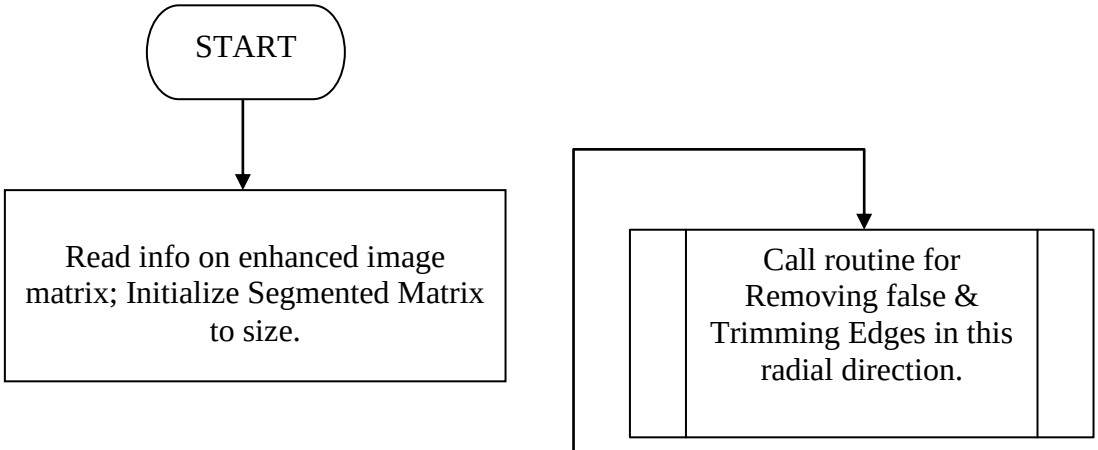
   Call routine for Row-wise scanning & analysis of pixels for edge detection in this radial direction.

   Call routine for Removing false & Trimming Edges in this radial direction.

   Set parameter limits for image matrix row, column and tables for Column-wise scanning in this direction.

   Call routine for Column-wise scanning & analysis of pixels for edge detection in this radial direction.

   Call routine for Removing false, Trimming & Fitting Edges in this radial direction.



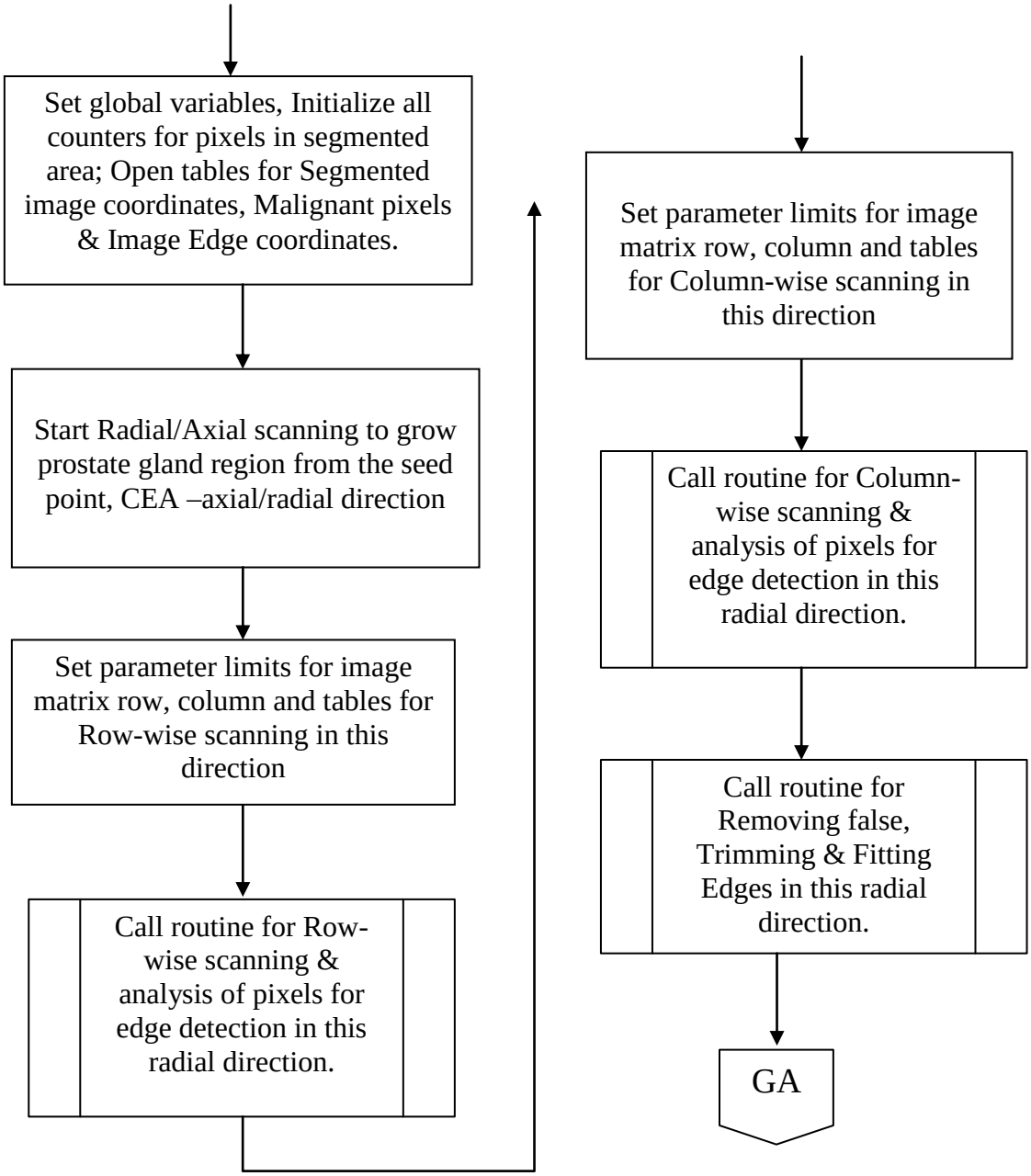
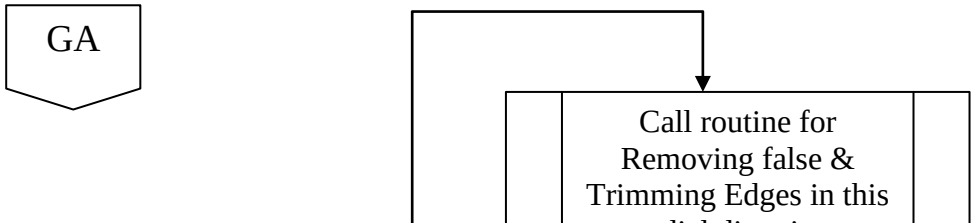


Figure 3.12A: Flowchart for segmentation files initialization and scanning image matrix in CEA – axial/radial direction.

The parameters for scanning in AGD – radial direction are set and control transferred to that module shown in Figure 3.12B. Figure 3.12B shows the section of the algorithm that continues growing prostate gland region by radial scanning in AGD - radial directions from seed point coordinates (see Figure 3.11) in the image matrix.



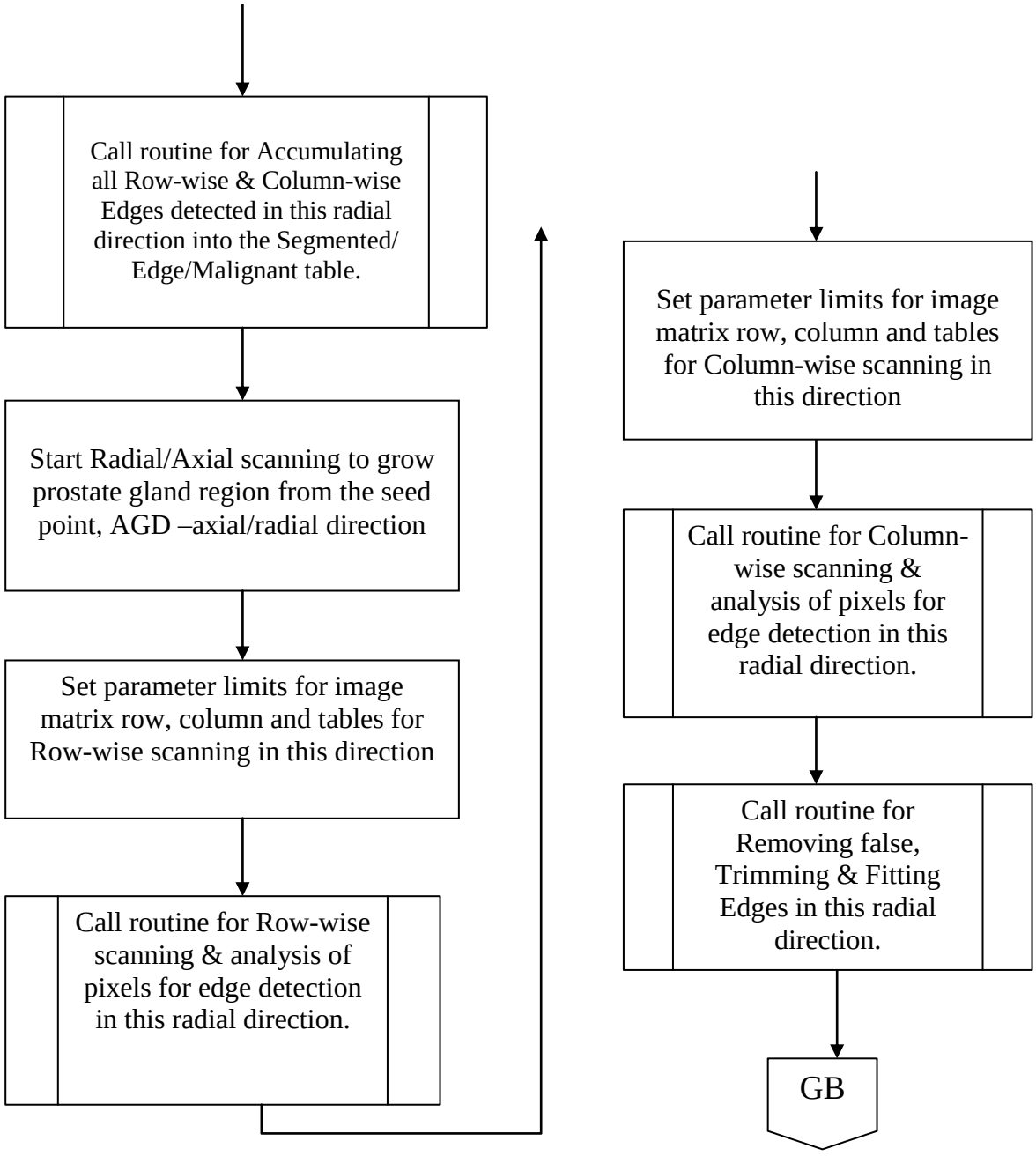


Figure 3.12B: Flowchart for scanning image matrix in AGD – radial direction.

The parameters for the direction are set for row-wise and column-wise scanning. Subroutines for row-wise and column-wise scanning are respectively called along with a routine for removing false edges, trimming and fitting edges. A routine to accumulate detected edges and detect malignant pixels within the prostate boundary detected are called. The Pseudo-code for scanning prostate gland image for boundary and malignant tissues – detection in AGD – Radial direction is as follows:

2. Call routine for Accumulating all Row-wise & Column-wise Edges detected in this radial direction into the Segmented/ Edge/Malignant table.

   Start Radial/Axial scanning to grow prostate gland region from the seed point, AGD – axial/radial direction about the seed point Rs, Cs.

   Set parameter limits for image matrix row, column and tables for Row-wise scanning in this direction.

   Call routine for Row-wise scanning & analysis of pixels for edge detection in this radial direction.

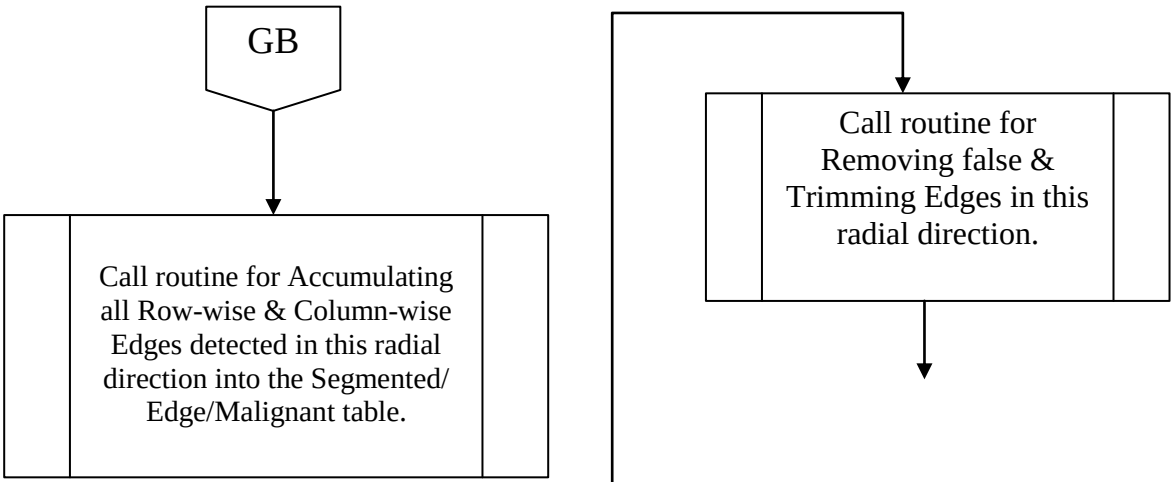
   Call routine for Removing false & Trimming Edges in this radial direction.

   Set parameter limits for image matrix row, column and tables for Column-wise scanning in this direction.

   Call routine for Column-wise scanning & analysis of pixels for edge detection in this radial direction.

   Call routine for Removing false, Trimming & Fitting Edges in this radial direction.

The parameters for scanning in DFB – radial direction are set and control transferred to that module shown in Figure 3.12C. Figure 3.12C shows the section of the algorithm that continues growing prostate gland region by radial scanning in DFB - radial directions from seed point coordinates in the image matrix. The parameters for the direction are set for row-wise and column-wise scanning. Subroutines for row-wise and column-wise scanning are respectively called along with a routine for removing false edges, trimming and fitting edges. A routine to accumulate detected edges and detect malignant pixels within the prostate boundary detected are called.



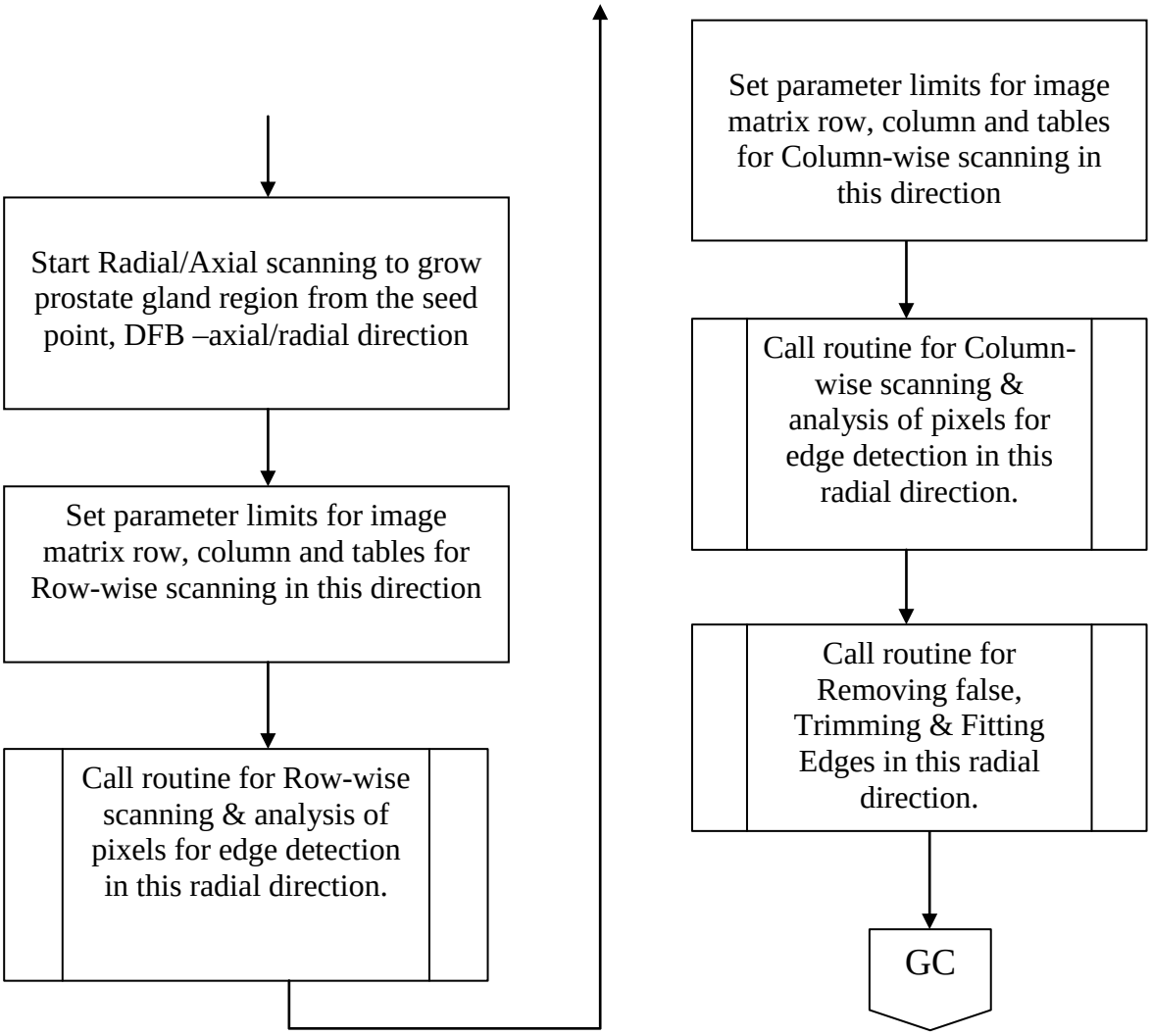


Figure 3.12C: Flowchart for scanning image matrix in DFB – radial direction.

The Pseudo-code for scanning prostate gland image for boundary and malignant tissues – detection in DFB – Radial direction is as follows:

3. Call routine for Accumulating all Row-wise & Column-wise Edges detected in this radial direction into the Segmented/ Edge/Malignant table.

   Start Radial/Axial scanning to grow prostate gland region from the seed point, DFB – axial/radial direction about the seed point Rs, Cs.

Set parameter limits for image matrix row, column and tables for Row-wise scanning in this direction.

Call routine for Row-wise scanning & analysis of pixels for edge detection in this radial direction.

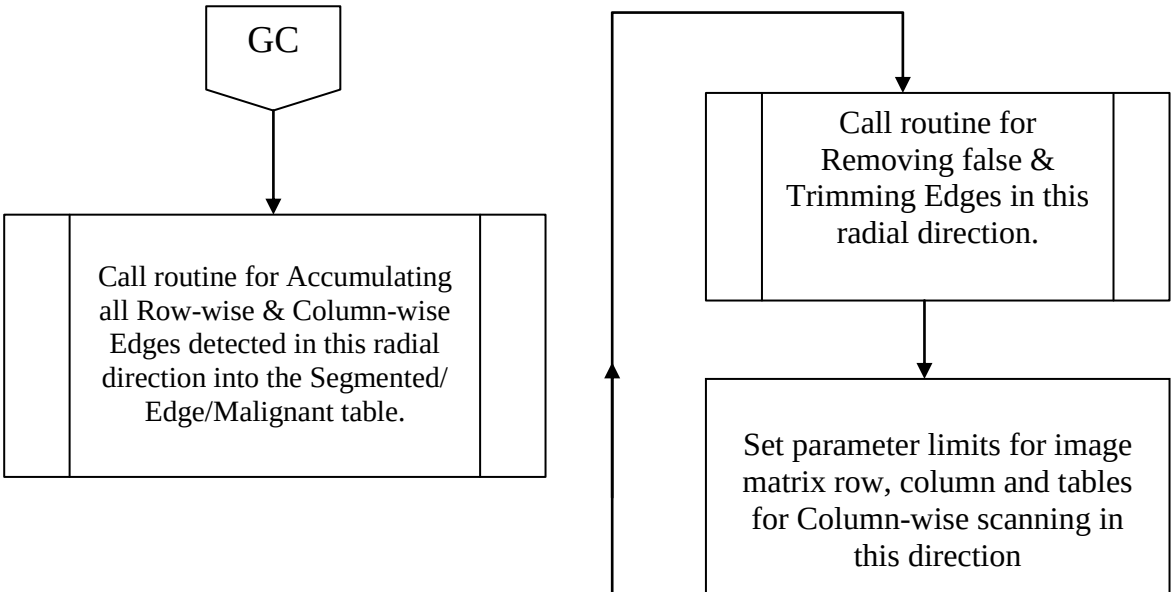
Call routine for Removing false & Trimming Edges in this radial direction.

Set parameter limits for image matrix row, column and tables for Column-wise scanning in this direction.

Call routine for Column-wise scanning & analysis of pixels for edge detection in this radial direction.

Call routine for Removing false, Trimming & Fitting Edges in this radial direction.

The parameters for scanning in BHC – radial direction are set and control transferred to that module shown in Figure 3.12D. Figure 3.12D shows the section of the algorithm that continues growing prostate gland region by radial scanning in BHC - radial directions from seed point coordinates (see Figure 3.11) in the image matrix. The parameters for the direction are set for row-wise and column-wise scanning. Subroutines for row-wise and column-wise scanning are respectively called along with a routine for removing false edges, trimming and fitting edges. A routine to accumulate detected edges and detect malignant pixels within the prostate boundary detected are called.



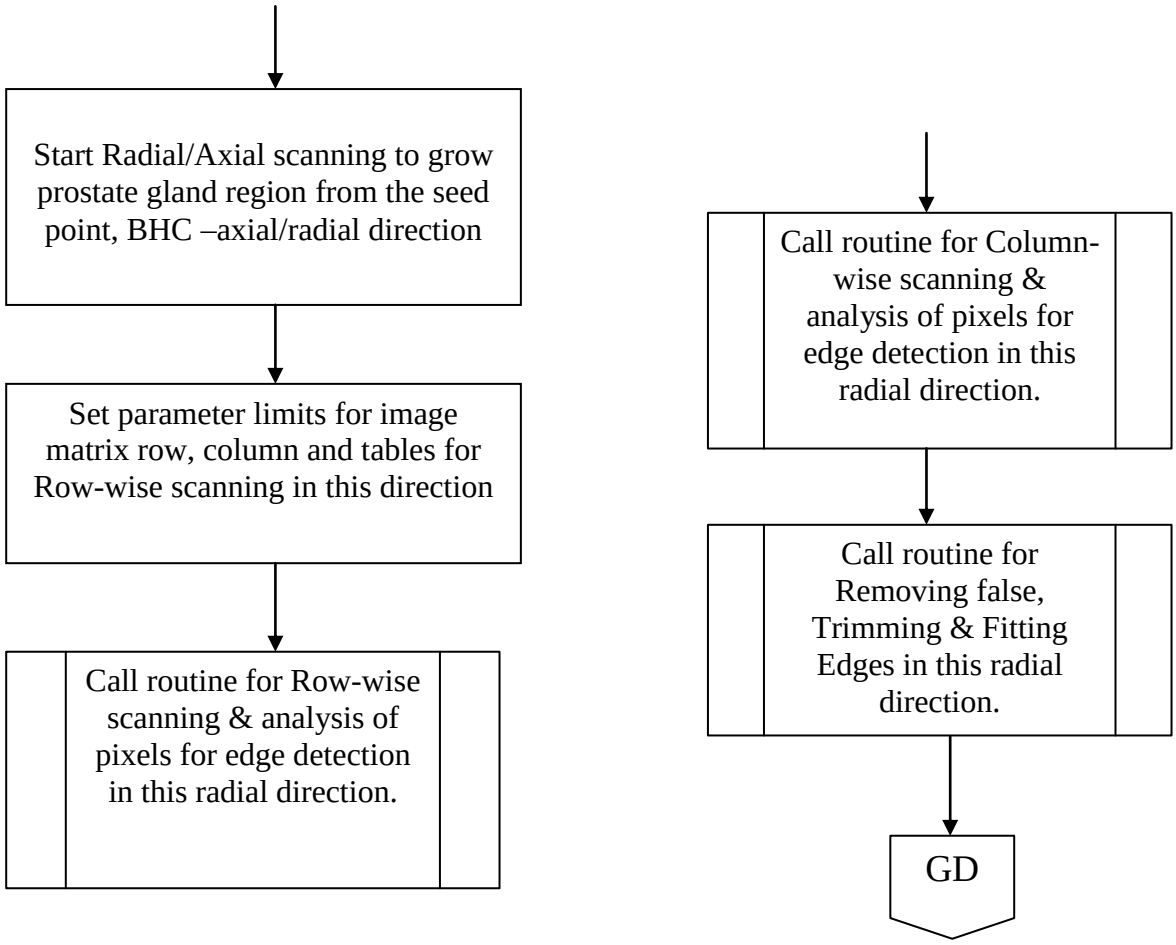


Figure 3.12D: Flowchart for scanning image matrix in BHC – radial direction.

The Pseudo-code for scanning prostate gland image for boundary and malignant tissues – detection in BHC – Radial direction is as follows:

4. Call routine for Accumulating all Row-wise & Column-wise Edges detected in this radial direction into the Segmented/ Edge/Malignant table.

   Start Radial/Axial scanning to grow prostate gland region from the seed point, BHC – axial/radial direction about the seed point Rs, Cs.

   Set parameter limits for image matrix row, column and tables for Row-wise scanning in this direction.

Call routine for Row-wise scanning & analysis of pixels for edge detection in this radial direction.

Call routine for Removing false & Trimming Edges in this radial direction.

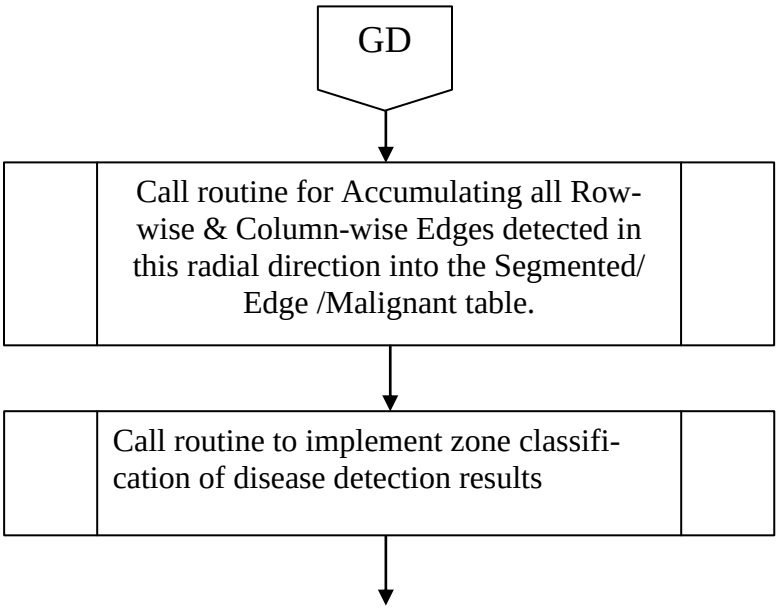
Set parameter limits for image matrix row, column and tables for Column-wise scanning in this direction.

Call routine for Column-wise scanning & analysis of pixels for edge detection in this radial direction.

Call routine for Removing false, Trimming & Fitting Edges in this radial direction.

The parameters for the direction are set for row-wise and column-wise scanning. Subroutines for row-wise and column-wise scanning are respectively called along with a routine for removing false edges, trimming and fitting edges. A routine to accumulate detected edges and detect malignant pixels within the prostate boundary detected are called.

Figure 3.12E shows concluding sections of the flowchart for image segmentation and detection of malignant pixels in prostate gland image. Routinesfor zone classification of disease detection results and displaying the result are called before this subsystem operation is terminated.



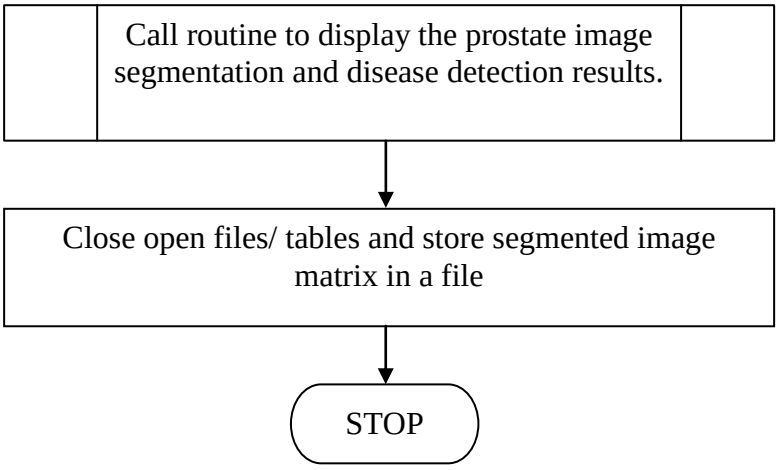
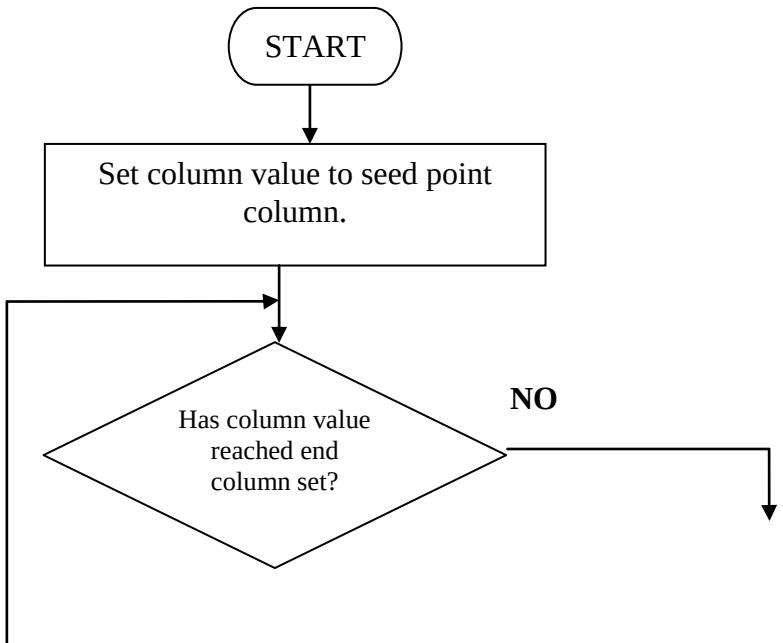


Figure 3.12E: Flowchart for concluding scanning, zoning, storing and displaying Segmentation and detection results.

The Pseudo-code for concluding scanning prostate gland image for boundary and malignant tissues and storing/displaying result is as follows:

5. Call routine for Accumulating all Row-wise & Column-wise Edges detected in this radial direction into the Segmented/ Edge /Malignant table.
   Call routine to implement zone classification of disease detection results
   Call routine to display the prostate image segmentation and disease detection results
   Store segmented file results and tables and Close files

Stop.



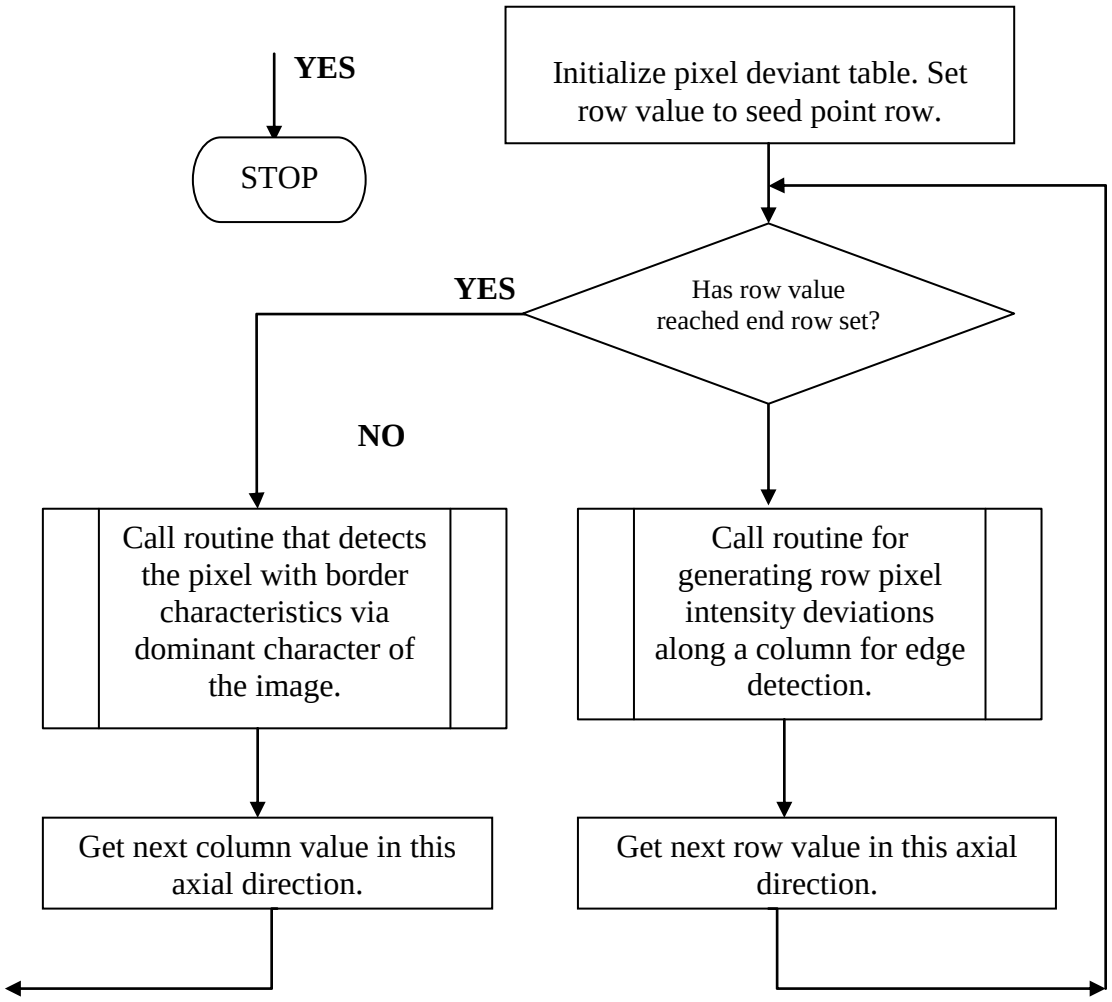


Figure 3.13A:Flowchart for algorithm that scans pixels in row-wise radial directions for edge and malignant intensity presence.

Figure 3.13A is the flowchart for the sub-routine that scans pixels in row-wise radial directions for edge and malignant intensity presence. It scans for all rows in the range specified by this radial direction for every selected column in the same direction starting from the seed point column and row. The subroutine that generates pixel intensity deviation along a column for all rows towards the boundary of the prostate gland image is invoked. When the limit of the rows is reached a second subroutine that detects the pixel with the boundary characteristics determined by the dominant characteristics of the image ascertained from the dominant intensity determination subroutine earlier described.

The pseudo code for the algorithm in Figure 3.13A is as follows:

1. Do while not end of column set for direction

   Initialize pixel deviation table

   Do while not end of row set for direction

   Call routine for generating pixel intensity deviations between rows along a column for edge detection.

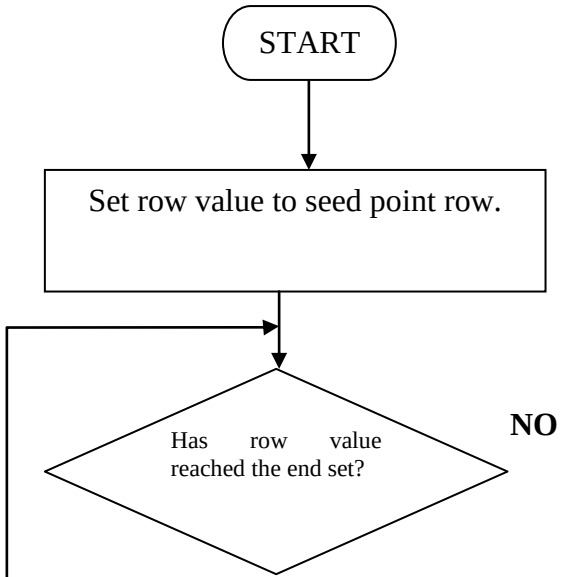
   Endwhile row limit reached

   Call routine that detects the pixel with border characteristics using dominant character of the image determined before.

   Endwhile column limit reached.

2. Stop

Figure 3.13B is the flowchart for the sub-routine that scans pixels in column-wise radial directions for edge and malignant intensity presence. It scans for all columns in the range specified by this radial direction for every selected row in the same direction starting from the seed point row and column. The subroutine that generates pixel intensity deviation along a row for all columns towards the boundary of the prostate gland image is invoked. When the limit of the columns is reached a second subroutine that detects the pixel with the boundary characteristics determined by the dominant characteristics of the image ascertained from the dominant intensity determination subroutine earlier described.



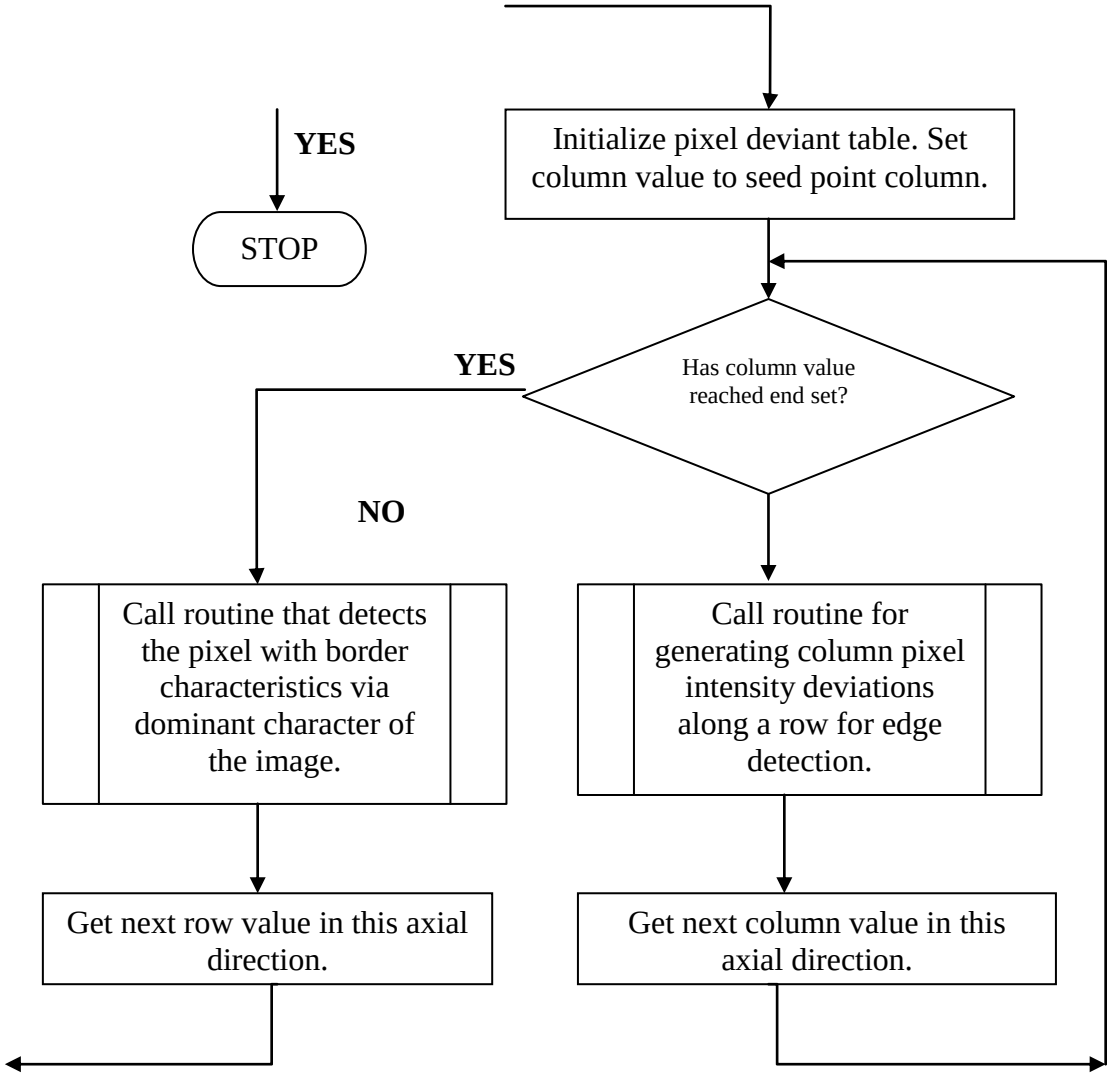


Figure 3.13B:Flowchart for algorithm that scans pixels in column-wise radial directions for edge and malignant intensity presence.

The pseudo code for the algorithm in Figure 3.13B is as follows:

1. Do while not end of row set for direction

   Initialize pixel deviation table

   Do while not end of column set for direction

   Call routine for generating pixel intensity deviations between columns along arow for edge detection.

   Endwhile column limit reached

Call routine that detects the pixel with border characteristics
using dominant character of the image determined before.

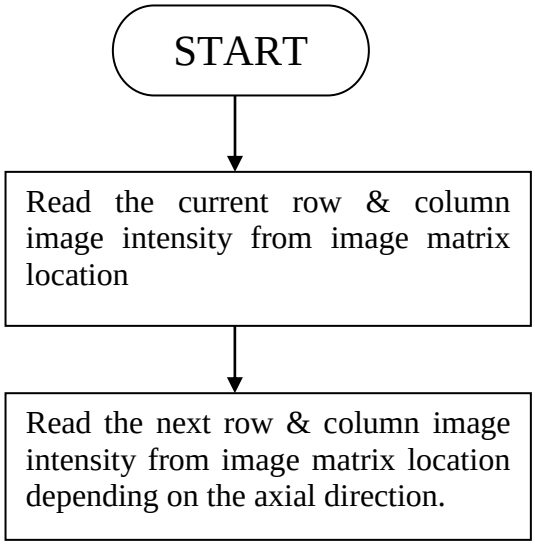
Endwhile row limit reached.

2. Stop

Figure 3.13C is the flowchart for the sub-routine that generates pixel intensity deviations between next neighbours along the row/column for edge detection. For each row and column identity for a pixel, it computes the new pixel intensity deviation by subtracting the next pixel intensity from the current pixel intensity. These values are stored in the pixel deviation table for the row-wise or column-wise scanning session.

The pseudo-code for the algorithm shown in Figure 3.13C is as follows:

1. Read the current row & column image intensity from image matrix location
   Read the nextrow & column image intensity from image matrix location depending
   on the axial direction in question
   Compute the current deviation, making the former the old and current new.
   Store the values row, column, current pixel intensity, next pixel intensity,
   old and new intensity deviation.
2. Stop.



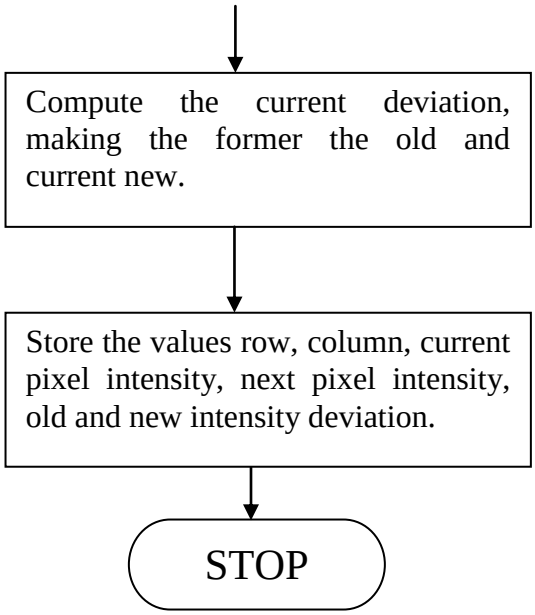
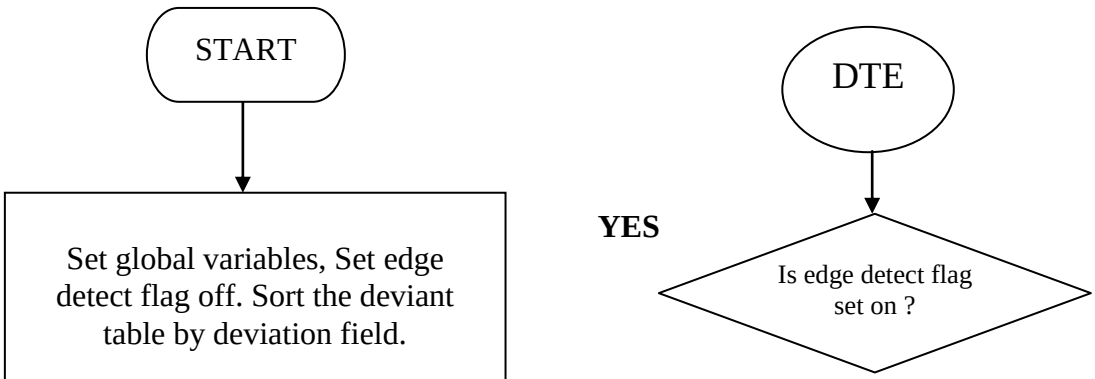


Figure 3.13C: Flowchart for routine that generates pixel intensity deviations between next neighbours along the row/column for edge detection.

Figure 3.13D is the flowchart for the sub-routine that detects the pixel with border characteristics using dominant parameters for the chosen image matrix. The parameters set include; current pixel minimum and maximum intensity range, next pixel minimum and maximum intensity range. This routine uses the generated table of pixel intensity deviations, sorts it in ascending order of the deviation. The algorithm checks the row and column values of highest deviation to ensure it falls within the region expected to have the prostate gland. This applies the knowledge that pixel intensities within the gland are lower, (darker) than that of the surrounding tissues. This means that transition from the gland pixel to surrounding tissue pixel will produce a positive and large intensity deviation. The routine sets boundaries for minimum row for row-wise and column for column-wise scanning for each axial direction of scanning for edges.



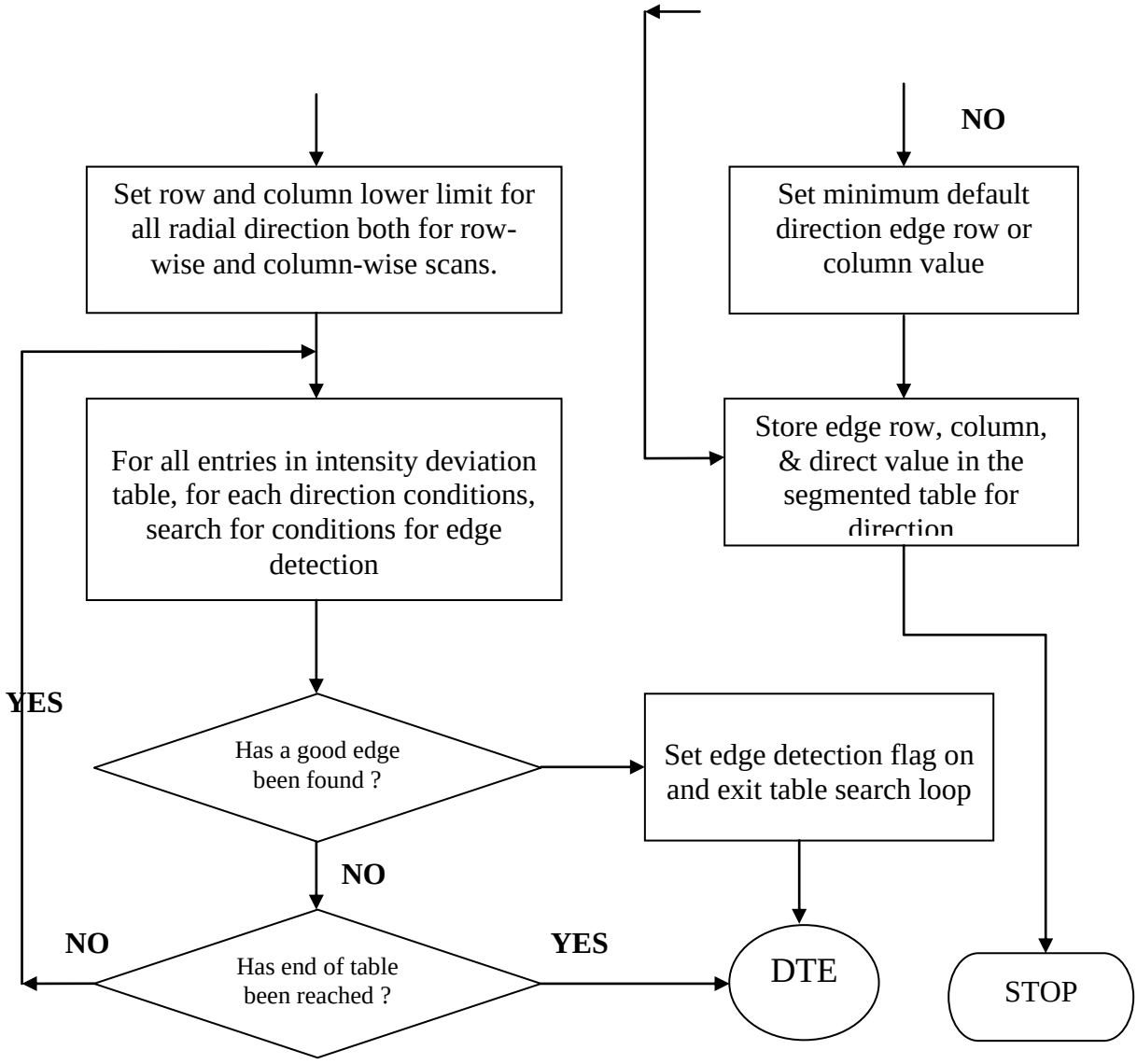


Figure 3.13D: Flowchart for detecting the best edge from pixel intensity deviation table.

The pseudo-code for the algorithm shown in Figure 3.13D is as follows:

1. Set global variables, Set edge detects flag off. Sort the deviant table by deviation field.

   Set row and column lower limit for all radial direction both for row-wise and column-wise scans.

   Do while not end of pixel intensity deviation table

   For every entries in intensity deviation table, for each direction conditions,

search for conditions for edge detection

If entry satisfies good edge conditions

Set edge detection flag on and exit table search loop

End

Enddo

If edge detection flag off

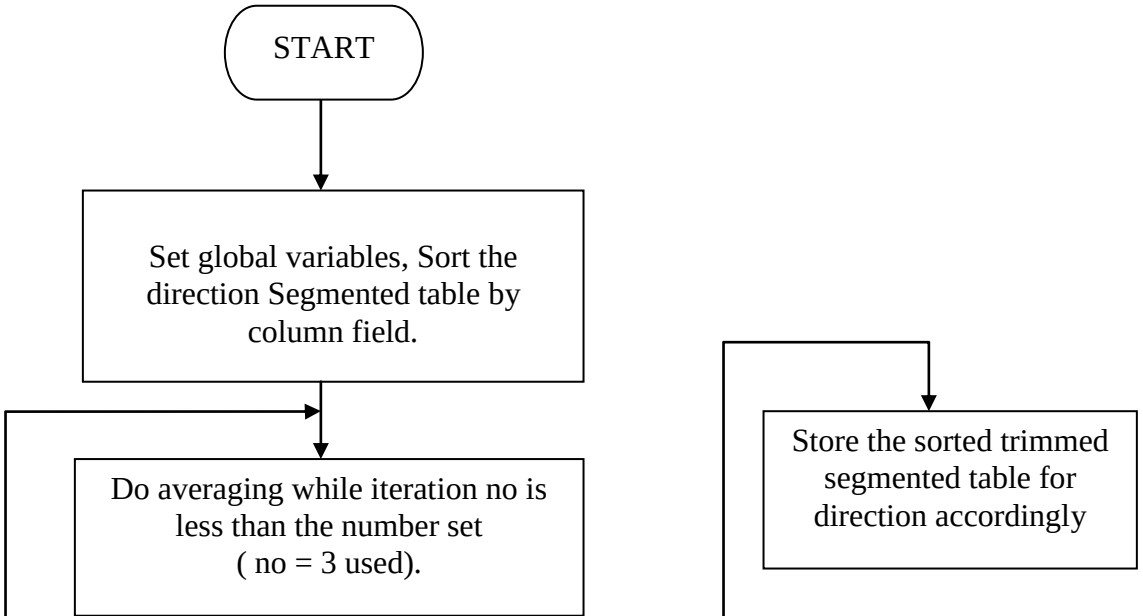
Set the default edge column for column-wise or row for row-wise for the current Row or Column respectively.

End

Store the row, column and direction values in the Segmented image table for the current direction of scan

2. Stop

Figure 3.13E is the flowchart for the sub-routine that implements removal of false and trimming of detected edges in the row-wise edge table in any direction. The algorithm uses averaging technique that iterates for a number of times. From the first element of the table of found edges in row-wise scanning average of current and next pixel intensities are used to trim down edges that are off the smooth curvature of the edges of the prostate gland into alignment (prior knowledge abound the shape of a typical prostate gland). The averaging is done until end of table and is repeated. By experiment three iterations was found to yield best result.



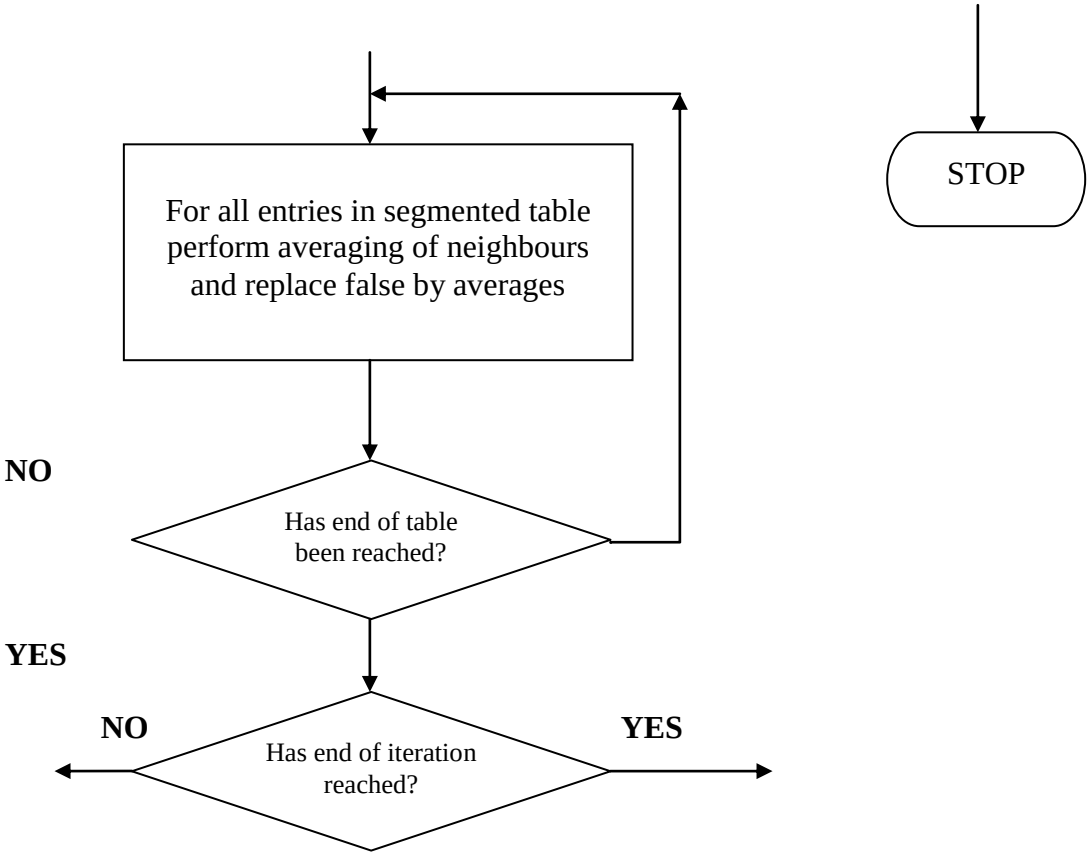


Figure 3.13E: Flowchart for removing & trimming edges in row-wise segmentation table.

The pseudo-code for the algorithm shown in Figure 3.13E is as follows:

1. Set global variables Sort the segmented image table by column field.
   While iteration number less than set value (no 3 used).

. Do averaging while not end of sorted segmented image table for direction
   For all entries in segmented table perform averaging of neighbours and replace false by averages
   Enddo

Endwhile

Store the sorted trimmed segmented coordinate table for corresponding direction of scan

2. Stop

Figure 3.13F is the flowchart for the sub-routine that implements removal of false, trimming and fitting of detected edges in the column-wise edge table in any direction. The algorithm uses averaging technique that iterates for a number of times. From the first element of the table of found edges in column-wise scanning average of current and next pixel intensities are used to trim down edges that are off the smooth curvature of the edges of the prostate gland into alignment (prior knowledge abound the shape of a typical prostate gland). The averaging is done until end of table and is repeated. By experiment, eight iterations was found to yield best result.

The pseudo-code for the algorithm shown in Figure 3.13F is as follows:

1. Set global variables Sort the segmented image table by row field.

   While iteration number less than set value (no = 8 used).

. Do averaging while not end of sorted segmented coord. image table for direction

   For all entries in segmented table perform averaging of neighbours and

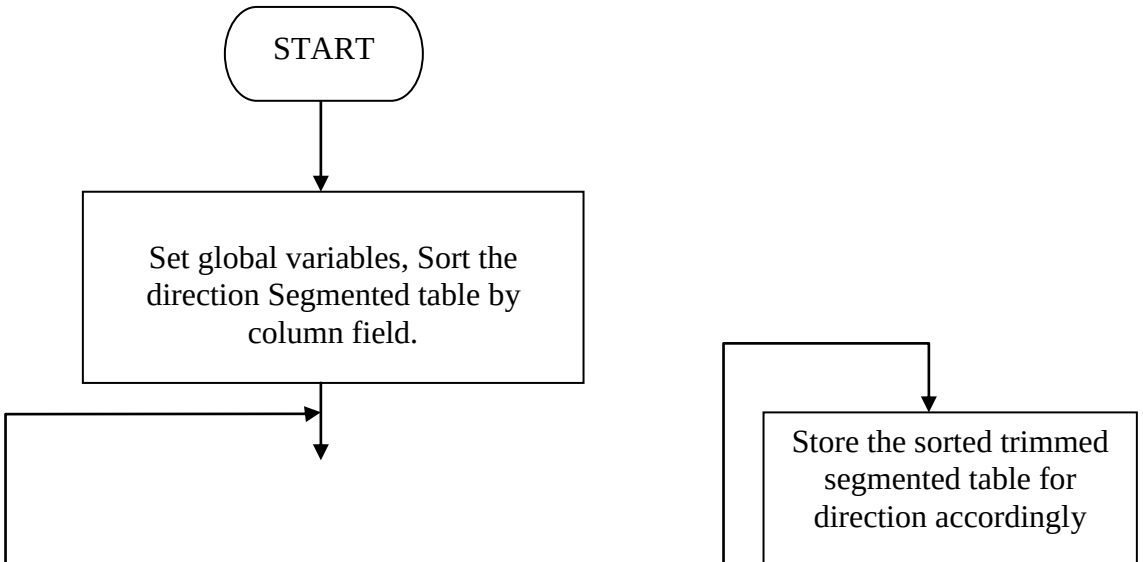
   Replace false by averages

   Enddo

   Endwhile

   Store the sorted trimmed segmented coordinate table for corresponding direction of scan

2. Stop



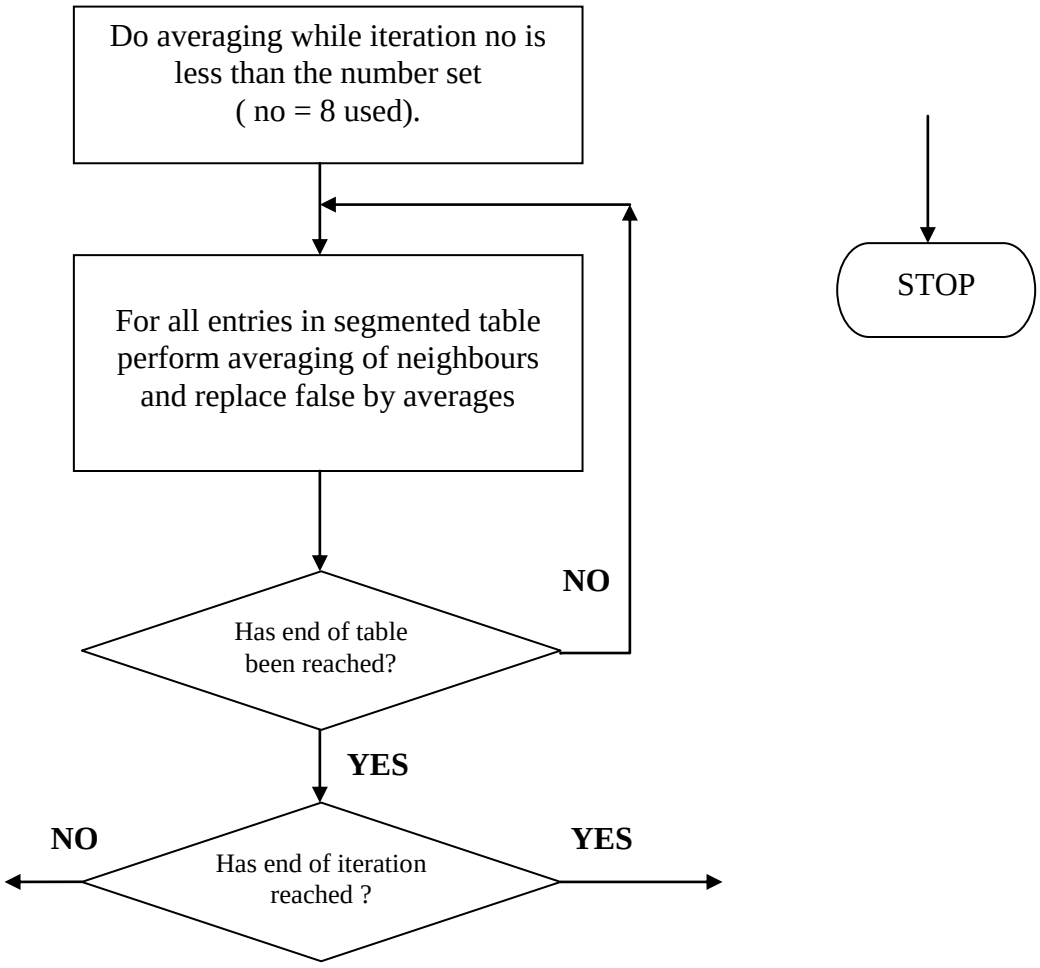


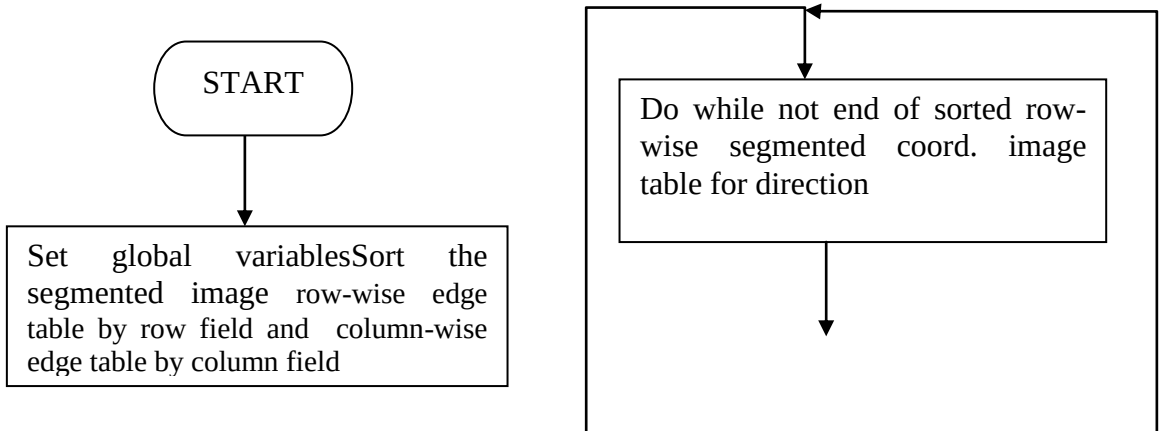
Figure 3.13F: Flowchart to remove, trim &fit edges in column-wise segmentation table

Figure 3.13G is the flowchart for the sub-routine that implements accumulation of pixels in row-wise, column-wise detected edges into the segmented edge coordinates table and also accumulates possible malignant pixels within the edge boundaries of the prostate gland using dominant intensity determined following knowledge rule. During this accumulation of edge coordinates, the total number of pixels within the segmented prostate boundary is obtained. The algorithm sorts the row-wise tables by rows and column-wise table by columns. It accumulates the edges in the row-wise scanned result table first followed by the edges in the column-wise scanned result table. For each edge coordinate located during the scanning, this algorithm starts from the seed point coordinate and accumulates all the pixels bound by that edge coordinate until all edges in that table is handled. It

also calls another subroutine that checks each pixel bound by the edge for malignancy properties, accumulates and stores results. In this way this algorithm obtains the number of pixels in the segmented prostate gland region of the image, hence the area in pixels of the segmented prostate gland by the proposed algorithm.

The pseudo-code for the algorithm shown in Figure 3.13G is as follows:

1. Set global variables
   Sort the segmented image row-wise obtained edge table by row field.
   Sort the segmented image column-wise obtained edge table by column field.
. Do while not end of sorted column-wise segmented coord. image table for direction
   Read (row, column) entries in segmented table perform
   For col from seed point column to column of edge
   Call routine to detect malignant pixel bound by current edge coord
   Increment count of the segmented pixels
   Store in Total Segmented Coord table the row of edge, col, & direction of scan
   Endfor
   Enddo
   Do while not end of sorted row-wise segmented coord. image table for direction
   Read (row, column) entries in segmented table perform
   For row from seed point row to row of edge
   Call routine to detect malignant pixel bound by current edge coord
   & Increment count of pixels
   Store in Total Segmented Coord table the col of edge, row, & direction of scan
   Endfor
   Enddo
2. Stop



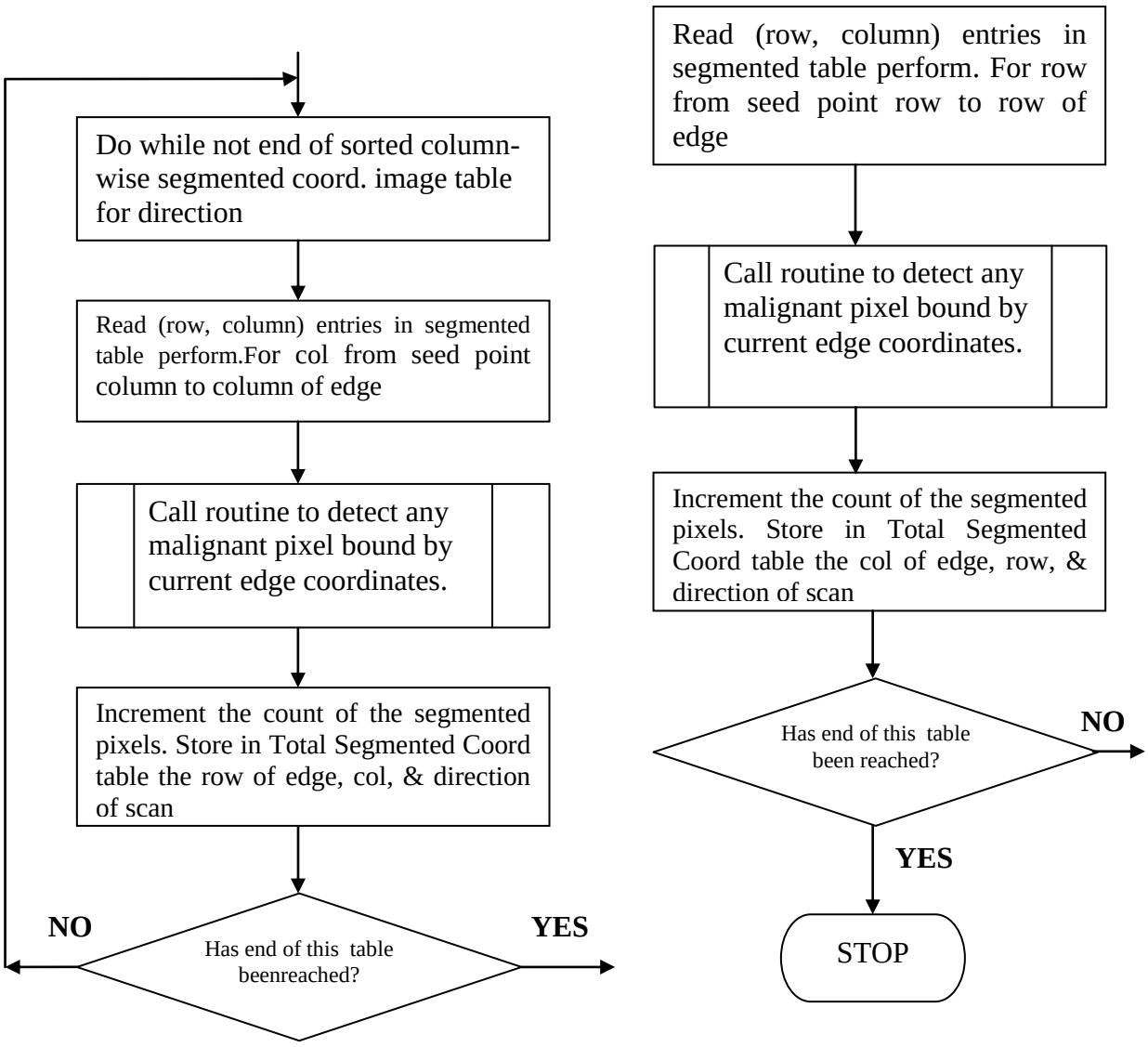


Figure 3.13G: Flowchart to accumulate pixels in segmented prostate gland region &detect malignant regions.

Figure 3.13H shows the flowchart for the sub-routine that detects pixels with malignant properties that are bound by the edge being accumulated within the segmented prostate gland image. The intensity of the edge pixel is compared with a malignancy intensity range determined by knowledge rule. For pixels with intensity within the range, their eight neighbours are compared too to validate that the region is consistently malignant. At least two immediate neighbours should possess same property before a pixel is suspected malignant. For a pixel confirmed to be in a region of malignant pixels, its row, column coordinates, its intensity, and direction of scan are stored in a malignancy coordinates table.

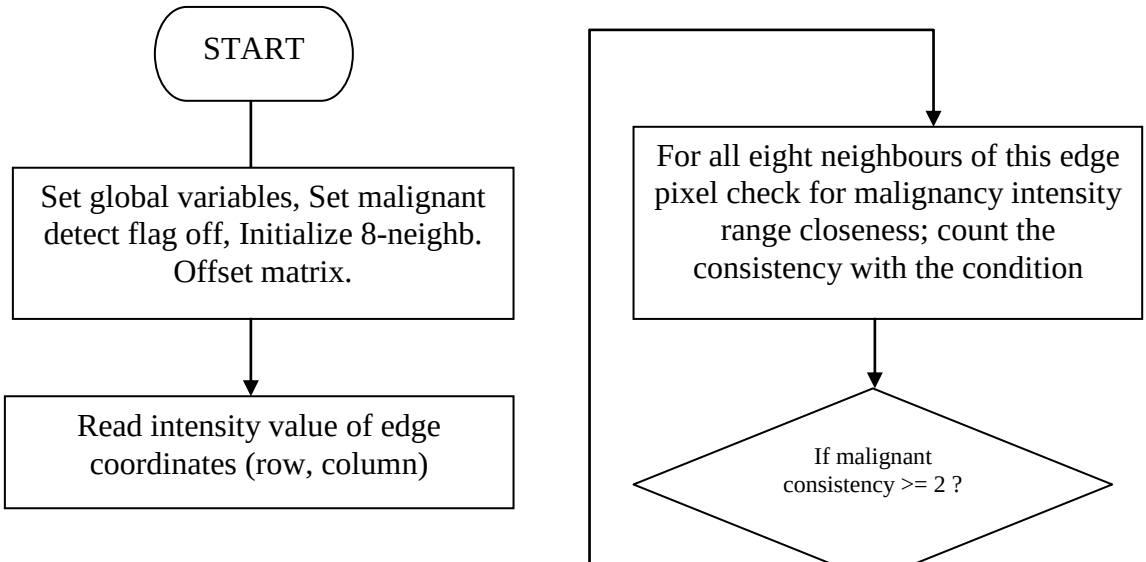
The pseudo-code for the algorithm shown in Figure 3.13H is as follows:

1. Set global variables;Set malignant detect flag off, Initialize 8-neighb. Offset matrix.
   Read the intensity value of the edge detected from the image matrix

   If intensity of edge pixel is less or equal tomalig_intesity_range ?
      For all eight neighbours of this edge pixel,
         check for malignancy intensity range closeness;
         count the consistency with the condition.

         If malignant consistency >= 2 ?
            Set malignancy flag on
         Endif
      Endfor
   Endif
   If malignancy flag is set on
      Increment count of malig pixels;
      Store row & column coordinates, direct & intensity
   Endif

2. Stop



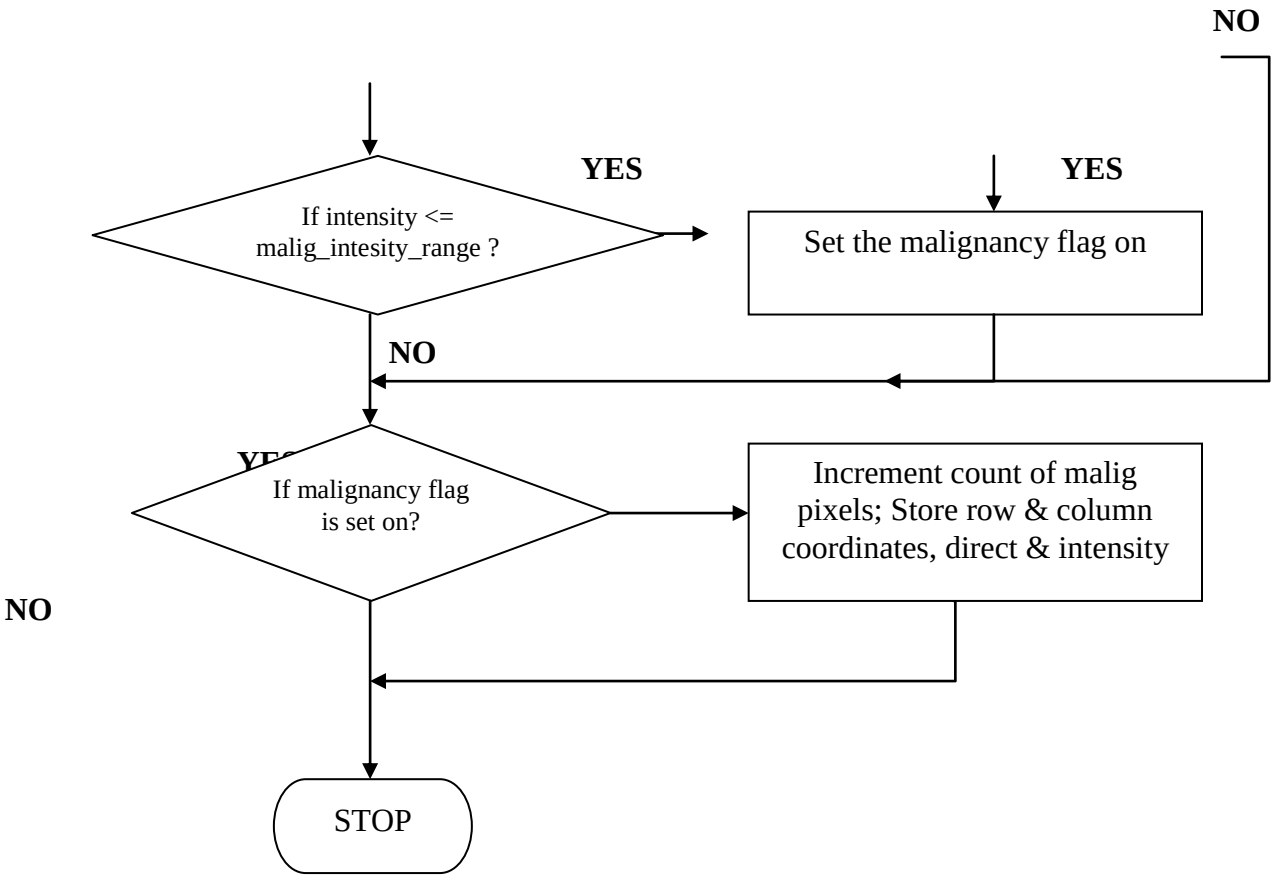


Figure 3.13H: Flowchart for routine that detects malignant pixels within segmented edge.

Figure 3.14 gives the description of the subroutine that displays the segmented prostate image. It reads the following input; original image file, segmented image file, table of coordinates of segmented pixels and the table of malignant tissue coordinates. It displays the original image side by side with the segmented image. Figure 3.14 shows the flowchart for this algorithm.

The Pseudo-code for the subroutine that displays the segmentation detection result is as follows:

1. Read the original image file, segmented image coordinate tabl, edge coord and malignant coord tables

   Use the edge coordinates table content to trace the boundary of the prostate gland

   Use the malignant cood tabl to show the malignant pixels or suspected cancerous tissues in white within the segmented region. Compute zone ratios

Display original, segmented& cancerous zones image files under separate axes

2. Stop

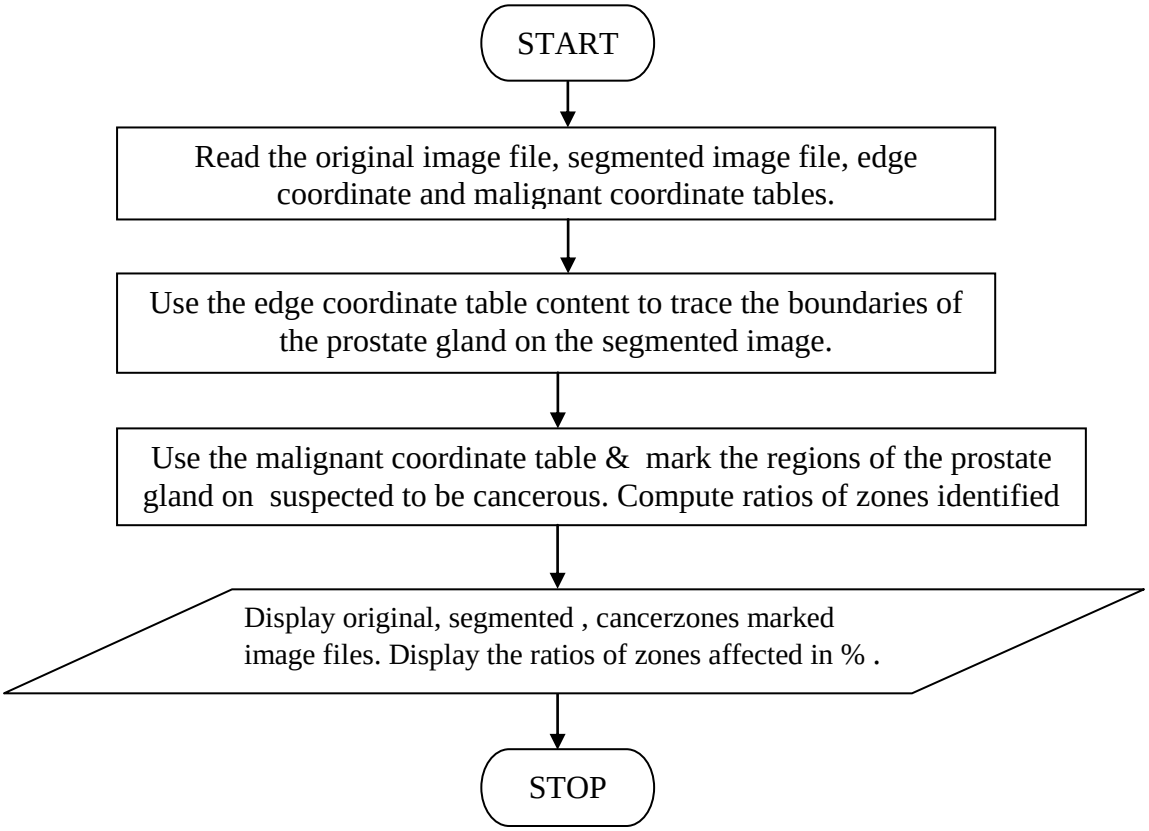


Figure 3.14: Flowchart/algorithm for displaying segmented image.

3.6.5.6 Zoning of Image Matrix by Ratio Based Metrics Design

The design here ensures the realisation of research objective iii. It uses the anatomical zone distribution of the prostate gland, namely peripheral (65%), central (25%) and transition (10%) given in section 2.1.1.1 and Figure 2.3. The seed point coordinate is assumed to be about the centre line of the segmented prostate gland. The segment of the prostate image that is likely to contain the prostate gland given by production rule 3 is split into 16 segments, of 22.5 degree, about the seed point centre axis (see Figure 3.15). The zones will share the segments in the ratio of 7:2:1, for peripheral-anterior, central, and transition zones respectively.

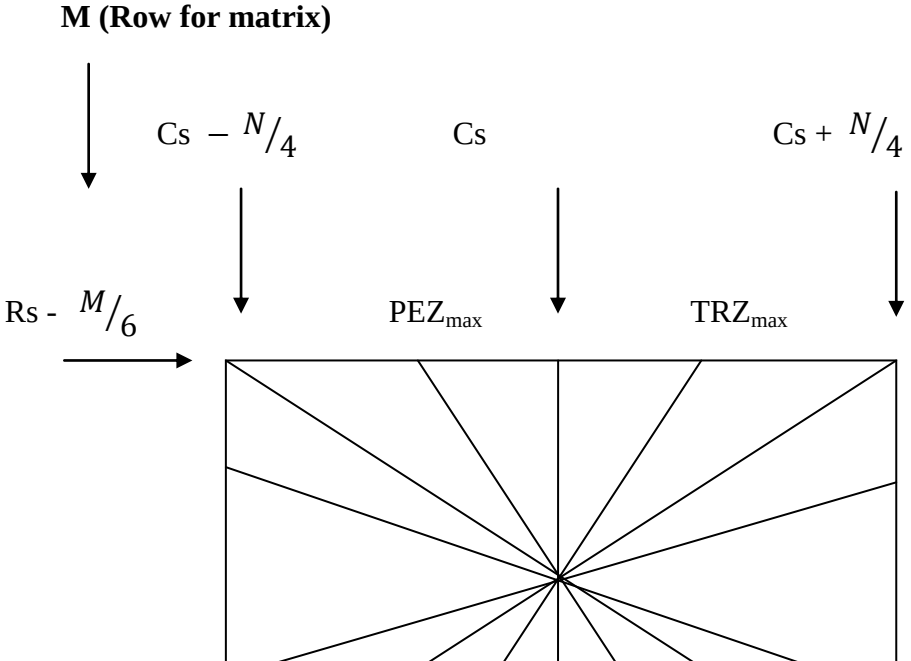
Peripheral zone will occupy, $^{7}/_{10} * {}^{16}/_{1}$ = 11.2, approx. 11 segments

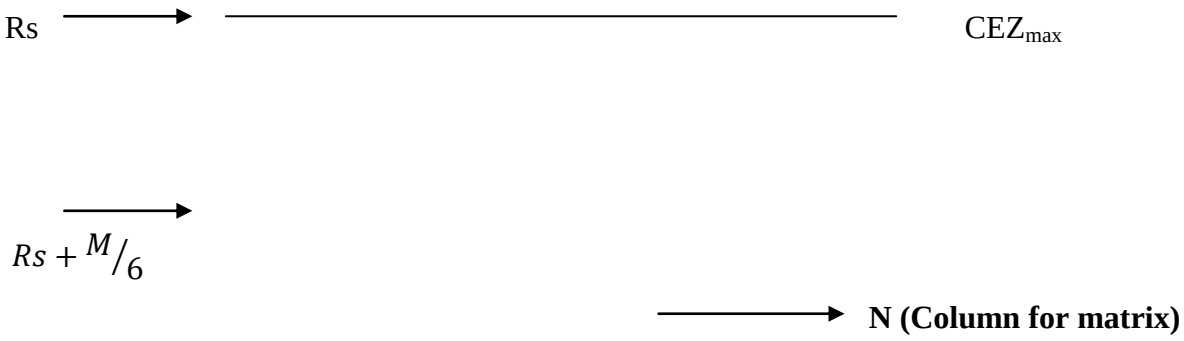
Central zone will occupy, $^{2}/_{10} * {}^{16}/_{1}$ = 3.2, approx. 3 segments

Transition zone occupy, $^{1}/_{10} * {}^{16}/_{1}$ = 1.6, approx. 2 segments

The Transition zone will occupy the 2 segments located in the 1st 22.5 degrees of the 1st quadrant and the 4th 22.5 degrees of the 4th quadrant of the axis shown in Figure 4.16. This position is in consonant with the location in Figure 2.3. The Central zone is next to the right of the Transition zone occupying the next 3 segments. It is next 3, 22.5 degrees of the 1st quadrant. The Peripheral-anterior zone occupies the remaining 11 segments spanning 2nd , 3rd and 1st 3, 22.5 degrees of the 4th quadrant of the axis about the seed point within the matrix of the image. These segments and rule 3 are used to determine the row and column limits for the pixel coordinates that will be stored for the different zones in the lookup table to be generated

Axial locations of the three zones in the image matrix are used to apportion pixel coordinates to the zones. The converted zone identities are stored in tables.





Rs & Cs are the row and column coordinates of the seed point,
PEZmax represents the end of the peripheral zone and the beginning of the transition zone
TRZmax represents the end of transition zone and the beginning of central zone
CEZmax represents the end of the central zone and the beginning of peripheral zone

Figure 3.15: Classifying the matrix of the prostate gland region into 16 segments.

.

The pseudo-codes for this algorithm is as follows:

1. Open segmented image coordinates table
   Use Rule 3 and expert ratio of zones of prostate gland as in Figure 2.3 to build a
   Lookup table for pixel coordinates in image matrix belonging to gland zones
   Rowsz = M/3, Colsz = N/2 , rule subset of gland region in matrix
   Gldtbsz = M/3 * N/2
   Create table, Gldzonlkuptb (Gldtbsz, 3) = 0; tblcnt = 1
   Accumulate Transition zone coordinates in matrix subset, E axial directn
   Rn = Rs-1; to avoid boundary of zones
   Cn = Cs-N/8; limit for TRZmin, same as PEZmax = Cs-(N/2*1/4)
2. Cnx = Cs,
   FOR Rnx = Rn to (Rs – M/6) step - 1, Generate neighbours along this radial path for inclusion in as pixels in zone
   Store table table contents, Gldzonlkuptb (Gldtbsz, 3) = “Transition”
   Gldzonlkuptb (Gldtbsz, 1) = Rnx, Gldzonlkuptb (Gldtbsz, 2) = Cnx

Tblcnt = tblcnt +1, next tabl locatn

Cnx = Cnx – 1

IF Cnx < Cn

Then Go to 4, start new axial accumulation of coords

Endif

3. Next Rnx

4. Rn = Rn -1; Cn = Cn + 1

IF (Rn < (Rs – M/6)) OR (Cn > Cs)

Then Go to 5, accumulate G axial side of transit zone

Else

Go to 2, continue next E axial coordinates in transition zone

Endif

Continue accumulatn in G axial of Transition zone

5. Rn = Rs-1; to avoid boundary of zones, Cn = Cs+N/4

Cn = Cs+N/8; limit for TRZmax, same as CEZmin = Cs+(N/2*1/4)

6. Cnx = Cs,

FOR Rnx = Rn to (Rs – M/6) step - 1, Generate neighbours along this radial path for inclusion in as pixels in zone

Store table table contents, Gldzonlkuptb (Gldtbsz, 3) = “Transition”

Gldzonlkuptb (Gldtbsz, 1) = Rnx, Gldzonlkuptb (Gldtbsz, 2) = Cnx

Tblcnt = tblcnt +1, next tabl locatn

Cnx = Cnx + 1

IF Cnx > Cn

Then Go to 7, start new axial accumulation of coords

Endif

Next Rnx

7. Rn = Rn -1; Cn = Cn - 1

IF Rn < (Rs – M/6) OR Cn < Cs

Then Go to 8, start accumulating central zone coords

Else

Go to 6, continue next G axial coordinates in transition zone

Endif

Start Accumulating Central zone using row-wise step

8. Cn = Cs ; to start at boundary of zones,
   Rn = Rs-M/6 ;
9. Rnx = Rs;
   FOR Cnx = Cn to (Cs + N/4) step 1, Generate neighbours along this radial path for inclusion in as pixels in zone
   Store table table contents, Gldzonlkuptb (Gldtbsz, 3) = "Central"
   Gldzonlkuptb (Gldtbsz, 1) = Rnx, Gldzonlkuptb (Gldtbsz, 2) = Cnx
   Tblcnt = tblcnt +1, next tabl locatn
   Rnx = Rnx - 1
   IF Rnx < Rn
   Then Go to 10, start new axial accumulation of coords
   Endif
   Next Cnx
10. Cn = Cn + 1
   IF (Cn > (Cs+ N/4))
   Then Go to 11, start accumulating peripheral - anterior zone coords
   Else
   Go to 9, continue next G axial coordinates in transition zone
   Endif

Start Accumulating Peripheral-anterior zone using

11. Rn = Rs-M/6 ;
   Cn = Cs ; to start at boundary of zones,
12. Rnx = Rs;
   FOR Cnx = Cn to (Cs - N/4) step - 1, Generate neighbours along this radial path for inclusion in as pixels in zone
   Store table table contents, Gldzonlkuptb (Gldtbsz, 3) = "Peripheral"
   Gldzonlkuptb (Gldtbsz, 1) = Rnx, Gldzonlkuptb (Gldtbsz, 2) = Cnx
   Tblcnt = tblcnt +1, next tabl locatn
   Rnx = Rnx - 1
   IF Rnx < Rn
   Then Go to 13, start new axial accumulation of coords

Endif

Next Cnx

13. Cn = Cn - 1

IF (Cn < (Cs - N/4))

Then Go to 14, accumulate peripheral - anterior zone cords in 2nd and 3rd

Else

Go to 12, continue next coordinates accumulatn

Endif

Start accumulating the cords in the 2nd and 3rd quadrants for peripheral zone

14. FOR Rnx = Rs to (Rs + M/6) step 1 ;

FOR Cnx = (Cs – N/4) to (Cs + N/4) step 1 ; Generate neighbrs along this path for inclusion in as pixels in zone

Store table table contents, Gldzonlkuptb (Gldtbsz, 3) = "Peripheral"

Gldzonlkuptb (Gldtbsz, 1) = Rnx, Gldzonlkuptb (Gldtbsz, 2) = Cnx

Tblcnt = tblcnt +1, next tabl locatn

Next Cnx

Next Rnx

15. Stop.

3.6.6 Expert Prostate Gland Segmentation/Cancer Detection Design

This system was designed to setup the control experiment for generating the parameters that was used to validate the performance of the cancer detection algorithm by applying the area based metrics defined in section 3.4.1. This algorithm takes as input an enhanced image whose dominant parameters has been determined and produces as output a segmented image, a file containing coordinates of the edges of the segmented prostate gland as marked by expert, a file containing coordinates of points within the segmented portion suspected to be malignant tissue as marked by expert, and a file containing the name of image file segmented and the number of pixels in the

segmented portion, representing the area of segmented prostate in pixels. The module enables the expert to mark the edges of the prostate gland for a selected image and also mark zones where suspected cancerous cells are detected.

3.6.6.1 Expert Main Menu Design

The expert control program for processing prostate images is also menu-driven. The options available in the menu are shown in Figure3.16. The flowchart of the algorithm for implementing the automation of the main menu is shown in Figure3.17.

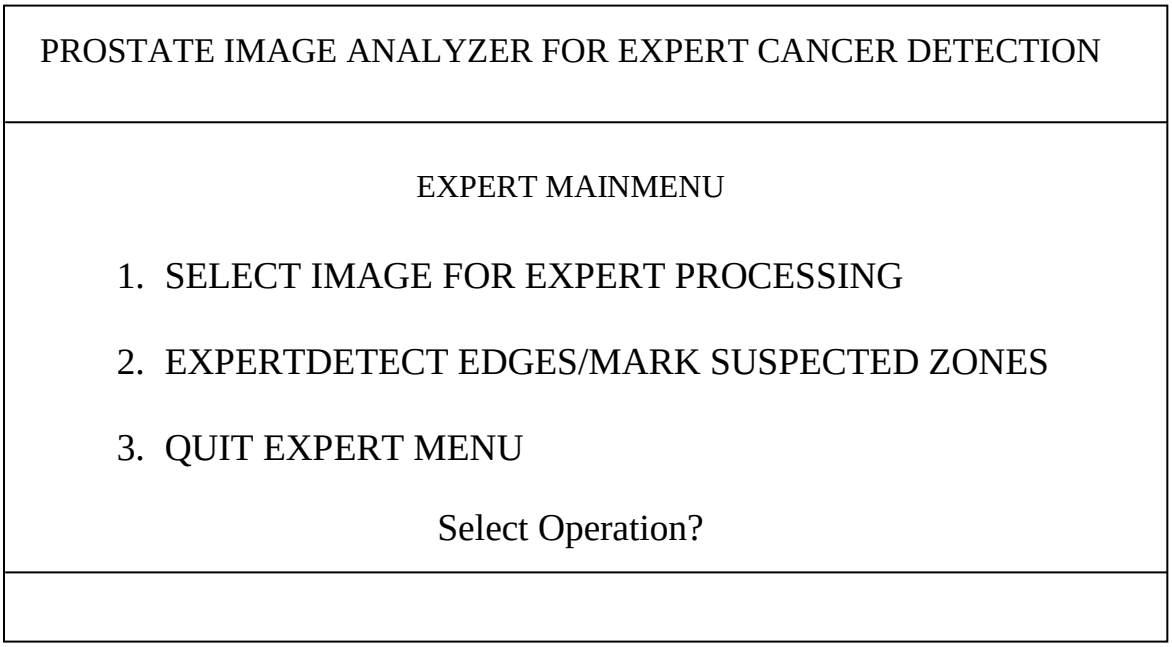
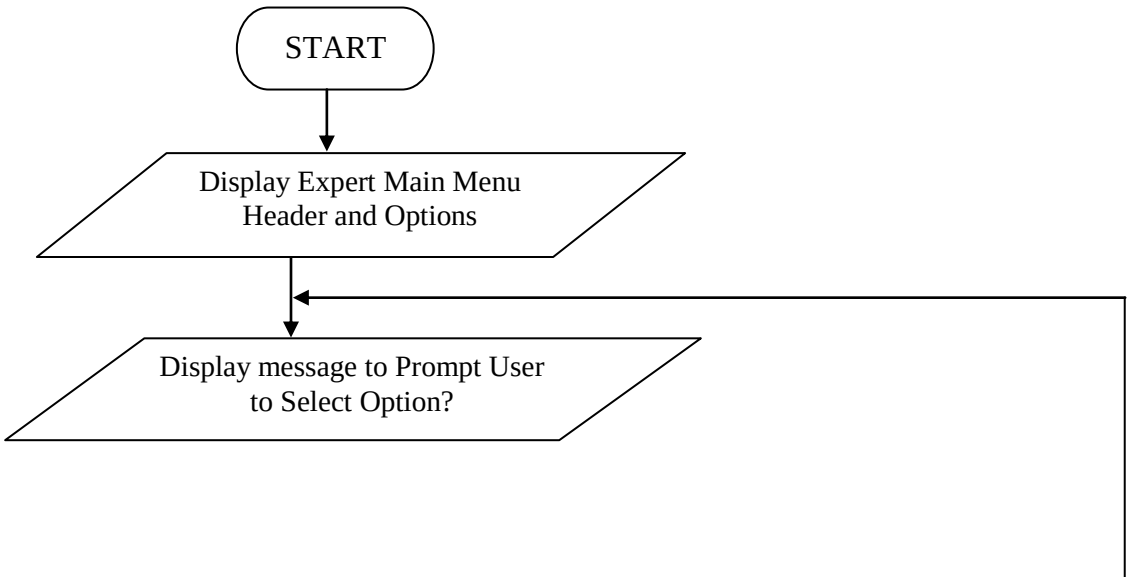


Figure 3.16: Expert main menu options.



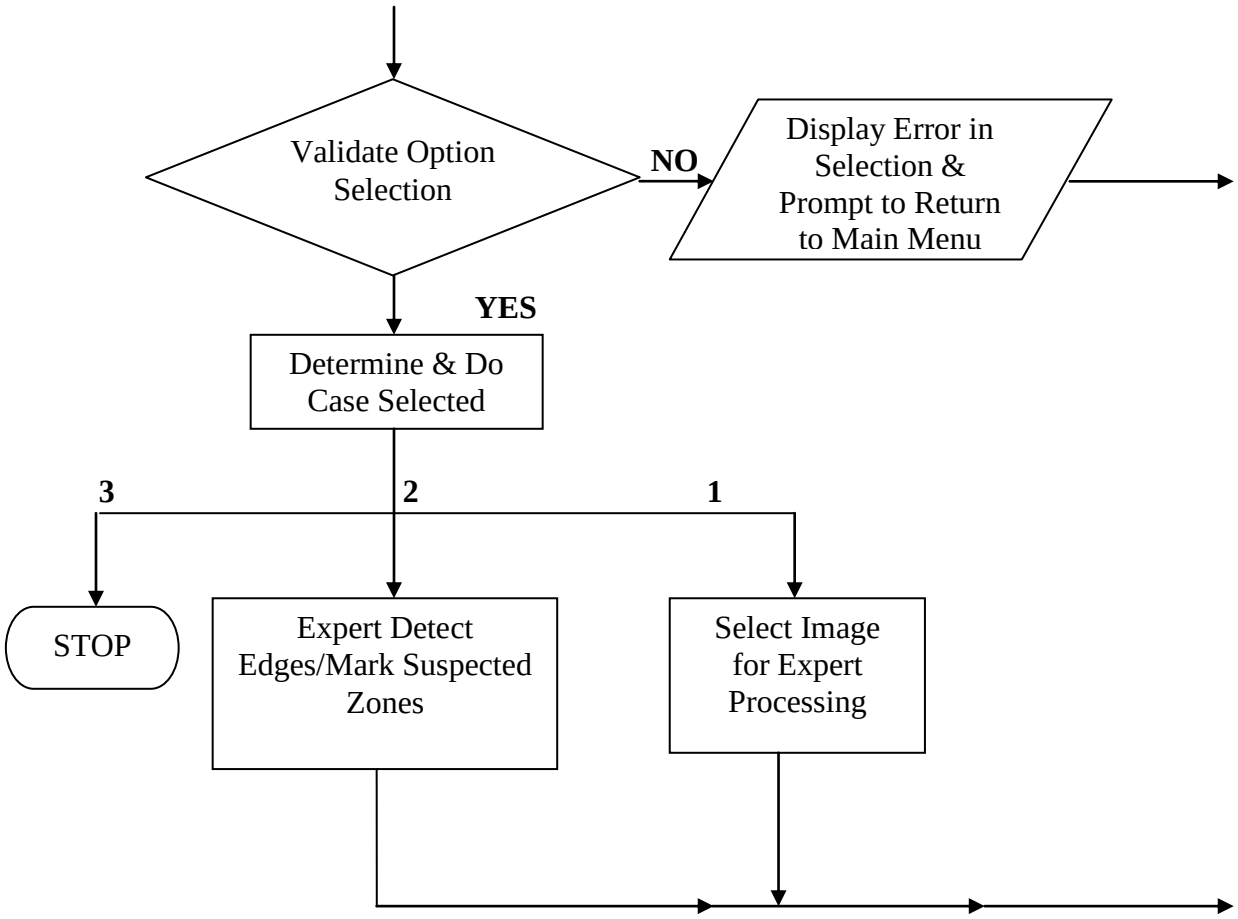


Figure 3.17 Expert main menu flowcharts.

The corresponding pseudo-code for this algorithm is as follows:

1. Display the menu layout and menu options and heading for expert prostate image analyzer for cancer detection

   Display message to prompt user to select option and receive input

   IF choice is invalid,

   Display error message and return to step 1

   Else

   IF choice is option 1

   Then call routine to Select image to be analyzed and return to step 1

   IF choice is option 2

   Then call routine forExpert Detect Edges/Mark Suspected Zones and display resulting image and return to step 1

   IF choice is option 3

Then Close menu and end.

Stop.

The algorithm that performs the image selection function of option 1 in the expert control menu is same as that described in section 3.6.4.3

3.6.6.2 Expert Detect Edges/Mark Suspected Cancerous ZonesDesign

This sub-module takes as input the image selected in section 3.6.6.1 and performs three similar pre-processingfunctions of removing labels, determining dominant intensity parameters and enhancement as described in sections3.6.5.1, 3.6.5.3 and 3.6.5.2 respectively. The enhancement aids the expert's visual observation of the images on the screen. The module calls a sub-module that performs expert segmentation. This sub-module presents the original image as well as the enhanced version to the expert to mark edges of the prostate gland. The program then uses the marked edges to extrapolate all other edge points between the marked edges. All the pixels bound by the edge points are computed as the area of the segmented gland in pixels and are stored along with the tables of the edges/pixels coordinates. An image showing the area of the segmented gland in highest intensity (white) is produced as output for display.

Subsequently another sub-module that performs expert suspected zone cancer marking was called. Theexpert was presented with the original image as well as the enhanced version to supply the number of sections in each of the three zones of the prostate with suspected cancerous cells, revealed by hyper-echoic pixels intensities (darker pixels that surrounding pixels). For each of the zones and sections supplied the program allows the expert to mark at least five points around that section in the zone. An edge table was generated for the edge coordinates of the sections marked. The program then uses the marked edges to extrapolate all other edge points between the marked edges. All the pixels bound by the edge points are computed as the area of the suspected section in the zone identified in pixels and are stored along with the tables of the edges/pixels coordinates. The process was repeated for all the zones of the prostate gland supplied as containing suspected cancerous cells. An image showing the sections in the selected marked as suspected for malignancy is generated as output for display. The results of the expert segmentation and detection of suspected cancerous zones are display along with the resultant images.

The pseudo-code of the algorithm of the sub-module for expert segmentation of the image is as follows:

Initialialize variables for global and local access –expsegmtedcoordtabl, expsegmtededgetabl, expsegmtedareacnt, pointcoord1, pointcoord2

1. Show original image side by side with enhanced image

   Prompt expert to mark at least 10 points around the identified edges of the prostate gland within the image

   Capture the points in an array

   If points array is empty, expert made no marks,

      Display message to alert expert no marks were made

      Return to 1

   End

   Prepare the array in a table for processing, Display processing message

   Determine Seed/Centre point within the points marked as gland area

   Connect the edges points marked and store edge table

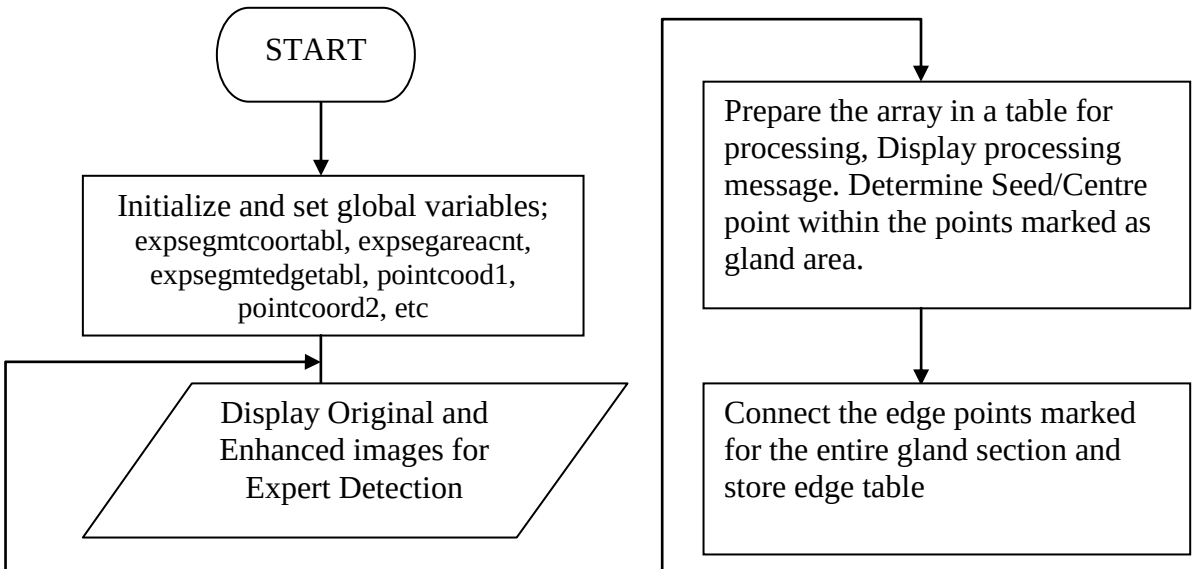
   Accumulate all pixels in image bound by edge points generated in the table

   Set these pixels to highest intensity to differentiate segmented gland in the image and

      create a segmented image file for display

2. Stop.

The corresponding flowchart is shown in Figure 3.18



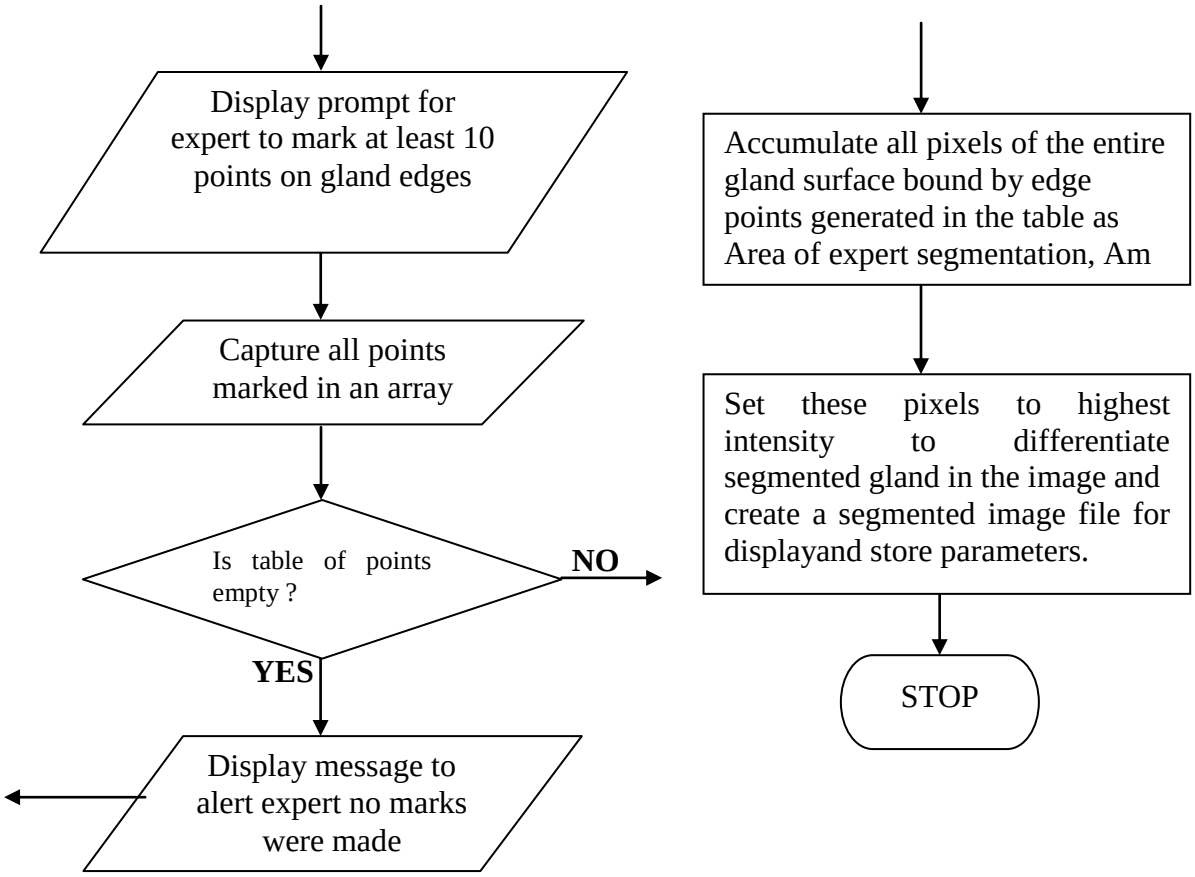


Figure 3.18:Expert gland segmentation algorithm flowcharts

The pseudo-code of the algorithm of the sub-module for expert gland zones cancerous cell detection on the image is as follows:

Initialize variables for global and local access – expsegmtedcoordtabl, expsegmtededgetabl, expsegmtedareacnt, pointcoord1, pointcoord2

1. While zone no is less than or equal 3

   Show original image side by side with enhanced image

   Show message for expert to mark at least 5 points around the identified edges of the section in the zone of the prostate gland within the image,

   Prompt user to enter (0 – 3) suspected cancerous sections in current zone

   If no of section = 0

   Set the area count for that zone =0; Go for another zone

Else

2. While sectn count is not 0

Prompt user to mark points around the section no within the gland

Capture the points in an array

If points array is empty, expert made no marks,

Display message to alert expert no marks were made

Return to 2

End

Prepare the array in a table for processing, Display processing message

Determine Seed/Centre point within the points marked as section in zone area

Connect the edges points marked in zone and store edge table

Accumulate all pixels of the entire gland surface bound by edge points generated in the table as Area of expert segmentation, Am

Sectn count decrease by 1

Endwhile sectn count

Endif

Zone no. increase by 1

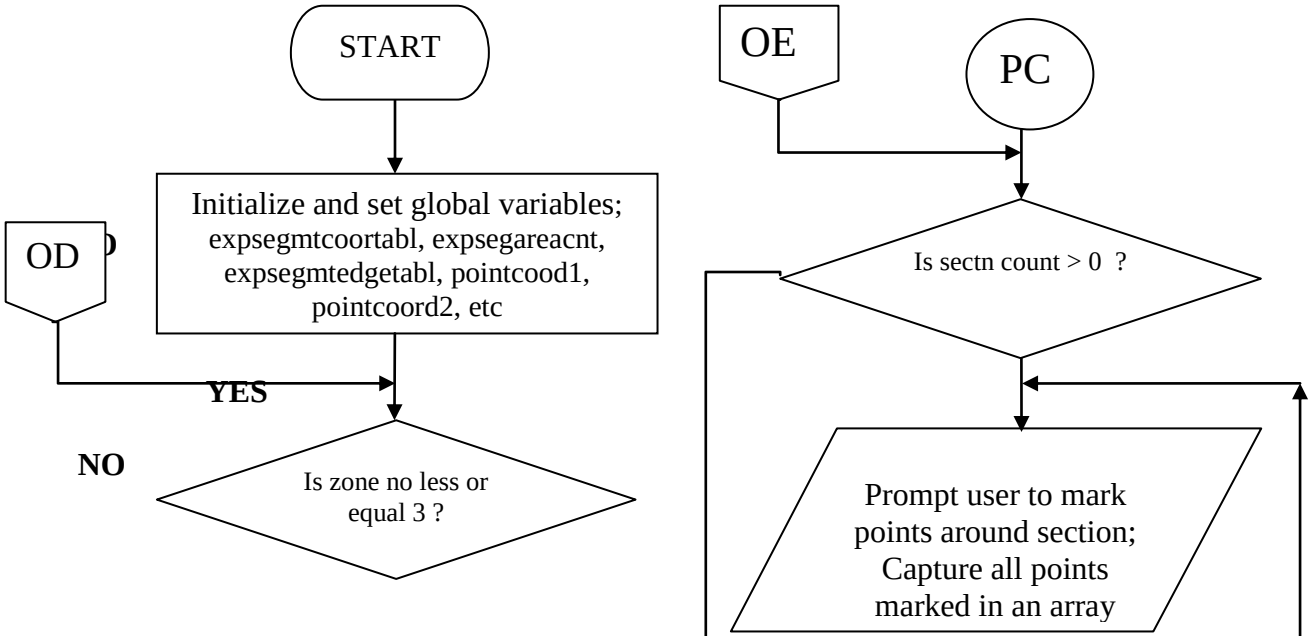
Endwhile zone no.

Set these pixels to highest intensity to differentiate section in zone of gland

in the image and create a zones marked image file for display and store.

3. Stop.

The corresponding flowchartsare shown in Figures3.19 and 3.20.



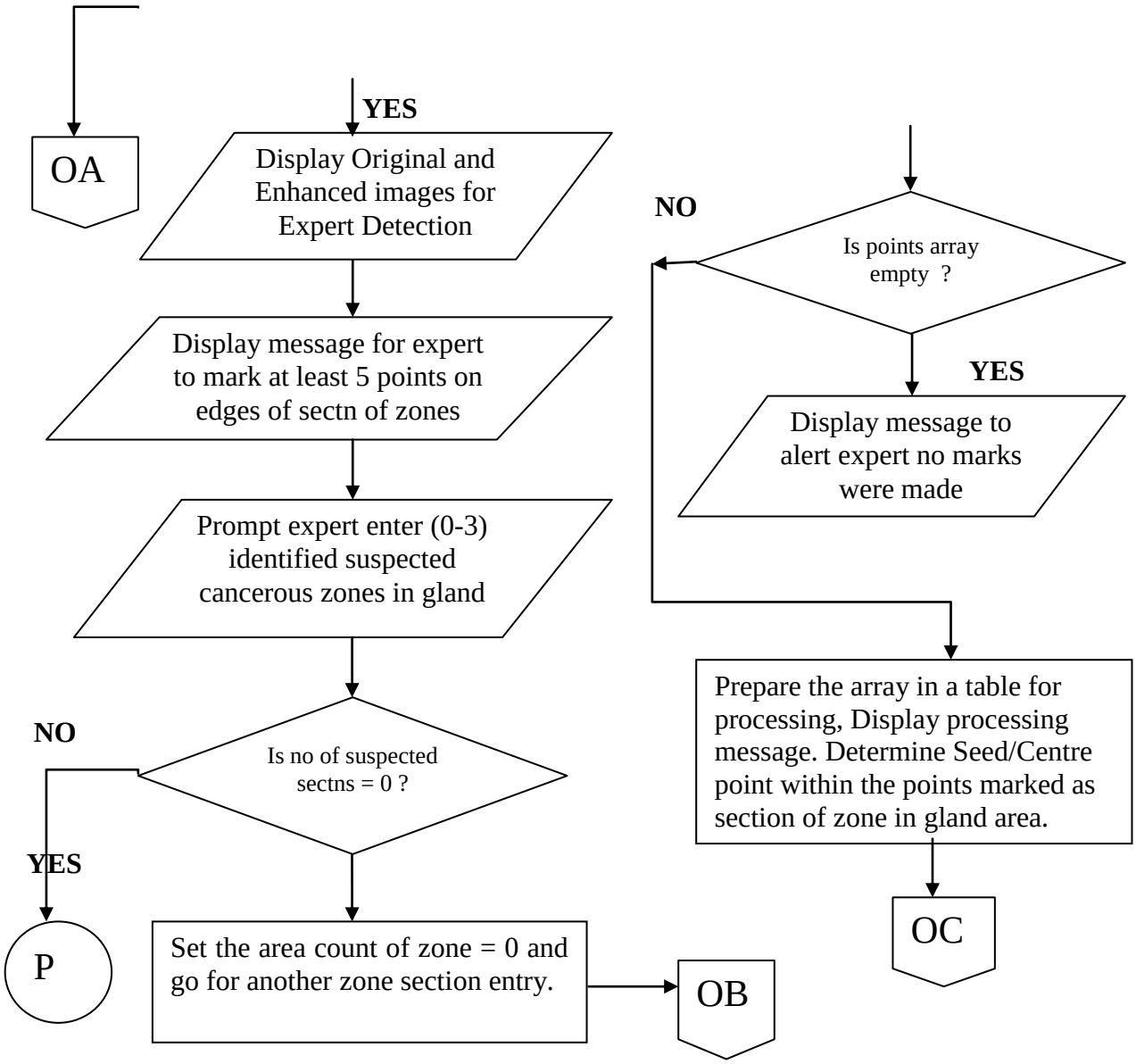
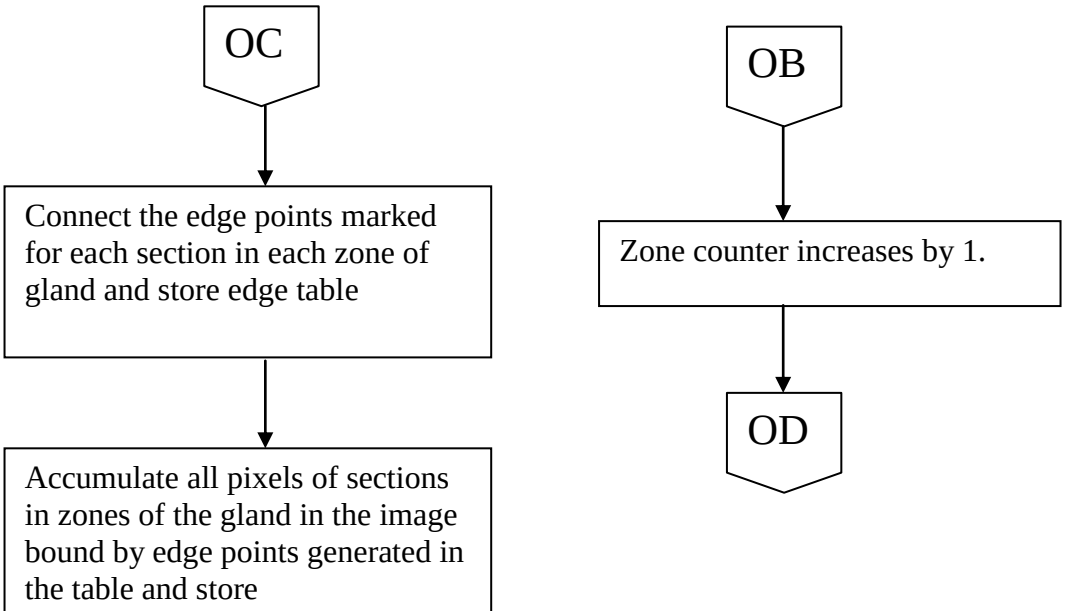


Figure 3.19: Expert gland zones cancerous cell detection algorithm flowcharts (1)



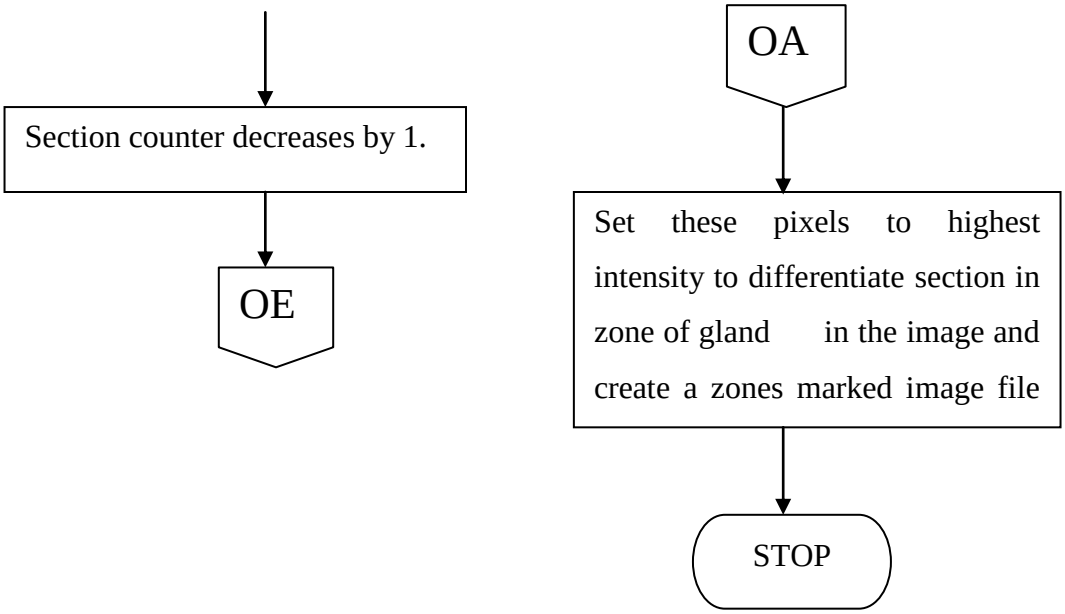


Figure 3.20: Expert gland zones cancerous cell detection algorithm flowcharts (2)

3.6.7 Design of System Result Validation

The design here facilitates the realisation of research objective vi. It implements option three of the main menu. The design, in Figure 3.21 shows steps to be taken when a particular image has been segmented both by the new system and the human experts. The algorithm enables the user to select the image whose algorithm segmentation result was compared with expert result. If the image selected has both segmentation results stored, the comparison starts, otherwise, an error message was displayed accordingly and system returns control to the main menu.

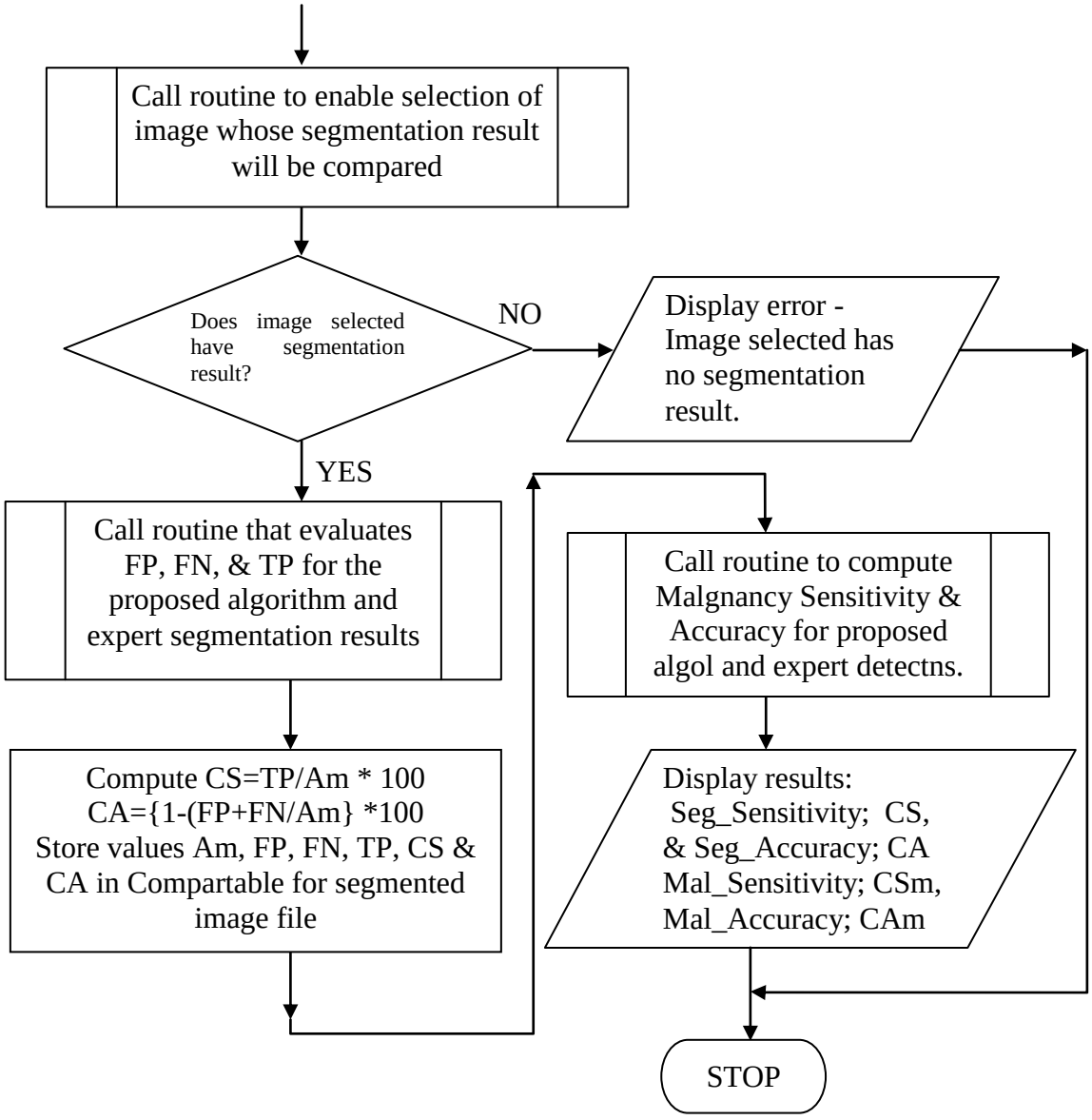


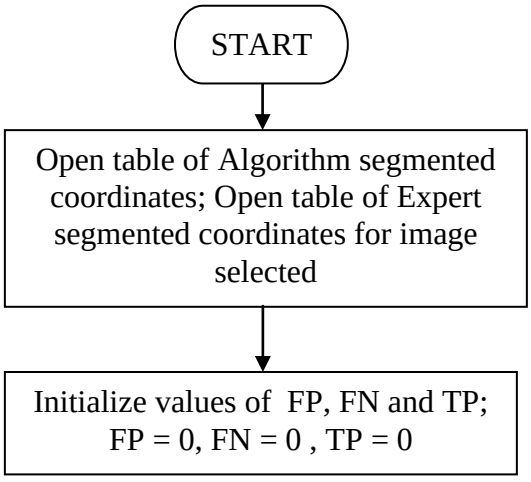
Figure 3.21: Flowchart/Algorithm for comparing segmentation results

The validation/comparison implements the area based metrics described in section 3.5.2. The algorithm calls a subroutine, shown in Figure 3.22 that evaluates FP, the area inside the proposed algorithm result but outside the expert manual result, FN, which is represented by the area inside the expert manual result but outside proposed algorithm result, and TP, which is the common area between both expert manual and the proposed algorithm results. The vales of sensitivity CS and accuracy, CA given by equations 3.1 and 3.2 respectively are computed for the segmentation of prostate gland image. Another subroutine, shown in Figure 3.21 which computes the sensitivity CSm and accuracy Cam for the detection of malignant tissues in the prostate image by the algorithm

and expert manual results. The values generated for the current segmented image validation/comparison is stored in appropriate tables to be referenced later.

Pseudo-code for the sub-module in the inference engine that is responsible for comparing expert and proposed systems result are as follows:

1. Call subroutine to enable selection of image whose segmented results will be compared
   IF image selected have no segmentation results
      Then Display error – "Image selected has no segmentation result"
      Go to 2
   Else
      Call subroutine that evaluates the segmentation parameters for sensitivity,
         and Accuracy for proposed algorithm and expert results, (TP, FP, FN & Am)
      Compute, Specificity, CS = TP/Am * 100, Accuracy,
                  CA = {1-(FP+FN/Am} * 100
      Store values of Am, FP, FN, TP, CS, CA in a table, Compartbl, for segmented image
      Call subroutine to evaluate/compute the malignancy parameters & sensitivity,
         and Accuracyproposed algorithm and expert results, (TP, FP, FN & Am)
      Display ;"Segmentation of Image:"; "Sensitivity = "; CS, ; Accuracy = "; CA
      "Malignancy detection on Image:"; "Sensitivity = "; CSm,; Accuracy = "; CAm
   Endif
2. Stop



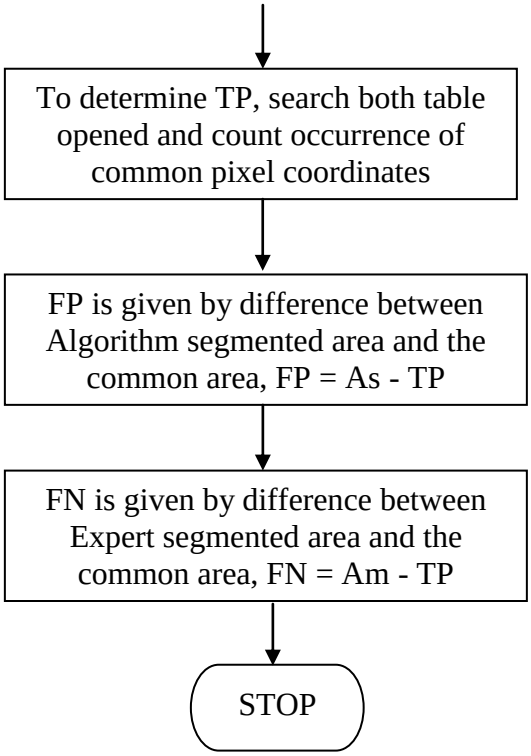


Figure 3.22: Flowchart/algorithm for evaluating FP, FN and TP for segmented images

This subroutine shown in Figure 3.22was called from the main algorithm of Figure 3.21. Its design was to compute the possible difference in surface areas of the algorithm segmented prostate image and expert manually segmented prostate image. It searches through the tables containing the pixel coordinates of the segmented image files. The counts of pixels satisfying the conditions for determining values of FP, FN and TP shown pictorially in Figure 3.2 and defined in section 3.5.2 are kept. The count for the common area, TP, was obtained, while the other values, FP and FN were computed from already generated and stored values of areas of algorithm segmented image, As and expert segmented image, Am. The pseudo-codes for the algorithm described are as follows:

1. Open the tables containing the coordinates of segmented image by the proposed algorithm and expert segmented coordinates for selected image file.

   Initialize the values of the parameters, TP = 0, FP = 0, FN = 0

   Determine TP by searching both tables and counting occurrences of common coordinates, that

gives the common area by both expert and proposed algorithm segmentation result

Determine FP given by the difference between the proposed algorithm segmented area, As, and TP; FP = As - TP

Determine FN given by the difference between the expert segmented area, Am, and TP; FN = Am - TP

Stop

The algorithm described in Figure 3.23is a subroutine called from algorithm in Figure 3.21. It performs comparison of malignant tissue detection by proposed algorithm and expert manual results. It also implements the area based metrics described in section 3.4.1 for validating the result of system. First the table containing the coordinates of the malignant pixels generated from proposed algorithm and captured for expert manual detection is opened. The tables are checked for at least one record of malignant tissue coordinate. Messages are displayed to indicate No malignant tissues detected for either or both methods. When one or both tables have records the parameters are computed for comparing performance of methods. Use Malignant coordinates tables for algorithm detection to determine tissue zones from lookup table and store accordingly.

Compute Am represented by the total number of pixel coordinates detected by expert as having malignant tissues & As, represented by the total number of pixel coordinates detected by algorithm as having malignant tissues. Determine TP, represented by number of common zones detected by both methods. Determine FP, represented by number of coordinates or zones detected by algorithm alone and FN, represented by number of coordinates or zones detected by expert manual alone. The values of sensitivity CSm and accuracy, CAm given by equations (3.1) and (3.2) respectively are computed for the detected malignant tissues of the prostate gland.

The pseudo-codes for this algorithm are as follows:

1. Open malignant table coordinates for algorithm cancer detection

   IF No record of coordinates by algorithm

   Then Display ; “No malignant tissues detected by Algorithm”

   Endif

   IF No record of coordinates by experts detection

Then Display ; “No malignant tissues/pixels detected by Expert results”

Endif

Use malignant coordinates table for algorithm detection to determine the gland tissue zones from lookup table and store same in tables.

Open SysMalgzonfl.mat, Open ExpMalgzonfl.mat

Compute Am & As areas in counts of detected zones for expert and proposed algorithm respectively. Removing redundancy in zones count

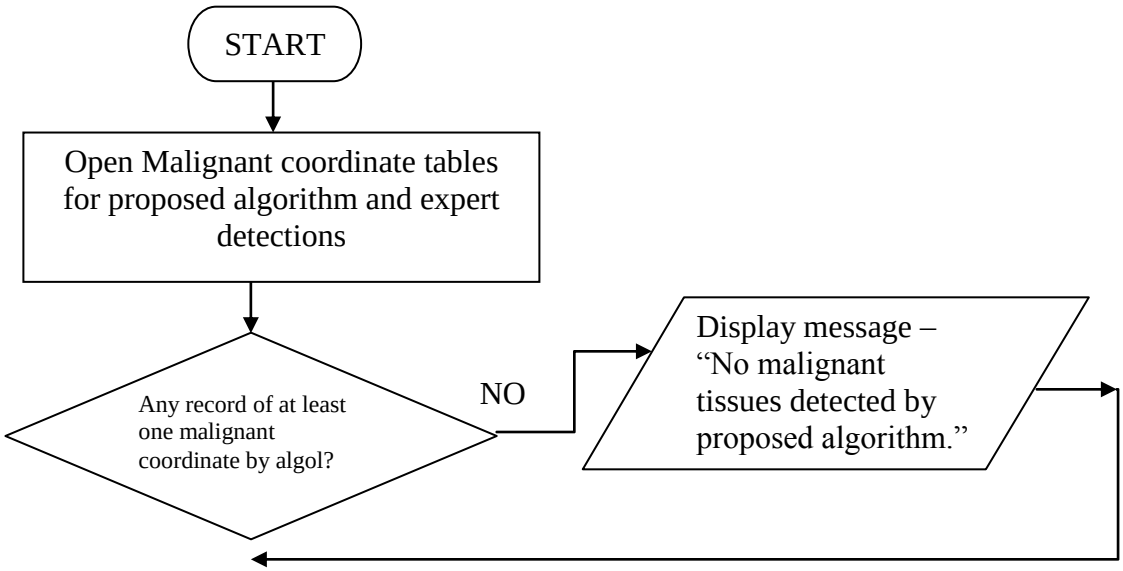
Determine TP, count of common zones detected by both expert and proposed algorithm

Determine FP, zones detected by proposed algorithm alone

Determine FN, zones detected by expert results alone

Compute Sensitivity, CSm = TP/Am * 100; Accuracy, CAm = {1-(FP + FN)/Am * 100}

15. Stop



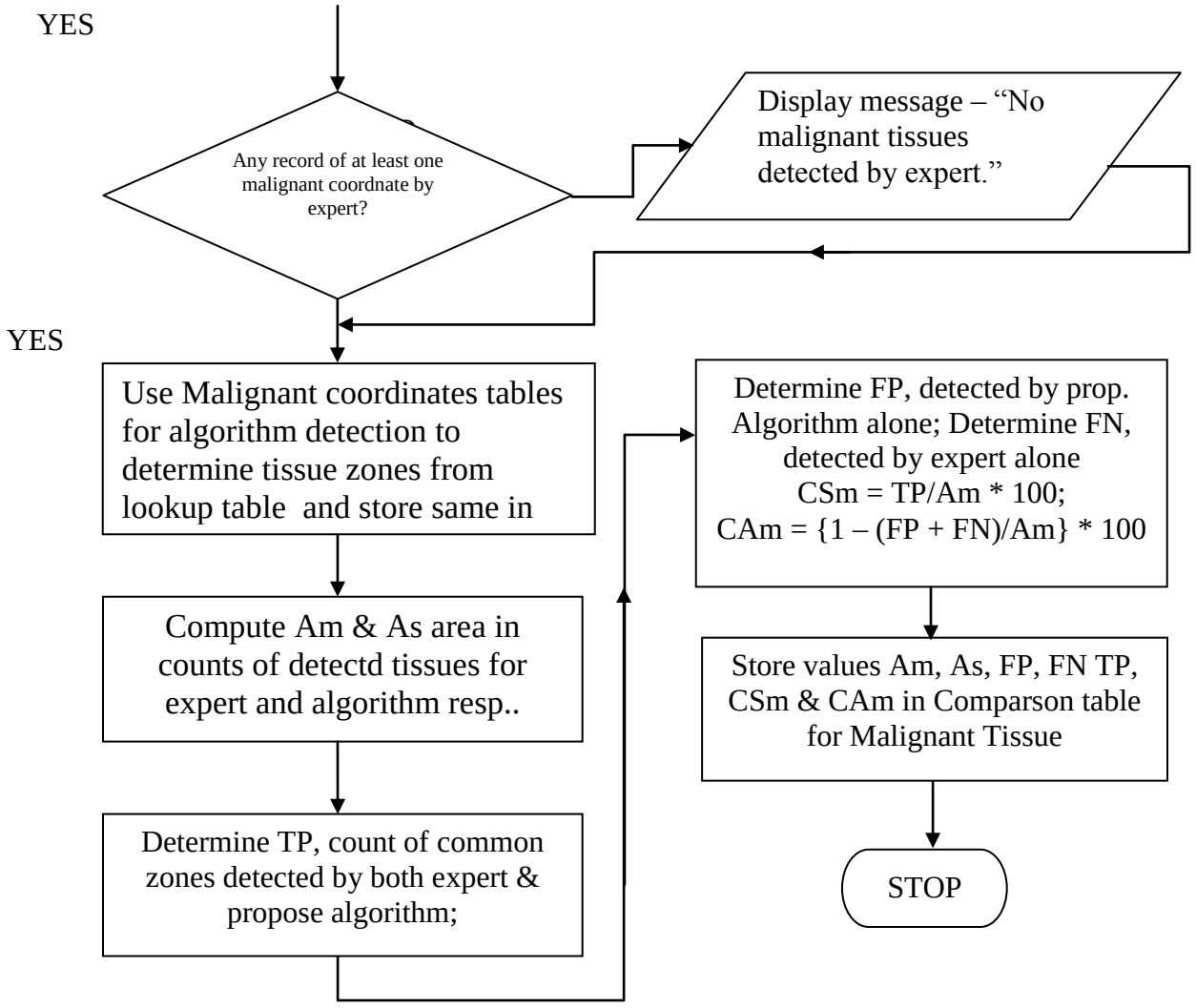


Figure 3.23: Flowchart/Algorithm for comparing malignant tissue detection results

CHAPTER FOUR

RESULTS AND DISCUSSION

In this chapter the software and hardware environment for implementing the systems designed in chapter three of this dissertation are presented. The test samples and test results showing algorithm segmented images alongside the expert segmented images are also presented and discussed. Results

for validating theimplemention of the system/algorithm using area-based metrics for segmentation and cancer detection are presented and discussed.

4.1 Software Operating Environment

The codes for the algorithm were written using MATLAB programming language. The codes for the various modules and routines are listed in appendix A. This implementation of this set of software requires MATLAB version 7.5 or higher version running on WINDOWS 7 or higher version of windows.

4.2 Hardware Environment

The minimum hardware requirement for the implementation of this work is as follows:

Processor : 2.0GHz, 64-bit Operating System and above.
Installed Memory (RAM) : 2.00GBytes and above.
Hard drive capacity: 200 GBytes and above

4.3 System Implementation

The various modules designed were implemented in the MATLAB environment with MATLABprogramming scripts/codes following the procedure described in section 3.5. The implementation which was handled module by module and integrated using the matlab function call features shall be briefly described.

4.3.1 User Interface/Explanation Subsystem Implementation

This subsysytem designed in section 3.6.4 with system main menu in Figure3.5 and corresponding flowchart of Fig. 3.6 was implemented by coding the instructions accordingly to produce the control menu and organize the graphic user interface to display the necessary user messages/explanation of the result of the processing steps as the system responds to user's requests.

The implementation script for this section is listed in Appendix B as a function namedANALYZ_ANALPROSM.

4.3.2 Image Acquisition Implementation

Some online images used in a previous work by Awad (2007) were downloaded and stored in memory as JPEG (.jpg) format digital images using Microsoft Photo Viewer to review and resize the image accordingly. We note that other digital image formats were considered but JPEG format was chosen because it occupies less memory and amenable to grey level conversion for image analysis and processing required for the algorithm.

4.3.3 Image Selection Implementation

This submodule was implemented by coding the algorithm and flowchart shown in Figure3.7 using Matlab scripts as designed in section 3.6.4.3. The codes written for thid submodule is a function called from within the main function, ANALYZ_ANALPROSM. It is listed in Appendix B as the function named, ANALYZ_IMAGESELC.

4.3.4 Image Preparation Implementation

This submodule was implemented by coding the algorithm and flowchart shown in Figure3.8 using Matlab scripts as designed in section 3.6.5.1. The codes written for this submodule is a function

called from within the main function, ANALYZ_ANALPROSM. It is listed in Appendix B as the function named, ANALYZ_REMVLABL.

4.3.5 Image Enhancement Implementation

This submodule was implemented by coding the algorithm using Matlab scripts as designed in section 3.6.5.2. The codes written for this submodule is a function called from within the main function, ANALYZ_ANALPROSM.m. It is listed in Appendix B as the function named, ANALYZ_SEQSTICK.

4.3.6 Dominant Intensity Parameter Determination Implementation

This submodule was implemented by coding the algorithm and flowchart shown in Figure3.9A and 3.9B using Matlab scripts as designed in section 3.6.5.3. The codes written for this submodule is a function called from within the main function, ANALYZ_ANALPROSM. It is listed in Appendix B as the function named, ANALYZ_MALGRANG.

4.3.7 Seed Point Localization Implementation

This submodule was implemented by coding the algorithm and flowchart shown in Figure3.10 using Matlab scripts as designed in section 3.6.5.4. The codes written for this submodule is a function called from within the main function, ANALYZ_ANALPROSM. It is listed in Appendix B as the function named, ANALYZ_SEEDLOCN.

4.3.8 Segmentation by Enhanced Region Growing and Malignant Tissue Detection Implementation

This submodule was implemented by coding the algorithms and flowcharts shown in Figures3.12A, 3.12B, 3.12C, 3.12D and 3.12E using Matlab scripts as designed in section 3.6.5.5. The codes

written for this submodule is a function called from within the main function, ANALYZ_ANALPROSM. It is listed in Appendix B as the function named, ANALYZ_IMAGEANAL.

Other subfunctions called from within the function, ANALYZ_IMAGEANAL, that were designed in flowcharts of Figures 3.13A, 3.13B, 3.13C, 3.13D, 3.13E and 3.13F to ensure scanning in rowwsie and columnwise radial directions with their corresponding algorithmswere implemented with the Matlab scripts listed as functions named ANALYZ_SCANROWSE andANALYZ_SCANCOLWSE. The subfunction for storing the grown and detected region pixel coordinates designed in Figs. 3.13G and 3.13H are listed as ANALYZ_ACUMPXSGCD.

The subfunction that displays the result of segmentation and detection designed in Figure3.14 and corresponding algorithm is implemented with matlab scripts listed as the last line of codes in the function named ANALYZ_IMAGEANAL, listed in Appendix B.

4.3.9 Expert Main Menu Implementation

This main module for expert marking of gland segments was implemented by coding the algorithms and flowcharts shown in Figures 3.16 and 3.17 using Matlab scripts as designed in section 3.6.6.1. The codes written for this module is listed in Appendix B as the function named, EXPERT_ANALPROSM.

4.3.10 Expert Detect Edges/Mark Suspected Cancerous Zones Implementation

This submodule was implemented by coding the algorithms and flowcharts shown in Figures 3.18, 3.19 and3.20, using Matlab scripts as designed in section 3.6.6.2. The codes written for this submodule is a function called from within the main function, EXPERT_ANALPROSM. It is listed in Appendix B as the function named, EXPERT_IMAGANALYS.

Other subfunctions called from within the function, EXPERT_IMAGANALYS, that facilitates expert marking of segments and sections of suspected cancerous zones are implemented with the Matlab scripts listed as functions named EXPERT_GLANDSEGMTG and EXPERT_GLANDMALGZN.

4.3.11 System Result Validation/Comparison Implementation

This submodule was implemented by coding the algorithms and flowcharts shown in Figures 3.21, 3.22, 3.23and 3.24, using Matlab scripts as designed in section 3.6.7. The codes written for this submodule is a function called from within the main function, ANALYZ_ANALPROSM. It is listed in Appendix B as the function named, ANALYZ_IMAGCMPARE.

4.4 System Testing

The various modules designed and implemented with MATLAB code were tested module by module to verify that their performance was according to design. Each module that was tested were debugged and retested with relevant input parameters. The modules were integrated and tested with image samples.

4.4.1 Test Samples

Randomy selected samples previously used in the work by Award (2007) were chosen for testing the performance of this present work. It was considered appropriate since the samples satisfy the scope of this work. Moreover the samples have proved to be amenable to digital image analysis. Figure 4.1 shows some samples of original TRUS 2D-images used in this work.

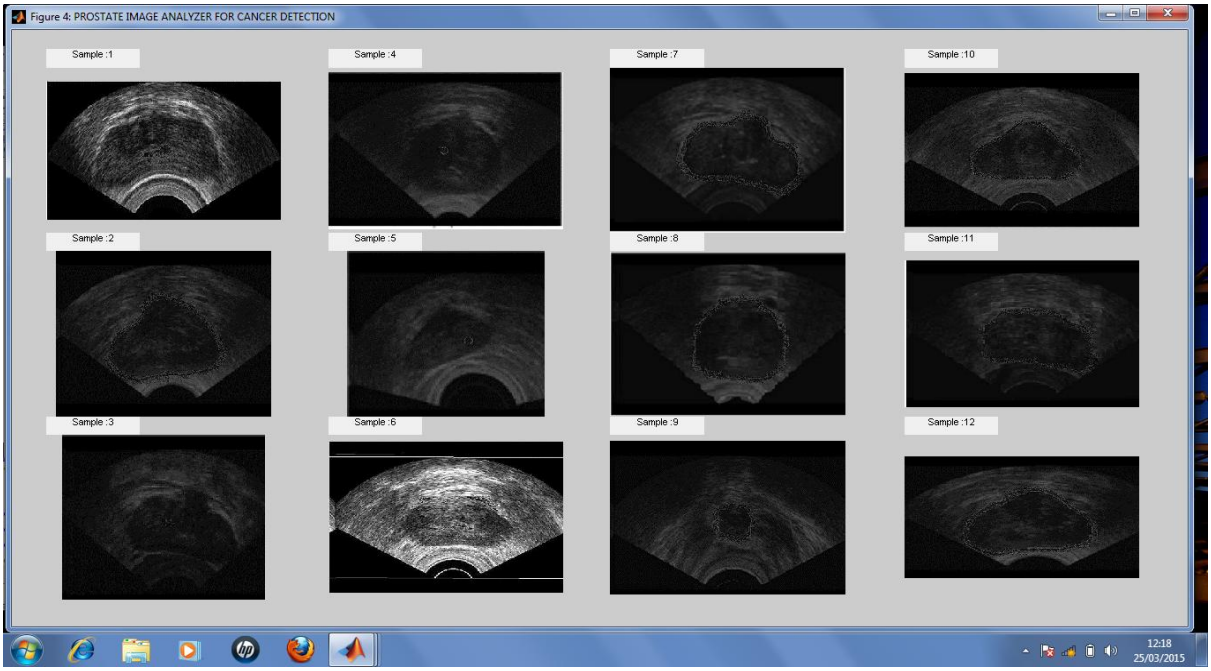


Figure 4.1: Samples of un-segmented original TRUS 2D-images of the prostate gland

4.4.2 Test Results by Modules

All modules were tested and those modules that have results that can be displayed are here presented. The rest of the results are shown by the test result of integrated algorithm.

4.4.2.1 Test Results by Modules - User Interface/Explanation Subsystem

The results of implementing the user interface/explanantion subsystem for both the system and expert menus are presented in the screen shots shown in Figures4.1A and 4.1B.

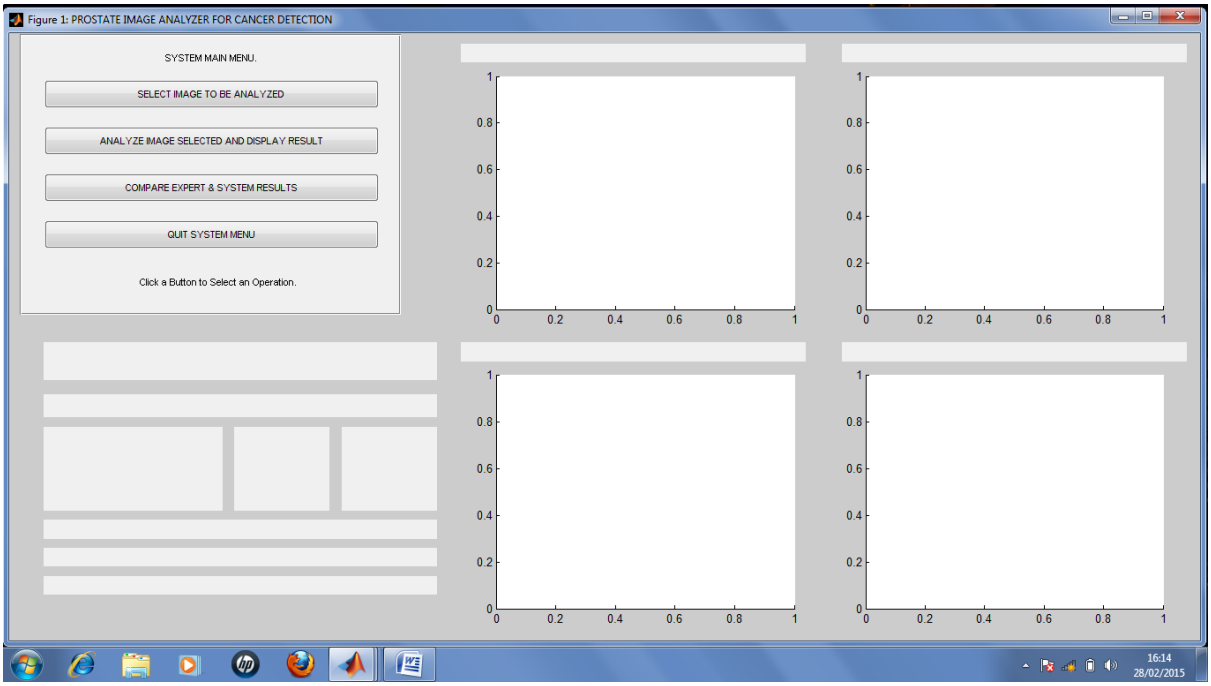


Figure 4.1A:System main menu screen shot

The screen shots in Figures 4.1A and 4.1B shows six sections of the opening screen of the Prostate image analyzer for cancer detection. The first section, left top has the system main menu having four options as designed. The second section, middle top is an axes created for displaying the image selected for analysis. The third section, right top is also an axes created for displaying output of image enhancement. The fourth section, left down has partitions for created for displaying the result of image analysis/detection for zones of suspected cancer tissues. The fifth section, middle down is an axes created for displaying output axial/radial segmentation of prostate image sample. The last section, right down is also an axes created for displaying the output of detection of zones marked as containing possible cancerous tissues.

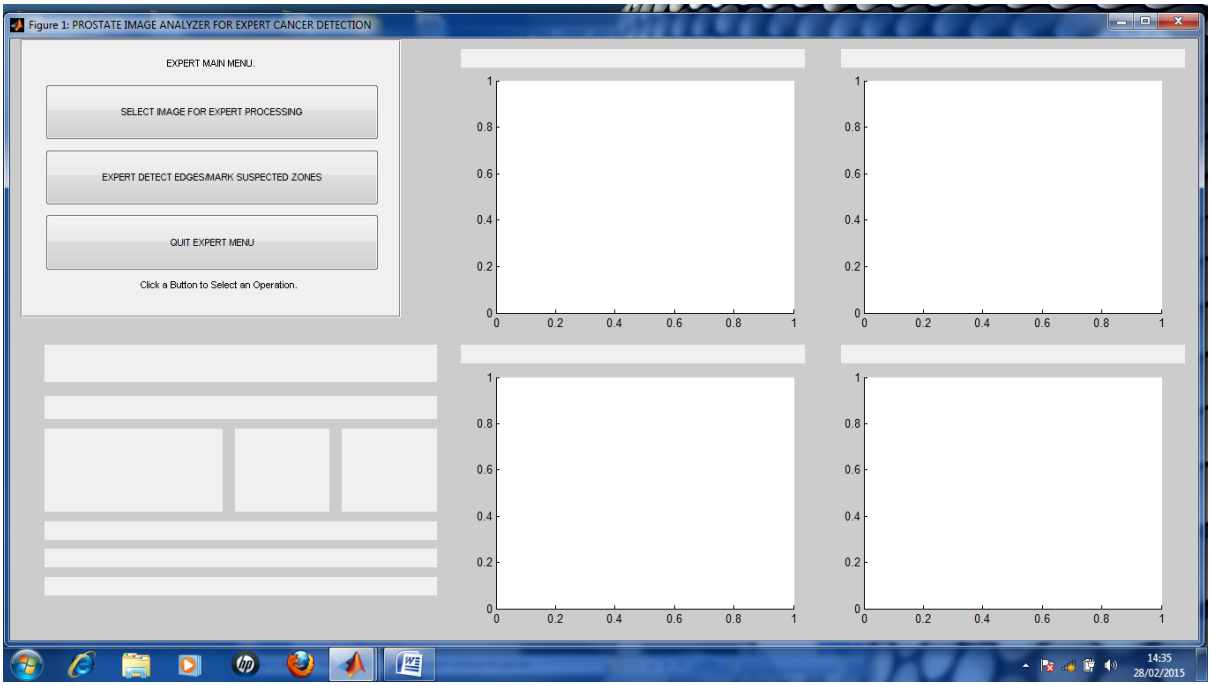


Figure 4.1B: Expert main menu screen shot

4.4.2.2 Test Results by Modules - Image Selection

The results of implementing the image selection subsystem for both the system and expert operations are presented in the screen shots shown in Figures 4.2A and 4.2B. The screen shots show a pop-up menu containing samples of images loaded into the subdirectory PROSTATEFILES in the MATLAB working directory. The images shown in the pop-up menu includes sample of original, segmented, detection, expert segmentation and detection image files. The user/expert was prompted to select the original image sample desired for analysis using the algorithm/expert skills.

4.4.3 Test Results of Integrated Algorithm

The modules were integrated and when the system menu was displayed and option 2 of the system main menu was selected the test result are presented in two stages, namely – intermediate processing, completed processing. Appendix A shows the screen shots of integrated algorithm results.

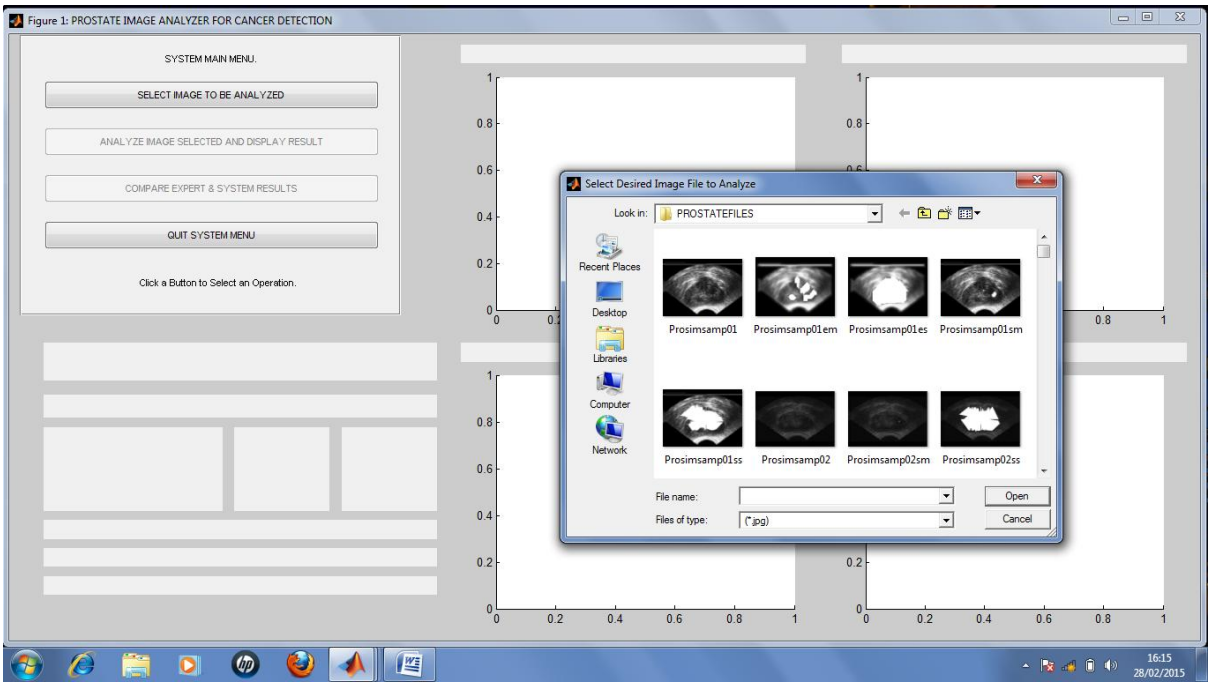


Figure 4.2A:System image selection process screen shot

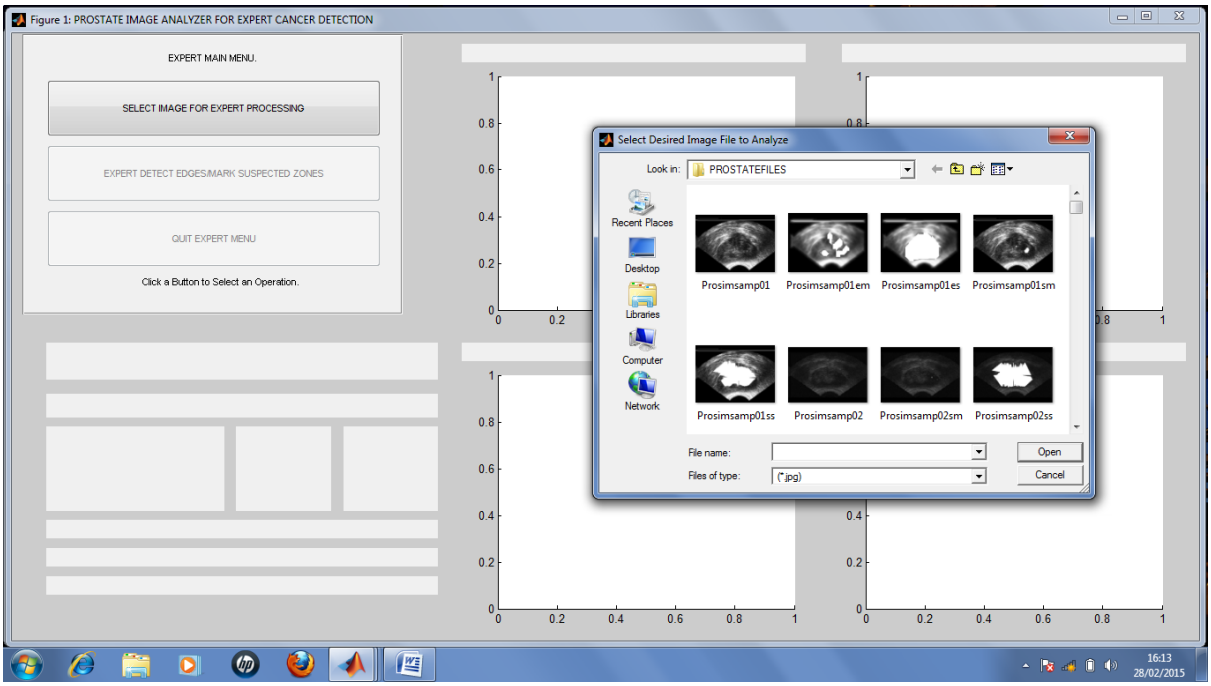


Figure 4.2B: Expert image selection process screen shot

4.4.3.1 Intermediate Processing Results of Algorithm - System

The imtermediate testing results captured and presented include the localization of seed point and image enhamcement along with the processing explanation messages implemented in sections 4.3.7 and 4.3.8 respectively. The screen shots shown in Figure 4.3A shows Figure 4.1Amodified. The second section, middle top have the original image selected for analysis displayed in the axes with the seed point located at the center shown as a cross at the center of the image. The third section, right top has output of image enhancement displayed in the axes. Also in the first line of the fourth section a system process message is displayed, showing what process is going on as at the time the screen shot was captured. The first section, left top has the second item in the menu highlighted – '**ANALYZE IMAGE SELECTED AND DISPLAY RESULT'**.

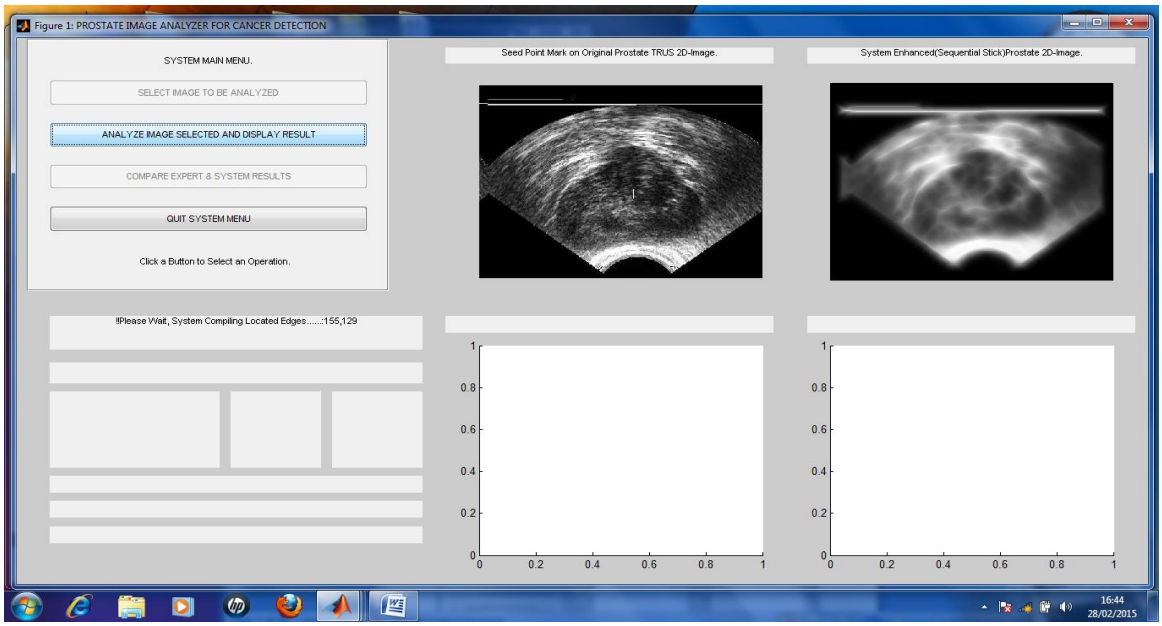


Figure 4.3A:System image segmentation/analysis process screen shot

4.4.3.2 Intermediate Processing Results of Algorithm - Expert

The imtermediate testing results captured and presented include the localization of seed point and image enhamcement along with the processing explanation messages implemented in sections 4.3.9 and 4.3.10 respectively. The screen shots shown in Figure 4.3B shows Figure 4.1Bmodified. The second section, middle top have the original image selected for analysis displayed in the axes. The

third section, right top has output of image enhancement displayed in the axes. Also in the first and second lines of the fourth section contains expert processing instruction/message as displayed, showing what expert should do as the processing was going on as at the time the screen shot was captured. The first section, left top has the second item in the menu highlighted – '**EXPERT DETECT EDGES/MARK SUSPECTED ZONES**'.

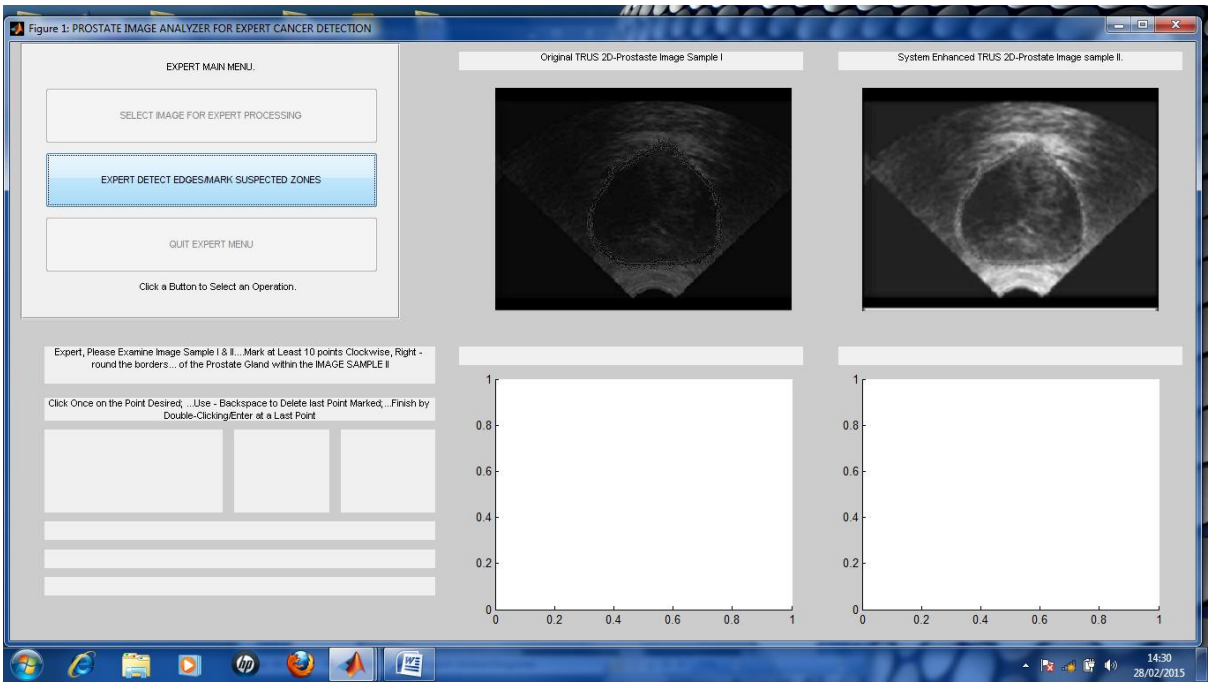


Figure 4.3B: Expertimage segmentation/detection marking process screen shot

4.4.3.3 Completed Process Results of Algorithm - System

Different samples of images were used to test the completed segmentation algorithm as well as the detection of suspected cancerous zone algorithm to ensure that the system was capable of handling the images of varying complexity.

The results of testing the system waspresented by showing the original image, enhanced image, segmented image and the image marked with suspected cancerous sections. It shows screen shots for algorithm segmentation and detection of suspected cancerous zones within region of the

segmented prostate gland. Each screen shot result shows a table of summary segmentation/detection result of the form:

Zone of Gland	No of Affected Pixels	% Affected Pixels
CENTRAL (1st Quad)	NNNN	NN
TRANSITION (4th - 1st Quad)	NNNN	NN
PERIPHERAL (2nd – 4thQuad)	NNNN	NN

TOTAL PIXEL IN SUSPECTED AREA ……..……:NNNN

TOTAL PIXEL IN THE SEGMENTED GLAND REGION ..……..:NNNN

PERCENTAGE IN SUSPECTED AREA ……..:NN%

This form was shown on the second line, fouth section, first left down, of the screen. Here are screen shots of the proposed algorithm results for some of the samples of images segmented and marked for suspected cancerous zones are shown in the Figures 4.4A to 4.4D.

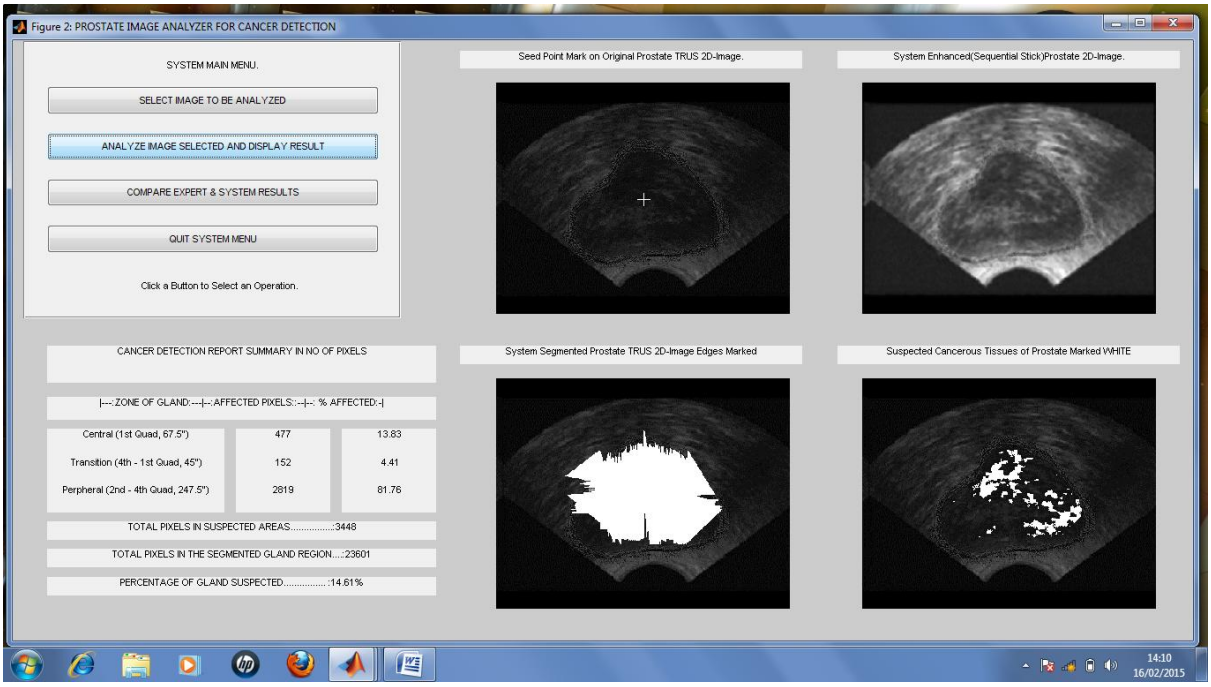


Figure 4.4A: Sample 05- algorithm segmentation and detection result

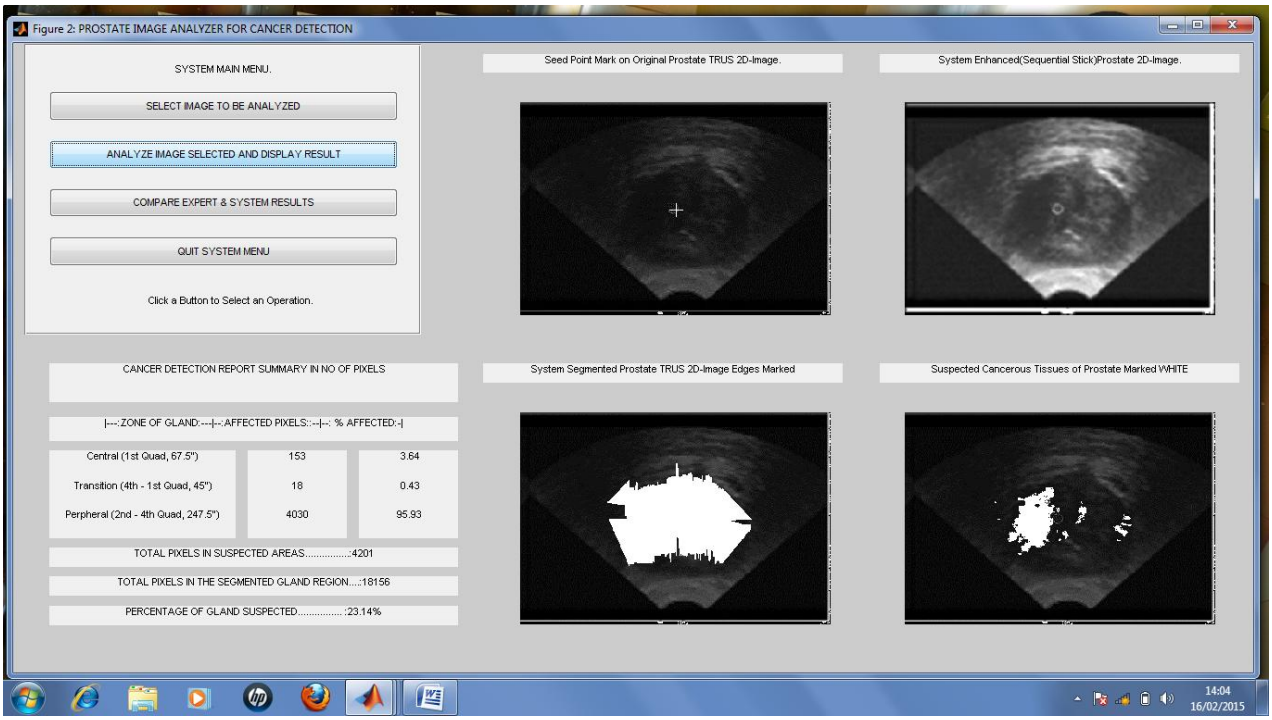


Figure 4.4B: Sample 07- algorithm segmentation and detection result

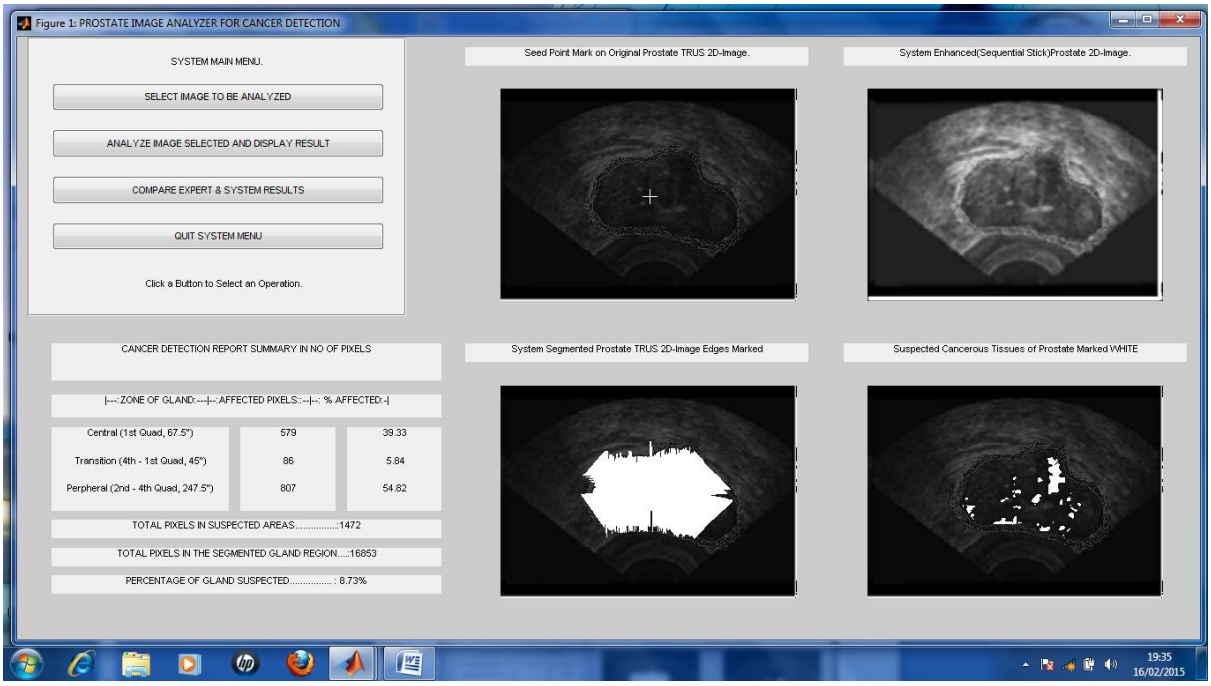


Figure 4.4C: Sample 11- algorithm segmentation and detection result

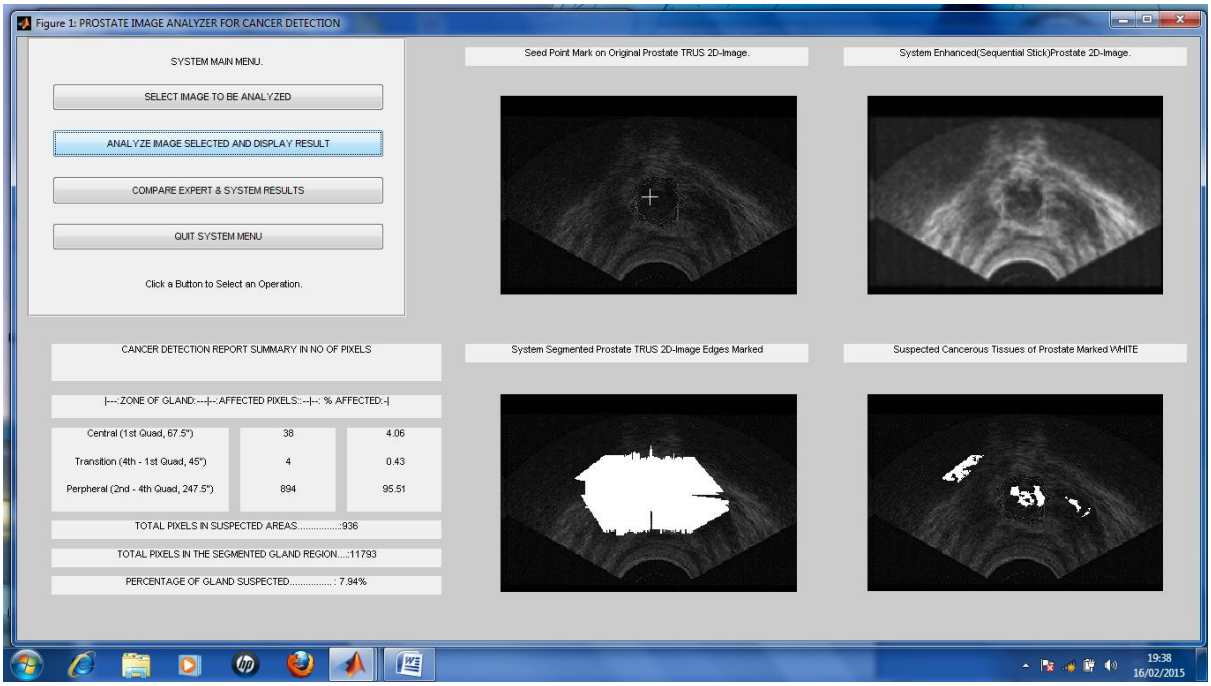


Figure 4.4D: Sample 13- algorithm segmentation and detection result

The first image on the first row shows the original sample with seed point mark, the second image on the first row shows the enhanced sample image, the first image on the second row shows the resulting image after segmentation by the proposed algorithm with the segmented prostate gland section shown in highest intensity, and the second image on the second row shows the result of cancer detection by the proposed algorithm. Also the highest intensity, white section (hyperechoic), shows the suspected cancerous cells/portions of the prostate gland sample under investigation.

The results reveal that the algorithm was weak in that it segments the edges of images of minimum axial diameter of one hundred pixels. It was robust in that it was able to detect specific tissue pixels suspected to be cancerous to a great degree.The content of the summary of segmentation and detection in the samples showing no of pixels in areas for several images were accumulated and listed together in Table 4.1. The record in each row of Table 4.1 was extracted from the summary of segmentation/detection for the image sample, listed in the column named 'SAMPLE NO.'. The other columns show; the area of gland segmented, the area of gland suspected to be cancerous, the area of gland suspected to be cancerous within the Transition, Central and Peripheral Zones.

4.4.3.4 Completed Process Results of Algorithm - Expert

Different samples of images were used to test the expert segmentation algorithm as well as the detection of suspected cancerous zone to ensure that the expert was capable of handling the images of varying complexity. The results of testing the expert marking was presented by showing the original image, enhanced image, segmented image and the image marked with suspected cancerous sections. It shows screen shots for expert segmentation and detection of suspected cancerous zones within region of the segmented prostate gland.

Each screen shot result shows a table of summary of segmentation/detection result of the form:

Zone of Gland	No of Affected Pixels	% Affected Pixels
CENTRAL (1^{st} Quad)	NNNN	NN
TRANSITION (4^{th} - 1^{st} Quad)	NNNN	NN
PERIPHERAL (2^{nd} – 4^{th}Quad)	NNNN	NN

TOTAL PIXEL IN SUSPECTED AREA ……..……:NNNN

TOTAL PIXEL IN THE SEGMENTED GLAND REGION ..……..:NNNN

PERCENTAGE IN SUSPECTED AREA …….:NN%

This form was shown on the second line, fouth section, first left down, of the screen. Here are screen shots of the proposed algorithm results for some of the samples of images segmented and marked for suspected cancerous zones are shown in the Figures 4.5A to 4.5D.

Table 4.1 System gland segmentation/cancer detection summary

SAMPLE NO.	AREA OF GLAND SEGMENTED (PIXELS)	AREA OF GLAND TOTAL (PIXELS)	ZONES SUSPECTED CANCEROUS. TRANSITION (PIXELS)	CENTRAL (PIXELS)	PERIPHERAL (PIXELS)
2	17817	16	0	0	16
3	13199	502	0	0	502
4	13132	646	120.4	180.6	345
5	23601	3448	251.6	377.4	2819
6	17705	4955	436.8	655.2	3863
7	18156	4201	68.4	102.6	4030
8	13218	852	195.6	293.4	363
9	24980	10817	874.4	1311.6	8631
10	16565	0	0	0	0
11	16853	1472	266	399	807
12	16668	6342	875.2	1312.8	4154
13	11793	936	16.8	25.2	894
14	19036	1217	181.6	272.4	763
15	13269	1179	68.4	102.6	1008
16	14320	1822	138.8	208.2	1475
17	13418	911	211.2	316.8	383
18	16114	2083	299.6	449.4	1334
19	20163	9893	384.8	577.2	8931
21	28287	672	45.6	68.4	558
22	18818	548	24.8	37.2	486
23	15531	2115	340	510	1265
27	23744	3332	464.4	696.6	2171
29	16310	2705	354.4	531.6	1819

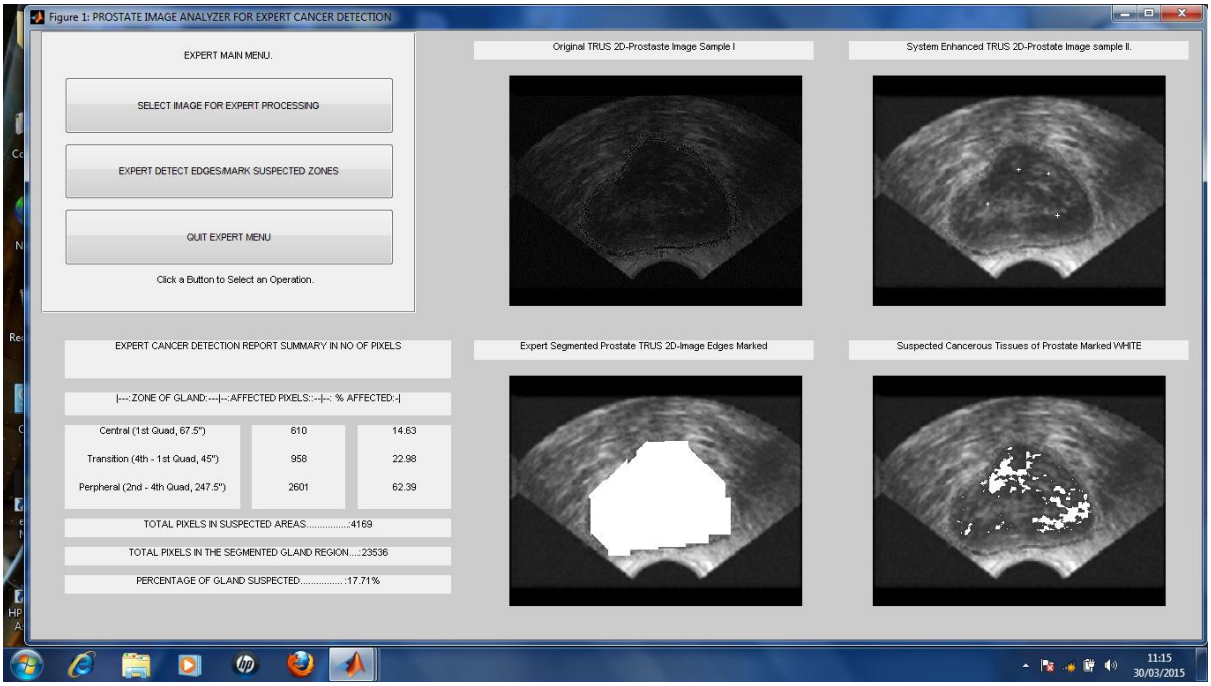


Figure 4.5A: Sample 05- expert segmentation and detection result

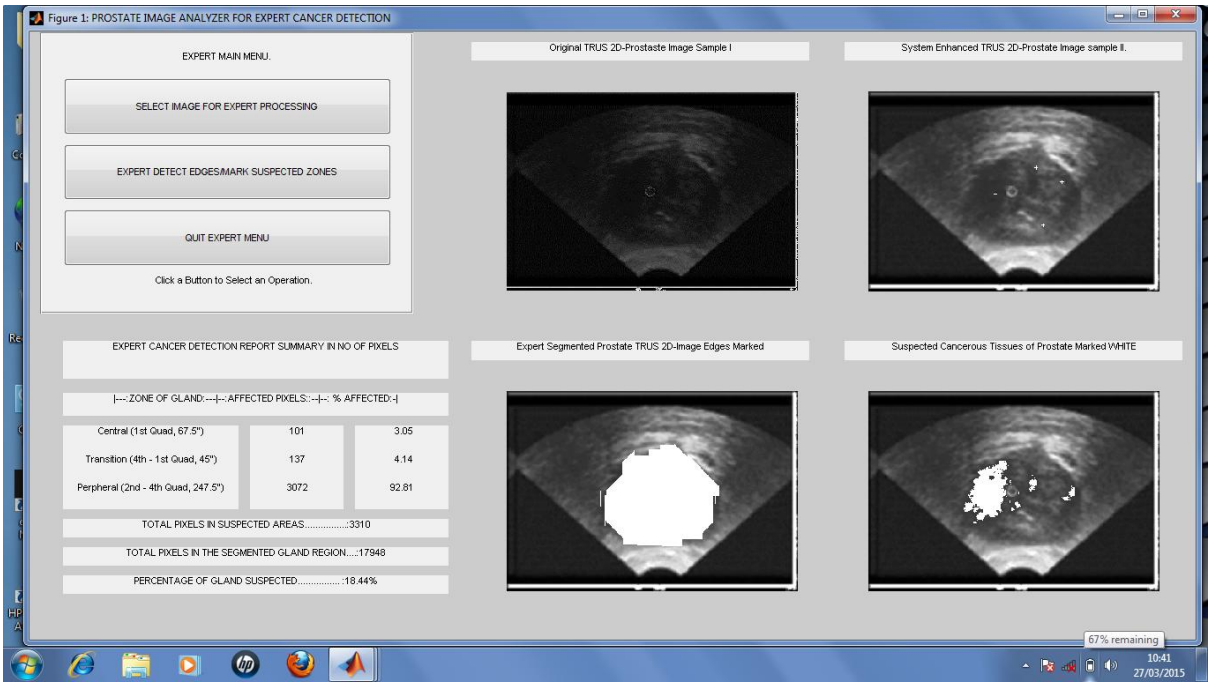


Figure 4.5B: Sample 07- expert segmentation and detection result

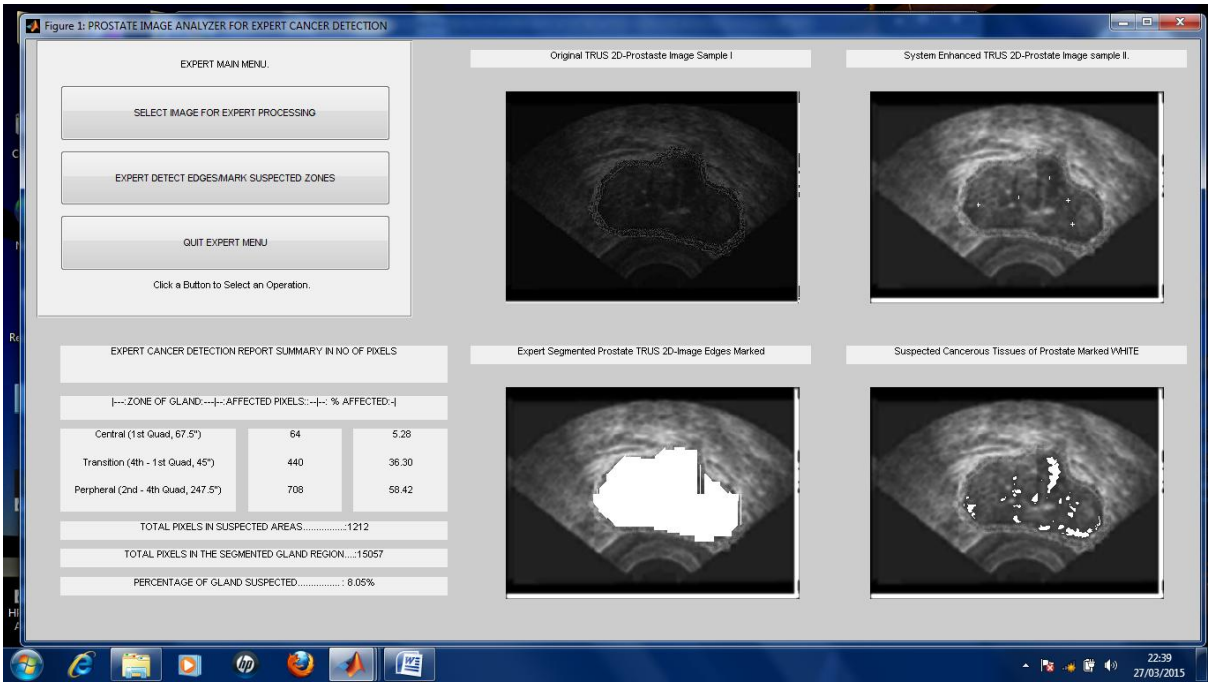


Figure 4.5C: Sample 11- expert segmentation and detection result

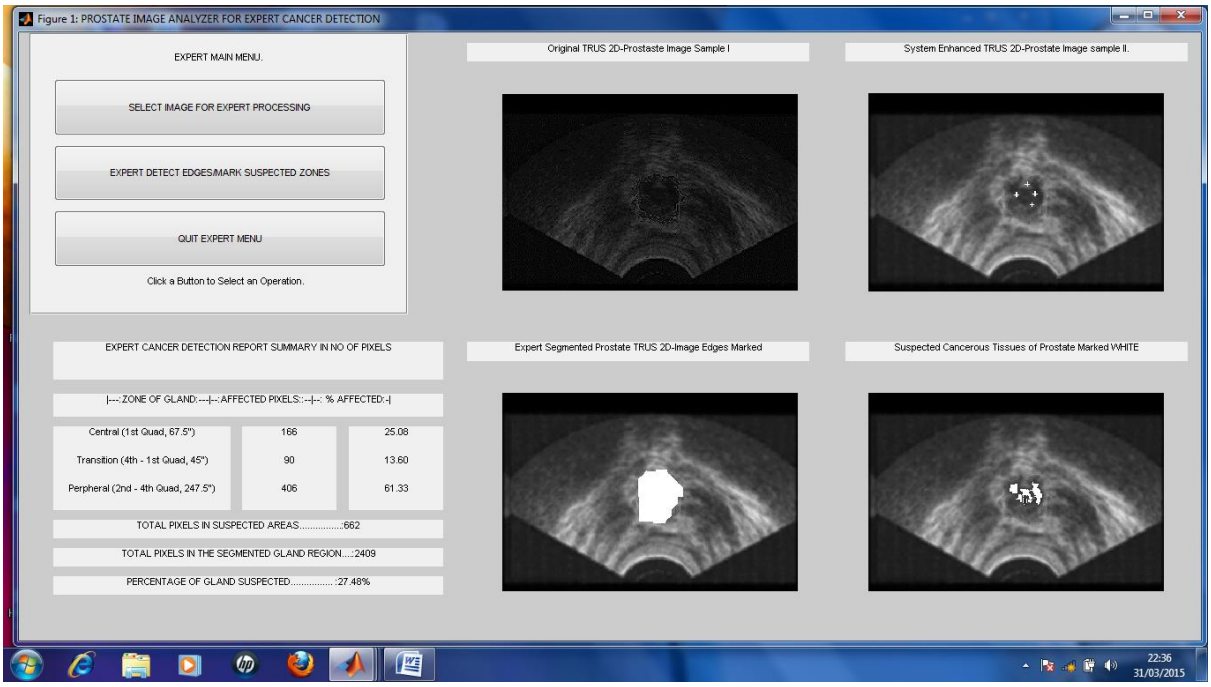


Figure 4.5D: Sample 13- expert segmentation and detection result

The first image on the first row shows the original sample with seed point mark, the second image on the first row shows the enhanced sample image, the first image on the second row shows the resulting image after segmentation by the algorithm with the segmented prostate gland section shown in highest intensity, and the second image on the second row shows the result of cancer detection by the algorithm. Also the highest intensity, white section (hyperechoic), shows the suspected cancerous cells/portions of the prostate gland sample under investigation.

Table 4.2 Expert gland segmentation/cancer detection summary

SAMPLE NO.	AREA OF GLAND (PIXELS) TOTAL	AREA OF GLAND ZONES SUSPECTED CANCEROUS		
		TRANSITION (PIXELS)	CENTRAL (PIXELS)	PERIPHERAL (PIXELS)
2	13994	1692	10	320	1362
3	9405	2268	0	233	2035
4	16097	811	304	112	395
5	23536	4169	958	610	2601
6	20530	9147	1049	1707	6391
7	19189	3846	144	372	3330
8	8060	1069	496	237	336
9	26369	8056	851	2187	5018
10	14051	0	0	0	0
11	15230	1668	366	379	923
12	11040	5643	968	1495	3180
13	2409	662	90	166	406
14	14957	1082	45	480	557
15	14187	3089	276	206	2607
16	12222	2509	159	514	1836
17	9646	1200	645	375	180
18	15673	2505	512	507	1486
19	23474	10282	700	1151	8431
21	25563	602	132	145	325
22	13875	478	0	94	384
23	13098	1749	224	691	834
27	30609	5058	969	1089	3000
29	14091	2584	441	425	1718

Expert result was strong in that it was capable of segmenting all sizes of prostate gland images and very robust in that it follows the experts mark precisely to spot suspected cancerous cells. The content of the summary of segmentation and detection in the samples showing no of pixels in areas for several images were accumulated and listed together in Table 4.2. The record in each row of Table 4.2 was extracted from the summary of segmentation/detection for the image sample, listed in the column named 'SAMPLE NO.'. The other columns show; the area of gland segmented, the area of gland suspected to be cancerous, the area of gland suspected to be cancerous within the Transition, Central and Peripheral Zones.

4.5 Validation of Algorithm/Comparison with Expert Solutions

The algorithm implemented the validation metrics presented in chapter three using area based metrics. The values for accuracy and sensitivity were obtained for each image sample analysed for presence of cancer cells by the algorithm. This required that the same image be subjected to expert analysis and area parameters obtained for the computation. The result presented here were generated from the system when option three (3) was selected from the system main menu.

The process of validation was termed comparison in the graphic user interface shown for the expert system screen. When the comparison was done for one image sample whose segmentation and detection results was known for both algorithm and expert the system producedthe result as shown in Figures 4.6A to 4.6D for the corresponding algorithm and expert results of Figures 4.4A to 4.4D and Figures 4.5A to 4.5D respectively.

The content of the result displayed on the graphic user interface screen for each figure shows on the first row or line, first the segmented image and second the image with suspected cancerous cells marked white by the algorithm. The second row or line shows same for the expert result. The table containing the result of the area based metrics for validation/comparison showing values of accuracy and sensitivity for segementation and detection displayed under the System main menu options has the format:

COMPARISON	ACCURACY	SENSITIVITY
SEGMENTATION	NNN	NNN
DETECTION	NNN	NNN

COMPARISON RESULT DISPALYED FOR : xxxxxxxx.jpg

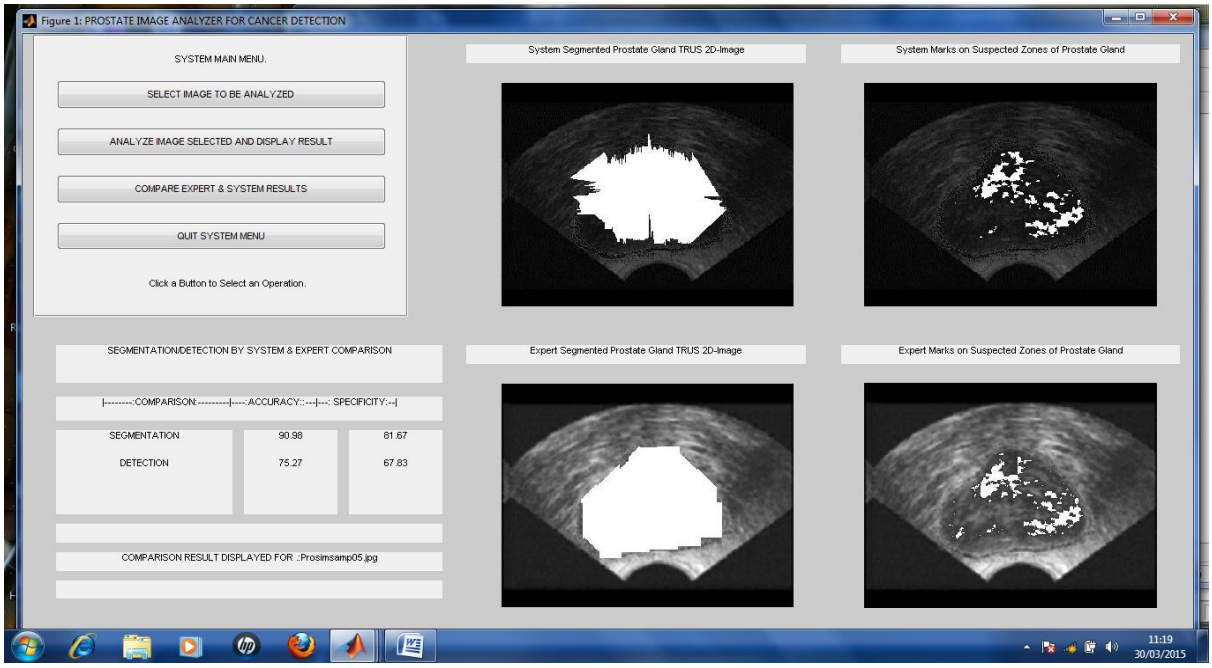


Figure 4.6A: Sample 05 – comparison of segmentation and cancer detection results

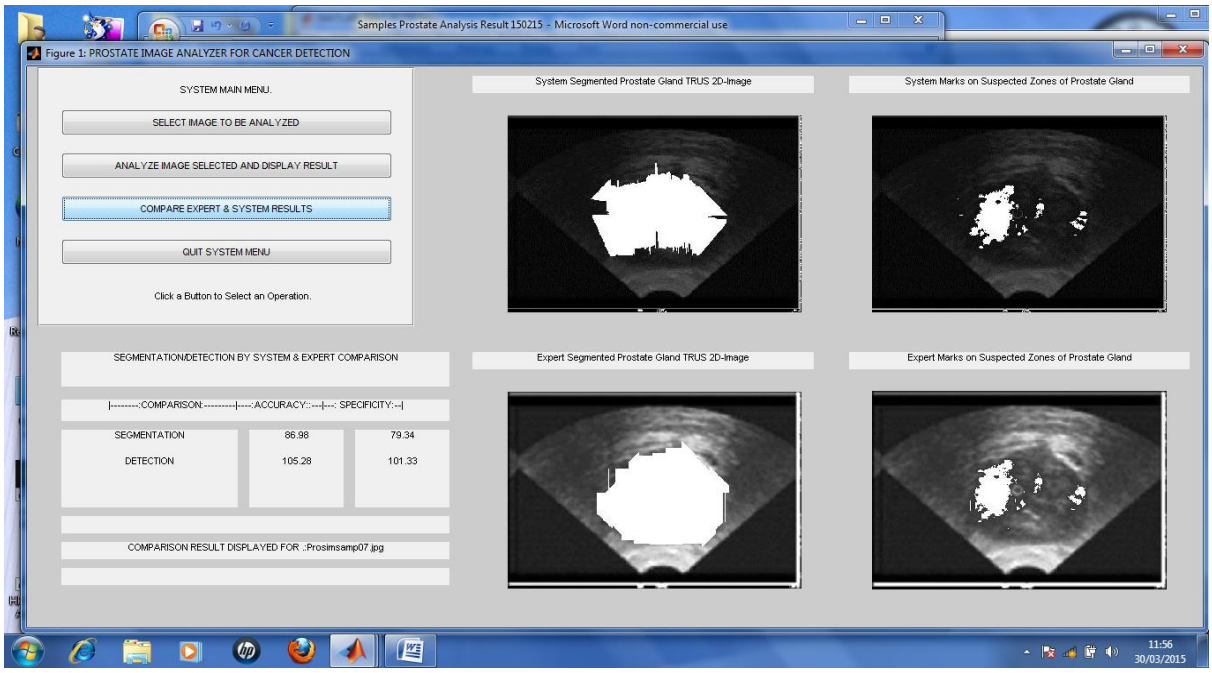


Figure 4.6B: Sample 7 – comparison of segmentation and cancer detection results

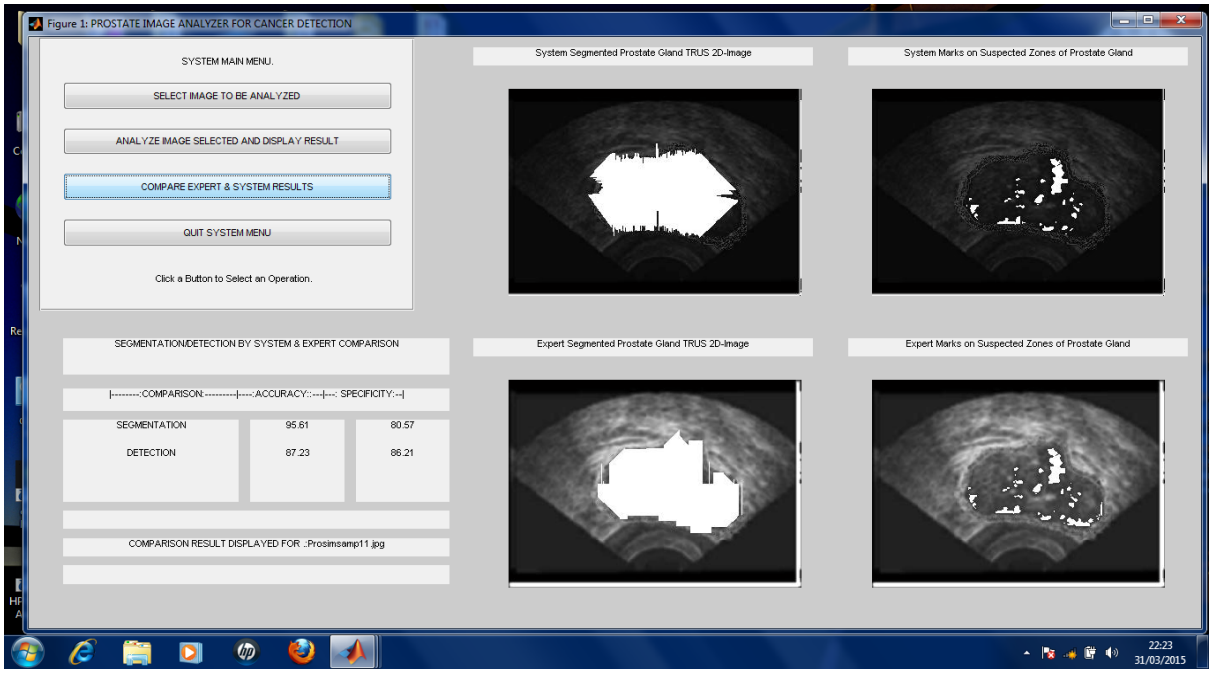


Figure 4.6C: Sample 11 – comparison of segmentation and cancer detection results

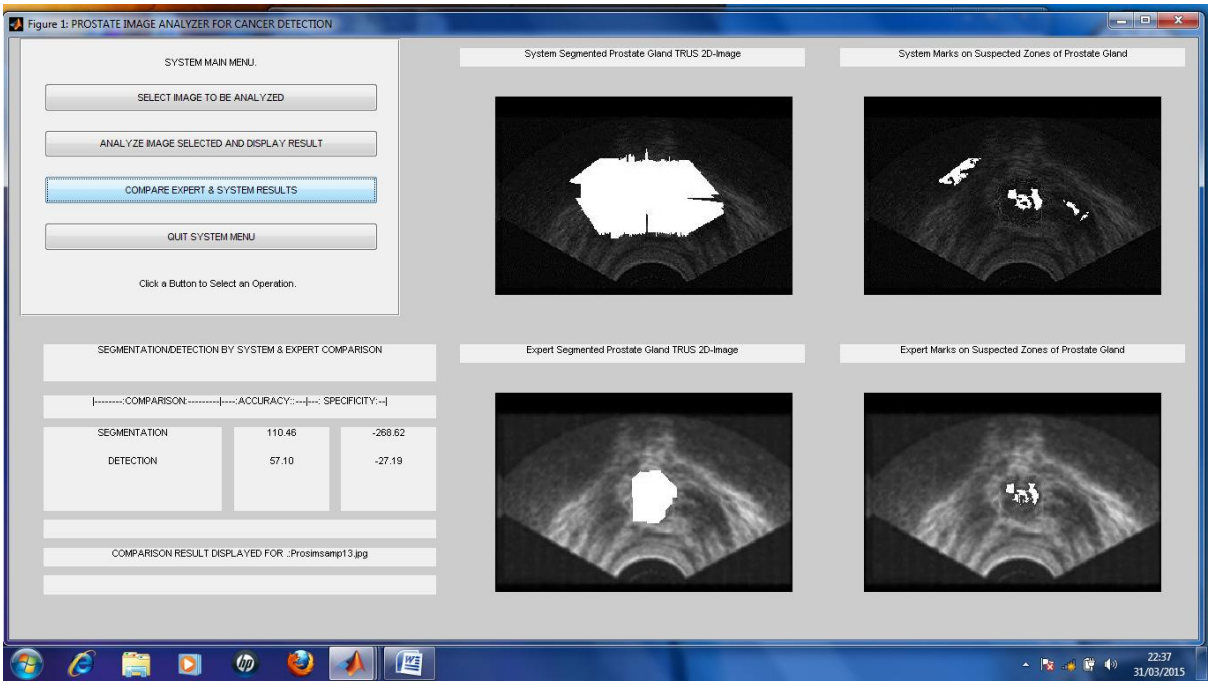


Figure 4.6D: Sample 13 – comparison of segmentation and cancer detection results

The validation/comparison algorithm produced values. These values for each image sample tested were extracted and stored as rows of Table 4.3. Columns two and three shows the values for accuracy and sensitivity for segmentation respectively, while columns four and five shows the values of accuracy and sensitivity for cancer detection for a give image sample shown in column one. The average values for the 20 samples are shown at the bottom of the table along with the standard deviation.

Table 4.3 Algorithm validation/comparison result summary

SAMPLE NO.	SEGMENTATION VALIDATION ACCURACY	SEGMENTATION VALIDATION SENSITIVITY	CANCER DETECTION VALIDATION ACCURACY	CANCER DETECTION VALIDATION SENSITIVIY
4	77.02	72.46	79.16	78.67
5	90.98	81.67	75.27	67.83
6	59.28	32.33	52.61	51.04
7	86.98	79.34	97.65	88.75
8	90.2	71.14	79.7	79.7
9	77.54	60.35	96.65	84.07
11	95.61	80.57	87.23	86.21
12	96.52	53.26	97.45	98.1
14	95.81	86.33	98.23	97.66
16	97.55	77.94	68.07	63.53
17	91.22	67.49	74.33	72.75
18	90.87	78.93	79.68	76.21
19	79.09	72.28	83.72	71.22
20	84.12	56.1	97.44	69.44
21	95.18	79.7	98.65	88.98
22	97.31	83.19	96.76	97.07
23	99.72	80.86	95.87	95.44
25	96.56	73.44	70.86	65.48
27	71.29	65.01	61.86	57.85
29	98.15	80.55	92.26	79.84
AVERAGE VALUES =	88.55	71.65	84.17	78.49
STANDARD DEVIATIONS =	10.74	13.04	13.84	13.68

In the table columns two and three show the accuracy and sensitivity for image segmentation while the fourth and fifth columns show the accuracy and sensitivity for cancer detection. From the summary result obtained in the table, the algorithm achieved for segmentation an average Accuracy of 88.55%, and average Sensitivity of 71.65% with standard deviation of 10.74 and 13.04 respectively, while for Cancer detection it achieved an average Accuracy of 84.17% and average Sensitivity of 78.49% with standard deviation of 13.84 and 13.68 respectively. The average values, $\bar{A}$, for each data set in the table were computed for the 20 samples recorded in the table using the expression given in equation (5.1):

$$\bar{A} = \frac{\sum_{i=1}^{20} a_i}{20} \qquad (5.1)$$

Also the standard deviation, STD, for each data set in the table was computed using the expression in equation (5.2):

$$\text{STD} = \sqrt{\frac{\sum_{i=1}^{20}(a_i - \bar{A})^2}{20}} \qquad (5.2)$$

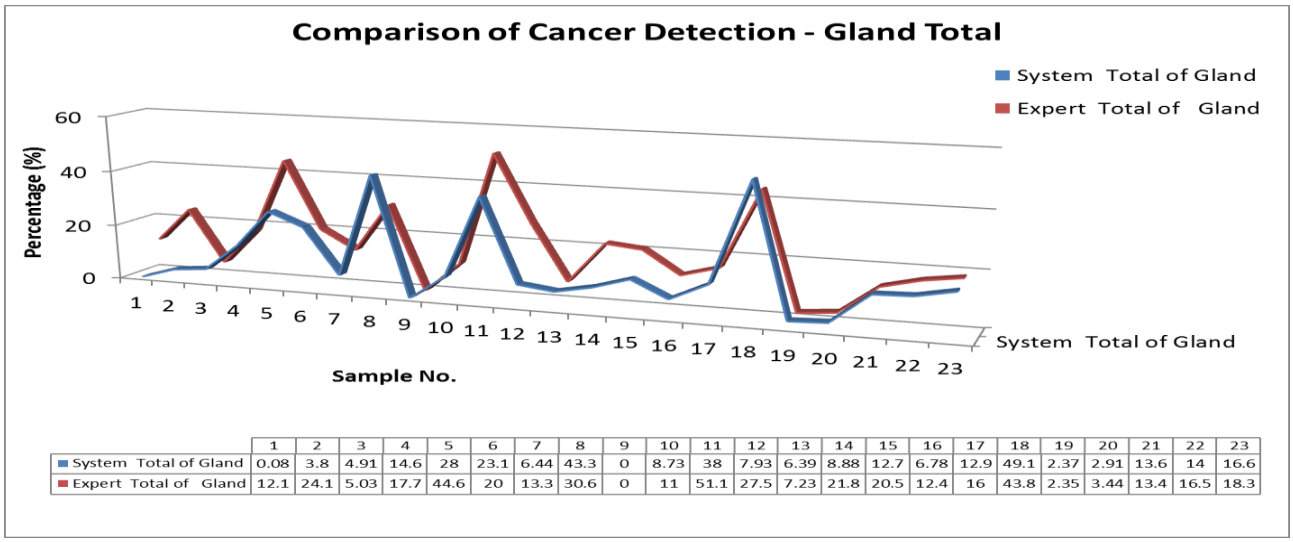
Table 4.4was the result of accumulating the percentage of the total gland, and zones of the gland suspected to be cancerous as generated from the result of section 4.4.3.3 and 4.4.3.4. The values for segmentation and detection are stored side by side for each sample in the table for ease of comparison by observation.

The results in the table show that algorithm's percentages for the total gland section detected, closely compared with that of the expert for most of the samples especially, 18 – 29, 5-10 and varied differently upward for expert result because of the effect of non-homogeneity of pixel distribution in most ultrasound images which is the major limitation of the region growing technique. The table also revealed that the percentages for zones detected by the algorithm and experts also agreed along the same line of image samples. However it was noted that the pattern of distribution of percentages agreed either in ascending or descending values for both expert and algorithm in most of the image samples.

Graphs were usedto represent the behaviour of the detection results obtained by the algorithm in common axes with the expert resultsto show the performance of the zoning subsystem of the algorithm. This further validated the performance of the entire system by revealing the closeness of the patterns and the cureves.The content of Table 4.4 were built from columns 3,4,5 and 6 of Tables 4.1 and 4.2 converted to percentages with respect to the segmented gland area for system and expert results respectively.

Table4.4Summary of percentage of cancer detection by gland zones

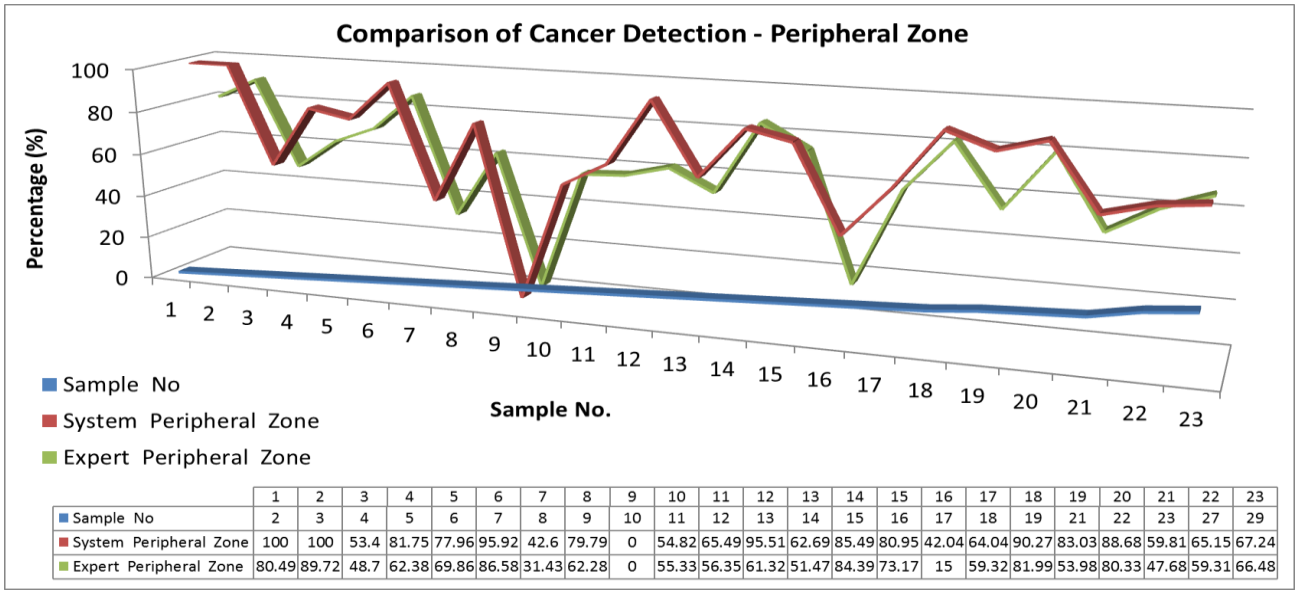
Sample No	Expert Cancer Detection (%)				System Cancer Detection (%)			
	Transitn Zone	Central Zone	Peripheral Zone	Total of Gland	Transitn Zone	Central Zone	Peripheral Zone	Total of Gland
2	0.59	18.91	80.49	12.09	0	0	100	0.08
3	0.00	10.27	89.72	24.11	0	0	100	3.80
4	37.48	13.81	48.70	5.03	23.295	27.954	53.40	4.91
5	22.97	14.63	62.38	17.71	9.115	10.938	81.75	14.60
6	11.46	18.66	69.86	44.55	11.015	13.218	77.96	27.98
7	3.74	9.67	86.58	20.04	2.03	2.436	95.92	23.13
8	46.39	22.17	31.43	13.26	28.695	34.434	42.60	6.44
9	10.56	27.14	62.28	30.55	10.1	12.12	79.79	43.30
10	0.00	0.00	0.00	0.00	0	0	0.00	0.00
11	21.94	22.72	55.33	10.95	22.585	27.102	54.82	8.73
12	17.15	26.49	56.35	51.11	17.245	20.694	65.49	38.04
13	13.59	25.07	61.32	27.48	2.235	2.682	95.51	7.93
14	4.15	44.36	51.47	7.23	18.65	22.38	62.69	6.39
15	8.93	6.66	84.39	21.77	7.25	8.7	85.49	8.88
16	6.33	20.48	73.17	20.52	9.52	11.424	80.95	12.72
17	53.75	31.25	15.00	12.44	28.975	34.77	42.04	6.78
18	20.43	20.23	59.32	15.98	17.975	21.57	64.04	12.92
19	6.80	11.19	81.99	43.8	4.855	5.826	90.27	49.06
21	21.92	24.08	53.98	2.35	8.48	10.176	83.03	2.37
22	0.00	19.66	80.33	3.44	5.655	6.786	88.68	2.91
23	12.80	39.50	47.68	13.35	20.09	24.108	59.81	13.61
27	19.15	21.53	59.31	16.52	17.42	20.904	65.15	14.03
29	17.06	16.44	66.48	18.33	16.37	19.644	67.24	16.58



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
System Total of Gland	0.08	3.8	4.91	14.6	28	23.1	6.44	43.3	0	8.73	38	7.93	6.39	8.88	12.7	6.78	12.9	49.1	2.37	2.91	13.6	14	16.6
Expert Total of Gland	12.1	24.1	5.03	17.7	44.6	20	13.3	30.6	0	11	51.1	27.5	7.23	21.8	20.5	12.4	16	43.8	2.35	3.44	13.4	16.5	18.3

Figure 4.7A: Comparison of cancer detection results graph - gland total

The detection graph, Figure 4.7A, for the entire prostate gland for both proposed algorithm and expert results show approximately the same pattern.This reveals that the algorithm appropriately detects presence of cancer characteristics in similar measures within the prostate gland.

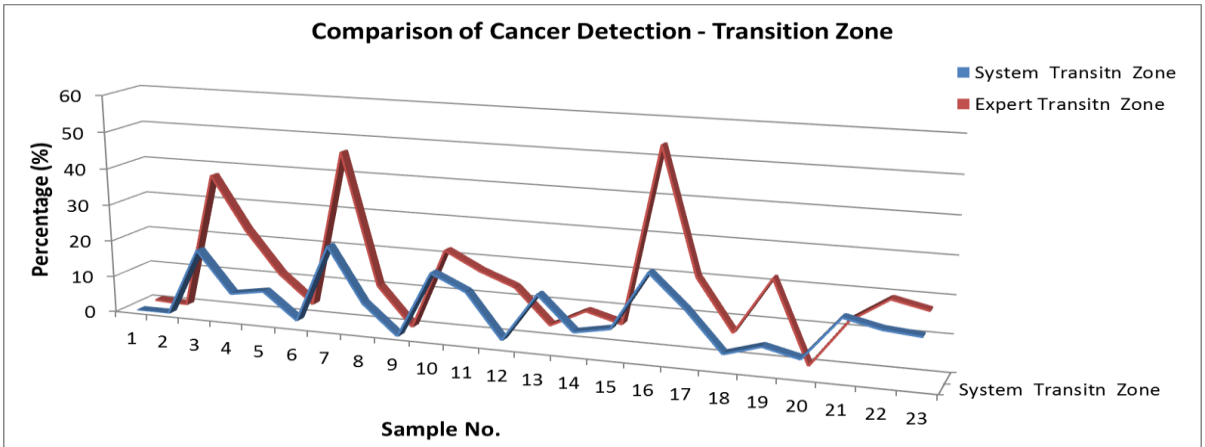


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sample No	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	21	22	23	27	29
System Peripheral Zone	100	100	53.4	81.75	77.96	95.92	42.6	79.79	0	54.82	65.49	95.51	62.69	85.49	80.95	42.04	64.04	90.27	83.03	88.68	59.81	65.15	67.24
Expert Peripheral Zone	80.49	89.72	48.7	62.38	69.86	86.58	31.43	62.28	0	55.33	56.35	61.32	51.47	84.39	73.17	15	59.32	81.99	53.98	80.33	47.68	59.31	66.48

Figure 4.7B: Comparison of cancer detection results graph – peripheral zone

The graph in Figure 4.7B shows that the pattern of detection in the peripheral zone for both proposed algorithm and expert follows a close sequence and size. Since more than 70% of cancers

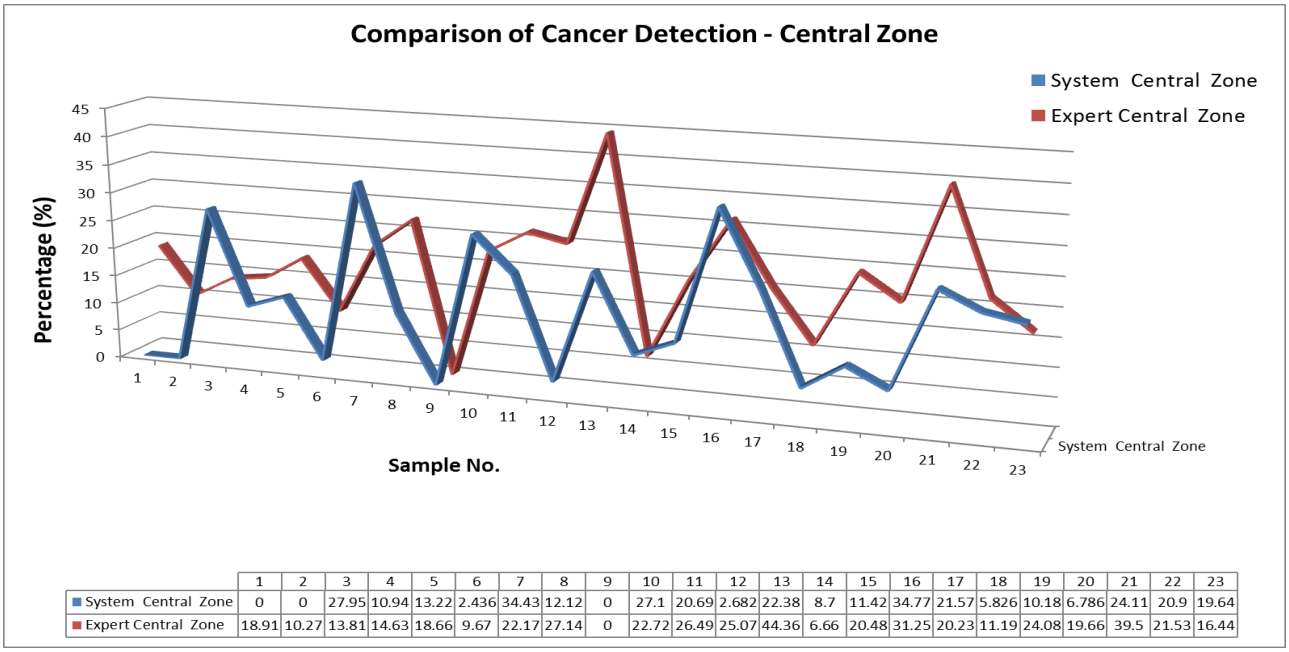
originate in this zone(see section 2.1.1.1), the system guarantees the detection of any such incidence of cancer in the prostate gland tissues.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
System Transitn Zone	0	0	18.6	7.29	8.81	1.62	23	8.08	0	18.1	13.8	1.79	14.9	5.8	7.62	23.2	14.4	3.88	6.78	4.52	16.1	13.9	13.1
Expert Transitn Zone	0.59	0	37.5	23	11.5	3.74	46.4	10.6	0	21.9	17.2	13.6	4.15	8.93	6.33	53.8	20.4	6.8	21.9	0	12.8	19.2	17.1

Figure 4.7C: Comparison of cancer detection results graph – transition zone

The graph reveals that the algorithm's performance in detecting cancer characteristics in the Transition zone is close except in four cases where the Expert record higher values. This was partly due to the shortfalls of the zoning algorithm. It can also be due to the maximum malignant dominant intensity value chosen. The result can be improved by adjusting this value upward. Only between 10% -20% of cancer originates in this zone.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
System Central Zone	0	0	27.95	10.94	13.22	2.436	34.43	12.12	0	27.1	20.69	2.682	22.38	8.7	11.42	34.77	21.57	5.826	10.18	6.786	24.11	20.9	19.64
Expert Central Zone	18.91	10.27	13.81	14.63	18.66	9.67	22.17	27.14	0	22.72	26.49	25.07	44.36	6.66	20.48	31.25	20.23	11.19	24.08	19.66	39.5	21.53	16.44

Figure 4.7D: Comparison of cancer detection results graph – central zone

Figure 4.7D reveals that the two curves had similar patternin the Central zone. The system recorded high in two cases while the Expert recorded high in five cases. The variance in this case was primarily due to the shortfalls of the zoning algorithm. This result can be improved by adjusting the boundary between the Transition and Central zones in the zoning algorithm. Only about 5% -10% of cancers originate in this zone.

4.6 Comparison of Performance of Algorithm with Others

The performance of the algorithm on segmentation of prostate gland and that of others were compared as summarized in Table 4.5. The table shows that the algorithm achieved accuracy values close to that of the other two by Ladak et al.(2000) and Awad (2007). The accuracy and sensitivity values of these algorithms on segmenting the prostate gland from an ultrasound image were compared.

Table 4.5Comparisonof the performance of segmentation by the system and two otheralgorithms.

	System		Ladak et al Algorithm		Awad Algorithm	
Measures	Average	Standard Deviation	Average	Standard Deviation	Average	Standard Deviation
Accuracy	88.55	10.74	90.10	3.2	91.03	2.6
Sensitivity	72.93	13.04	94.50	2.7	95.60	2.5

From the table it was observed that the system yielded high accuracy very close to the other two algorithms of Ladak et al. (2000) and Awad (2007). However, the others yielded slightly higher sensitivity values than the system. The system has higher standard deviation value than the other two algorithms.

It is worthy to note that the algorithm based on region growing performs at higher speed than the other two based on deformable model or snake algorithm. The algorithm was simple and less complex while the other two are very complex and cumbersome involving a lot of computations. The algorithm was therefore better because it was faster less than 40s for segmentation of the gland and has performance that is highly comparable.

The performance of the new system on detection of cells suspected to be cancerous (hyperechoic pixels) compared in the light of that of another was summarized in Table 4.6.

Table 4.6Comparison of the performance of detection of suspected cancerous cells by the system and that proposed by Awad (2007).

	System		Awad's Algorithm	
Measures	Average	Standard Deviation	Average	Standard Deviation
Accuracy	84.17	13.84	84.16	6.46
Sensitivity	78.49	13.68	66.56	14.65

The table shows the algorithm gives higher accuracy than that proposed by Awad(2007) while the later produced lower standard deviation. Also the system has higher sensitivity that that of Awad (2007), while the algorithm has slightly lower standard deviations. The algorithm therefore yields better result in detection of suspected cancerous cells or hyperechoic pixels..

CHAPTER FIVE
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Several techniques have been applied to segment the regions of the prostate gland and detect suspected cancerous cells (hyperechoic regions) with various degrees of accuracy and sensitivity. This work has designed and developed a new algorithm for segmenting and detecting suspected cancerous cells in the prostate gland from a TRUS 2D-image. In this work enhancing region growing segmentation method with prior knowledge and radial/axial scanning of the pixels from anautomated chosen centre of a TRUS 2D-image sample of the prostate gland was used to both segment the boundaries of the prostate gland and detect suspected cancerous cells in specific zones of the segmented prostate gland.Artificial intelligence techniques (expert system and image segmentation) were used to develop a novel algorithm for consistently and reliably diagnosing prostate cancer from TRUS 2D-images. Structured systems analysis and design methodology was adoptedin modelling the new algorithm. MATLAB programming tool was used to code the developed algorithm.

The algorithm for segmenting TRUS 2D-images of the prostate was designed, developed and tested with several life image samples from an earlier work using a different segmentation technique. The algorithm is strong in that it was able to detect the boundaries of most image samples correctly. However, it was weak because it placed a limit on the smallest diameter of prostate gland it could detect its boundaries. The area of segmented gland were displayed. The detection of suspected cancerous sections was successfully implemented. The test results revealed that the algorithm could detect details of clusters of pixels that show cancer properties which the expert's visual ability could not catch. The diagnosis results displayed by the algorithm showed the detected suspected cancerous areaand the total gland area.The algorithm also successfully used ratio-based metrics to partition the gland into three regions and the result of cancer detection was shown in area according to these three zones. The zoning helps to localize the result of diagnosis to specific zones of the prostate gland.

Also percentages of the zones affected by cancer in relation to the total affected in the prostate gland were computed and displayed in the result of diagnosis. The result of detection revealed in the partitioning of zones agreed with the expert view of zones likely to manifest cancer according to percentages. Also the curve plotted for partitioning the gland into zones by the algorithm and expert results showed close similarity. The algorithm for capturing expert's marking of the boundaries of the prostate gland and suspected cancerous sections of the three zones was implemented successfully.The average processing time of the segmentation and detection of suspected cancerous pixels algorithm was 40s approximately.

The algorithm was validated using area-based metrics. The areas of the algorithm and experts' segmentation and detection results were used to compute the validation and evaluation parameters. An average accuracy of 88.55% and sensitivity of 71.647% were recorded for segmentation of prostate gland while accuracy of 84.17% and sensitivity of 78.49% were recorded for detecting the hyperechoic (suspected cancer cells) regions of the prostate gland. These results compares well with that of previous work using same life image samples but applying parametric deformable model segmentation technique, (Gradient Vector Flow), which recorded average accuracy of 91.03% and sensitivity of 95.60% for segmentation and average accuracy of 84.16% and sensitivity of 66.56% for detecting the hyperechoic regions of the prostate gland.The overall certainty factor of the detection algorithm using the association of the significant certainty factors that propagated through the system, namely cf_8= 0.9 and cf_9 = 0.9,was 0.81 (according to conjuntive rule of eqtn 2.40). The other uncertainty factors were intangible as stated in section 2.1.6.5 (iii). This result was equivalent to 81% which was close to the accuracy valuesof 88.55% and 84.17% obtained using the area-based evaluation metrics. It implies that for every detection result by the algorithm it was about 80% certain that the patient with the prostate gland image was likely to have cancer cells in the regions identified.It was also interpreted to mean that the result obtained by the system at any given time for any image sample was almost certainly true.

The analysis of medical images using artificial intelligence techniques like segmentation therefore can accurately and successfully assist medical experts in diagnosing disease especially cancer of the prostate.It was therefore the view of the researcher that this work is another important milestone in the search for a reliable, consistent and robust image segmentation technique. It is believed that the

results obtained in this work will go a long way to assisting medical professionals involved in managing prostate disease to provide both routine services and required reliable diagnosis.

5.2 Summary of Achievements

The research work has made the following achievements:

i. Successfully completed the development and testing of a novel algorithm for the segmentation of TRUS 2D-images of the prostate gland towards area estimation.

ii. Successfully developed and testedanovel algorithm for detecting suspected cancerous regions of the prostate gland from TRUS 2D-images

iii. Successfullydeveloped and tested a novel algorithm for zoning the matrix of TRUS 2D-images of the prostate gland using ratio-based matrices towards localization of suspected cancerouspixels/tissues.

iv. Successfully developed and tested a novel algorithm for capturing expert marking of the segments of the prostate gland and suspected sections of the three zones on a TRUS 2D-image of the prostate.

v. Successfully applied the algorithms for detection of prostate cancer from TRUS 2D-images of the prostate.

vi. Successfully validated and evaluated the algorithm for the segmentation of TRUS 2D-images of the prostateand cancer detection using area-based metrics that applied expert’s results.

5.3 Contributions to Knowledge

This work has made a number of contributions to knowledge which includes:

i. A novel algorithm based on region growing for segmenting and detecting cancerous regions of the prostate gland from TRUS 2D-images was developed.

ii. Developed a novel algorithm performing radial/axial scanning of TRUS 2D-image matrix towards region growing image segmentation.

iii. Developed a novel technique for zoningthe matrix of TRUS 2D-images of the prostate gland using ratio-based metrics towards localization of malignant pixel/tissues.

iv. Developed a novel algorithm for capturing the expert's marking of the segments of the prostate gland and suspected sections of the three zones on a TRUS 2D-image of the prostate.

v. Developed a novel algorithm based on statistical analysis and sequential stick technique for detecting malignant sections of a prostate gland from ultrasound 2D-images.

vi. This work provides radiologists with ready, reliable and consistent diagnostic assistant for detecting regions/zones of suspected cancerous cells from 2D TRUS images.

vii. This work facilitates reduction in the frequency of biopsies since sites of suspected cancerous cells are readilyand reliably known, consequently biopsy can be conducted on sites selected with scientific accuracy.

5.4 Suggestions for Further Improvements

In the light of the experiences and findings/observations in the course of this research, the following were suggested for further improvement on the results of this direction of research in image analysis towards medical diagnosis:

i. The algorithm that trims and fits the edges detected to give a smooth curvature around the boundaries of the regions of the prostate gland can be improved upon to yield even smoother edges,

ii. Incorporate an interface to an ultrasound machine to read online an image of the prostate organ captured during investigation to process real-time the image and give diagnosis result immediately.

iii. The malignancy detection algorithm can be enhanced by incorporating the feature that also determines the accuracy and sensitivity of the area in pixels of all the separate zones that are marked malignant within the region of the segmented prostate gland.

iv. To introduce an additional algorithm to estimate the area/ volume of the segmented prostate and subsequently the area/volume of the suspected malignant sections of the gland in metric units and not in no of pixels.

v. To repeat suggestion in iv. above using ultrasound 3D-images of the prostate.

vi. Extend the work to the detection of breast, ovarian cancer etc.

5.5 Problems Encountered and Solutions

The problems encountered include:

i. Delay and difficulty in locating a diagnostic centre where trans-rectal ultrasound imaging is performed. Also unavailability of archive digital record of images of patients with history of prostate condition. Unwillingness of patient being treated of prostate cases to submit themselves to be scanned. After prolonged waiting and persuasions and collaboration some samples were obtained on-line from a previous work (Award, 2007) to enable testing and completion of this work.

ii. Challenge of learning how to program with MATLAB. The challenge was overcome by making several references to the online command help. Several dummy programs were written to test the functionality, features and syntax of some command, especially with respect to matrix/array/table manipulation.

iii. Irregular power supply. Had to resort to use of generators. Sometimes had to work late into the night when power supply from the Power Holding returns.

5.6 Recommendations

The following are recommended:

i. Medical professional should take advantage of the fast and robust feature of the algorithm presented in this work to make consistent diagnosis from ultrasound 2D-images.

ii. The cancer detection functionality should be handy for determining sections of the gland prior to biopsy procedure to guarantee selection of sites for the investigation.

iii. This software can be interfaced to the ultrasound machine to provide on-line real-time analysis of 2D images captured for patients with raised PSA levels to detect presence or absence of cancerous cells in the prostate gland image.

5.7 Publications on the Research

i. **Chijindu V. C.**, "*Search in Artificial Intelligence Problem Solving*", Afr J. of Comp & ICTs. Vol. 5, No. 5. pp 37-42, September, 2012.

ii. **Chijindu V. C., Inyiama H. C. and Uzedhe G. O.** (2012), "*Medical Image Segmentation Methodologies – An Overview*", Afr J. of Comp & ICTs. Vol. 5, No. 5. pp 100-108, September, 2012.

iii. **Inyiama H. C. , Chijindu V. C. and Ufoaroh S. C.** (2012), "*Intelligent Agents – Autonomy Issues*", Afr J. of Comp & ICTs. Vol. 5, No. 4, pp 49-53, June, 2012.

iv. **Chijindu V. C. and Inyiama H. C.**, "*Social Implication of Robots – An Overview*", International Journal of Physical Sciences Vol 7(8), pp. 1270-1275, 16 Feb. 2012.

References

Aarnink R. G., Giesen R. J. B.,and Huynen A. L. (1994), *A Practical Clinical Method for Contour Determination in Ultrasonographic Prostate Images*, Ultrasound Med Biol 20(1):133-142.

Aarnink R. G., Pathak S. D. and de la Rosette J. J. (1998).*Edge Detection in Prostate Ultrasound Images Using Integrated Edge Maps*, Ultrasonics, 36(1):212-223.

Aarnink R. G., Beerlage H. P., de la Rosette J. J., Debruyne F. M. J. and Wijkstra H. (1998). *Transrectal Ultrasound of the Prostate: Innovations and Future Applications*, J Urol, 159(1):311-321.

Al-tarawneh M. S. (2012*). Lung Cancer Detection Using Image Processing Techniques.*Leonardo Electronic Journal of Practices and Technologies. 20(1):147-158.

Amayeh G. and Tavakkoli A. (2009*). Accurate and Efficient Computation of Gabor Features in Real-Time Applications.* In Advances in Visual Computing, pp. 243-252.

Anderson M. andTrahey G. (2000).*A Seminar on k-space Applied to Medical Ultrasound*, Department of Biomedical Engineering, Duke University.

Andreasen N.C., Rajarethinam R.and Cizadlo T. (1996).*Automatic Atlas-Based Volume Estimation of Human Brain Regions from MR Images,* J. Comp. Assist. Tom., 20(1):98-106.

Awad J.G.E. (2007). *Prostate Segmentation and Regions of Interest Detection in Transrectal Ultrasound Images*, A Doctor of Philosophy Thesis, University of Waterloo, Waterloo, Ontarion, Canada.

Balakrishnan K., Sunil K., Sreedhanya A. V. and Soman K. P. (2012). *Effect Of Pre-Processing On Historical Sanskrit Text Documents.*International Journal of Engineering Research and Applications. 2(4):1529-1534.

Barney Smith E. H., Likforman-Sulem L. andDarbon J. (2010*). Effect of Preprocessing on Binarization.* In Proceedings of SPIE.7534, pp.75340H-75340H.

Benoudjit N., Lendasse A., Lee J. and Verleysen M. (2002). *Width Optimization of the Gaussian Kernels in Radial Basis Function Networks.*In Esann' 00 Proceedings-European Symposium on Artificial Networks, pp. 425-432.

Betrouni N., Vermandel M., Pasquier D., Maouche S. and Roussea J. (2005). *Segmentation of*

Abdominal Ultrasound Images of the Prostate Using Priori Information and an Adapted Noise Filter. Computerized Medical Imaging and Graphics, 29(1):258-265.

Bezdek J.C., Hall L.O.and Clarke L.P. (1993).*Review of MR Image Segmentation Techniques Using Pattern Recognition,* Med. Phys., 20(1):1033-1048.

Bo J.and Wei H. (2007). *Adaptive Threshold Median Filter for Multiple-Impulse Noise*,Journal of Electronic Science and Technology of China.5(1):70-74.

Boncelet C. (2005).''*Image Noise Models*". In Alan C.Bovik. Handbook of Image & Video Processing, pp. 88-92.

Bovik A. C. (1988). *On Detecting Edges in Speckle Imagery*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 36(10):422-435.

Brodic D. and Milivojevic Z. (2010). *Optimization of the Gaussian Kernel Extended by Binary Morphology for Text Line Segmentation.* RadioEngineering. 19(4):718-724.

Canada Cancer Society (2007).*National Cancer Institute of Canada: Canada Cancer Statistics 2006*. Retrieved on 12.09.2012from http//www.cancer.ca.

Castanho M. J. P., Hernandes F., De Ré A. M., Rautenberg S. and Billis A. (2013). "Expert Systems with Applications Fuzzy Expert System for Predicting Pathological Stage of Prostate Cancer". 40(1):232-245.

Chan R. H., Hu C. and Nikolova M. (2004). *An Iterative Procedure for Removing Random-Valued Impulse Noise.* IEEE Signal Processing Letters. 11(12):921-924.

Chaudhary A. and Singh S. S. (2012*). Lung Cancer Detection Using Digital Image Processing.*International Journal of Research in Engineering & Applied Sciences. 2(2):1351-1359.

Cheng H. D., Cai X., Chen X., Hu L.andLou X. (2003).*Computer-Aided Detection and Classification of Micro-calcifications in Mammograms: A Survey*, Pattern Recognition Society Journal, 36(12):2967-2991.

Chiu B. (2003). *A New Segmentation Algorithm for Prostate Boundary Detection in 2D Ultrasound Images*, Master's thesis, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada.

Ciabattoni A., Muiño D. P., Vetterlain T. and El-zekey M. (2013). "Formal Approach to Rule-Based Systems in Medicine: The case of CADIAG-2". Int. J. Approx. Reason. 54(1):876-889.

Comaniciu D., Meer P. and Member S. (2002). *Mean Shift : A Robust Approach Toward Feature Space Analysis.*IEEE Transaction on Pattern Analysis and Machine Intelligence. 24(5):454-465.

Conor H. (2010).*Generation and Application of Recombinant Antibody Fragments for Prostate Cancer Detection*. PhD Thesis, Dublin City University.

Czerwinski R. N., Jones D. L. and O'Brien W. D. Jr. (1993).*An Approach to Boundary Detection in Ultrasound Imaging.* In: 1993 IEEE Ultrasonics Symposium Proceedings, Piscataway, NJ:Institute of Electrical and Electronics Engineers, pp. 180-191.

Czerwinski, R. N., Jones, D. L. and O'Brien, Jr., W. D. (1998). *Line and boundary detection in speckle images.*IEEE Trans. on Image Process. 7(12):231-242.

Czerwinski, R. N., Jones D. L. and O'Brien W. D. (1999). *Detection of Lines and Boundaries in Speckle Images – Application to Medical Ultrasound,* IEEE Transaction on Medical Imaging, 18(2):126-136.

Daugman J. G. (1985). *Uncertainty Relation For Resolution In Space, Spatial Frequency, And Orientation Optimized By Two-Dimensional Visual Cortical Filters*. Journal of the Optical Society of America. 2(7):1160-1169.

Davatzikos C. (1996). *Spatial Normalization of 3D Images Using Deformable Models*, J. Comp.Assist. Tom., 20(1):656-665.

Davatzikos C., Vaillant, M., Resnick S., Prince J. L., Letovsky S.and Bryan R. N. (1996). *A Computerized Method for Morphological Analysis of the Corpus Callosum*, J. Comp. Assist. Tom., 20(1):88-97.

Dougherty G. (2009).*Digital Image Processing for Medical Applications*, Cambridge University Press NY, pp. 11, 155-189,252.

Dubey S., Pandey R. K., and Gautam S. S. (2014).*Dealing with Uncertainty in Expert Systems*. International Journal of Soft Computing and Engineering (IJSCE). 4(3):105-111.

Eltoft T. (2003).*Speckle: Modeling and Filtering*, In Norwegian SignalProcessing Symposium, Norway. Retrieved on 14.05.2014 from http://www.norsig.no/norsig2003/.

Eskandari H., Talebpour A., Tabrizi S. H. and Nowroozi M. R. (2012). Development of a Fast Algorithm for Automatic Delineation of Prostate Gland on 2D Ultrasound Images. Proceedings of the 19th Iranian Conference on Biomedical Engineering (ICBME 2012), pp. 1122-1132.

Evangelista P. F., Embrechts M. J. and Szymanski B. K. (2007). *Some Properties of the Gaussian Kernel for One Class Learning*. In Proceeding of the International Conference on Artificial Neutral Networks.4668:269-278.

Farooque M. A. and Rohankar J. S. (2013).*Survey on Various Noises & Techniques for Denoising the Color Image*. International Journal of Application or Innovation in Engineering & Management (IJAIEM). 2(11):.

Garcia-Consuegra J., Cisneros G.and Navarro E. (2000). *A Sequential Echo Algorithm Based on the Integration of Clustering and Region Growing Techniques*, Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International, Vol 2, pp 648-650.

Garraway M. (1995).*Epidemiology of Prostate Disease*, New York: Springer-Verlag.

Gasteratos A., Andreadis I. and Tsalides P. (1998). *A parallel architecture for implementation of filters based on order statistics.* Pattern Recognition Letters. 19(9):815-820.

Gelenbe E., Feng Y.and Krishnan K.R.R. (1996). *Neural Network Methods for Volumetric Magnetic Resonance Imaging of the Human Brain*. Proc. IEEE, 84(1):1488-1496.

Gomez-Moreno H. (2001). *A Modified Median Filter for the Removal of Impulse Noise Based on the Support Vector Machines*. In Proceedings of the Advances in Signal Processing and Computer Technologies WSEAS 2001, pp. 9-14.

Gonzalez R. C. and Woods R. E. (2002). *Digital Image Processing (2nd Edition)*, Prentice Hall Inc. Upper Saddle River, NJ, pp. 661-663.

Goyal G., Bansal A. K. and Singhal M. (2013). *Review Paper on Various Filtering Techniques and Future Scope to Apply These on TEM Images*. International Journal of Scientific and Research Publications. 3(1):1-11.

Gyorgy Toth (2005).*Diagnosis and Treatment of Prostate Cancer,* PhD Dissertation, University of Debrecen 2005. Retrieved on 12.09.2012 from http://dea,lib.unideb.hu/dea/bitstream/2437/2454/2/Toth_Gyorgy_tezis_angol.pdf

Hall L.O., Bensaid A.M., Clarke L.P., Velthuizen R.P., Silbiger M.S.andBezdek J.C. (1992). *A Comparison of Neural Network and Fuzzy Custering Techniques in Segmenting Magnetic Resonance Images of the Brain*. IEEE Trans. Neural Networks3:672-682.

Hamarneh G. (2001).*Towards Intelligent Deformable Models for Medical Image Analysis,* Ph.D. Thesis, Department of Signals and Systems, School of Electrical and Computer Engineering, Chalmers University of Technology.

Hebert T. J. (1997). Fast Iterative Segmentation of High Resolution Medical Images. IEEE Trans. Nucl. Sci., 44(1):1363-1367.

Hedayati T.andKeegan M. (2009). *Prostatitis*. eMedicine.

Hu N., Downey D. B., Foster A.and Ladak H. M. (2002).*Prostate Surface Segmentation from 3D Ultrasound Images,* Proceedings of IEEE International Symposium on Biomedical Imaging.

Isram R.,and Simri W.I.W. (2013).*Rule-Based Expert System for Diagnosing Toddler Disease Using Certainty Factor and Forward Chaining*. The Proceedings of The 7th Icts, Bali.

Jawadekar, W. S. (2004). *Software Engineering Principles and Practice*, Tata McGraw-Hill, pp. 200-210.

Jensen J. A. (2006).*Medical Ultrasound Imaging*. Journal of Progress in Biophysicsand Molecular Biology, 93(1).

Karray F.andSilva C. (2004). *Soft Computing and Intelligent System Design*, Addison Wesley, pp. 88-94.

Keles A. & Yavuz U. (2011). "Expert System Based on Neuro-Fuzzy Rules for Diagnosis Breast Cancer" . Expert Syst. Appl. 38(5).

Kirkham A., Emberton M.and Allen C. (2006). *How good is MRI at Detecting and CharacterisingCancer Within the Prostate?*, European Association of Urology, 50(6):78-85.

Ladak H. M., Mao F. and Wang Y. Q. (2000). *Prostate Segmentation from 2D Ultrasound Images*. Med Phys 27(1):44-53.

Lakare S.and Kaufman A. (2000).*3D Segmentation Techniques for Medical Volumes, Center for Visual Computing*, Department of Computer Science, State University of New York.

Lee C., Huh S., Ketter T. A.and Unser M. (1998). Unsupervised Connectivity-Based Thresholding Segmentation of Midsagittal Brain MR Images, Computers in Biology and Medicine, 28(3):309-338.

Lee F., Bahn D. K., Siders D.B. and Greene C. (1998). *The Role of TRUS-Guided Biopsies for Determination of Internal &External Spread of Prostate Cancer*. Semin Urol Oncol, 16(1).

Leung Y., Chinese T., Kong H. and Kong H. (2009). "Fuzzy Set and Fuzzy Logic", pp. 234-245.

Li L., Gong J. and Chen W. (1997).*Gray-Level Image Thresholding Based on Fisher Linear Projection of Two-Dimensional Histogram*, Pattern Recognition, 30(5):98-108.

Liu L. andDai T. S. (2006). *Ridge Orientation Estimation and Verification Algorithm for Fingerprint Enhancement*.Journal of Universal Computer Science. 12(10):1426-1438.

Lobregt S. and Viergever M. A. (1995).*A Discrete Dynamic Contour Model*. IEEE Trans Med Imaging, 14(1):12-24.

Maintz J.B.A. and Viergever M.A. (1998).*A Survey of Medical Image Registration*. Med. Im. Anal., 2(1):1-36.

Mathur U., Gill S. S. and Rattan M. (2011). *Removal of Speckle Noise from Eye Images through Bacterial Foraging Optimization*. International Journal of Advanced Engineering Technology. 2(1):112-119.

Mcandrew A. (2004). *An Introduction to Digital Image Processing with Matlab,*pp. 156-165.

Miljkovic O. (2009). *Image Pre-Processing Tool*. Kragujevac Journal of Mathematics. 32(1):97-107.

Murali M. S. and Dinesh M. (2012). *Classification of Mass in Breast Ultrasound Images using Image Processing Techniques*. International Journal of Computer Applications. 42(10).

National Cancer Institute (2010). *Transrectal Ultrasound*. Retrieved on 10.09.2012 from http://www.cancer.gov/Templates/db_alpha.aspx?CdrID=46632

Negnevitsky M. (2002). *Uncertainty Management in Rule-Based Expert Systems.* Pearson Education.

Nilsson, N. J. (1998). *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann Pub. Inc., San Francisco, CA, pp. 280-282.

Noble J. A. and Boukerroui D. (2006). *Ultrasound Image Segmentation: A Survey, IEEE Transactions on Medical Imaging,* 25(8).

Ohm J.R. and Ma P. (1997).*Feature-Based Cluster Segmentation of Image Sequences.* Proc. IEEE International Conference on Image Processing, pp. 178-181.

Padmavathi G., Subashini P., Kumar M. M. and Thakur S. K. (2010). *Comparison of Filters used for Underwater Image Pre-Processing*. International Journal of Computer Science and Network Security. 10(1):58-65.

Pan W. (2004). *Application of Region Growing Algorithm on Handwritten Numerical Recognition*, Thesis for Master of Science, Department of Computer Science and Engineering, Tatung University.

Pathak S.D, Aarnink R.G. and de la Rosette J.J. (1998). *Quantitative Three-Dimensional Trans-Rectal Ultrasound for Prostate Imaging.* In: Proceedings of SPIE, Bellingham, WA: SPIE.3335(1):115-127.

Pathak S. D., Grimm P. D., Chalana V.and Kim Y. (1998). Pubic Arch Detection in TransrectalUltrasound Guided Prostate Cancer Therapy, IEEE Transactions on Medical Imaging, 17(5):762-771.

Pathak S. D., Chalana V., Haynor D. R.and Kim Y. (2000).*Edge-Guided Boundary Delineation in Prostate Ultrasound Images*, IEEE Transactions on Medical Imaging, 19(12):543-551.

Patidar, P., Gupta M., Srivastava S. and Nagawat A. K. (2010).*Image De-noising by Various Filters for Different Noise*. International Journal of Computer Applications (0975 – 8887)9(4).

Pavlin C. J. &Foster F. S. (1995).*Ultrasound Biomicroscopy of the Eye.* First Edition, SpringerVerlag,NewYork, pp. 282-286.

Paul F., Huggel C. and Ka A. (2004). *Combining Satellite Multispectral Image Data And A Digital Elevation Model For Mapping Debris-Covered Glaciers.* Remote Sensing of Environment. 89(1):510-518.

Peng Y. (2010).*Computer-Aided Histological Analysis for Prostate Cancer Diagnosis*. Doctoral Thesis/Dissertation, University of Chicago.Retrieved on 10.09.2012 from http://www.grin.com/en/doc/237742/computer-aided-histological-analysis-for-prostate-cancer-diagnosis

Perona P. and Malik J. (1990).*Scale-Space and Edge Detection Using Anisotropic Diffusion*. IEEE Trans PatterMachine Intell, 12(1):112-122.

Pham D. L., Xu C. and Prince J. L. (1998). *A Survey of Current Methods in Medical Image Segmentation,* Annual Review of Biomedical Engineering, pp. 256-265.

Pham D. L., Xu C. andPrince J. L. (2000).*Current Methods in Medical Image Segmentation*. Annual Review of Biomedical Engineering, vol. 2(1):315-338.

Pitas I. and Venetsanopoulos A. N. (1999). *Order Statistics in Digital Image Processing*. In *Proceedings of The IEEE*, 1992, vol. 80, no. 12, pp. 1893–1921.

Posada-gómez R. and Sandoval-Gonzalez O. O. (2001). *Digital Image Processing Using LabView*. In Practical Applications and Solutions Using LabVIEW Software, pp. 297-317.

Potocnik B., Zazula D.andSolina F. (1997).*Classical Image Processing vs. Computer Vision Techniques in Automated Computer-Assisted Detection of Follicles in Ultrasound Images of Ovary*. 6th International Conference on Image Processing and its Applications, 2(1):551-555.

Prasad V. S. N. and Domke J. (2005). *Gabor Filter Visualization.*Technical Report, University of Maryland.

Prater J. S.and Richard W. D. (1992). *Segmenting Ultrasound Images of the Prostate Using Neural Network.* Ultrason Imaging, pp. 100-112.

Pressman R. S. (2005). *Software Engineering: A Practitioner's Approach*, McGraw-Hill, pp. 211-212.

Rajinikannan M., Kumar A. D. and Muthuraj R. (2010*). Estimating the Impact of Fingerprint Image Enhancement Algorithms for Better Minutia Detection.* International Journal of Computer Applications. 2(1):36-42.

Rastgarpour M. and Shanbehzadeh J. (2011). *Application of AI Techniques in Medical Image Segmentation and Novel Categorization of Available Methods and Tools*, Proceedings of the International MultiConference of Engineers and Computer Scientists 2011, Hong Kong, pp. 1112-1120.

Reddick W.E., Glass J.O., Cook E.N., Elkin T.D.and Deaton R.J. (1997).*Automated Segmentation and Classification of Multispectral Magnetic Resonance Images of Brain Using Artificial Neural Networks*. IEEE Trans. Med. Imag., 16(1):911-918.

Sahoo P. K., Farag A. A.and Yeap Y. P. (1992).*Threshold Selection Based on Histogram Modeling,* IEEE International Conference on Systems, Man and Cybernetics, 1:351-356.

Sandor S.and Leahy R. (1997).*Surface-Based Labelling of Cortical Anatomy Using a Deformable Atlas.* IEEE Trans. Med. Imag., 16(1):41-54.

Sarra C. (2007). *Understanding DPI and Pixel Dimensions*. Retrieved on 20.10.2014 from http://www.iprintfromhome.com.

Sachs J. (1999). *Digital Image Basics.* In Digital Light and Color, pp. 1-14.

Shao F., Ling K. V., Ng W. S. and Wu R. Y. (2003). *Prostate Boundary Detection from Ultrasonographic Images*, J Ultrasound Med 22(1):123-128.

Schalkoff R.J. (1992). *Pattern Recognition: Statistical, Structural and Neural Approaches*, John Wiley and Sons, pp. 134-145.

Sedef Kent, Osman Nuri Oçan, and Tolga Ensari (2004)."*Speckle Reduction of Synthetic Aperture Radar Images Using Wavelet Filtering*". In ASTRIUM. EUSAR 2004Proceedings, 5th European Conference on SyntheticAperture Radar, May 25–27, 2004, Ulm, Germany, pp. 654-660.

Shapiro L. G. and Stockman G. C. (2001). *Computer Vision*. Prentice-Hall, New Jersey, pp. 92-99.

Shen D., Zhan Y. and Davatzikos C. (2003). *Segmentation of Prostate Boundaries from Ultrasound Images Using Statistical Shape Model*, IEEE Transactions on Medical Imaging, 22(4):234-244.

Sikchi S. S. and Ali M. S. (2013). "Fuzzy Expert Systems (FES) for Medical Diagnosis". 63(11).

Southern Cross Medical Library(2005). *Prostate Cancer (Symptoms, diagnosis, treatment).* New Zealand. Retrieved on 12.09.2012 from http://www.southerncross.co.nz/AboutTheGroup/HealthResources/MedicalLibrary/tabid/178/vw/1/ItemID/127/Prostate-Cancer-Symptoms-diagnosis-treatment.aspx

Stefan D. (2007). Prostate Ultrasound Image Processing. Spring 2007. 13(3). Retrieved from www.acm.0rg/crossroads

Stork M. (2003). *Median Filters Theory And Applications*. In Third International Conference on Electrical and Electronics Engineering Papers-Chamber of Electrical Engineering, Bursa, Turkey. 3(1):1-5.

Sudha S., G. R. Suresh G. R. and Sukanesh R. (2009). *Speckle Noise Reduction in Ultrasound Images by Wavelet Thresholding based on Weighted Variance.*International Journal of Computer Theory and Engineering. 1(1):723-729.

Szabo T. L. (2004). *Diagnostic Ultrasound Imaging: Inside Out*, Elseveir, pp. 238-245.

Talairach J.and Tournoux P. (1998).*Co-Planar Stereotaxic Atlas of the Human Brain. 3-D Proportional System: An Approach to Cerebral Imaging*. Thieme Medical Publisher, Inc., Stuttgart, pp. 44-49.

Terzopoulos D.and Szeliski R. (1992).*Tracking with Kalman Snakes*. In Blake A. and Yuille A.,Editors, Active Vision, Artificial Intelligence, The MIT Press, Cambridge, Massachusetts, pp 3-20.

Tewari M. D. A. (1998).*Prostate Cancer: A Step-by-Step Interactive Monogram*, CD-ROM, Macromedia Interactivity and Medical Illustration.

The Mathworks Inc. (2015), *R2015b Documentation – Image Enhancement*. Retrieved on 20.10.2014 from http://www.mathworks.com/help/images/image-enhancement-and-restoration.html

The Mathworks Inc. (2015b), *R2015b Documentation – Noise Removal.*Retrieved on 20.10.2014 from http://www.mathworks.com/help/images/noise-removal.html

The Mathworks Inc. (2015c), *R2015b Documentation – Canny Edge Function.*Retrieved on 20.10.2014 from http://www.mathworks.com/help/images/Canny-Edge-Detection.html

Thompson P.and Toga A.W. (1997).*Detection, Visualization and Animation of Abnormal Anatomic Structure with a Probabilistic Brain Atlas-Based on Random Vector Field Transformations.* Med. Im. Anal.,1:271-294.

Turban E., Rainer R. K. Jr. and Potter R. E. (2005).*Introduction to Information Technology*, 2nd Edition, John Wiley & Sons, Inc., New Delhi, pp. 234-245.

Turban E., Aronson J. E., Liang Ting-Peng and Sharda R. (2005).*Decision Support and Business Intelligence Systems*,7thEdition, Pearson Education, Inc., Upper Saddle River, New Jersey, pp. 45-48.

Van Den Boomgaard R. and Van Der Weij R. (2001). *Gaussian Convolutions Numerical Approximation Based on Ineterpolation*, pp. 537-540.

Vilarino D.L., Brea V.M., Cabello D.andPardo J.M. (1998). *Discrete-Time CNN for Image Segmentation by Active Contours*, Patt. Rec. Let., 19(1):721-734.

Wahba M. (2008). *An Automated Modified Region Growing Technique for Prostate Segmentation in Trans-Rectal Ultrasound Images*, Master's Thesis, Department of Electrical

and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada.

Wang J., Lu H., Plataniotis K. N. andLu J. (2009*). Gaussian Kernel Optimization for Pattern Classification*.Pattern Recognition. 42(7):1237-1247.

Withey D. J. and Kole Z.J.(2007).*Medical Image Segmentation: Methods and Software,* Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging, pp. 612-628.

Wu P., Liu Y., Li Y. and Liu B. (2015). *Robust Prostate Segmentation Using Intrinsic Properties of TRUS Images*. IEEE Transactions on Medical Imaging. 34(6):331-343.

Xiao C. Y., Zhang S. and Chen Y. Z. (2004).*A Diffusion Stick Method for Speckle Suppression in Ultrasonic Images*. Pattern Recognition Letter,25(16):232-246.

Xin L., Langer D. L., Haider M. A., Yang Y., Wemick M. N.and Yetick I. S. (2009). *Prostate Cancer Segmentation with Simultaneous Estimation of Markov Random Field Parameters and Class*, IEEE Trans. on Medical Imaging, 28(6):211-222.

Xu C. and Prince J. L. (1998a). *Generalized Gradient Vector Flow External Forces For Active Contours.*Signal Process.: An Int. J.71(2):34-46.

Xu C.and Prince J. L. (1998b). *Snakes, Shapes, and Gradient Vector Flow*. IEEE Trans. Image Process. 7(3):56-64.

Xu R.andWunsch D. (2005). *Survey of Clustering Algorithms*. IEEE Transactions on Neural Networks, 16(3):67-75.

Youmaran R. (2005). *Automatic Measurement of Features in Ultrasound Images of the Eye*, Masters Thesis, School of Information Technology and Engineering, University of Ottawa, Canada.

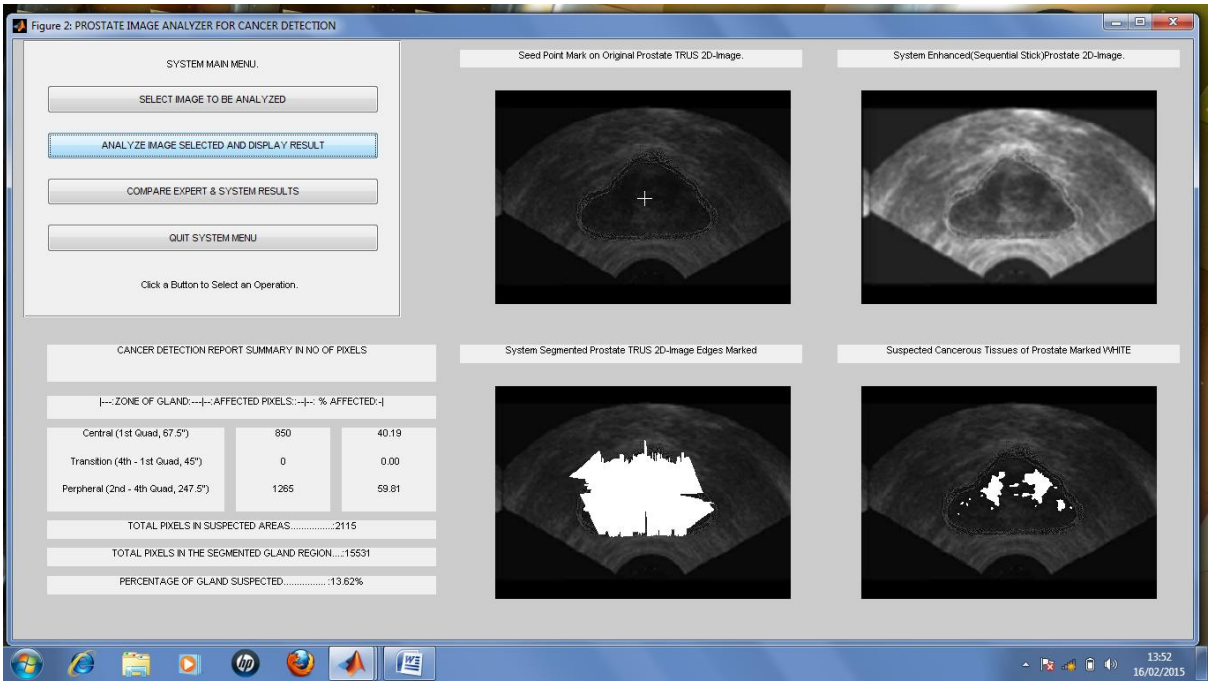
Yu Y., Cheng J., Li J., Chen W. and Chiu B. (2014). *Automatic Prostate Segmentation from Transrectal Ultrasound Images*. IEEE, Bio CAS.2014.6981659.

Yunusoglu M. G. and Selim H. (2013). *Expert Systems with Applications A Fuzzy Rule Based Expert System for Stock Evaluation and Portfolio Construction: Application to Instanbul Stock Exchange,* Expert Syst. Appl. 40(3):286-299.

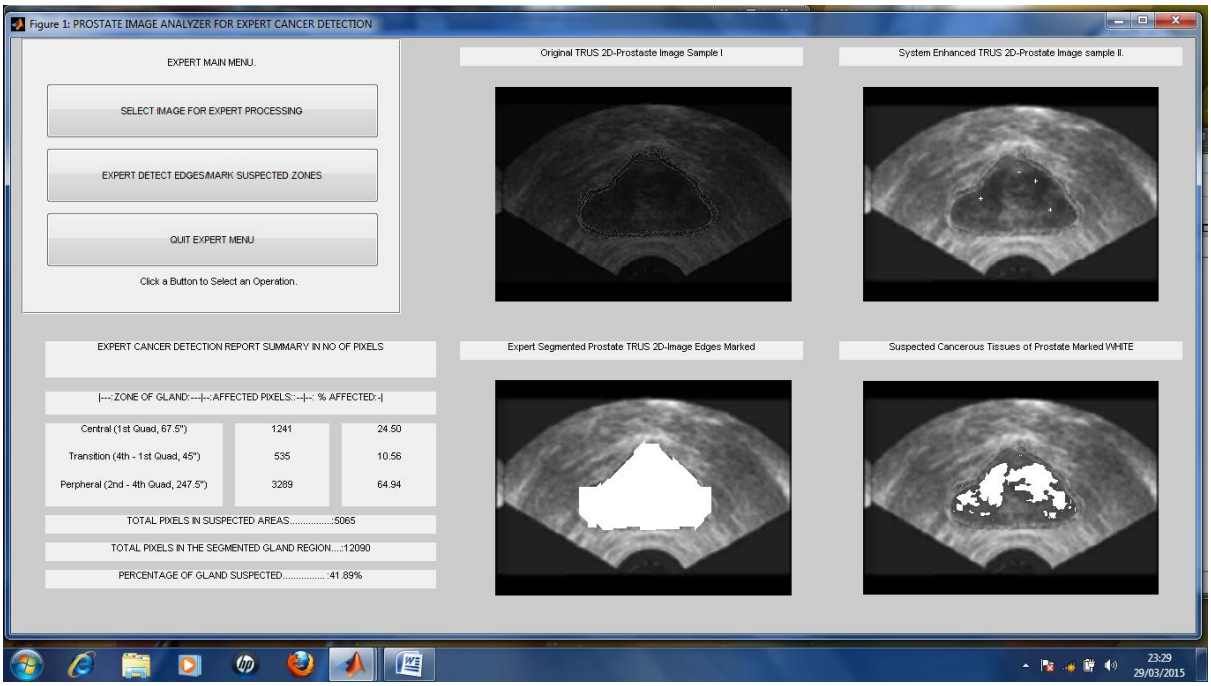
Zhang, Y. J. (2001). *An Overview of Image and Video Segmentation in the last 40 years*, Proceedings of the 6th International Symposium on Signal Processing and Its Applications, pp. 144-151.

APPENDIX A

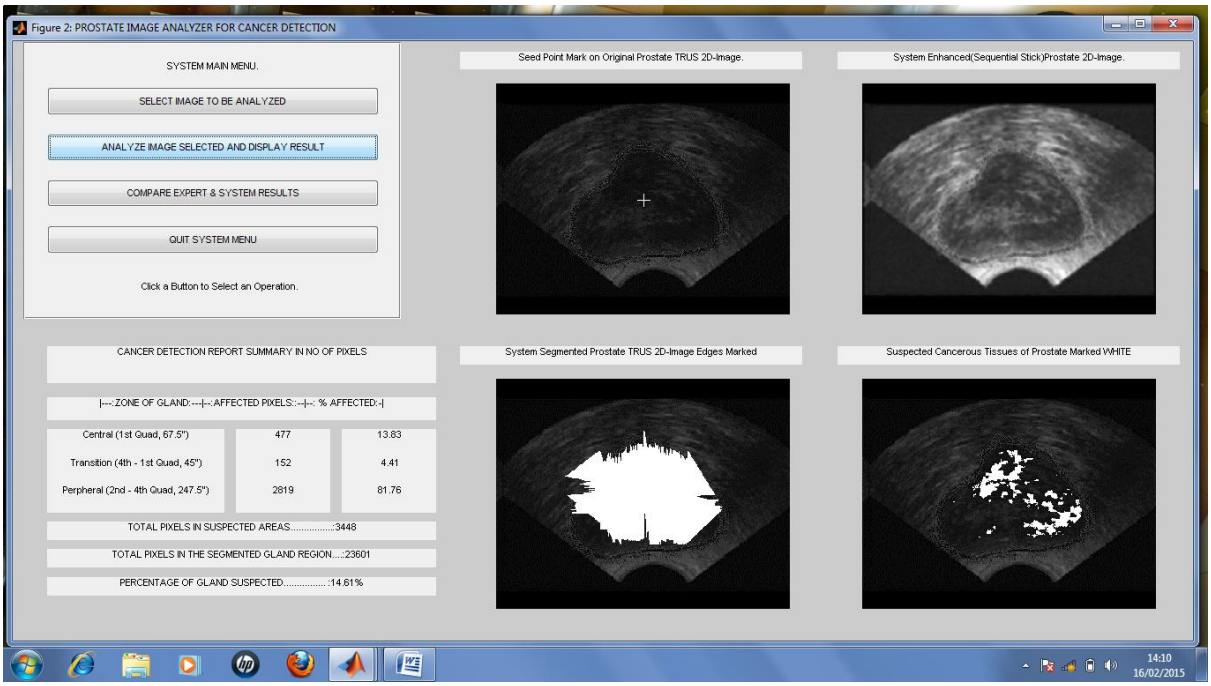
SCREEN SHOTS OF SEGMENTATION /DETECTION RESULT FOR SOME IMAGE SAMPLE FOR BOTH PROPOSED ALGORITHM AND EXPERT



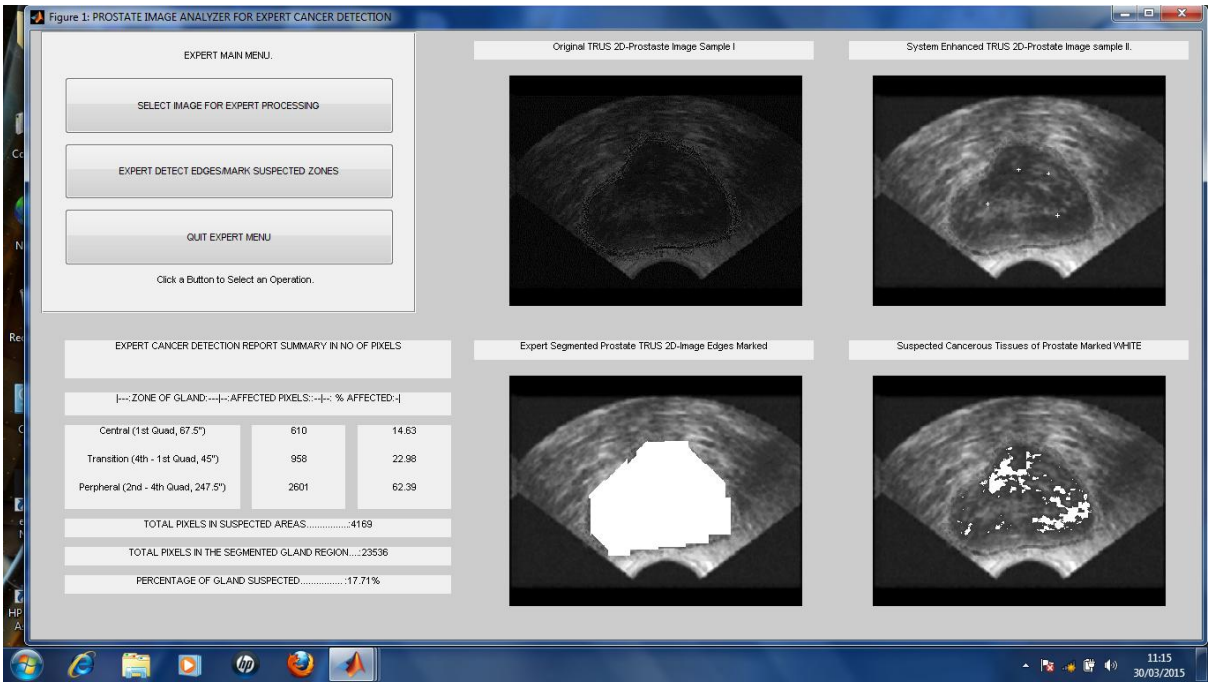
ALGORITHM RESULT – SAMPLE 23



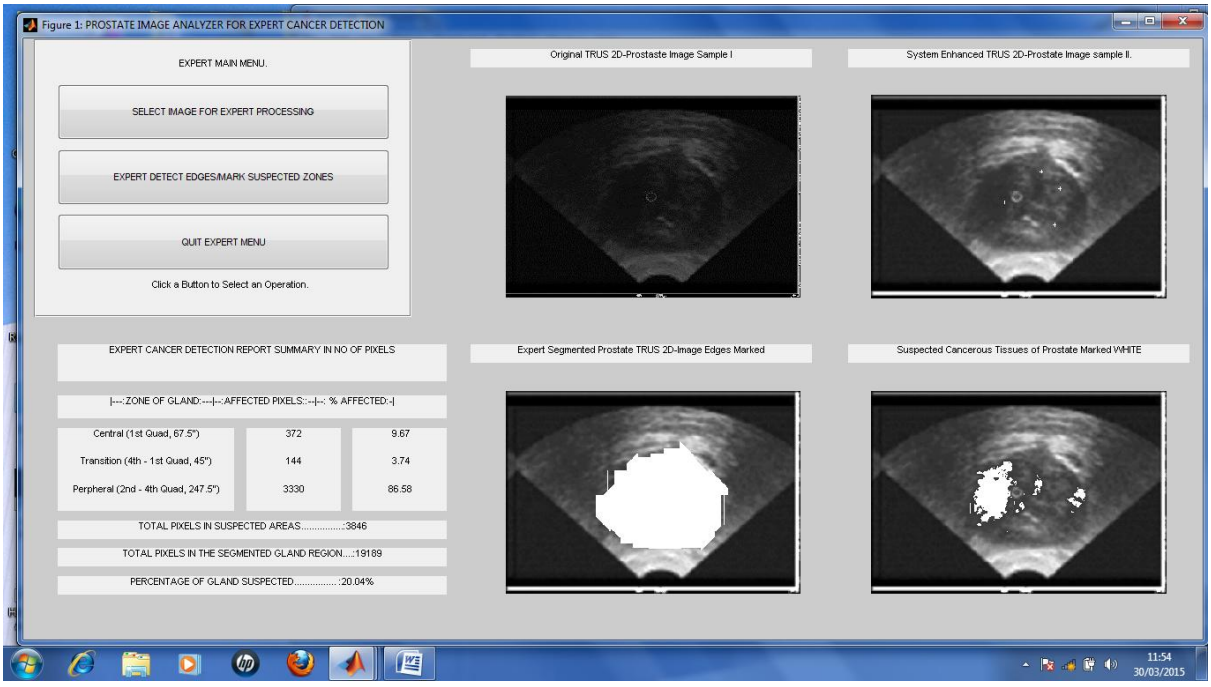
EXPERT RESULT - SAMPLE 23



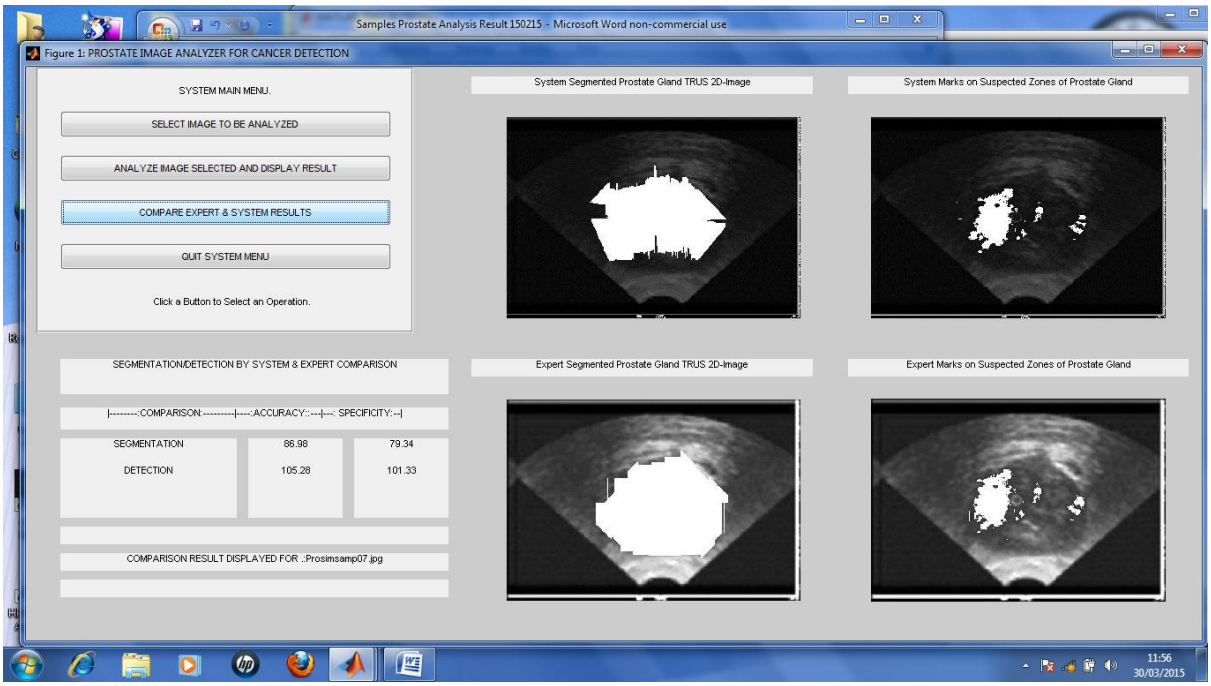
ALGORITHM RESULT - SAMPLE 05



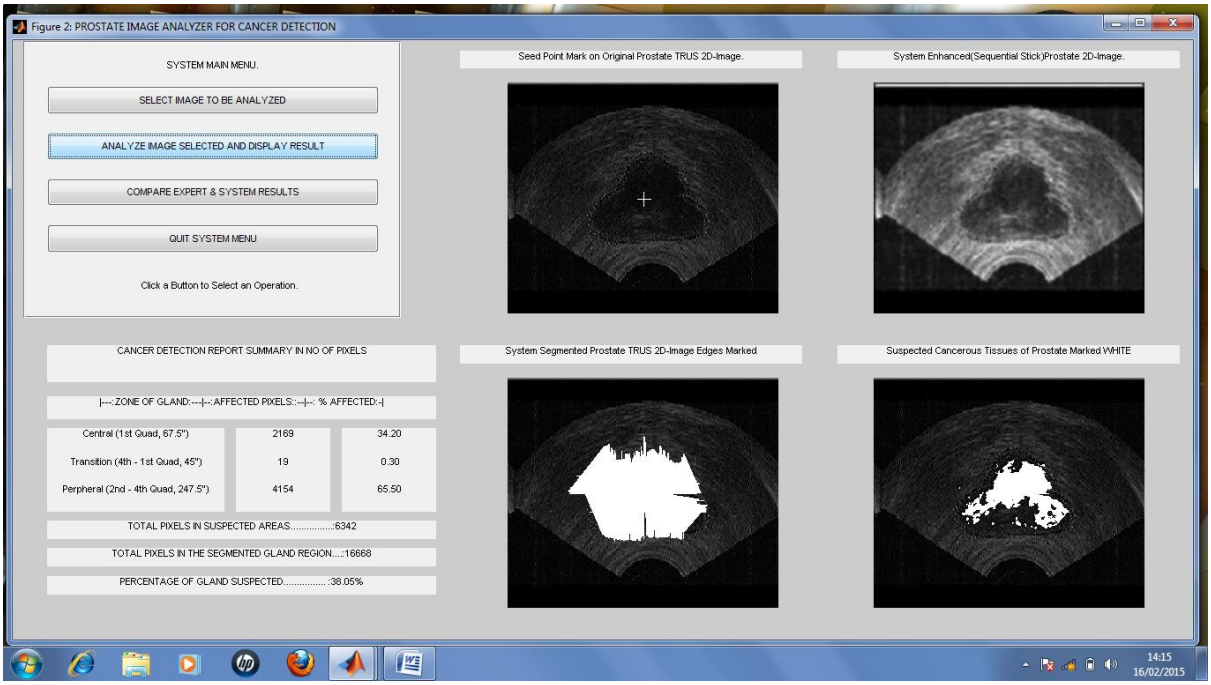
EXPERT RESULT - SAMPLE 05



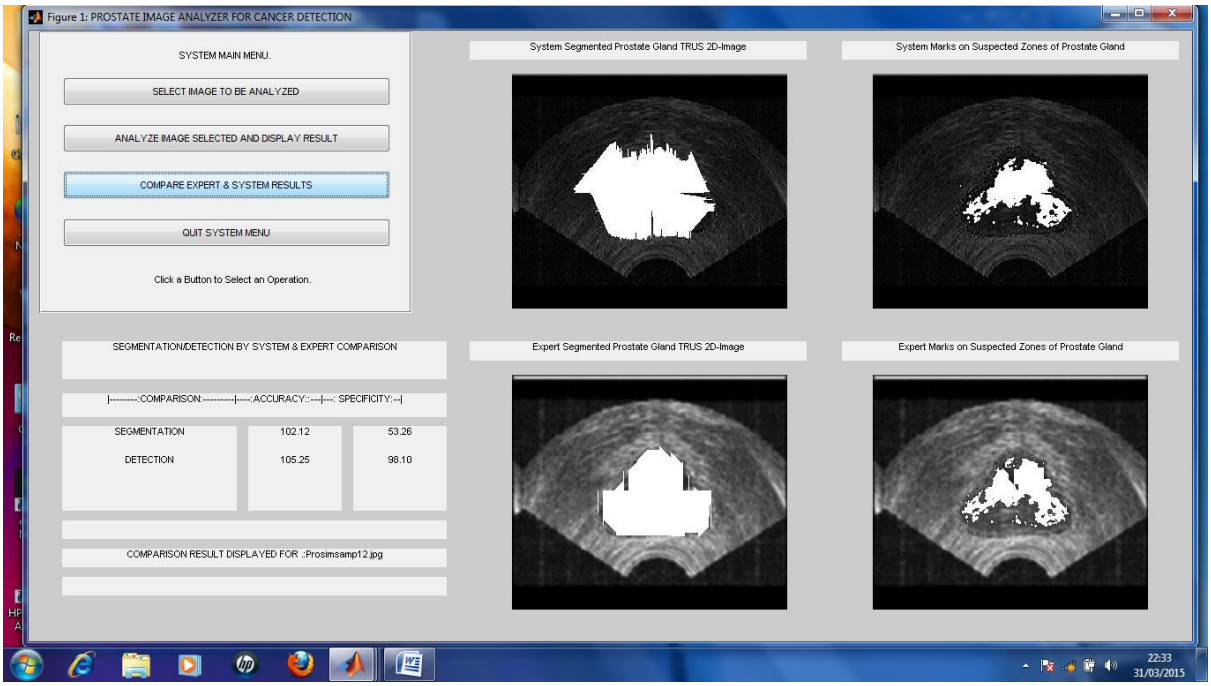
EXPERT RESULT - SAMPLE 07



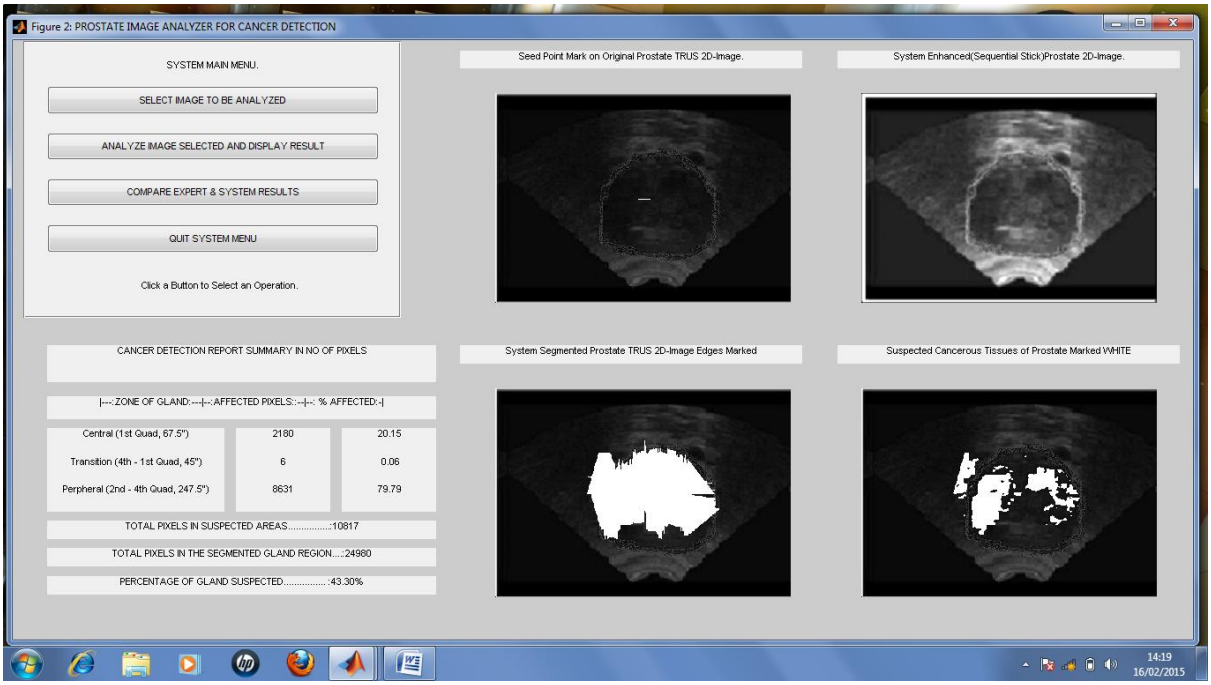
COMPARISON OF SAMPLE 07



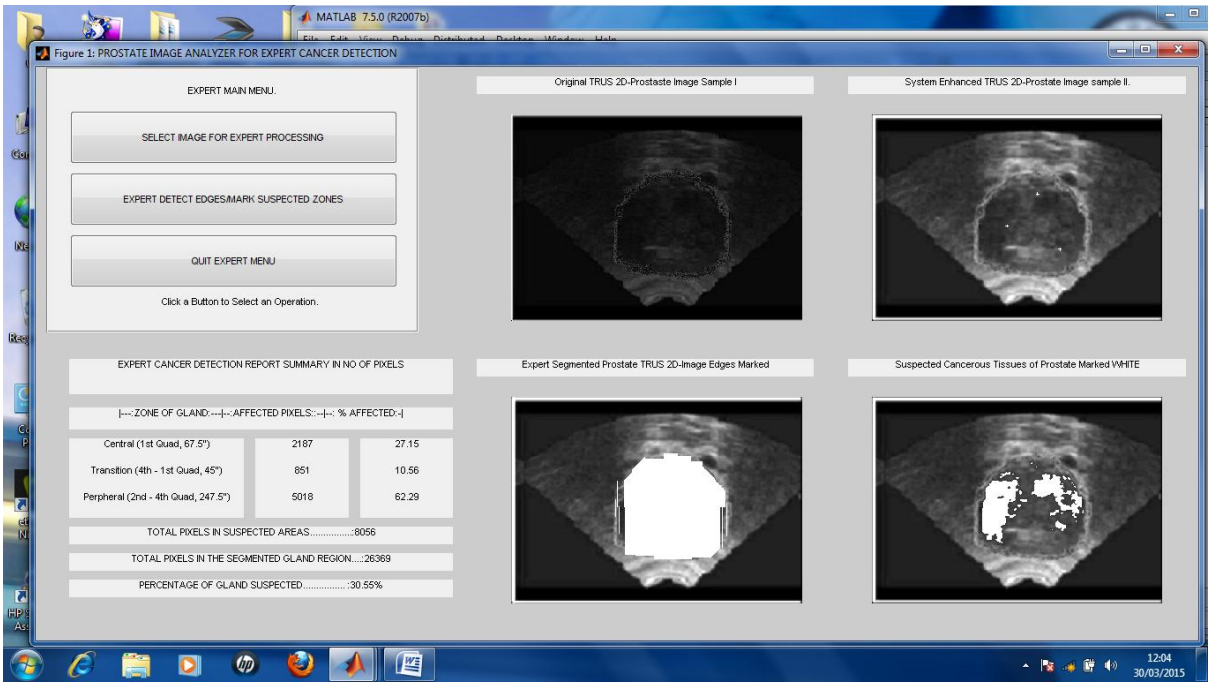
ALGORITHM RESULT - SAMPLE 12



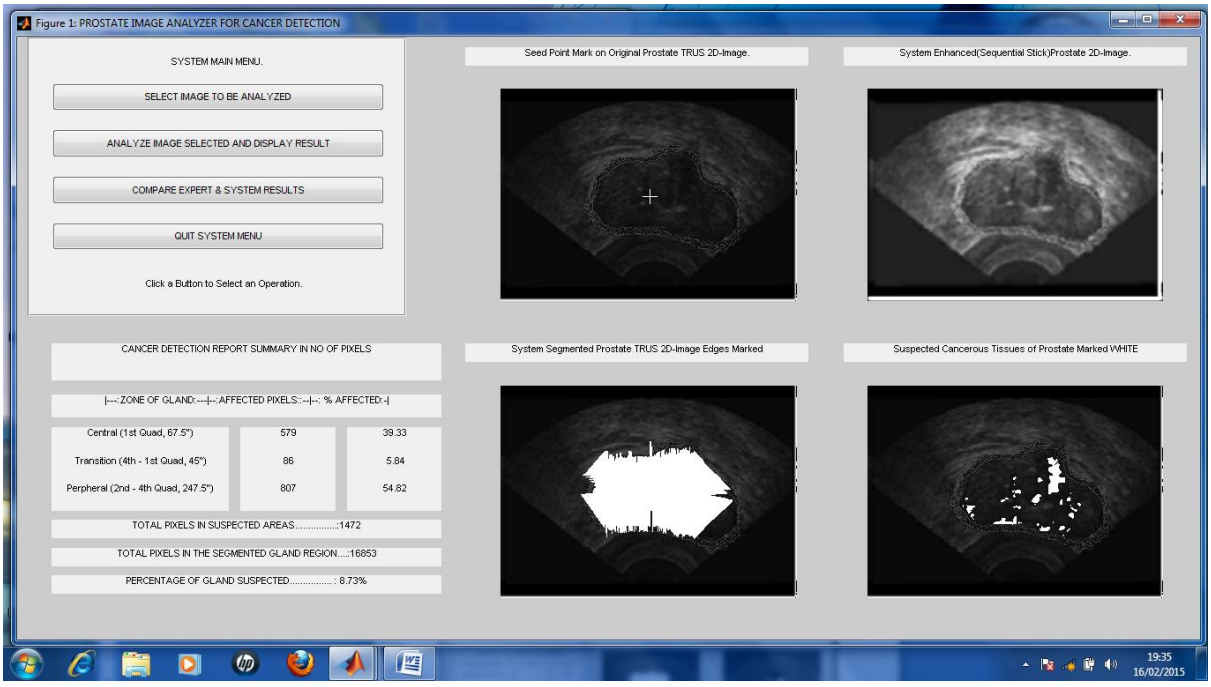
COMPARISON OF SAMPLE12



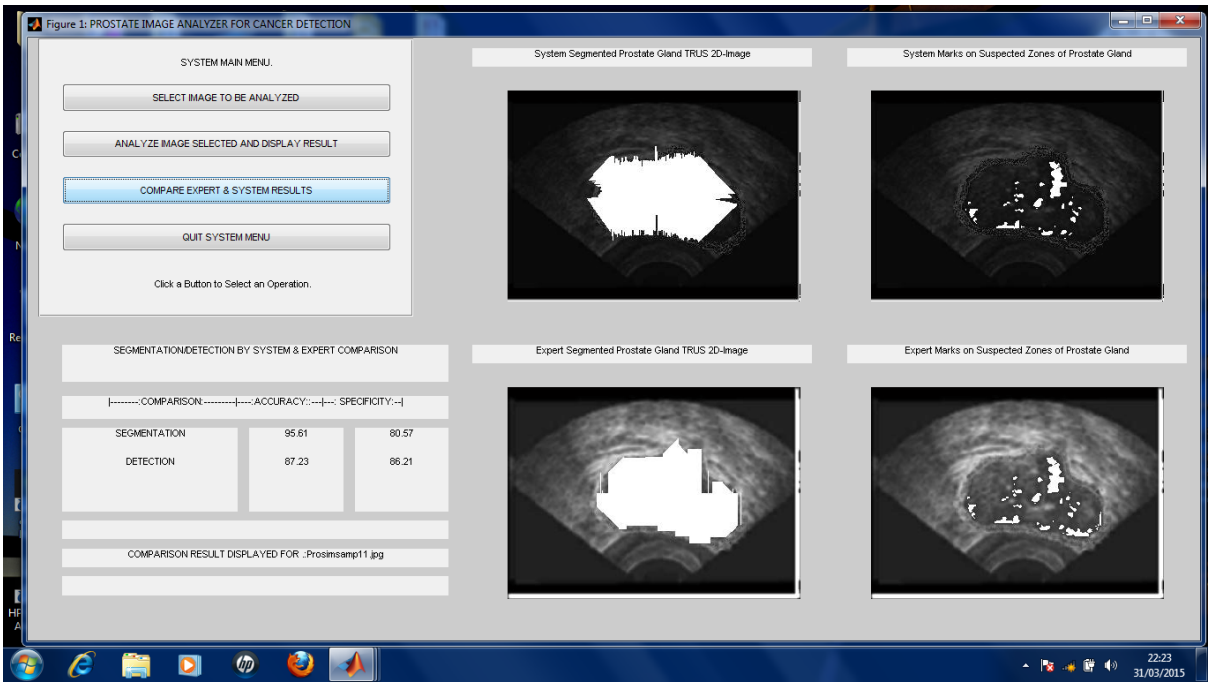
ALGORITHM RESULT - SAMPLE 09



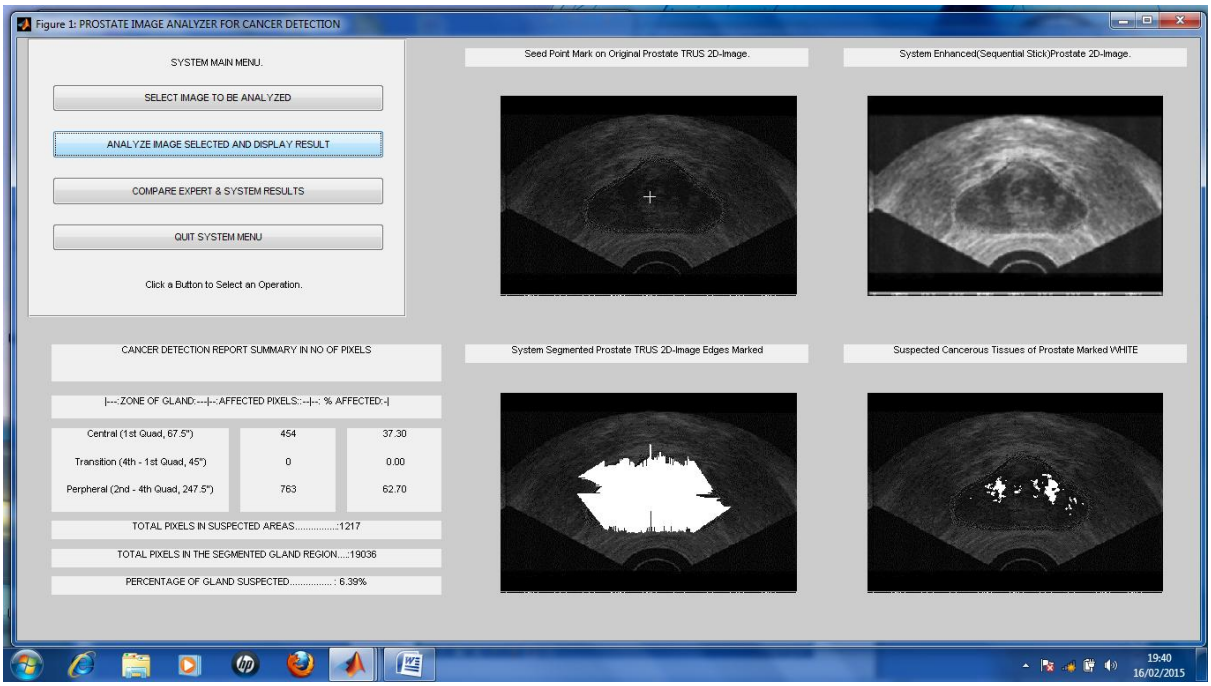
EXPERT RESULT - SAMPLE 09



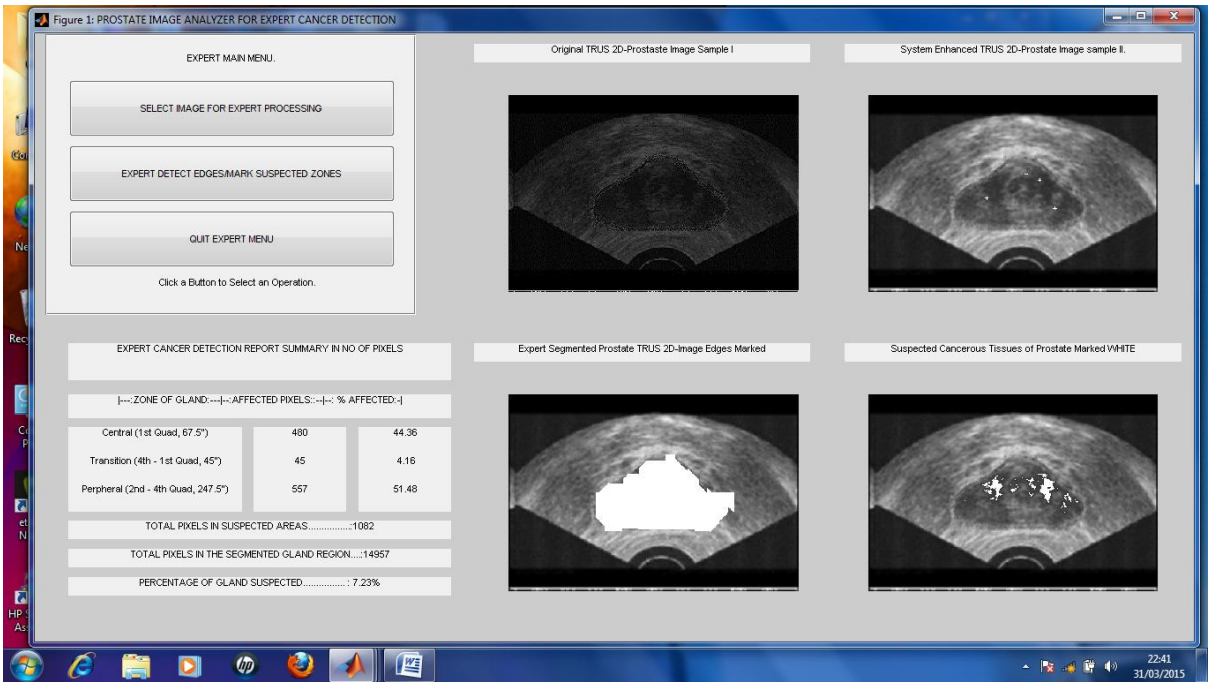
ALGORITHM RESULT - SAMPLE 11



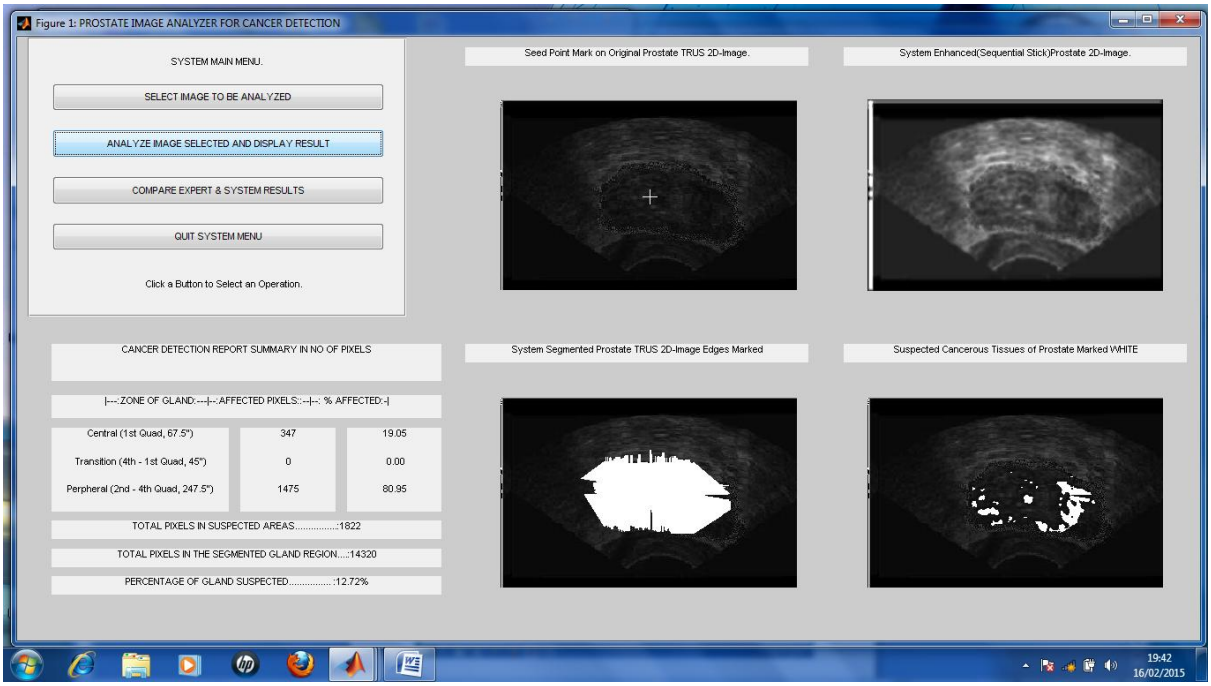
COMPARISON OF SAMPLE 11



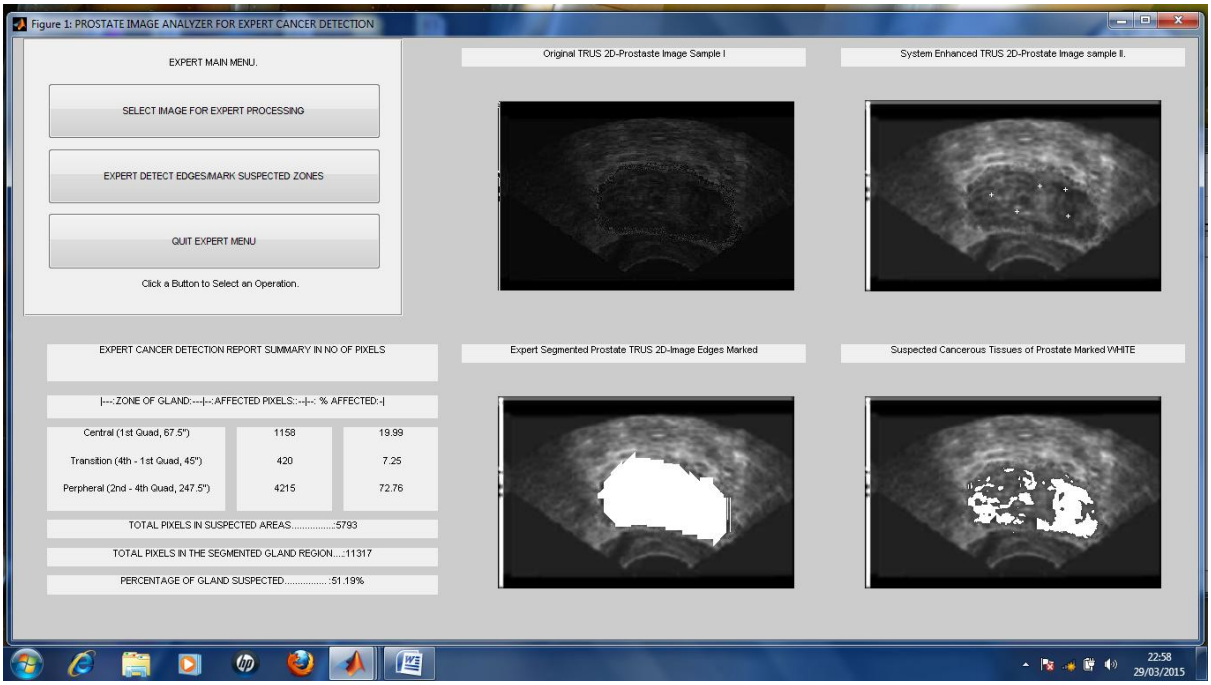
ALGORITHM RESULT - SAMPLE 14



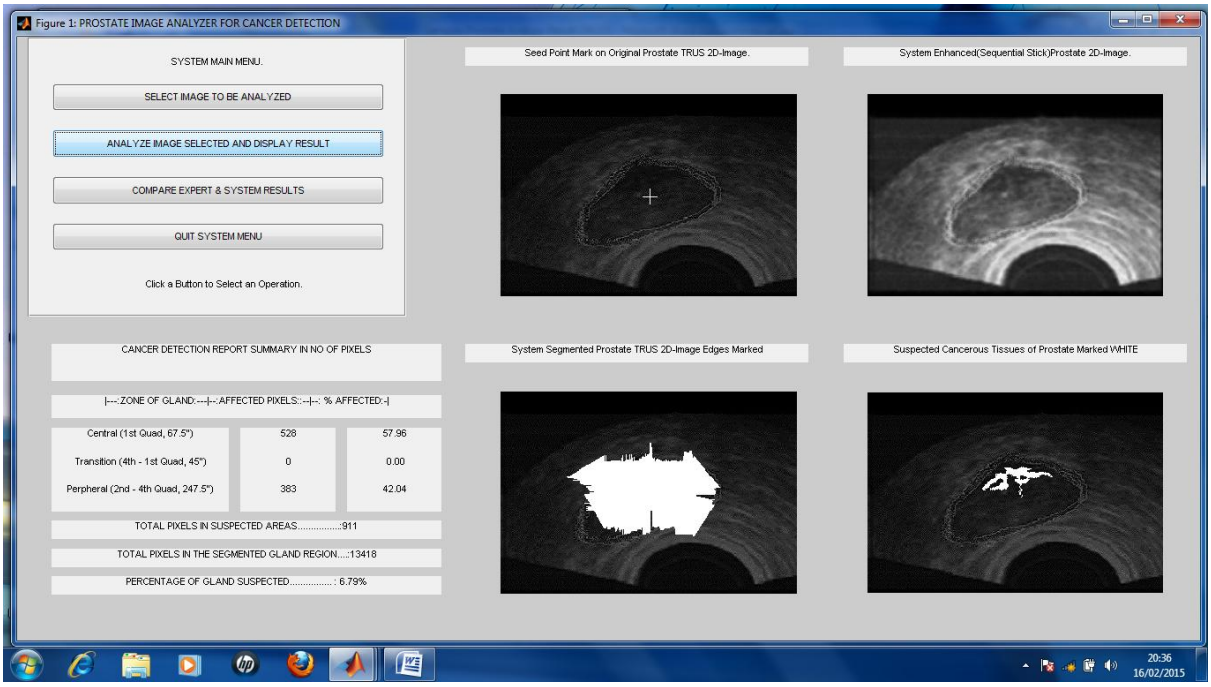
EXPERT RESULT - SAMPLE 14



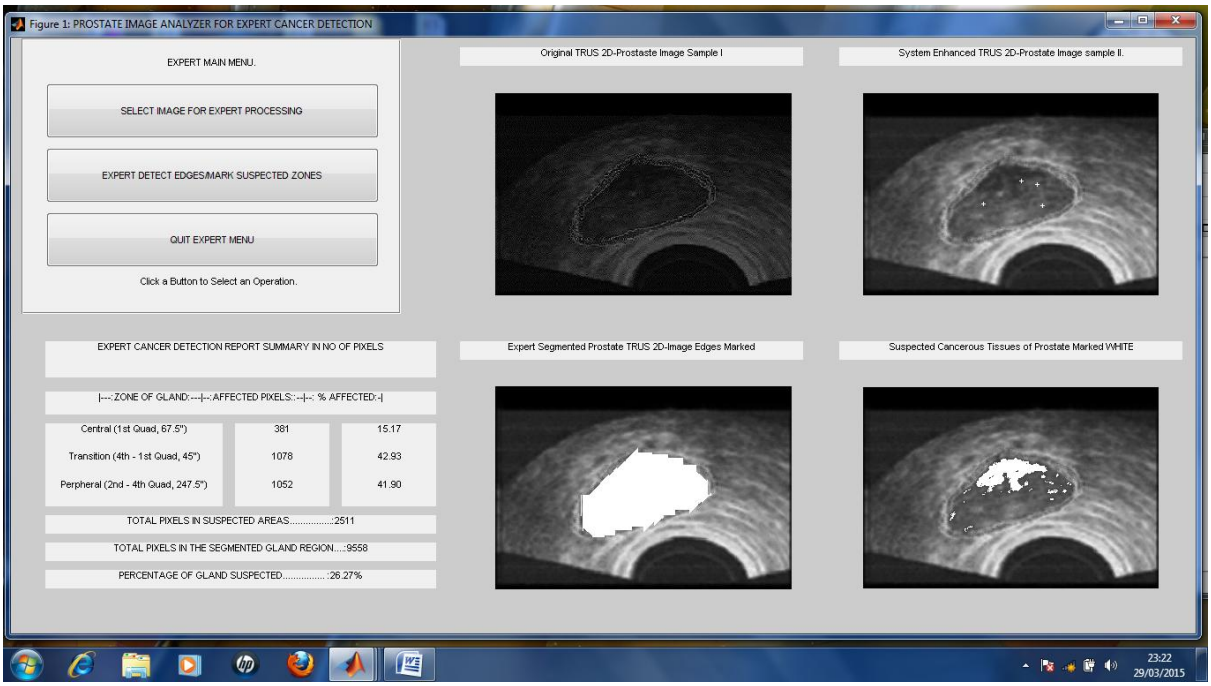
ALGORITHM RESULT - SAMPLE 16



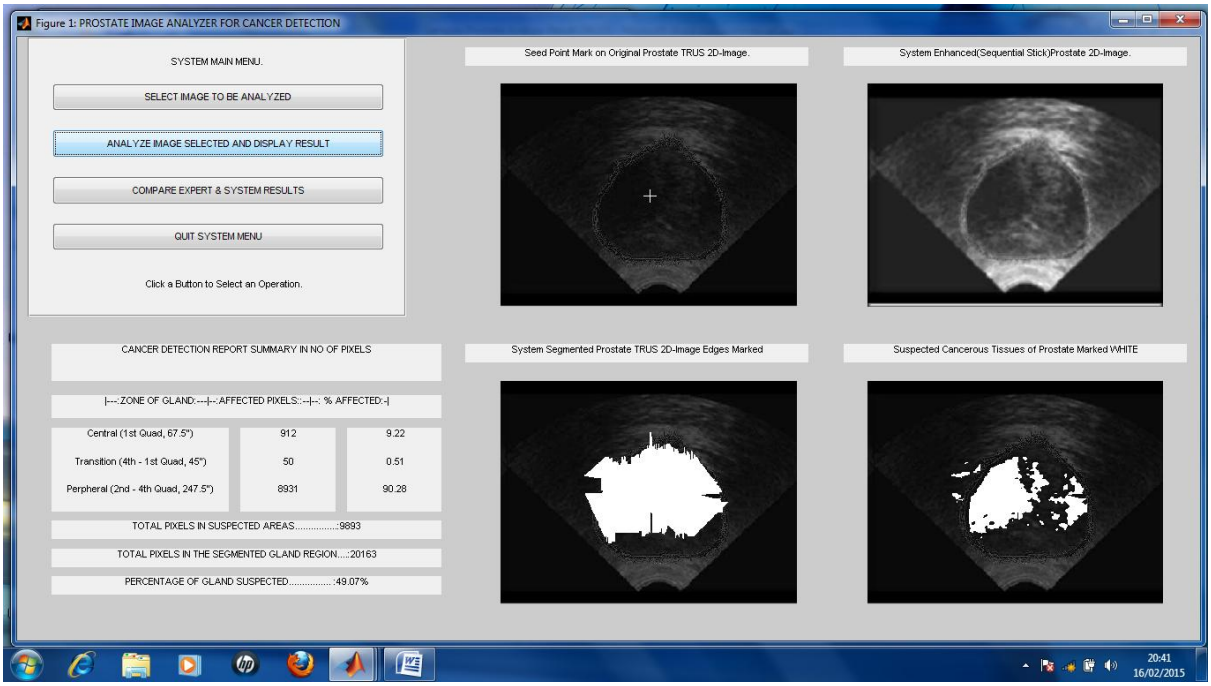
EXPERT RESULT OF SAMPLE 16



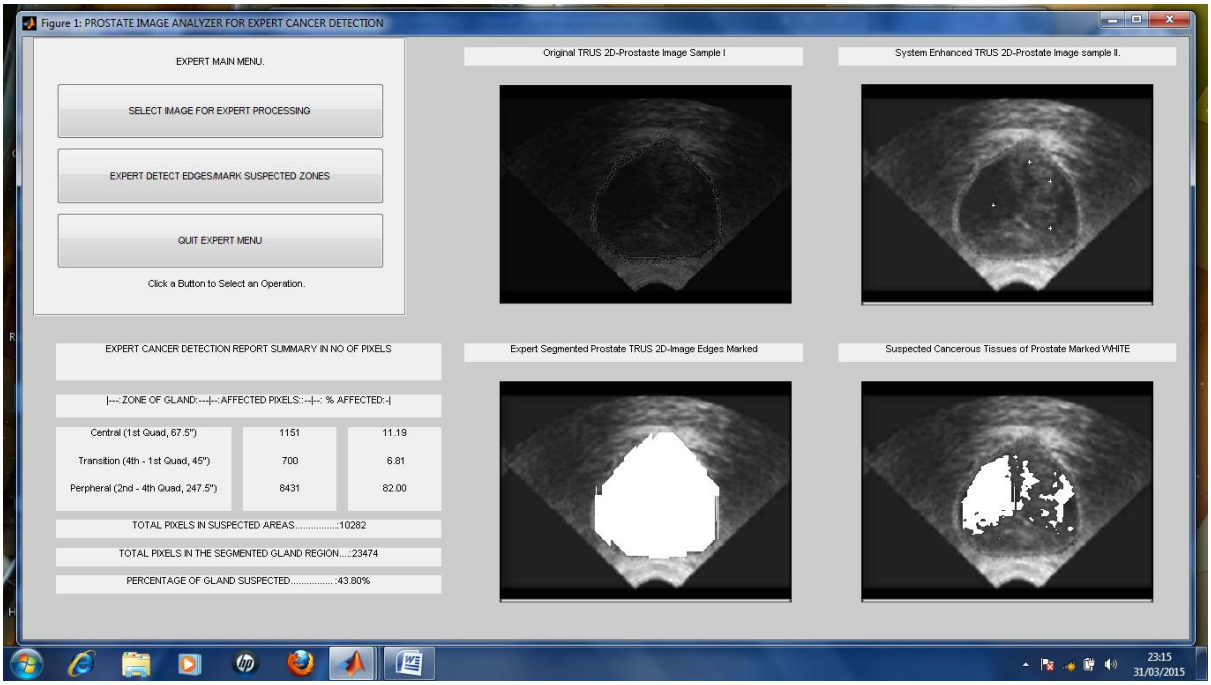
ALGORITHM RESULT - SAMPLE 17



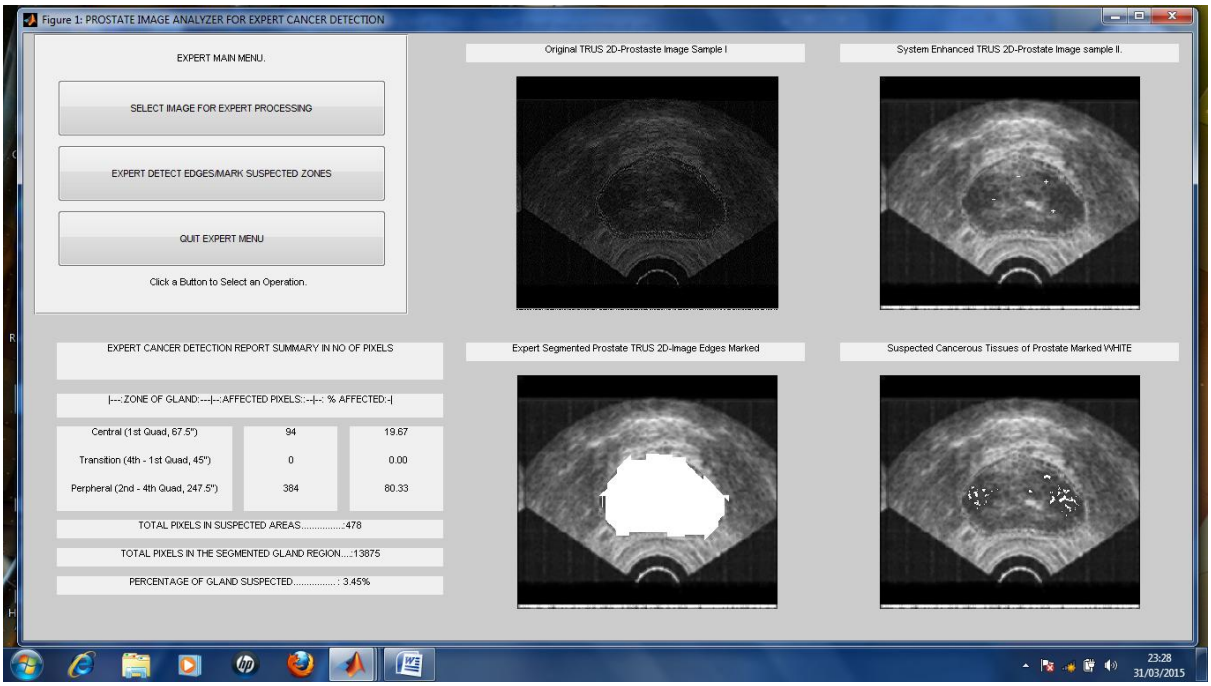
EXPERT RESULT - SAMPLE 17



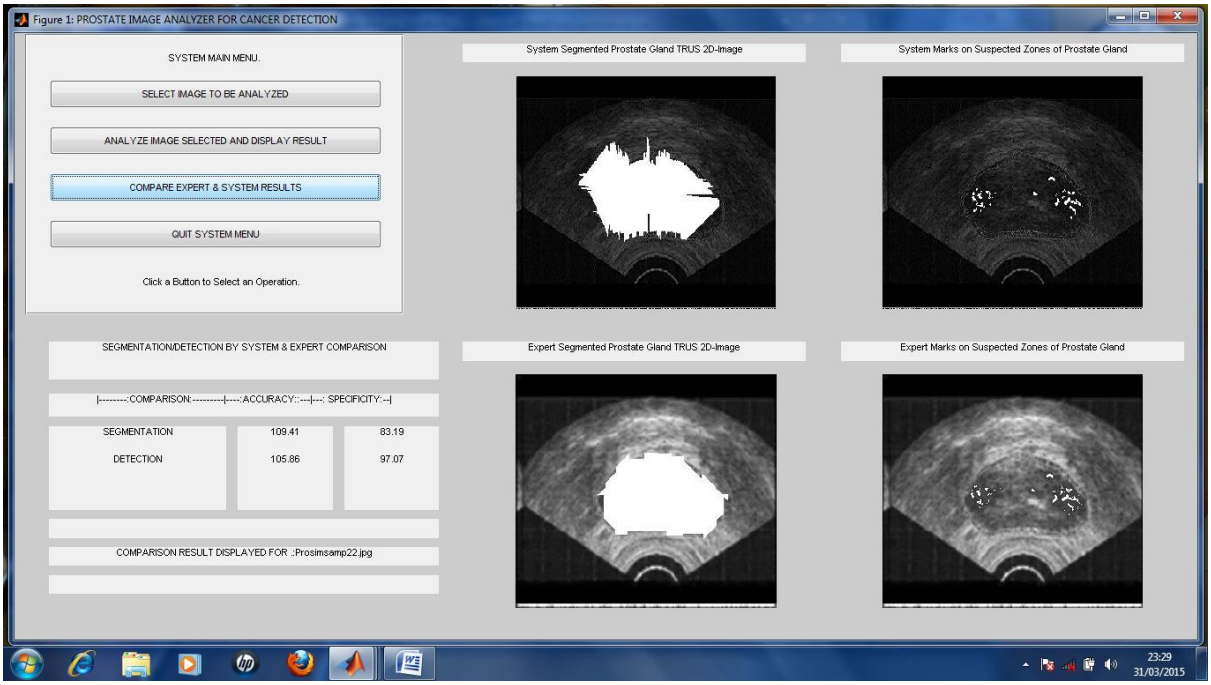
ALGORITHM RESULT - SAMPLE 19



EXPERT RESULT - SAMPLE 19



ALGORITHM RESULT - SAMPLE 22



COMPARISON OF SAMPLE 22

APPENDIX B

LISTING OF MATLAB PROGRAMS CODES FOR THE IMPLEMENTATION OF THE DESIGN FOR PROSTATE SEGMENTATION AND CANCER DETECTION

```
function analyz_analprosm
% ANALYZ_ANALPROSM is the Control Programme for Diagnostic Software
% For Prostate Cancer Using TRUS 2D-images of the Prostate
% Image Segmentation by implementing prior knowledge is used.
% Graphic User Interface tool in MATLAB drives the menu.
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.

% Initialize global variables here
global J L domintrng prost_imsamp;
global h_fig1 h_panel1 h_axes1 h_axes2 h_axes3 h_axes4 h_txts5 h_txts6;
global Centralcnt Transitcnt Periphcnt Centralp Transitp Periphp;
global segcount locmg;
global FSMSYS FMMSYS FSMEXP FMMEXP;
global Accs Specs Accm Specm;

domintrng = 'light';  % dominant intensity range indicator
prost_imsamp = ' '; % keeps original image filena5me and pathname
imagfilenam = ' ';  % keeps only original filename
```

```
diremty = 0; % directory empty is 0 when no jpg image files are stored

% Create and hide the figure as is being created.
h_fig1 = figure('Visible', 'off','Position',[50,50,1000,650],...
'Toolbar','none','MenuBar','none');
h_text = uicontrol(h_fig1,'Style','text',...
'Position',[15,610,310,20],'String',...
'SYSTEM MAIN MENU.');
% Construct the components put in a panel, panel1
h_panel1 = uipanel(h_fig1,'Units','pixels','Position',[10,350,320,300]);
h_menuopt1 = uicontrol(h_panel1,'Style','pushbutton',...
'Position',[20,220,280,30],...
'String','SELECT IMAGE TO BE ANALYZED',...
'TooltipString','Interruptible = off',...
'Interruptible','off',...
'Callback',@analyz_imagselect);
h_menuopt2 = uicontrol(h_panel1,'Style','pushbutton',...
'Position',[20,170,280,30],...
'String','ANALYZE IMAGE SELECTED AND DISPLAY RESULT',...
'TooltipString','Interruptible = off',...
'Interruptible','off',...
'Callback',@analyz_imaganalys);
h_menuopt3 = uicontrol(h_panel1,'Style','pushbutton',...
'Position',[20,120,280,30],...
'String','COMPARE EXPERT & SYSTEM RESULTS',...
'TooltipString','Interruptible = off',...
'Interruptible','off',...
'Callback',@analyz_imagcmpare);
h_menuopt4 = uicontrol(h_panel1,'Style','pushbutton',...
'Position',[20,70,280,30],...
'String','QUIT SYSTEM MENU',...
'TooltipString','Interruptible = off',...
'Interruptible','off',...
'Callback',@analyz_endsyst);
h_text1 = uicontrol(h_panel1,'Style','text',...
'Position',[40,20,250,20],'String',...
'Click a Button to Select an Operation.');
h_txts1 = uicontrol(h_fig1,'Style','text',...
'Position',[380,620,290,20],'String',...
'                                    ');
h_txts2 = uicontrol(h_fig1,'Style','text',...
'Position',[700,620,290,20],'String',...
'                                    ');
h_txts3 = uicontrol(h_fig1,'Style','text',...
'Position',[380,300,290,20],'String',...
'                                    ');
h_txts4 = uicontrol(h_fig1,'Style','text',...
'Position',[700,300,290,20],'String',...
'                                    ');
h_txts5 = uicontrol(h_fig1,'Style','text',...
'Position',[30,280,330,40],'String',...
'                                     ');
h_txts6 = uicontrol(h_fig1,'Style','text',...
'Position',[30,240,330,25],'String',...
'                                     ');
h_txts7 = uicontrol(h_fig1,'Style','text',...
'Position',[30,200,150,30],'String',...
'                                     ');
h_txts7a = uicontrol(h_fig1,'Style','text',...
```

```
'Position',[190,200,80,30],'String',...
'                                    ');
h_txts7b = uicontrol(h_fig1,'Style','text',...
'Position',[280,200,80,30],'String',...
'                                    ');
h_txts8 = uicontrol(h_fig1,'Style','text',...
'Position',[30,170,150,30],'String',...
'                                    ');
h_txts8a = uicontrol(h_fig1,'Style','text',...
'Position',[190,170,80,30],'String',...
'                                    ');
h_txts8b = uicontrol(h_fig1,'Style','text',...
'Position',[280,170,80,30],'String',...
'                                    ');
h_txts9 = uicontrol(h_fig1,'Style','text',...
'Position',[30,140,150,30],'String',...
'                                    ');
h_txts9a = uicontrol(h_fig1,'Style','text',...
'Position',[190,140,80,30],'String',...
'                                    ');
h_txts9b = uicontrol(h_fig1,'Style','text',...
'Position',[280,140,80,30],'String',...
'                                    ');
h_txts10 = uicontrol(h_fig1,'Style','text',...
'Position',[30,110,330,20],'String',...
'                                    ');
h_txts11 = uicontrol(h_fig1,'Style','text',...
'Position',[30,80,330,20],'String',...
'                                    ');
h_txts12 = uicontrol(h_fig1,'Style','text',...
'Position',[30,50,330,20],'String',...
'                                    ');

%
% Create axis for diplaying the images
% axes1 for Displaying original unsegmented image of prostate
% axes2 for displaying segmented image of prostate
% axes3 for displaying original with seed point mark
% axes4 for displaying original with malignant regions marked


h_axes1 = axes('Parent',h_fig1,'Units','pixels',...
'Position',[410,355,250,250]);
h_axes2 = axes('Parent',h_fig1,'Units','pixels',...
'Position',[720,355,250,250]);
h_axes3 = axes('Parent',h_fig1,'Units','pixels',...
'Position',[410,35,250,250]);
h_axes4 = axes('Parent',h_fig1,'Units','pixels',...
'Position',[720,35,250,250]);

% --------------------------------------------------------------------

align(h_panel1, h_text,'Center','None');

% Initialize the GUI.
% Change units to normalized so components resize automatically.
%
set([h_fig1,h_panel1,h_axes1,h_axes2,h_axes3,h_axes4,h_menuopt1,...
```

```
    h_menuopt2,h_menuopt3,h_menuopt4,h_text,h_text1,h_txts1,h_txts2,...
    h_txts3,h_txts4,h_txts5,h_txts6,h_txts7,h_txts8,h_txts9,h_txts10,...
    h_txts7a,h_txts7b,h_txts8a,h_txts8b,h_txts9a,h_txts9b,...
    h_txts11,h_txts12],'Units','normalized');
% Assign the GUI a name to appear in the window title.
set(h_fig1,'Name','PROSTATE IMAGE ANALYZER FOR CANCER DETECTION')
% Move the GUI to the center of the screen.
movegui(h_fig1,'center')
% Make the GUI visible.
set(h_fig1,'Visible','on');
%
% Callbacks for the GUI in h_fig1
% --------------------------------------------------------------------
% Executes when you click the menuopt1 push button.
function analyz_imagselect(hObject,eventdata)
% Disable the other keys.
        set(h_menuopt2,'Enable','off'); set(h_menuopt3,'Enable','off')
% Call the routine to automate selection of an image file.
      [imagfilenam,diremty] = analyz_imageselc;
      cla(h_axes3,'reset');
if diremty == 0;
        directnam = cd; % Read current directory pathname
        warndlg ({directnam;...
'Please Copy .jpg Prostate Image Files into this Directory';...
'!!! Then Try this Option Aagain !!! '},...
'No Image Files For Selection !');
else
if imagfilenam == ' ';
            set(h_txts5,'String',...
'User selected No File to Analyze.');
            set(h_txts2,'String',...
'                                                    ');
else
            set(h_txts5,'String',...
             strcat('User Selected:...', imagfilenam, '...Image File'));
            I = imread(prost_imsamp);
            cla(h_axes1,'reset');
            axes(h_axes1); imshow(I);
            cla(h_axes2,'reset'); cla(h_axes3,'reset');

            set(h_txts6,'String',...
'                                                    ');
            set(h_txts7,'String',...
'                                              ');
            set(h_txts7a,'String',...
'                                    ');
            set(h_txts7b,'String',...
'                                    ');
            set(h_txts8,'String',...
'                                              ');
            set(h_txts8a,'String',...
'                                    ');
            set(h_txts8b,'String',...
'                                    ');
            set(h_txts9,'String',...
'                                              ');
            set(h_txts9a,'String',...
'                                    ');
            set(h_txts9b,'String',...
```

```
'                                          ');
            set(h_txts10,'String',...
'                                                  ');
            set(h_txts11,'String',...
'                                                  ');
            set(h_txts12,'String',...
'                                                  ');

            cla(h_axes4,'reset');
            set(h_txts4,'String',...
'                                                        ');

            set(h_txts3,'String',...
'                                                        ');
            set(h_txts2,'String',...
'                                                        ');
            set(h_txts1,'String',...
'Selected Original/Unsegmented Prostate TRUS 2D-Image.');
end
end
% Enable the other keys
      set(h_menuopt2,'Enable','on'); set(h_menuopt3,'Enable','on');
end
% --------------------------------------------------------------------
% Executes when you click the menuopt2 push button.
function analyz_imaganalys(hObject,eventdata)

%
% Disable the other keys.
        set(h_menuopt1,'Enable','off'); set(h_menuopt3,'Enable','off')
% check flag for image filename selection
if imagfilenam == ' ';
            set(h_txts5,'String',...
'User selected No File to Analyze.');
else
             cla(h_axes2,'reset');cla(h_axes3,'reset');cla(h_axes4,'reset')
             set(h_txts5,'String',...
'!!! Wait Analysis of Image is in Progress ....... !!!');

% Call routines to start analysis of selected image.
% J = analyz_remspecmk(prost_imsamp);% remove specs mark
% call routine to remove labels in ultrasound images for
% manual segmentation/interpretation

             J = analyz_remvlabl;% remove labels

%cla(h_axes1,'reset');
             axes(h_axes1); imshow(J);

% Call to determine dominant intensity range of image & set
% parameters for image processing accordingly
             [domintrng]=analyz_malgrang(J);

             cla(h_axes3,'reset');cla(h_axes4,'reset')

% Call routiine to enhance image removing noise
             K = analyz_seqstick(J); % enhance image by seq stick
```

```
            cla(h_axes2,'reset'); % Clear any dislay of previous result
            axes(h_axes2); imshow(K); %
            set(h_txts2,'String',...
'System Enhanced(Sequential Stick)Prostate 2D-Image.');

            set(h_txts5,'String',...
            strcat('Wait Analysis of ', '...Image File in Progress'));
% Call seed point determination routine
            [L,sedptrow,sedptcol] = analyz_seedlocn(K);%

            cla(h_axes1,'reset'); % Clear any dislay of previous result
            axes(h_axes1); imshow(L);
            set(h_txts1,'String',...
'Seed Point Mark on Original Prostate TRUS 2D-Image.');

            set(h_txts5,'String',...
            strcat('Seed Point row:.:',int2str(sedptrow),':.col:.:',...
               int2str(sedptcol),':..Dominant Intensity Id of this..',...
'Prostate Image is..:', domintrng, ...
':..Please Wait Analysis of Image File in Progress...'));

            axes(h_axes1); imshow(L);

% Call routine to start region growing by radial scanning
% and detection of malignant tissues at the same time
            [PS, PM]=analyz_imageanal(K,sedptrow,sedptcol);

            cla(h_axes3,'reset'); % Clear any dislay of previous result
            axes(h_axes3); imshow(PS);
            set(h_txts3,'String',...
'System Segmented Prostate TRUS 2D-Image Edges Marked');

            cla(h_axes4,'reset'); % Clear any dislay of previous result
            axes(h_axes4); imshow(PM);
            set(h_txts4,'String',...
'Suspected Cancerous Tissues of Prostate Marked WHITE');

% Call routine to save segmented & malignant image file
% with original filenames & store results in a summary table
            analyz_resfilsave(imagfilenam,PS,PM);

% Display Report of Cancer/malignacy

            format bank;
if segcount > 0;
                malpctge = ((locmg * 100)/segcount);
else
                malpctge = 0;
end

            set(h_txts5,'String',...
'CANCER DETECTION REPORT SUMMARY IN NO OF PIXELS');

            set(h_txts6,'String',...
            strcat('|---:ZONE OF GLAND',':---|--:AFFECTED PIXELS:',...
':--|--: % AFFECTED:-|'));
```

```
            set(h_txts7,'String','Central (1st Quad, 67.5")');
            set(h_txts7a,'String',int2str(Centralcnt));
            set(h_txts7b,'String',sprintf('%5.2f',Centralp));

            set(h_txts8,'String','Transition (4th - 1st Quad, 45")');
            set(h_txts8a,'String',int2str(Transitcnt));
            set(h_txts8b,'String',sprintf('%5.2f',Transitp));

            set(h_txts9,'String','Perpheral (2nd - 4th Quad, 247.5")');
            set(h_txts9a,'String',int2str(Periphcnt));
            set(h_txts9b,'String',sprintf('%5.2f',Periphp));

            set(h_txts10,'String',...
            strcat('TOTAL PIXELS IN SUSPECTED AREAS.................:',...
                   int2str(locmg)));

            set(h_txts11,'String',...
            strcat('TOTAL PIXELS IN THE SEGMENTED GLAND REGION....:',...
                  int2str(segcount)));

            set(h_txts12,'String',...
            strcat('PERCENTAGE OF GLAND SUSPECTED.............. :',...
                  sprintf('%5.2f',malpctge),'%'));

%close(h_wait); % to allowed to fuction when analyz_ is actv
%    Enable the other keys
end
        set(h_menuopt1,'Enable','on'); set(h_menuopt3,'Enable','on')
end
% --------------------------------------------------------------------
% Executes when you click the menuopt3 push button.
function analyz_imagcmpare(hObject,eventdata)
% Disable the other keys.
        set(h_menuopt1,'Enable','off'); set(h_menuopt2,'Enable','off')
        cla(h_axes1,'reset'); cla(h_axes2,'reset')

% Call routine to start comparing expert and system results.
        Accs = 0.0; Specs = 0.0; Accm = 0.0; Specm = 0.0;

        [imagfilenam,nofilepar] = analyz_imagecmpr;

if nofilepar == 0;
        IFSMSYS = imread(FSMSYS);  IFMMSYS = imread(FMMSYS);
        IFSMEXP = imread(FSMEXP);  IFMMEXP = imread(FMMEXP);

        set(h_txts1,'String',...
'System Segmented Prostate Gland TRUS 2D-Image');
        cla(h_axes1,'reset'); % Clear any dislay of previous result
           axes(h_axes1); imshow(IFSMSYS);

        set(h_txts2,'String',...
'System Marks on Suspected Zones of Prostate Gland');
        cla(h_axes2,'reset'); % Clear any dislay of previous result
            axes(h_axes2); imshow(IFMMSYS);

        set(h_txts3,'String',...
```

```
'Expert Segmented Prostate Gland TRUS 2D-Image');
        cla(h_axes3,'reset'); % Clear any dislay of previous result
            axes(h_axes3); imshow(IFSMEXP);

        set(h_txts4,'String',...
'Expert Marks on Suspected Zones of Prostate Gland');
        cla(h_axes4,'reset'); % Clear any dislay of previous result
            axes(h_axes4); imshow(IFMMEXP);
end
            set(h_txts5,'String',...
'SEGMENTATION/DETECTION BY SYSTEM & EXPERT COMPARISON');

             set(h_txts6,'String',...
             strcat('|--------:COMPARISON',':---------|----:ACCURACY:',...
':---|---: SPECIFICITY:--|'));

             set(h_txts7,'String','SEGMENTATION');
             set(h_txts7a,'String',sprintf('%5.2f',Accs));
             set(h_txts7b,'String',sprintf('%5.2f',Specs));

             set(h_txts8,'String','DETECTION');
             set(h_txts8a,'String',sprintf('%5.2f',Accm));
             set(h_txts8b,'String',sprintf('%5.2f',Specm));

             set(h_txts11,'String',...
             strcat('COMPARISON RESULT DISPLAYED FOR .:',...
                   imagfilenam));

% close(h_wait); % to allowed to fuction when analyz_ is runing
% Enable the other keys
        set(h_menuopt1,'Enable','on'); set(h_menuopt2,'Enable','on')
end
% --------------------------------------------------------------------
% Executes when you click the menuopt4 push button.
function analyz_endsyst(hObject,eventdata)
% close all open figures
        close all
end
% --------------------------------------------------------------------
% Create and update a waitbar.
%   function create_update_waitbar
%   h_wait = waitbar(0,'Please wait Process in Progress...',...
%            'Position',[150,100,270,50]);
%   end
% Create and update a waitbar.
function create_update_waitbar
    h_wait = waitbar(0,'Please wait...',...
'Position',[150,100,270,50]);
% 'CloseRequestFcn',@close_waitbar);
for i=1:10000,
if ishandle(h_wait)
            waitbar(i/10000,h_wait)
else
break
end
end
% When waitbar reaches max, close it.
if ishandle(h_wait)
```

```
            close(h_wait)
end
end
% --------------------------------------------------------------------
% C%lose the waitbar. Executes in response to BREAK and CLOSE commands.
% function close_waitbar(hObject,eventdata)
%     delete(gcbf)
% end
% --------------------------------------------------------------------
end

function [PS, PM]=analyz_imageanal(K,sedptrow,sedptcol)

% ANALYZ_IMAGEANAL is the Routine that automates the analysis of selected
% Image file - TRUS 2D-images of the Prostate stored in the working
% Disk directory of MATLAB.
% The File types possible/accepatble include jpg, tif, png or bmp.
% However jpg is prefered for this work
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.

% Develop procedure to analyze the image by perfoming the following sub-
% rountine functions:
%   Commence image segmentation and spoting malignant tissues
%         (pixels) by radial/axial scanning while growing gland region
%
% --------------------------------------------------------------------

global segimcoodtbl segcount maligcoodtbl locmg Segimedgetbl loced;
global thrshcntmx directn daxis Cs Rs imsize Scl Srw Ecl Erw;
global J L prost_imatrx rowm rowmin rowmax colm colmin colmax;
global maligcoodtbln locmgn zonelkuptabl h_axes1 h_txts5;
global loceddfb1 locedbhc1 loceddfb2 locedbhc2 edsubtsze;
global locedcea1 locedagd1 locedcea2 locedagd2 edgtbsize;
global Segedgetblcea1 Segedgetblagd1 Segedgetbldfb1 Segedgetblbhc1;
global Segedgetblcea2 Segedgetblagd2 Segedgetbldfb2 Segedgetblbhc2;

loceddfb1 = 0; locedbhc1 =0; locedcea1 =0; locedagd1 =0;
loceddfb2 = 0; locedbhc2 =0; locedcea2 =0; locedagd2 =0;

loced=0; segcount=0; % initialize these global counters
locmgn = 0; locmg=0;
prost_imatrx=K;
[rowm,colm] = size(K);
imsize=rowm*colm;

segimcoodtbl = zeros(imsize,3); % segmented pixels coord table
% Table => row, column, directn: 1- 'CEA', 2- 'AGD', 3- 'DFB', 4- 'BHC'
% For segmented pixels in region of prostate gland of image
maligcoodtbl = zeros(imsize,5); % malignant coordnate table
% Table => row, column, directn, malig pixel intensity, zoneid
% where directn: 1- 'CEA', 2- 'AGD', 3- 'DFB', 4- 'BHC',
% where zoneid = 'Central, Transition, Peripheral
% For Suspected malignant pixels within segmented region of prostate
zonelkuptabl = zeros(imsize,3); % zone lookup pixels coord table
% Table => row, column, zoneid: '1 - Central', '2-Peripheral',
%                               '3 - Transition'
```

```
% For looking up zones of detected malignant pixels in the region
% of the prostate gland

maligcoodtbln = zeros(imsize,3);
% Table => row, column,zoneid
% where zoneid = 'Central, Transition, Peripheral

edgtbsizei = idivide(int32(imsize), int32(24));
edgtbsize = double(edgtbsizei);
Segimedgetbl=zeros(edgtbsize,3);% total edge table

edsubtszei = idivide(int32(imsize), int32(42));
edsubtsze = double(edsubtszei);
Segedgetblcea1 = zeros(edsubtsze,3);% edgetable for radial dir - CEA
Segedgetblcea2 = zeros(edsubtsze,3);% 2

Segedgetblagd1 = zeros(edsubtsze,3);% edgetable for radial dir - AGD
Segedgetblagd2 = zeros(edsubtsze,3);% 2

Segedgetbldfb1 = zeros(edsubtsze,3);% edgetable for radial dir - DFB
Segedgetbldfb2 = zeros(edsubtsze,3);% 2

Segedgetblbhc1 = zeros(edsubtsze,3);% edge table for radial dir - BHC
Segedgetblbhc2 = zeros(edsubtsze,3);% 2


%Assign seedpoint coordinate for use
Rs = sedptrow; Cs = sedptcol;

% Setup a loop to scan in 4 radial/axial directions

for dirnumb = 1 : 1 : 8,
switch dirnumb;
case 1;
% Start Radial/Axial scanning/grow gland regn frm seed point/ detect for
% CEA-direction (-1,0; ...) about the seed point Rs, Cs left upward.

        directn = 'CEA';
        daxis = 'Rowwise';
        thrshcntmx = Rs - rowmin +1; % lenght of edgtranstbl
        Scl = -1; Ecl = colmin; Srw = -1; Erw = rowmin;
%disp('..Radial Scanning - CEA axis - Left Up from Seed');
%disp (daxis); % Segtabl = Segedgetblcea2; Segcnt = locedcea2;
        analyz_scanrowse; %(Segtabl,Segcnt);%Scan rowwise for each column
% Call routine- false edges in rowwise edge table
        analyz_remflsedr;

case 2;

        directn = 'CEA';
        daxis = 'Columnwise';
        thrshcntmx = Cs - colmin + 1; % lenght of edgtranstbl
        Scl = -1; Ecl = colmin; Srw = -1;
        Erw = rowmin + ((Rs - rowmin) - idivide(Rs - rowmin, int32(3)));
%disp('..Radial Scanning - CEA axis - Left Up from Seed ');
%disp (daxis); %Segtabl = Segedgetblcea1; Segcnt = locedcea1;
        analyz_scancolwse; %(Segtabl,Segcnt); %Scan columwise for each row
```

```
% Call routine- false edges in columwise edge table
        analyz_remflsedc;
% Call routine to accumulate the total edges and pixels segmented
        analyz_acumpxsgcd;

case 3;
% Start Radial/Axial scanning/grow gland regn frm seed point/detectn for
% AGD-direction (0, 1; ...) about the seed point Rs, Cs right upward.

        directn = 'AGD';
        daxis = 'Rowwise';
        thrshcntmx = Rs - rowmin +1; % lenght of edgtranstbl
        Scl = 1; Ecl = colmax; Srw = -1; Erw = rowmin;
%disp('..Radial Scanning - AGD axis - Right Up from Seed');
%disp (daxis); %Segtabl = Segedgetblagd2; Segcnt = locedagd2;
        analyz_scanrowse; %(Segtabl,Segcnt);%Scan columwise for each row
% Call routine- false edges in rowwise edge table
        analyz_remflsedr;

case 4;

        directn = 'AGD';
        daxis = 'Columnwise';
        thrshcntmx = (colmax - Cs + 1); % lenght of edgtranstbl
        Scl = 1; Ecl = colmax; Srw = -1;
        Erw = rowmin + ((Rs - rowmin) - idivide(Rs - rowmin, int32(3)));
%disp('..Radial Scanning - AGD axis - Right Up from Seed ');
%disp (daxis); %Segtabl = Segedgetblagd1; Segcnt = locedagd1;
        analyz_scancolwse; %(Segtabl,Segcnt); %Scan columwise for each row
% Call routine- false edges in columwise edge table
        analyz_remflsedc;
% Call routine to accumulate the total edges and pixels segmented
        analyz_acumpxsgcd;

case 5;
% Start Radial/Axial scanning to grow gland region from seed point
% and detect for DFB-direction (1, 1; ...) about the seed point Rs, Cs.

        directn = 'DFB';
        daxis = 'Rowwise';
        thrshcntmx = (rowmax - Rs+1); % lenght of edgtranstbl
        Scl = 1; Ecl = colmax; Srw = 1; Erw = rowmax;
%disp('..Radial Scanning - DFB axis - Right Down from Seed ..');
%disp (daxis); % Segtabl = Segedgetbldfb2; Segcnt = loceddfb2;
        analyz_scanrowse; %(Segtabl,Segcnt);% Scan columwise for each row
% Call routine- false edges in rowwise edge table
        analyz_remflsedr;

case 6;

        directn = 'DFB';
        daxis = 'Columnwise';
        thrshcntmx = (colmax - Cs + 1); % lenght of edgtranstbl
        Scl = 1; Ecl = colmax; Srw = 1;
        Erw = rowmax - ((rowmax - Rs) - idivide(rowmax - Rs, int32(3)));
%disp('..Radial Scanning - DFB axis - Right Down from Seed .');
%disp (daxis); % Segtabl = Segedgetbldfb1; Segcnt = loceddfb1;
        analyz_scancolwse; %(Segtabl,Segcnt); %Scan columwise for each row
```

```
% Call routine- false edges in columwise edge table
        analyz_remflsedc;
% Call routine to accumulate the total edges and pixels segmented
        analyz_acumpxsgcd;

case 7;
% Start Radial/Axial scanning to grow gland region from seed point
% and detect for BHC-direction (1, - 1; ...) about the seed point Rs, Cs.

        directn = 'BHC';
        daxis = 'Rowwise';
        thrshcntmx = (rowmax - Rs+1); % lenght of edgtranstbl
        Scl = -1; Ecl = colmin; Srw = 1; Erw = rowmax;
%disp('..Radial Scanning - BHC axis - Left Down from Seed .');
%disp (daxis); %Segtabl = Segedgetblbhc2; Segcnt = locedbhc2;
        analyz_scanrowse; %(Segtabl,Segcnt);%Scan columwise for each row
% Call routine- false edges in rowwise edge table
        analyz_remflsedr;

otherwise; % case 8

        directn = 'BHC';
        daxis = 'Columnwise';
        thrshcntmx = (Cs - colmin + 1); % lenght of edgtranstbl
        Scl = -1; Ecl = colmin; Srw = 1;
        Erw = rowmax - ((rowmax - Rs) - idivide(rowmax - Rs, int32(3)));
        analyz_scancolwse; %(Segtabl,Segcnt); %Scan columwise for each row
% Call routine- false edges in columwise edge table
        analyz_remflsedc;
% Call routine to accumulate the total edges and pixels segmented
        analyz_acumpxsgcd;

end

end


% Use the segmented edge tables and mark the location of
% the edge with white to differentiate in the original image

% Get the table and work on the matrix PS to be displayed with the
% white on the original image as image with boundary segmented

set(h_txts5,'String',...
strcat('!!Please Wait, System Compiling Detected Gland Pixels -',...
 directn,'- direction - ',daxis));

axes(h_axes1); imshow(L);

PS=J;
for  edcnt = 1 : segcount
      pxr = segimcoodtbl(edcnt,1);  pxc = segimcoodtbl(edcnt,2);
      PS(pxr,pxc) = 255;
end

% Use the spotted malignant location tables and mark the location of
% the edge with white to differentiate in the original image
```

```
PM=J;
for malcnt = 1 :locmg
    pxr = maligcoodtbl(malcnt,1);  pxc = maligcoodtbl(malcnt,2);
    PM(pxr,pxc)=255;
end

% check locmg and locmgn to set a flag for no malignant found

% Call routine to determine the zones of the
% identified malignant sections
analyz_genzontbl;

% and display report table
%               CANCER DETECTION REPORT FROM PROSTATE TRUS 2D-IMAGE
%    ------------------------------------------------------------------
%    |   No of Pixels |  Lobe/Zone of Gland  | % of Zone Affected  |
%    |-----------------------------------------------------------------
%    |                |   Central            |                     |
%    |                |   Peripheral         |                     |
%    |                |   Posterior          |                     |
%    ------------------------------------------------------------------
%    | Total Pixels in the Gland =                                   |
%    | % of Gland Affected =                                         |
%    |-----------------------------------------------------------------
%    Summary:---% of gland image is detected as suspected cancer cells
%

% Call routine to use sequential stick and statistical technique
% to detect malignant pixels
%
%    analyz_dthpecopxnew(J)
%
% PN=J;
% for malcntn = 1 :locmgn
%    disp('!!! Please Wait, Compiling Detected Malignant Pixels !!!');
%    pxr = maligcoodtbln(malcntn,1); pxc = maligcoodtbln(malcntn,2);
%    PN(pxr,pxc)=255;
% end

end

function analyz_scanrowse
%(Segtabl,Segcnt)
global L h_axes1 thrshcntmx thrshcnt edgflg edgtranstbl
global Cs Scl Ecl Rs Srw Erw h_txts5 daxis directn

set(h_txts5,'String',...
strcat('!!Please Wait, System Locating Edges in -',...
 directn,'- direction - ',daxis));

axes(h_axes1); imshow(L);

for Cn = Cs : Scl : Ecl,
% generate neighbours along this radial path
% to search for inclusion as region and/or malignat pixel
```

```
    edgtranstbl=zeros(thrshcntmx,6); %initialz table for de col searchg
    thrshcnt = 0; edgflg = 0;
for Rn = Rs : Srw : Erw,
% Call subroutine to compute neighbors mean deviation for
% edge determination parameter and add to table accordingly
% determine whether it belongs to a malignant pixel

       analyz_chkedgpxd(Rn, Cn);
end%Next row along column being searched

% Call subroutine to determine the edge nearest to the gland
% borders along the current column and store the pixels bound by
% it from seed point as belonging to the grown region

    analyz_setedgpx(Rn, Cn);
end%Next Column left upward from seed ponit of image matrix

end
%--------------------------------------------------------------------

%--------------------------------------------------------------------
function analyz_scancolwse

%(Segtabl,Segcnt)

global L h_axes1 thrshcntmx thrshcnt edgflg edgtranstbl
global Cs Scl Ecl Rs Srw Erw h_txts5 daxis directn

set(h_txts5,'String',...
strcat('!!Please Wait, System Locating Edges in -',...
 directn,'- direction - ',daxis));

 axes(h_axes1); imshow(L);


for Rn = Rs : Srw : Erw,
% generate neighbours along this radial path
% to search for inclusion as region and/or malignat pixel

    edgtranstbl=zeros(thrshcntmx,6); %initialz table for de row searchg
    thrshcnt = 0; edgflg = 0;
for Cn = Cs : Scl : Ecl,
% Call subroutine to compute neighbors mean deviation for
% edge determination parameter and add to table accordingly
% determine whether it belongs to a malignant pixel

       analyz_chkedgpxd(Rn, Cn);
end%Next column along row being searched

% Call subroutine to determine the edge nearest to the gland
% borders along the current row and store the pixels bound by
% it from seed point as belonging to the grown region

    analyz_setedgpx(Rn, Cn);
end%Next Row left upward from seed ponit of image matrix
```

```
%disp (Segtabl);disp('Edge table for column found, seqtl - rows');
%disp ('Edge count tab1= '); disp (Segcnt );
%reply = input('Press Enter to Continue!! ', 's');
%       if isempty(reply)
%            % okay
%       end

end

function [domintrng]=analyz_malgrang(I)
% ANALYZ_DOMIMALG is the Routine  to determine malignant
%          intensity pixel range  and dominant intensity
%          for this image sample
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
% ------------------------------------------------------------------------

global curtintmin curtintmax nextintmin nextintmax malgintrg;
global rowm4 rowmax rowmin colm5 colmax colmin h_axes1 h_txts5;

% Obtains the dimensions of input image matrix say (m,n)
[rowm,colm] = size(I);

%Mask out image elements portions not to be considered for seed point
%using RULE 3 to propose a boundary. The result is stored in a table
%Seedpointbl(loc,3) to store intensity, row, col

rowm4 = idivide(int32(rowm), int32(4));
rowmaxi = rowm - rowm4 + 1;%rowmax = rowm -(1/4 * (rowm)) + 1; integer
rowmax = double(rowmaxi); %double precision
%rowmax is submatrix maximum row selected from ROW of image matrix of ROW
rowmin = double(rowm4); %(1/4 * (rowm));
%rowmin is submatrix minimum row selected from ROW of image matrix of ROW

colm5 = idivide(int32(colm), int32(5));
colmaxi = colm - colm5 + 1;% colmax = colm -(1/5 * (colm)) + 1;
colmax = double(colmaxi);
%colmax is submatrix maximum Col selected from COL of image matrix of COL
colmin = double(colm5); %(1/5 * (colm));
%colmin is submatrix maximum col selected from COL of image matrix of COL

minintst=0; maxintst=255; thrshint=1;
samprang=maxintst-minintst; samplsiz=samprang/thrshint;
freqdisttbl(samplsiz,2)=zeros;

samplival=minintst;locn=1;

while locn <= samplsiz;
  freqcnt=0;
for coln=colmin:colmax,

for rown=rowmin:rowmax,
if (I(rown,coln)- samplival) == 0;
         freqcnt = freqcnt + 1;
```

```
end
end

end
% store intensity and freq count
  freqdisttbl(locn,1)=samplival; freqdisttbl(locn,2)=freqcnt;

% get next sample of intensity val to accumm its freq.
  samplival=samplival+thrshint; locn=locn+1;
end

freqtabsort = sortrows(freqdisttbl,1);
% sort rows with intensity, column 1 in ascending order
% highest intensity is in last item

% Note intensities 0 to 95 is chosen as dark gray level range
% 96 to 255 as the light gray range

dakrngcnt = 0;litrngcnt = 0;
for intslm = 1 : 1 : 255 ; % check each table for the int range

if (freqtabsort(intslm,1)>=0 && freqtabsort(intslm,1)<= 95);
              dakrngcnt = int32(dakrngcnt)...
                  + int32(freqtabsort(intslm,2));
else
if (freqtabsort(intslm,1)>=96 && freqtabsort(intslm,1)<= 255);
              litrngcnt  = int32(litrngcnt)...
                  + int32(freqtabsort(intslm,2));
end
end



end
set(h_txts5,'String',...
 strcat('!!Please Wait, Dominant Parameters Processing in Progress!!'));

axes(h_axes1); imshow(I);


rngratio = idivide(litrngcnt,dakrngcnt);

%disp ('litrngcnt = '); disp(litrngcnt);
%disp ('dakrngcnt = '); disp(dakrngcnt);
%disp ('light/dark = '); disp(rngratio);

if (rngratio > 2);% light dominant gray level image selected
        domintrng = 'Light';
        curtintmin = 70;  curtintmax = 180;
        nextintmin = 90;  nextintmax = 255;
        malgintrg = 50; % % malignancy max range of intensity

else%dark dominant gray level image selected
        domintrng = 'Dark';
        curtintmin = 30;  curtintmax = 150;
      nextintmin = 90;  nextintmax = 255;
        malgintrg = 55; % malignancy max range of intensity
```

```
end


end
%
% ----------------------------------------

function analyz_chkedgpxd(Rn, Cn)
% ANALYZ_CHKEDGMA is the Subroutine that determines possible edge
% pixel by producing two differences in intensity
% prsdev betwen previous location and curent,
% nextdev betwen next location and next. These shall be used by next
% subroutine for selecting the positve difference  sections
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
% ------------------------------------------------------------------
global thrshcnt edgtranstbl newpxldev oldpxldev;
global prost_imatrx directn Cs Rs daxis;

%-------------------------------------------------------------------

% EDGE & MALIGNANT PARAMETERS TABLE GENERATION FOR NEIGBORS
% OF A PIXEL, AND LOCATION ACCUMULATION AND STORAGE


if strcmp(daxis,'Columnwise') ;% x-axis scanning for edge pixel

if (Cn == Cs);
if strcmp(directn,'CEA') || strcmp(directn,'BHC');
% movement to zero of Cn
            nextpxlint =  int32(prost_imatrx(Rn,Cn));
            curtpxlint =  int32(prost_imatrx(Rn,Cn+1));
            oldpxldev = nextpxlint - curtpxlint;
else
% movement to max of Cn
% strcmp(directn,'DFB') || strcmp(directn,'AGD');
               nextpxlint =  int32(prost_imatrx(Rn,Cn));
               curtpxlint =  int32(prost_imatrx(Rn,Cn-1));
               oldpxldev = nextpxlint - curtpxlint;
end
else
        oldpxldev = newpxldev; % assign former new as old

if strcmp(directn,'CEA') || strcmp(directn,'BHC');
% movement to zero of Cn
            nextpxlint =  int32(prost_imatrx(Rn,Cn-1));
            curtpxlint =  int32(prost_imatrx(Rn,Cn));
else
% strcmp(directn,'DFB') || strcmp(directn,'AGD');
% movement to max of Cn
            nextpxlint =  int32(prost_imatrx(Rn,Cn+1));
            curtpxlint =  int32(prost_imatrx(Rn,Cn));
end
```

```
end

else
% strcmp(daxis,'Rowwise') - y-axis scanning for edge pixel

if (Rn == Rs);
if strcmp(directn,'CEA') || strcmp(directn,'AGD');
% movement to zero of Rn
            nextpxlint =  int32(prost_imatrx(Rn,Cn));
            curtpxlint =  int32(prost_imatrx(Rn+1,Cn));
            oldpxldev = nextpxlint - curtpxlint;
else
% movement to max of Rn
% strcmp(directn,'BHC') || strcmp(directn,'DFB');
               nextpxlint =  int32(prost_imatrx(Rn,Cn));
               curtpxlint =  int32(prost_imatrx(Rn-1,Cn));
               oldpxldev = nextpxlint - curtpxlint;
end
else
        oldpxldev = newpxldev; % assign former new as old

if strcmp(directn,'CEA') || strcmp(directn,'AGD');
% movement to zero of Rn
            nextpxlint =  int32(prost_imatrx(Rn-1,Cn));
            curtpxlint =  int32(prost_imatrx(Rn,Cn));
else
% strcmp(directn,'BHC') || strcmp(directn,'DFB');
% movement to max of Rn
            nextpxlint =  int32(prost_imatrx(Rn+1,Cn));
            curtpxlint =  int32(prost_imatrx(Rn,Cn));
end

end

end


   newpxldev = nextpxlint - curtpxlint; % get a new pixel devitn

% save this current pixel particulars for examination for edge
% store the pixel row,col,deviatn values for
% a possible edge of gland region on this row
          thrshcnt = thrshcnt + 1;   edgtranstbl(thrshcnt,1)= Rn;
          edgtranstbl(thrshcnt,2)= Cn; edgtranstbl(thrshcnt,3)= curtpxlint;
edgtranstbl(thrshcnt,4)= nextpxlint; edgtranstbl(thrshcnt,5)= oldpxldev;
   edgtranstbl(thrshcnt,6)= newpxldev;

end

% ------------------------------------------------------------------------

function analyz_setedgpx(Rn, Cn)
% ANALYZ_SETEDGPX is the Subroutine that selects the edge pixel from
% from many possible with pixel-deviation parameters table using the
% dominant intensity parameters
%
```

```
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
% ------------------------------------------------------------------------

global Cs Rs Srw Scl edgflg h_txts5;
global thrshcntmx edgtranstbl directn daxis;
global rowmin rowmax rowedprs colmin colmax coledprs;
global curtintmin curtintmax nextintmin nextintmax;
global loceddfb1 locedbhc1 loceddfb2 locedbhc2;
global locedcea1 locedagd1 locedcea2 locedagd2;
global Segedgetblcea1 Segedgetblagd1 Segedgetbldfb1 Segedgetblbhc1;
global Segedgetblcea2 Segedgetblagd2 Segedgetbldfb2 Segedgetblbhc2;

% EDGE DETECTION PIXEL LOCATION,SEGMENTED REGION ACCUMULATION
% AND STORAGE

%-------------------------------------------------------------------------
edgflg = 0;% set edge fag off
threshtablsort = sortrows (edgtranstbl,6);

%-----------------------------------------------------------------
% Check row and column values of highest deviation
% To ensure it falls within the region expected to have the
% Prostate gland within rowmin and rowmax and colmin and colmax


if strcmp(daxis,'Columnwise');

% set edge condition for missed edge at beginng of rowss Rs
if (Rn == Rs);
if strcmp(directn,'CEA') || strcmp(directn,'BHC');
                coledprs = (Cs-60);
else
%strcmp(directn,'AGD') || strcmp(directn,'DFB');
                coledprs = (Cs+50);% or 40
end
end

else
%strcmp(daxis,'Rowwise')
% set edge condition for missed edge at beginng of Column Cs
if (Cn == Cs);
if strcmp(directn,'CEA') || strcmp(directn,'AGD');
                rowedprs = (Rs-35);
else
%strcmp(directn,'BHC') || strcmp(directn,'DFB');
                rowedprs = (Rs+35);
end
end
end


for x = thrshcntmx : -1 : 1,

if strcmp(directn,'CEA');
```

```
if (threshtablsort(x,3) >= curtintmin )...
&& (threshtablsort(x,3) <= curtintmax)...
&& (threshtablsort(x,4) >= nextintmin)...
&& (threshtablsort(x,4) <= nextintmax);

if (threshtablsort(x,1) >= (rowmin+10))...
&& (threshtablsort(x,2) >= (colmin+10))...
&& (sign(threshtablsort(x,6))== 1);

if strcmp(daxis,'Columnwise');
if (threshtablsort(x,2) <= (Cs-60));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
% Get out of the table search loop
end
else
% strcmp(daxis,'Rowwise')
if(threshtablsort(x,1) <= (Rs-35));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
% Get out of the table search loop
end
end
end
end
%Rn >= rowmin && Cn >= colmin
else
if strcmp(directn,'AGD');

if (threshtablsort(x,3) >= curtintmin )...
&& (threshtablsort(x,3) <= curtintmax)...
&& (threshtablsort(x,4) >= nextintmin)...
&& (threshtablsort(x,4) <= nextintmax);

if (threshtablsort(x,1) >= (rowmin+10))...
&& (threshtablsort(x,2) <= (colmax-10))...
&& (sign(threshtablsort(x,6))== 1);

if strcmp(daxis,'Columnwise')
if (threshtablsort(x,2) >= (Cs+50));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
% Get out of the table search loop
end
else
% strcmp(daxis,'Rowwise')
if(threshtablsort(x,1) <= (Rs-35));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
% Get out of the table search loop
end
end
end
```

```
end
%Rn >= rowmin && Cn <= colmax
else
if strcmp(directn,'DFB');

if (threshtablsort(x,3) >= curtintmin )...
&& (threshtablsort(x,3) <= curtintmax)...
&& (threshtablsort(x,4) >= nextintmin)...
&& (threshtablsort(x,4) <= nextintmax);

if (threshtablsort(x,1) <= (rowmax-10))...
&& (threshtablsort(x,2) <= (colmax-10))...
&& (sign(threshtablsort(x,6))== 1);

if strcmp(daxis,'Columnwise')
if (threshtablsort(x,2) >= (Cs+50));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
% Get out of the table search loop
end
else
% strcmp(daxis,'Rowwise')
if(threshtablsort(x,1) >= (Rs+35));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
% Get out of the table search loop
end
end
end
end
% Rn <= rowmax && Cn <= colmax
else
% if strcmp(directn,'BHC');

if (threshtablsort(x,3) >= curtintmin )...
&& (threshtablsort(x,3) <= curtintmax)...
&& (threshtablsort(x,4) >= nextintmin)...
&& (threshtablsort(x,4) <= nextintmax);

if (threshtablsort(x,1) <= (rowmax-10))...
&& (threshtablsort(x,2) >= (colmin+10))...
&& (sign(threshtablsort(x,6))== 1);

if strcmp(daxis,'Columnwise')
if (threshtablsort(x,2) <= (Cs-50));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
% Get out of the table search loop
end
else
% strcmp(daxis,'Rowwise')
if(threshtablsort(x,1) >= (Rs+35));
                              edgflg=1; %set edge flag on
%disp ('Edge pixel Found');
break;
```

```
% Get out of the table search loop
end
end
end
end
% Rn <= rowmax && Cn >= colmin
end
end
end

% Go check the next higher deviation for edge pixel conditions
end

% Setting common parameter for both missed and normal edge
% before computation storage and checking for malign

if (edgflg == 1); %set edge flag on;

%disp('Found an edge at At Row : Column');
    nr = threshtablsort(x,1); nc = threshtablsort(x,2);

    set(h_txts5,'String',...
      strcat('!!Please Wait, System Locating/Storing Edges......:',...
         int2str(nr),',',int2str(nc)));

   Csele = int32(threshtablsort(x,2));
   Rsele = int32(threshtablsort(x,1));
   coledprs = Csele + Scl + Scl; % assign col value for nxt prset edge
   rowedprs = Rsele + Srw + Srw; % assign row value for nxt prset edge
else
if strcmp(daxis,'Columnwise');
% set edge parameter for missed edge at other rows
        Csele = coledprs; % - Scl - Scl;
        Rsele = Rn; % use last preset edge

else
%strcmp(daxis,'Rowwise')
% set edge parameters for missed edge at other columns
        Rsele = rowedprs; % + Srw + Srw;
        Csele = Cn; % use last preset edge

end

end

% save the edge particulars found/preset
if strcmp(directn,'CEA'); drnum=1;
if strcmp(daxis,'Columnwise');
            locedcea1 = locedcea1 + 1; % increment edge counter
            Segedgetblcea1(locedcea1,1) = Rsele;
            Segedgetblcea1(locedcea1,2) = Csele;
            Segedgetblcea1(locedcea1,3) = drnum;

else
% strcmp(daxis,'Rowwise')
            locedcea2 = locedcea2 + 1; % increment edge counter
            Segedgetblcea2(locedcea2,1) = Rsele;
```

```
            Segedgetblcea2(locedcea2,2) = Csele;
            Segedgetblcea2(locedcea2,3) = drnum;
end
else
if strcmp(directn,'AGD');drnum=2;
if strcmp(daxis,'Columnwise');
            locedagd1 = locedagd1 + 1; % increment edge counter
            Segedgetblagd1(locedagd1,1) = Rsele;
            Segedgetblagd1(locedagd1,2) = Csele;
            Segedgetblagd1(locedagd1,3) = drnum;
else
% strcmp(daxis,'Rowwise'
             locedagd2 = locedagd2 + 1; % increment edge counter
             Segedgetblagd2(locedagd2,1) = Rsele;
             Segedgetblagd2(locedagd2,2) = Csele;
             Segedgetblagd2(locedagd2,3) = drnum;
end
else
if strcmp(directn,'DFB');drnum=3;
if strcmp(daxis,'Columnwise');
                  loceddfb1 = loceddfb1 + 1; % increment edge counter
                  Segedgetbldfb1(loceddfb1,1) = Rsele;
                  Segedgetbldfb1(loceddfb1,2) = Csele;
                  Segedgetbldfb1(loceddfb1,3) = drnum;
else
% strcmp(daxis,'Rowwise'
                  loceddfb2 = loceddfb2 + 1; % increment edge counter
                  Segedgetbldfb2(loceddfb2,1) = Rsele;
                  Segedgetbldfb2(loceddfb2,2) = Csele;
                  Segedgetbldfb2(loceddfb2,3) = drnum;
end
else
% if strcmp(directn,'BHC');
              drnum=4;
if strcmp(daxis,'Columnwise');
                  locedbhc1 = locedbhc1 + 1; % increment edge counter
                  Segedgetblbhc1(locedbhc1,1) = Rsele;
                  Segedgetblbhc1(locedbhc1,2) = Csele;
                  Segedgetblbhc1(locedbhc1,3) = drnum;
else
% strcmp(daxis,'Rowwise'
                  locedbhc2 = locedbhc2 + 1; % increment edge counter
                  Segedgetblbhc2(locedbhc2,1) = Rsele;
                  Segedgetblbhc2(locedbhc2,2) = Csele;
                  Segedgetblbhc2(locedbhc2,3) = drnum;
end
end
end
end


end
% --------------------------------------------------------------------

function analyz_dthpecopx(nr, nc, Drpxl)
% ANALYZ_DTHPECOPX is the Subroutine that detect the pixel
% with hyper echoic intensities, property of malignant tissues
%
%
```

```
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
%
% ------------------------------------------------------------------------

global maligcoodtbl locmg malgintrg;
global prost_imatrx h_txts5;

neighbnx8 = [-1,0;-1,1;0,1;1,1;1,0;1,-1;0,-1;-1,-1];

% NOTE COMPARE RESULT OBTAINED WITH THIS PROCEDURE WITH NEW
% PROEDURE USING IMAGE SEQUENTIAL STICK IMAGE ENHANCEMENT AND SCANNING
% CHOOSE BETTER AND COMMENT ON PROS AND CONS AS RESEARCH FINDING

   malgfd=0; % set malign flag off
% check intensity range within malign & proximity

if (prost_imatrx(nr, nc) <= malgintrg);

% where malinthresh <= 20 / >= -20, can be varied +/- to fintune result
% &&<= malgintrg, 60 - 70) can be varied;
% check neigbors consistency for malign features

           malgcfc=0; % initialize malignancy feature counter
for negb = 1 : 8,
               nrow = nr + neighbnx8(negb,1);
               ncol = nc + neighbnx8(negb,2);
if (prost_imatrx(nrow,ncol) <= malgintrg);
                  pxtd1 = int32(prost_imatrx(nr,nc));
                  pxtd2 = int32(prost_imatrx(nrow,ncol));
                  pxdev = pxtd1 -pxtd2;
if (pxdev >= -5) && (pxdev <= 5);
% if difference is within -5 and +5
                     malgcfc = malgcfc+1; % increase counter
end
end
end

if (malgcfc >= 1);
               malgfd = 1; % set malignancy found flag
end
end

if (malgfd == 1); %

      set(h_txts5,'String',...
      strcat('!!Please Wait, System Compiling Malignant Pixels......:',...
        int2str(nr),',',int2str(nc)));

%disp('Test of Malignancy True !!');

%Storecoords of pixel in malignant cordinates tabl
        locmg = locmg + 1;
        maligcoodtbl (locmg, 1) = nr;   maligcoodtbl (locmg, 2) = nc;
```

```
        maligcoodtbl (locmg, 3) = Drpxl;
        maligcoodtbl (locmg, 4) = prost_imatrx(nr, nc);

end

end% Get next column item along this row edged


% ------------------------------------------------------------------------


function analyz_acumpxsgcd
% ANALYZ_ACUMPXSGCD is the Subroutine that accumulates the segemented
% edges, segmented pixels for the columnwise and rowsise results
% for the four directions of radial scanning using the tables
%
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
% ------------------------------------------------------------------------

global segimcoodtbl segcount Segimedgetbl loced Cs Scl Srw rowmax;
global loceddfb1 loceddfb2 locedbhc1 locedbhc2 directn Rs rowmin;
global locedcea1 locedcea2 locedagd1 locedagd2 edsubtsze;
global Segedgetblcea1 Segedgetblagd1 Segedgetbldfb1 Segedgetblbhc1;
global Segedgetblcea2 Segedgetblagd2 Segedgetbldfb2 Segedgetblbhc2;

global L h_axes1 h_txts5;

%-------------------------------------------------------------------------
% Accumulate the pixels within the edge located
% from seed point for column wise and reference Erwn for row wise
% accumulation to avoid double counting of segmented pixels
%-------------------------------------------------------------------------

if strcmp(directn,'CEA');
       acumtbl1 = Segedgetblcea1;acumtbl2 = Segedgetblcea2;
       numedg1 = locedcea1;   numedg2 = locedcea2;
else
if strcmp(directn,'AGD');
         acumtbl1 = Segedgetblagd1; acumtbl2 = Segedgetblagd2;
         numedg1 = locedagd1; numedg2 = locedagd2;
else
if strcmp(directn,'DFB');
              acumtbl1 = Segedgetbldfb1;acumtbl2 = Segedgetbldfb2;
              numedg1 = loceddfb1;     numedg2 = loceddfb2;
else
% if strcmp(directn,'BHC');
                   acumtbl1 = Segedgetblbhc1; acumtbl2 = Segedgetblbhc2;
                   numedg1 = locedbhc1; numedg2 = locedbhc2;
end
end
end

acumtbl1sort = sortrows (acumtbl1 ,1);
```

```
acumtbl2sort = sortrows (acumtbl2, 2);

% Acumulate rows in both tables acumtbl1, acumtbl2  using their average
% column values into a temp file, acumtmp and add rows in tab 1 alone
% with their column values into the temp table


%Set the limits of the rows for directns along a column
if strcmp(directn,'CEA') || strcmp(directn,'AGD');
        Rwb = rowmin + ...
             ((Rs - rowmin) - idivide(Rs - rowmin, int32(3)));
else
% if strcmp(directn,'BHC') || strcmp(directn,'DFB');
       Rwb = rowmax - ...
            ((rowmax - Rs) - idivide(rowmax - Rs, int32(3)));
end

% Acumulate the pixel coordinates bound by the edges in acumtbl1 sorted
% and acumulate the total pixels in segmented area, segcount

if numedg1 > 0;
for cntcol = (edsubtsze - numedg1 +1) : 1 : edsubtsze, %loop for tabl1

     Rpxl = acumtbl1sort(cntcol,1); Cpxl = acumtbl1sort(cntcol,2);
     Drpxl = acumtbl1sort(cntcol,3);

if (Rpxl > 0) && (Cpxl > 0);

% acumulate total edgetabl
         loced = loced +1;
         Segimedgetbl(loced,1)= Rpxl; Segimedgetbl(loced,2)= Cpxl;
         Segimedgetbl(loced,3)= Drpxl;

if strcmp(directn,'CEA') || strcmp(directn,'AGD');
if Rpxl < Rwb;
                    rowacm = 1; %set rowwise accum on
else
                    rowacm = 0; %set columnwise  accum on
end
else
% if strcmp(directn,'BHC') || strcmp(directn,'DFB');
if Rpxl > Rwb;
                   rowacm = 1; %set rowwise accum on
else
                   rowacm = 0; %set columnwise  accum on
end
end

if rowacm == 0;
%accumulate segmented coord tabe, column-wise, along a row
for nc = Cs : Scl : Cpxl,
                nr = Rpxl;
                analyz_dthpecopx(nr, nc, Drpxl);
% Routine to detect hypoechoic pixels, as malignant

%cla(h_axes1,'reset'); % Clear any dislay of previous result
        segcount = segcount+1; segimcoodtbl(segcount,1) = Rpxl;
```

```
              segimcoodtbl(segcount,2) = nc;  segimcoodtbl(segcount,3) = Drpxl;

end;    % Get next column item along this row edge
          set(h_txts5,'String',...
          strcat('!!Please Wait, System Compiling Located Edges......:',...
          int2str(nr),',',int2str(nc)));

          axes(h_axes1); imshow(L);

else
% rowacm == 1
%accumulate segmented coord tabe, row-wise, along a column

for nr = Rwb : Srw : Rpxl,

                nc = Cpxl;
                analyz_dthpecopx(nr, nc, Drpxl);
% Routine to detect hypoechoic pixels, as malignant
              segcount = segcount+1;   segimcoodtbl(segcount,1) = nr;
              segimcoodtbl(segcount,2) = Cpxl;segimcoodtbl(segcount,3) = Drpxl;

end;    % Get next column item along this row edge

          set(h_txts5,'String',...
          strcat('!!Please Wait, System Compiling Located Edges......:',...
          int2str(nr),',',int2str(nc)));

          axes(h_axes1); imshow(L);

end

end

end
end

% Acumulate de pixel coordinates bound by de edges in acumtbl2 sorted
% and acumulate the total pixels in segmented area, segcount

if numedg2 > 0;
for cntrow = (edsubtsze - numedg2 +1) : 1 : edsubtsze, %loop for tabl 2,

     Rpxl = acumtbl2sort(cntrow,1); Cpxl = acumtbl2sort(cntrow,2);
     Drpxl = acumtbl2sort(cntrow,3);

if (Rpxl > 0) && (Cpxl > 0);

% acumulate total edgetabl
          loced = loced +1;
          Segimedgetbl(loced,1)= Rpxl; Segimedgetbl(loced,2)= Cpxl;
          Segimedgetbl(loced,3)= Drpxl;

%accumulate the segmented coord tabe, row-wise, along a column

for nr = Rwb : Srw : Rpxl,
               nc = Cpxl;
```

```
                analyz_dthpecopx(nr, nc, Drpxl);
% Routine to detect hypoechoic pixels, as malignant

               segcount = segcount+1;   segimcoodtbl(segcount,1) = nr;
               segimcoodtbl(segcount,2) = Cpxl;segimcoodtbl(segcount,3) = Drpxl;

end;   % Get next column item along this row edge

          set(h_txts5,'String',...
          strcat('!!Please Wait, System Compiling Located Edges......:',...
          int2str(nr),',',int2str(nc)));

          axes(h_axes1); imshow(L);
end
end
end

end

%--------------------------------------------------------------------------
% -------------------------------------------------------------------------


function K=analyz_seqstick(J)
% ANALYZ_SEQSTICK is the Routine that remove noise and enhance image
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
% -------------------------------------------------------------------------

I=J; %assign original image to I that will be changing in sequences
% Subroutine call to use sequential stick algorithm remove noise
%     and enhance image features for digital analysis

global rng domintrng h_txts5;

% setup a sequential for loop that will call the stick algorithm
%
if strcmp(domintrng, 'Light')
    k=1; nmax=13; % for light images set in malign routine
%disp('Light gray level image selected...Wait...Image Enhancement!!');
else
if strcmp(domintrng, 'Dark')
      k=3; nmax=5; % for dark images
%disp('Dark gray level image selected...Wait...Image Enhancement!!');
else
if rng == 1;
            k=3; nmax=5; % for light malig image section extraction
else
% rng = 2
           k=3; nmax=7; % for dark malig image section extraction
end
end
end
```

```
%k=3;% fixed value of stick thickness

for n = 5 : 2: nmax,

    G =aiprocsf(I,n,k);
    I=G; % make enhanced image go another sequence
end
  K=G; % final result is in I and G at this point

  set(h_txts5,'String',...
     strcat('!!Please Wait, Sequential Stick Enhancing of Image....:',...
         int2str(n),',',int2str(nmax)));

end
% --------------------------------------------------------------------

%--------------------------------------------------------------------
function G = aiprocsf(I,n,k)
% AIPROCSF Sticks Filtering
%    AIPROCSF(I,N,K) Sticks filtering for speckle noise reduction
%    This filter works for greyscale images
%
%    I is the image to be filtered
%    N is the size of the filter
%    K is the sticks thickness

%   Using sticks filters to reduce speckle noise : exemples.
%   Increasing "stick" length leads to a more smoothly filtered image,
%   at the expense of weakly highlighting tightly bound curves - a
%   result of the "stick" being longer than some of the boundary edges.
%   Similarly, thicker "sticks" suppress more noise the expense of making
%   thin boundaries less visible. A thick "stick" can be used to smooth
%   noise similar to a low-pass filter, with the addition of highlighting
%   broad region differences.
%
%   Source : "Prostate Ultrasound Image Processing" by Deian Stefan
%   and Pr. Hong Man
%   08/07 T. Mervin


s = aiprocsticks(n,k);  % generate a set of sticks filters
m = 2*n-2;        % number of filters

[mi,ni] = size(I);
mh = mi+n-1;  % size of filtered image
nh = ni+n-1;  % after convolution

H = zeros(m,mh,nh);
G = zeros(mh,nh);

% H = {Hi | i = 1 . . . 2n - 2}, where Hi = I * Si
for i = 1:m,
    H(i,:,:) = conv2(double(I),double(s(:,:,i)));
end;

% G(x,y) = max{Hi(x,y)}, for i = 1 . . . 2n - 2.
for i = 1:mh,
```

```
for j = 1:nh,
        G(i,j) = max(H(:,i,j));
end;
end;

G = uint8(round(G));  % get back to int
end
%-------------------------------------------------------------------------

%-------------------------------------------------------------------------
function s = aiprocsticks(n,k)
%AIPROCSTICKS  Sticks Filter Generation
%   AIPROCSTICKS(N,K) returns a set of sticks filters
%   N is the stick length
%   K is the sticks' thickness
%   Exemple : aiprocsticks(5,1) provide 8 5x5 matrix with stick
%   thickness equal to one

%   04/07 T. Mervin

% check that args are odd positive numbers and that
% the sticks' thickness does not exceed filter size
if (mod(n,2) == 0) || (mod(k,2) == 0) || (k > n)
    disp('Wrong arguments. Expecting odd positive numbers, k < n')
else
% N x N x 2N-2 matrix for the 2*N-2 sticks
    s = zeros(n,n,2*n-2);

    t = pi/(2*n-2); % angular gap pi/(2n-2)
    s0 = 1/n;   m = round(n/2);     l = (k-1)/2;

% particular cases :  tan(pi/2) and tan(0)
    s(m-l:m+l,:,1) = s0; % horizontal, tan(0)
    s(:,m-l:m+l,n) = s0; % vertical, tan(pi/2)

for j = 2 : (n-1)/2 + 1,
for i = 1 : n,
            a = round(tan((j-1)*t)*(i-m) + m); % ]0,pi/4]
            b = round(-tan((j-1)*t)*(i-m) + m); % symmetry
            c = round(cot(((n-1)/2+j-1)*t)*(i-m) + m); % beyond pi/4
            d = round(-cot(((n-1)/2+j-1)*t)*(i-m) + m); % symmetry

            A1 = a-l;A2 = a+l;B1 = b-l;B2 = b+l;
            C1 = c-l;C2 = c+l; D1 = d-l; D2 = d+l;

if A1 <= 0     % we check that the determined value
                A1 = 1;    % don't exceed the matrix size
end% we truncate the values if it is so
if A2 > n;   A2 = n;       end
if B1 <= 0;   B1 = 1;      end
if B2 > n;    B2 = n;      end
if C1 <= 0;   C1 = 1;      end
if C2 > n ;   C2 = n;      end
if D1 <= 0;   D1 = 1;      end
if D2 > n;    D2 = n;      end

            s(A1:A2,i,j) = s0;     s(B1:B2,i,2*n-j) = s0;
```

```
            s(i,C1:C2,(n-1)/2+j) = s0;  s(i,D1:D2,2*n-((n-1)/2+j)) = s0;
end
end
end
end

%-----------------------------------------------------------------------



function analyz_genzontbl
% ANALYZ_GENZONTBL is the Subroutine that generates the zone identity
% for pixels in the referenced prostate gland image
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
%
% ------------------------------------------------------------------------
global maligcoodtbl locmg Segimedgetbl loced;
global loctbl zoneid zonelkuptabl edgtbsize;
global Centralcnt Transitcnt Periphcnt Centralp Transitp Periphp;
global endcol begcol endrow begrow;
%

Segimedgetblrows = sortrows (Segimedgetbl,1);
% Segimedgetbl table sorted in ascending rows
Segimedgetblcols = sortrows (Segimedgetbl,2);
% Segimedgetbl table sorted in ascending cols


% the lst edge coordinates row, is the loced position
% from the highest row in the last location of the sorted table
%valdmir = 0;% validity set off
for tcnt = (edgtbsize - loced + 1) :  1 : edgtbsize,
    gldminrow = Segimedgetblrows(tcnt,1);%highest row

if (gldminrow > 0) ;
break; % valid values found
end
end

% the last edge coordinates row in table, is the highest row location
% from the last location of the sorted table by rows
%valdmxr = 0;% validity set off
for tcnt = edgtbsize : - 1 :(edgtbsize - loced + 1),
    gldmaxrow = Segimedgetblrows(tcnt,1);%highest row

if (gldmaxrow > 0) ;
break; % valid values found
end
end



% the lst edge coordinates col, is the loced position
% from the highest col in the last location of the sorted table
```

```
%valdmic = 0;% validity set off
for tcnt = (edgtbsize - loced + 1) :  1 : edgtbsize,
    gldmincol = Segimedgetblcols (tcnt,2);%highest col

if (gldmincol > 0) ;
break; % valid values found
end
end

% the last edge coordinates col in table, is the highest col location
% from the last location of the sorted table by cols
%valdmxc = 0;% validity set off
for tcnt = edgtbsize : - 1 :(edgtbsize - loced + 1),
    gldmaxcol = Segimedgetblcols (tcnt,2);%highest col

if (gldmaxcol > 0) ;
break; % valid values found
end
end
%
%
%-------------------------------------------------------
% Divide Columns into 4 segments for 22.5degrees ratios
%-------------------------------------------------------
colrange = (int32(gldmaxcol) - int32(gldmincol) + 1);
colintval = idivide(colrange,int32(4));
col1pemax = int32(gldmincol) + colintval;
col2trsg1 = int32(gldmincol) + (2*colintval);
col3trmax = int32(gldmincol) + (3*colintval);

% Divide Rows into 4 segments for 22.5degrees ratios
rowrange = (int32(gldmaxrow) - int32(gldminrow) +1);
rowintval = idivide(rowrange,int32(4));
row1cesg2 = int32(gldminrow) + rowintval;
row2cemax = int32(gldminrow) + (2*rowintval);

loctbl = 0; % Initialize lookup table counter

% Buildup Transition zone pixel coordinates in the table
begcol = col1pemax; endcol = col3trmax;
begrow = int32(gldminrow); endrow = row1cesg2;
zoneid = 3; % 'Transition';
if (begcol>0)&& (endcol>0)&& (begrow>0)&& (endrow>0)&& (zoneid>0);

    analze_storezoneid;
% call routine to build and store
% zoneid with the set parameters
end

% Buildup Central zone pixel coordinates in the table
% 1st section
begcol = col3trmax; endcol = int32(gldmaxcol);
begrow = int32(gldminrow); endrow = row1cesg2;
zoneid = 1; %'Central';
if (begcol>0)&& (endcol>0)&& (begrow>0)&& (endrow>0)&& (zoneid>0);

    analze_storezoneid;
```

```
% zoneid with the set parameters
% call routine to build and store
end

%2nd section
begcol = col2trsg1; endcol = int32(gldmaxcol);
begrow = row1cesg2; endrow = row2cemax;
if (begcol>0)&& (endcol>0)&& (begrow>0)&& (endrow>0)&& (zoneid>0);

    analze_storezoneid;
% zoneid with the set parameters
% call routine to build and store
end

% Buildup Peripheral zone pixel coordinates in the table
% 1st section
begcol = int32(gldmincol); endcol = int32(gldmaxcol);
begrow = row2cemax; endrow = int32(gldmaxrow);
zoneid = 2;% 'Peripheral';
if (begcol>0)&& (endcol>0)&& (begrow>0)&& (endrow>0)&& (zoneid>0);

    analze_storezoneid;
% zoneid with the set parameters
% call routine to build and store
end

%2nd section
begcol = int32(gldmincol); endcol = col2trsg1;
begrow = row1cesg2; endrow = row2cemax;
if (begcol>0)&& (endcol>0)&& (begrow>0)&& (endrow>0)&& (zoneid>0);

    analze_storezoneid;
% zoneid with the set parameters
% call routine to build and store
end

%3rd section
begcol = int32(gldmincol); endcol = col1pemax;
begrow = int32(gldminrow); endrow = row1cesg2;
if (begcol>0)&& (endcol>0)&& (begrow>0)&& (endrow>0)&& (zoneid>0);

    analze_storezoneid;
% zoneid with the set parameters
% call routine to build and store
end

%-------------------------------------------------------------------------
% Assign zoneids to malignant coordinates table
% and count the nos for each zone
%-------------------------------------------------------------------------

Centralcnt = 0; Transitcnt = 0; Periphcnt = 0;
Centralp = 0; Transitp = 0; Periphp = 0;

if locmg > 0;
for malcnt = 1 : 1 : locmg,
       rwmal = maligcoodtbl(malcnt,1);
```

```
       clmal= maligcoodtbl(malcnt,2);

for tblcnt = 1 : 1 : loctbl,
if (rwmal == zonelkuptabl(tblcnt,1))...
&& (clmal == zonelkuptabl(tblcnt,2));
% if lookup table row,col is equal to maligcood table
% then read the tabe zoneid store in maligtable
                maligcoodtbl(malcnt,5) = zonelkuptabl(tblcnt,3);
end
end
end


for malcnt = 1 : 1 : locmg,

if maligcoodtbl(malcnt,5)== 1; %'Central'
            Centralcnt = Centralcnt + 1;
else
if maligcoodtbl(malcnt,5) == 2; %'Peripheral'
                Periphcnt = Periphcnt + 1;
else
if maligcoodtbl(malcnt,5) == 3; % 'Transition'
                   Transitcnt = Transitcnt + 1;
end
end
end
end

format bank;

Centralp = ((Centralcnt*100)/locmg); Transitp = ((Transitcnt*100)/locmg);
Periphp  = ((Periphcnt*100)/locmg);

end
end

function analze_storezoneid
%-----------------------------------------------------------
% Stores the zoneids in table with set parameters
%-----------------------------------------------------------

global loctbl zonelkuptabl zoneid L h_axes1 h_txts5;
global endcol begcol endrow begrow;

for rowcnt = begrow : 1 : endrow,
for colcnt = begcol : 1 : endcol,

          loctbl = loctbl + 1; zonelkuptabl(loctbl,1) = rowcnt;
          zonelkuptabl(loctbl,2) = colcnt;zonelkuptabl(loctbl,3) = zoneid;

end
end

  set(h_txts5,'String',...
 strcat('!!Please Wait, System Compiling Gland Zones Lookup table ..:',...
     int2str(rowcnt),',',int2str(colcnt)));
```

```
  axes(h_axes1); imshow(L);

end

% --------------------------------------------------------------------



function expert_analprosm
% EXPERT_ANALPROSM is the Control Programme for Expert Software
% For Prostate Cancer Detection and Segmentation of TRUS 2D-images
% of the Prostate.
% Graphic User Interface tool in MATLAB drives the menu.
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.

% Initialize global variables here
global J domintrng prost_imsamp h_axes4 h_txts5 h_txts6;
global h_fig1 h_panel1 h_axes1 h_axes2 h_axes3 Centrzncnt Peripzncnt Transzncnt
global Centralp Periphp Transitp expsegcount explocmg;

domintrng = 'light';  % dominant intensity range indicator
prost_imsamp = ' '; % keeps original image filename and pathname
imagfilenam = ' ';  % keeps only original filename
diremty = 0; % directory empty is 0 when no jpg image files are stored

% Create and hide the figure as is being created.
h_fig1 = figure('Visible', 'off','Position',[50,50,1000,650],...
'Toolbar','none','MenuBar','none');
h_text = uicontrol(h_fig1,'Style','text',...
'Position',[15,610,310,20],'String',...
'EXPERT MAIN MENU.');
% Construct the components put in a panel, panel1
h_panel1 = uipanel(h_fig1,'Units','pixels','Position',[10,350,320,300]);
h_menuopt1 = uicontrol(h_panel1,'Style','pushbutton',...
'Position',[20,190,280,60],...
'String','SELECT IMAGE FOR EXPERT PROCESSING',...
'TooltipString','Interruptible = off',...
'Interruptible','off',...
'Callback',@expert_imagselect);
h_menuopt2 = uicontrol(h_panel1,'Style','pushbutton',...
'Position',[20,120,280,60],...
'String','EXPERT DETECT EDGES/MARK SUSPECTED ZONES',...
'TooltipString','Interruptible = off',...
'Interruptible','off',...
'Callback',@expert_imaganalys);
h_menuopt3 = uicontrol(h_panel1,'Style','pushbutton',...
'Position',[20,50,280,60],...
'String','QUIT EXPERT MENU',...
'TooltipString','Interruptible = off',...
'Interruptible','off',...
'Callback',@expert_endsyst);
h_text1 = uicontrol(h_panel1,'Style','text',...
'Position',[40,20,250,20],'String',...
```

```
'Click a Button to Select an Operation.');
h_txts1 = uicontrol(h_fig1,'Style','text',...
'Position',[380,620,290,20],'String',...
'                                    ');
h_txts2 = uicontrol(h_fig1,'Style','text',...
'Position',[700,620,290,20],'String',...
'                                    ');
h_txts3 = uicontrol(h_fig1,'Style','text',...
'Position',[380,300,290,20],'String',...
'                                    ');
h_txts4 = uicontrol(h_fig1,'Style','text',...
'Position',[700,300,290,20],'String',...
'                                    ');
h_txts5 = uicontrol(h_fig1,'Style','text',...
'Position',[30,280,330,40],'String',...
'                                    ');
h_txts6 = uicontrol(h_fig1,'Style','text',...
'Position',[30,240,330,25],'String',...
'                                    ');
h_txts7 = uicontrol(h_fig1,'Style','text',...
'Position',[30,200,150,30],'String',...
'                                    ');
h_txts7a = uicontrol(h_fig1,'Style','text',...
'Position',[190,200,80,30],'String',...
'                                    ');
h_txts7b = uicontrol(h_fig1,'Style','text',...
'Position',[280,200,80,30],'String',...
'                                    ');
h_txts8 = uicontrol(h_fig1,'Style','text',...
'Position',[30,170,150,30],'String',...
'                                    ');
h_txts8a = uicontrol(h_fig1,'Style','text',...
'Position',[190,170,80,30],'String',...
'                                    ');
h_txts8b = uicontrol(h_fig1,'Style','text',...
'Position',[280,170,80,30],'String',...
'                                    ');
h_txts9 = uicontrol(h_fig1,'Style','text',...
'Position',[30,140,150,30],'String',...
'                                    ');
h_txts9a = uicontrol(h_fig1,'Style','text',...
'Position',[190,140,80,30],'String',...
'                                    ');
h_txts9b = uicontrol(h_fig1,'Style','text',...
'Position',[280,140,80,30],'String',...
'                                    ');
h_txts10 = uicontrol(h_fig1,'Style','text',...
'Position',[30,110,330,20],'String',...
'                                    ');
h_txts11 = uicontrol(h_fig1,'Style','text',...
'Position',[30,80,330,20],'String',...
'                                    ');
h_txts12 = uicontrol(h_fig1,'Style','text',...
'Position',[30,50,330,20],'String',...
'                                    ');

%
% Create axis for diplaying the images
h_axes1 = axes('Parent',h_fig1,'Units','pixels',...
```

```
'Position',[410,355,250,250]);
h_axes2 = axes('Parent',h_fig1,'Units','pixels',...
'Position',[720,355,250,250]);
h_axes3 = axes('Parent',h_fig1,'Units','pixels',...
'Position',[410,35,250,250]);
h_axes4 = axes('Parent',h_fig1,'Units','pixels',...
'Position',[720,35,250,250]);

% --------------------------------------------------------------------

align(h_panel1, h_text,'Center','None');

% Initialize the GUI.
% Change units to normalized so components resize automatically.
%
set([h_fig1,h_panel1,h_axes1,h_axes2,h_axes3,h_axes4,h_menuopt1,...
    h_menuopt2,h_menuopt3,h_text,h_text1,h_txts1,h_txts2,...
    h_txts3,h_txts4,h_txts5,h_txts6,h_txts7,h_txts8,h_txts9,h_txts10,...
    h_txts7a,h_txts7b,h_txts8a,h_txts8b,h_txts9a,h_txts9b,...
    h_txts11,h_txts12],'Units','normalized');
% Assign the GUI a name to appear in the window title.
set(h_fig1,'Name','PROSTATE IMAGE ANALYZER FOR EXPERT CANCER DETECTION')
% Move the GUI to the center of the screen.
movegui(h_fig1,'center')
% Make the GUI visible.
set(h_fig1,'Visible','on');
%
% Callbacks for the GUI in h_fig1
% --------------------------------------------------------------------
% Executes when you click the menuopt1 push button.
function expert_imagselect(hObject,eventdata)

% Disable the other keys.
        set(h_menuopt2,'Enable','off'); set(h_menuopt3,'Enable','off')
% Call the routine to automate selection of an image file.
      [imagfilenam,diremty] = analyz_imageselc;

      cla(h_axes3,'reset');
if diremty == 0;
        directnam = cd; % Read current directory pathname
        warndlg ({directnam;...
'Please Copy .jpg Prostate Image Files into this Directory';...
'!!! Then Try this Option Aagain !!! '},...
'No Image Files For Selection !');
else
if imagfilenam == ' ';
            set(h_txts5,'String',...
'User selected No File for Expert Processing!!');
            set(h_txts2,'String',...
'                                                        ');
else
            set(h_txts5,'String',...
             strcat('User Selected:...', imagfilenam, '...Image File'));
            I = imread(prost_imsamp);
            cla(h_axes1,'reset');
            axes(h_axes1); imshow(I);
            cla(h_axes2,'reset'); cla(h_axes3,'reset');
```

```
            set(h_txts6,'String',...
'                                                    ');
            set(h_txts7,'String',...
'                                                ');
            set(h_txts7a,'String',...
'                                          ');
            set(h_txts7b,'String',...
'                                          ');
            set(h_txts8,'String',...
'                                                ');
            set(h_txts8a,'String',...
'                                          ');
            set(h_txts8b,'String',...
'                                          ');
            set(h_txts9,'String',...
'                                                ');
            set(h_txts9a,'String',...
'                                          ');
            set(h_txts9b,'String',...
'                                          ');
            set(h_txts10,'String',...
'                                                ');
            set(h_txts11,'String',...
'                                                ');
            set(h_txts12,'String',...
'                                                ');

            cla(h_axes4,'reset');
            set(h_txts4,'String',...
'                                                    ');

            set(h_txts3,'String',...
'                                                    ');
            set(h_txts2,'String',...
'                                                    ');
            set(h_txts1,'String',...
'Original TRUS 2D-Prostaste Image Sample I');
end
end
% Enable the other keys
      set(h_menuopt2,'Enable','on'); set(h_menuopt3,'Enable','on');
end
% --------------------------------------------------------------------
% Executes when you click the menuopt2 push button.
function expert_imaganalys(hObject,eventdata)

%
% Disable the other keys.
        set(h_menuopt1,'Enable','off'); set(h_menuopt3,'Enable','off')
% check flag for image filename selection
if imagfilenam == ' ';
            set(h_txts5,'String',...
'User selected No File to Analyze.');
else
             cla(h_axes2,'reset');cla(h_axes3,'reset');cla(h_axes4,'reset')
             set(h_txts5,'String',...
'!!! Wait Analysis of Image is in Progress ..... !!!');
```

```
% Call routines to start analysis of selected image.
% J = analyz_remspecmk(prost_imsamp);% remove specs mark
% call routine to remove labels in ultrasound images for
% manual segmentation/interpretation

            J = analyz_remvlabl;% remove labels

%cla(h_axes1,'reset');
            axes(h_axes1); imshow(J);

% Call to determine dominant intensity range of image & set
% parameters for image processing accordingly
            domintrng = analyz_malgrang(J);

            cla(h_axes3,'reset');cla(h_axes4,'reset')

% Call routiine to enhance image removing noise
            K = analyz_seqstick(J); % enhance image by seq stick

            cla(h_axes2,'reset'); % Clear any dislay of previous result
            axes(h_axes2); imshow(K); %
            set(h_txts2,'String',...
'System Enhanced TRUS 2D-Prostate Image sample II.');

% Call routine to start collecting expert border marking
% for segmenting prostate gland/processing of region table

            PS = expert_glandsegmtg(K);

            cla(h_axes3,'reset'); % Clear any dislay of previous result
            axes(h_axes3); imshow(PS);
            set(h_txts3,'String',...
'Expert Segmented Prostate TRUS 2D-Image Edges Marked');

% Call routine to start collecting expert border marking
% for suspected malignant zones by zones Transition,
% Central & Peripheral Zones,process and display SUMMARY

           PM = expert_glandmalgzn(K);

           cla(h_axes4,'reset'); % Clear any dislay of previous result
           axes(h_axes4); imshow(PM);
           set(h_txts4,'String',...
'Suspected Cancerous Tissues of Prostate Marked WHITE');

% Call routine to save original and the corresponding
% segmented & malignant file names & store results in sumfile
            expert_resfilsave(imagfilenam, PS,PM);

% Display Report of Cancer/malignacy

            format bank;
if expsegcount > 0;
               malpctge = ((explocmg * 100)/expsegcount);
else
               malpctge = 0;
```

```
end

if explocmg > 0;
                Centralp = ((Centrzncnt * 100)/explocmg);
                Periphp = ((Peripzncnt * 100)/explocmg);
                Transitp = ((Transzncnt * 100)/explocmg);
else
                Centralp = 0;Periphp = 0; Transitp = 0;
end

            set(h_txts5,'String',...
'EXPERT CANCER DETECTION REPORT SUMMARY IN NO OF PIXELS');

            set(h_txts6,'String',...
            strcat('|---:ZONE OF GLAND',':---|--:AFFECTED PIXELS:',...
':--|--: % AFFECTED:-|'));

            set(h_txts7,'String','Central (1st Quad, 67.5")');
            set(h_txts7a,'String',int2str(Centrzncnt));
            set(h_txts7b,'String',sprintf('%5.2f',Centralp));
            set(h_txts8,'String','Transition (4th - 1st Quad, 45")');
            set(h_txts8a,'String',int2str(Transzncnt));
            set(h_txts8b,'String',sprintf('%5.2f',Transitp));
            set(h_txts9,'String','Perpheral (2nd - 4th Quad, 247.5")');
            set(h_txts9a,'String',int2str(Peripzncnt));
            set(h_txts9b,'String',sprintf('%5.2f',Periphp));

            set(h_txts10,'String',...
            strcat('TOTAL PIXELS IN SUSPECTED AREAS.................:',...
                   int2str(explocmg)));

            set(h_txts11,'String',...
            strcat('TOTAL PIXELS IN THE SEGMENTED GLAND REGION....:',...
                  int2str(expsegcount)));

            set(h_txts12,'String',...
            strcat('PERCENTAGE OF GLAND SUSPECTED.............. :',...
                  sprintf('%5.2f',malpctge),'%'));

%close(h_wait); % to allowed to fuction when analyz_ is actv
%    Enable the other keys
end
        set(h_menuopt1,'Enable','on'); set(h_menuopt3,'Enable','on')
end
% --------------------------------------------------------------------

% Executes when you click the menuopt3 push button.
function expert_endsyst(hObject,eventdata)
% close all open figures
        close all
end
% --------------------------------------------------------------------
% Create and update a waitbar.

% Create and update a waitbar.
function create_update_waitbar
    h_wait = waitbar(0,'Please wait...',...
```

```
'Position',[150,100,270,50]);
% 'CloseRequestFcn',@close_waitbar);
for i=1:10000,
if ishandle(h_wait)
            waitbar(i/10000,h_wait)
else
break
end
end
% When waitbar reaches max, close it.
if ishandle(h_wait)
          close(h_wait)
end
end
% ---------------------------------------------------------------------
% C%lose the waitbar. Executes in response to BREAK and CLOSE commands.
% function close_waitbar(hObject,eventdata)
%     delete(gcbf)
% end
% ---------------------------------------------------------------------
end
```

```
function [PS]=expert_glandsegmtg(K)

% EXPERT_GLANDSEGMTG is the Routine that automates the analysis of selected
% Image file - TRUS 2D-images of the Prostate stored in the working
% Disk directory of MATLAB.
% Gives the Expert opportunity to mark the edges of the prostate gland
% on the displayed image. Further accumulates the pixels bound by the marked
% edge by the expert to estimate the area in pixels
% The File types possible/accepatble include jpg, tif, png or bmp.
% However jpg is prefered for this work
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.

global expsegimcoodtbl expsegcount explocmg
global expsegimedgetbl exploced Cs Rs imsize ;
global L rowm colm edsubtsze edgtbsize;
global h_axes2 h_txts5 h_txts6 pt1row pt1col pt2row pt2col;

explocmg=0; exploced=0; expsegcount=0; % initialize these global counters

[rowm,colm] = size(K); imsize=rowm*colm;L = K;K1=K;

expsegimcoodtbl = zeros(imsize,3); % expert segmented pixels coord table
% Table => row, column, directn: '
% For segmented pixels in region of prostate gland of image

edgtbsizei = idivide(int32(imsize), int32(24));
edgtbsize = double(edgtbsizei);
expsegimedgetbl=zeros(edgtbsize,3);% total edge table
```

```
edsubtszei = idivide(int32(imsize), int32(42));
edsubtsze = double(edsubtszei);

segmal = 1; %set flag for expert segmentation process

% Capture the Points Marked by Expert Sonographer/Radiologist
% on the isolated borders of the prostate gland shown in TRUS
% 2D-iamge of the prostate under investigation

prpt = 1;
while prpt == 1;

 set(h_txts5,'String',...
 strcat('Expert, Please Examine Image Sample I & II....',...
'Mark at Least 10 points Clockwise, Right - round the borders',...
'... of the Prostate Gland within the IMAGE SAMPLE II'));

 cla(h_axes2,'reset'); % Clear any dislay of previous result
 axes(h_axes2); imshow(K); %

 set(h_txts6,'String',...
 strcat('Click Once on the Point Desired; ...',...
'Use - Backspace to Delete last Point Marked;...', ...
'Finish by Double-Clicking/Enter at a Last Point'));

% Command to Capture Points on the Current Figure Axes, h_axes3
  cla(h_axes2,'reset'); % Clear any dislay of previous result
  axes(h_axes2); imshow(K); %
  [Cplist, Rplist]=getpts;% [x,y] where x is columns of matrix, y is rows

% Rpcols and Cpcols should all be equal to 1 one column list of points

    [Rprows,Rpcols]=size(Rplist);
% Get lenght, Rprows, of List of Rows coordinates of points selected
    [Cprows,Cpcols]=size(Cplist);
% Get length, Cprows, of List of Columns coordinates of points selected

if (Rprows > 1) && (Cprows > 1)...
&& (Rprows == Cprows) && (Rpcols == Cpcols);
        prpt =0;
        exptedgtabl = int32(round([Cplist, Rplist]));
% assign points list to a table
%disp(exptedgtabl); % display points marked

%reply = input('!!Press Enter - To Continue Process !! ', 's');
%if isempty(reply)
% okay
%end

        set(h_txts6,'String',...
'                                                          ');
        set(h_txts5,'String',...
        strcat('!!Please Wait, System Processing Expert Marked..',...
'Prostate Gland Segment!!!'));

        axes(h_axes2); imshow(K);
```

```
% Determine a seed point for detecting and displaying
% pixels in the marked region of the gland in image
%
         exptedgtablcol = sortrows(exptedgtabl,1);
         exptedgtablrow = sortrows(exptedgtabl,2);
         colrng = exptedgtablcol(Rprows,1)- exptedgtablcol(1,1) + 1;
         rowrng = exptedgtablrow(Rprows,2)- exptedgtablrow(1,2) + 1;

         rowrng2 = idivide(int32(rowrng),int32(2));
         colrng2 = idivide(int32(colrng),int32(2));

         Rs = rowrng2 + int32(exptedgtablrow(1,2));
         Cs = colrng2 + int32(exptedgtablcol(1,1));

for incr = -8 : 1 : +8,

             K1(Rs, (Cs+incr)) = 255;
             K1((Rs+incr), Cs) = 255;
end

         cla(h_axes2,'reset'); % Clear any dislay of previous result
         axes(h_axes2); imshow(K1);


% process the table, set up a loop to pick from the first point
% to the last point, calling routine to joint point current and
% next until end of table is reached.

for curtbl = 1 : Rprows,

if curtbl == Rprows;
%connect 1st and last point
          curtloc = Rprows; nextloc = 1;
          pt1row = exptedgtabl(curtloc,2); pt1col = exptedgtabl(curtloc,1);
          pt2row = exptedgtabl(nextloc,2); pt2col = exptedgtabl(nextloc,1);
% call routine to connect last points marked with first point
% to fit them and store all coordinates
          expert_pntconcurv (segmal);
break; % computation concluded
else
          curtloc = curtbl; nextloc = curtbl + 1;
          pt1row = exptedgtabl(curtloc,2); pt1col = exptedgtabl(curtloc,1);
          pt2row = exptedgtabl(nextloc,2); pt2col = exptedgtabl(nextloc,1);

% call routine to connect points marked with fitting point
% and store all coordinates
          expert_pntconcurv (segmal);

end

       set(h_txts5,'String',...
       strcat('Connecting Edge Points.:',int2str(pt1row),',',...
       int2str(pt1col)));

       axes(h_axes2); imshow(K1);
```

```
end

% call routine to accumulates coordinates
               expert_acumpxsgcd (segmal);

else
      prpt = 1;
% display message that experts did not mark desired edge point
% on the borders of the prostate gland within image II
% go back to menu
        set(h_txts5,'String',...
         strcat('Expert, Please you did not mark any points....',...
'On the edges of the Prostate Gland in the image',...
'... Do so now to enable process completion...'));

       axes(h_axes2); imshow(K1);

       reply = input('!!Press Enter - To Continue !! ', 's');
if isempty(reply)
% okay
end

end

% Get the table and work on the matrix PM to be displayed with the
% white on the original image as image with boundary segmented

set(h_txts5,'String',...
'!!Please Wait, System Compiling Detected Gland Pixels!!!');

axes(h_axes2); imshow(K1);

PS=L;
for  edcnt = 1 : expsegcount
      pxr = expsegimcoodtbl(edcnt,1); pxc = expsegimcoodtbl(edcnt,2);
      PS(pxr,pxc) = 255;
end


end


function [PM]=expert_glandmalgzn(K)

% EXPERT_GLANDMALGZN is the Routine that automates the analysis of selectd
% Image file - TRUS 2D-images of the Prostate BY THE EXPERT TO MARK
% SELECTED SUSPECTED CANCEROUS SECTIONS BY ZONES- TRANSITION, CENTRAL
% AND PERIPHERAL
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
%
% ------------------------------------------------------------------------

global expmaligcoodtbl explocmg;
```

```
global expmalsecedgetbl expsected;
global L rowm colm edsubtsze edgtbsize imsize;
global h_axes2 h_txts5 h_txts6 K1 exptedgtabl;
global Centrzncnt Peripzncnt Transzncnt zon;

% initialize these global counters
explocmg = 0; expsected = 0;
Centrzncnt = 0; Peripzncnt = 0; Transzncnt = 0;

[rowm,colm] = size(K);imsize=rowm*colm;
L = K; K1=K;PM=L;

edgtbsizei = idivide(int32(imsize), int32(24));
edgtbsize = double(edgtbsizei);
edsubtszei = idivide(int32(imsize), int32(42));
edsubtsze = double(edsubtszei);
expmalsecedgetbl = zeros(edsubtsze,3);% mal section edge table
expmaligcoodtbl = zeros(imsize,4); % malignant coordnate table
% Table => row, column, zoneid, pixel intensity
% For Suspected malignant pixels within segmented region of prostate
% Table => row, column, zoneid: '2 - Central', '3-Peripheral',
%                               '1 - Transition'

% Capture the Points Marked by Expert Sonographer/Radiologist
% on the isolated zones of the prostate gland shown in TRUS
% 2D-iamge of the prostate under investigation

 zon = 1; zoneid = 'TRANSITION ZONE..(4th -> 1st Quad, 45")' ;
while zon < 4;

    set(h_txts5,'String',...
    strcat('Please Examine Image Sample I & II....',...
'Mark in Sample II at Least 5 points Clockwise, round the sections',...
'... of the SUSPECTED CANCEROUS cells of ...',zoneid));

    cla(h_axes2,'reset'); % Clear any dislay of previous result
    axes(h_axes2); imshow(K); %

    prompt = {'Enter No. (0 - 3) of Suspected Sections Observed?'};
    dlg_title = 'Cancerous Sections in Zone';
    num_lines = 1; def = {'0'};
    options.Resize='on';  options.WindowStyle='normal';
    answer = inputdlg(prompt,dlg_title,num_lines,def);

    nbe = str2double(answer); %disp (nbe);
% check if no suspected sections exist
if nbe == 0;
if zon == 1;
           Centrzncnt = 0;
else
if zon == 2; Peripzncnt = 0;  else;  Transzncnt = 0;  end
end
% no suspected mal in zone
else
% some sections were observed, then capture the points
for nb = 1 : nbe,
```

```
          set(h_txts6,'String',...
          strcat('For Section:..', int2str(nb),...
'..Click Once on the Points around it:..',...
'Backspace to Delete last Point Marked;...', ...
'Double-Click/Enter to End'));

          Rprows = 0; Cprows = 0;
while (Rprows < 3) && (Cprows < 3);
% Command to Capture Points on the Current
% Figure Axes, h_axes2
            cla(h_axes2,'reset'); % Clear any dislay of previous result
            axes(h_axes2); imshow(K); %
            [Cplist, Rplist]=getpts;
% [x,y] where x is columns of matrix, y is rows

            [Rprows,Rpcols]=size(Rplist); [Cprows,Cpcols]=size(Cplist);

% Lenghts Rprows of List of Rows;Cprows of List of Columns
% for coordinates of points selected

if (Rprows >= 3) && (Cprows >= 3)...
&& (Rprows == Cprows) && (Rpcols == Cpcols);
              exptedgtabl = int32(round([Cplist, Rplist]));
%disp(exptedgtabl);  % display points marked

             set(h_txts6,'String',...
'                                                          ');
               set(h_txts5,'String',...
            strcat('!!Please Wait, System Processing Expert Marked..',...
'Prostate Gland Segment!!!'));

              axes(h_axes2); imshow(K);
% Call routine to locate Rs Cs for section
              expert_seedmal(Rprows);
% Call routine process the current table of marked points

              expsected = 0; expert_tablproc(Rprows);

% call routine to accumulates coordinates of marked points
              segmal = 2;
              expert_acumpxsgcd (segmal);

             set(h_txts5,'String',...
'!!Please Wait, System Compiling Detected Gland Pixels!!!');

             axes(h_axes2); imshow(K1);

% Get the table and work on the matrix PM to be displayed as
% white on the original image as image with boundary marked
% Use the spotted malignant location tables and mark the
% location of the edge with white to differentiate
% in the original image

for malcnt = 1 :explocmg
 pxr = expmaligcoodtbl(malcnt,1); pxc = expmaligcoodtbl(malcnt,2);
                  PM(pxr,pxc)=255;
end
```

```
else
% display message that experts did not mark
%
                set(h_txts6,'String',...
                strcat('For Section:..', int2str(nb),...
'..Mark (at Least 3 Points) around it:..',...
'Backspace to Delete last Point Marked;...', ...
'Double-Click/Enter to End'));
end% end of test to confirm 3 or more points were entered

end% loop to ensure points are marked

end% end of loop to capture points of sections in zone selected

end; % no suspects in this zone, go for next zone

    zon = zon + 1;
if zon == 2;
        zoneid = 'CENTRAL ZONE...(1st Quad, 67.5")';
else
if zon == 3;
           zoneid = 'PERIPHERAL ZONE...(2nd -> 4th Quad, 247.5")';
end
end
end
function expert_acumpxsgcd(segmal)
% EXPERT_ACUMPXSGCD is the Subroutine that accumulates the segemented
% edges, segmented pixels for the left and right results
% for the expert marked edges points
%
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.
% ------------------------------------------------------------------------

global expsegimcoodtbl expsegcount expsegimedgetbl exploced;
global J L h_axes1 h_txts5 Rs;
global expmalsecedgetbl expsected expmaligcoodtbl explocmg;
global Centrzncnt Peripzncnt Transzncnt zon malgintrg;

%-------------------------------------------------------------------------
% Accumulate the pixels within the edge located
% from seed point for column wise and reference Erwn for row wise
% accumulation to avoid double counting of segmented pixels
%-------------------------------------------------------------------------
% Acumulate the pixel coordinates bound by the edges in table sorted
% and acumulate the total pixels in segmented area, expsegcount

if segmal == 1;
    tblend = exploced; accumedgtbl = expsegimedgetbl;
else
%segmal == 2
    tblend = expsected; accumedgtbl = expmalsecedgetbl;
end

if tblend > 0;
```

```
for cntcol = 1 : 1 : tblend,
     Rpxl = accumedgtbl (cntcol,1); Cpxl = accumedgtbl (cntcol,2);
     zon =  accumedgtbl (cntcol,3);

if (Rpxl > 0) && (Cpxl > 0);
if (Rpxl >= Rs);
             Srw = 1;
else
             Srw = -1;
end
%accumulate segmented coord tabe, column-wise, along a row

         nc = Cpxl;
for nr = Rs : Srw : Rpxl,
if segmal == 1;
              expsegcount = expsegcount + 1; expsegimcoodtbl(expsegcount,1)= nr;
                expsegimcoodtbl(expsegcount,2) = nc;
else
% Call a routine that will determine malignancy
if L(nr, nc) <= (malgintrg); % is intensity within malrange
              explocmg = explocmg + 1;  expmaligcoodtbl(explocmg,1) = nr;
              expmaligcoodtbl(explocmg,2)=nc;expmaligcoodtbl(explocmg,3)= zon;
               expmaligcoodtbl(explocmg,4) = L(nr, nc);
if zon == 3;
                        Peripzncnt = Peripzncnt + 1;
else
if zon == 2;
                            Centrzncnt  = Centrzncnt + 1;
else
if zon == 1;
                                Transzncnt  = Transzncnt + 1;
end
end
end
end
end
end;   % Get next column item along this row edge

         set(h_txts5,'String',...
         strcat('!!Please Wait, System Compiling Located Edges......:',...
         int2str(nr),',',int2str(nc)));  axes(h_axes1); imshow(J);
end
end
end

end
%-------------------------------------------------------------------------

%-------------------------------------------------------------------------
function [imagfilenam,nofilepar] = analyz_imagecmpr
% ANALYZ_IMAGECMPR is the Routine that automates the comparing
% of the result of system and expert segmentation and malig detection
% results
% Created by VC Chijindu, 2013/2014 for PhD dissertation Work
% Department of Electronic & Computer Engineering
% Nnamdi Azikiwe University Awka, Nigeria.

global prost_imsamp Asysseg Asysmag Aexpseg Aexpmag;
```

```
global segcodtblss segcodtbles malcodtblsm malcodtblem;
global malrecsys malrecexp segrecsys segrecexp;
global h_axes1 h_axes2 h_axes3 h_axes4
global h_txts1 h_txts2 h_txts5 h_txts6;
global Accs Specs Accm Specm;

nofilepar = 0; % Assume file parameters present
% Call routine to automate selection of an image file to compare result.
  [imagfilenam,diremty] = analyz_imageselc;
  cla(h_axes3,'reset');
if diremty == 0;
   directnam = cd; % Read current directory pathname
   warndlg ({directnam;...
'Please Copy .jpg Prostate Image Files into this Directory';...
'!!! Then Try this Option Aagain !!! '},...
'No Image Files For Selection !');
else
if imagfilenam == ' ';
      nofilepar = 1; %no file for parameters
      set(h_txts5,'String',...
'User selected No File to Analyze.');
      set(h_txts2,'String',...
'                                                          ');
else
      set(h_txts5,'String',...
      strcat('User Selected:...', imagfilenam, '...Image File'));
      I = imread(prost_imsamp);
      cla(h_axes1,'reset'); axes(h_axes1); imshow(I);
      cla(h_axes2,'reset'); cla(h_axes3,'reset');cla(h_axes4,'reset');

      set(h_txts1,'String',...
'Prostate TRUS 2D-Image For Result Comparison');

% Call routine to get results filenmane and check contents
      analyz_genfilname(imagfilenam);

% test if segmtatn results that exist are valid

if (malrecsys == 0) || (segrecsys == 0)...
                  || (malrecexp == 0) || (segrecexp == 0);

              nofilepar = 1; % file parameters not present
% if so determine & display coresponding message & quit
if (malrecsys == 0) && (segrecsys == 0);
            set(h_txts6,'String',...
'Selected Image File Has No System Segmentation Results.');
            axes(h_axes1); imshow(I);
end

if (malrecexp == 0) && (segrecexp == 0);
             set(h_txts6,'String',...
'Selected Image File Has No Expert Segmentation Results.');
            axes(h_axes1); imshow(I);

end

if (malrecsys == 0) && (segrecsys == 0)...
```

```
&& (malrecexp == 0) && (segrecexp == 0);
             set(h_txts6,'String',...
'Selected Image File Has No System/Expert Segmentation Results.');
            axes(h_axes1); imshow(I);
end

else
          set(h_txts5,'String',...
         strcat('System Comparing Results of Segmentation .. & ..',...
'Detection of SUSPECTED CANCEROUS Cells'));
         axes(h_axes1); imshow(I); %

         fid1 = fopen (segcodtblss,'r');
         segimcoodtblss = fscanf (fid1,' %4i  %4i  %4i\n',[3,inf]);
         fid2 = fopen (segcodtbles, 'r');
         segimcoodtbles = fscanf (fid2,' %4i  %4i  %4i\n',[3,inf]);

         fid3 = fopen (malcodtblsm,'r');
         maligcoodtblss = fscanf (fid3,...
' %4i  %4i  %4i  %4i  %4i\n',[5,inf]);
         fid4 = fopen (malcodtblem, 'r');
         maligcoodtbles = fscanf (fid4,' %4i  %4i  %4i  %4i\n',[4,inf]);
         fclose ('all');

% Initialize values
         TPs = 0; % True Positive; Common area of expert & system segmtn
%FNs = 0; % False Negative; Area of Expertseg - TPs common area
%FPs = 0; % False Positive; Area of Systemseg - TPs common area

% Read the segmentation area pixels for system & expert
% to compute the common area, TP

         set(h_txts6,'String',...
          strcat('Please Wait,System Checking Common Areas;',...
'..in Segmentation Results:..'));
                    axes(h_axes1);  imshow(I);%

            [stblt,stbwt] = size(segimcoodtblss);
            [etblt,etbwt] = size(segimcoodtbles);

for sr = 1:stbwt,
for er = 1:etbwt,

if (segimcoodtblss (2,sr) == segimcoodtbles (2,er))...
&& (segimcoodtblss (3,sr) == segimcoodtbles (3,er));

                      TPs = TPs + 1;break;
end

end

end
           set(h_txts6,'String',...
               strcat('Please Wait,Checking Common Areas;',...
'..in Segmentation Results:..', int2str(sr),...
', ...',int2str(er)));
                    axes(h_axes1);  imshow(I);%
```

```
%disp ('TPs - Segmted common area = '); disp(TPs);
%reply = input('Press Enter to Continue!! ', 's');
%     if isempty(reply)
% okay
%     end
if (Asysseg == 0) || (Aexpseg == 0);
                TPs = 0;
end
            FPs = Asysseg - TPs; FNs = Aexpseg - TPs;

% check if Aexpseg ==0 & % Asysseg ==0
            format bank;
if (Aexpseg == 0) && (TPs > 0);
                Accs = 0.00;  Specs = 0.00; % not realistic case
else
if (Asysseg == 0) && (Aexpseg == 0) && (TPs == 0);
                    Accs = 90.00; Specs = 90.00;
else
                  Accs = (TPs/Aexpseg) * 100; % Accuracy computed
                  Specs = (1 - (FPs + FNs)/Aexpseg) * 100; % Specificity
end
end
if Accs > 100.00; Accs = 90.00; end
if Specs > 100.00; Specs = 90.00; end

% Initialize values

         TPm = 0; % True Positive; Common area of expert/system malg
%FNm = 0; % False Negative; Area of Expertmalg - TPm common area
%FPm = 0; % False Positive; Area of Systemmalg - TPm common area

% Read the malignancy detection area pixels for system & expert
% to compute the common area, TPm

         set(h_txts6,'String',...
         strcat('Please Wait, Checking Common Areas;',...
'..in Cancer Detection Results:..',...
              int2str(sr), ', ...',int2str(er)));
                    axes(h_axes1);  imshow(I);%

            [stblt,stbwt] = size(maligcoodtblss);
            [etblt,etbwt] = size(maligcoodtbles);

for sr = 1:stbwt,
for er = 1:etbwt,

if (maligcoodtblss (4,sr) == maligcoodtbles (3,er))...
&& (maligcoodtblss (5,sr) == maligcoodtbles (4,er));

                     TPm = TPm + 1;break;
end

end

end
            set(h_txts6,'String',...
```

```
            strcat('Please Wait, Checking Common Areas;',...
'..in Cancer Detection Results:..',...
              int2str(sr), ', ...',int2str(er)));
                    axes(h_axes1);  imshow(I);%

% check if Asysmag ==0

%disp('Asysmag = '); disp(Asysmag);disp('Aexpmag = '); disp(Aexpmag);
% disp ('TPm - Malign common area = '); disp(TPm);
% reply = input('Press Enter to Continue!! ', 's');
%      if isempty(reply)
% okay
%      end

if (Asysmag == 0) || (Aexpmag == 0);
                TPm = 0;
end
            FPm = Asysmag - TPm;  FNm = Aexpmag - TPm;
            format bank;
if (Aexpmag == 0) && (TPm > 0);
                Accm = 0.00;  Specm = 0.00; % not realistic case
else
if (Asysmag == 0) && (Aexpmag == 0) && (TPm == 0);
                    Accm = 90.00; Specm = 90.00;
else
                Accm = (TPm/Aexpmag) * 100; % Accuracy computed
                Specm = (1 - (FPm + FNm)/Aexpmag) * 100; % Specificity
end
end
if Accm > 100.00; Accm = 90.00;  end
if Specm > 100.00;  Specm = 90.00; end

% Save Comparison result and exit to Display
         analyz_savcompfil(imagfilenam);

end

end

end

end


function analyz_genfilname(imagfilenam)

% routine to get result file particulars,
% reading and content confirm

global prost_imsamp Asysseg Asysmag Aexpseg Aexpmag;
global malrecsys malrecexp segrecsys segrecexp sumrecsys sumrecexp;
global segcodtblss segcodtbles malcodtblsm malcodtblem
global FSMSYS FMMSYS FSMEXP FMMEXP;

[pathstr,name,ext,versn] = fileparts(prost_imsamp);
segcodtblss = strcat(name,'ss','.table');% ss - system segmented
SMSYS = strcat(name,'ss','.jpg');% ss - system segmented matrix
```

```
malcodtblsm = strcat(name,'sm','.table');% sm - system malign
MMSYS = strcat(name,'sm','.jpg');% sm - system malign matrix

segcodtbles = strcat(name,'es','.table');% es - expert segmented
SMEXP = strcat(name,'es','.jpg');% es - expert segmented matrix
malcodtblem = strcat(name,'em','.table');% em - expert ,,
MMEXP = strcat(name,'em','.jpg');% em - expert malign matrix

FSMSYS = fullfile(pathstr,SMSYS); FMMSYS = fullfile(pathstr,MMSYS);
FSMEXP = fullfile(pathstr,SMEXP); FMMEXP = fullfile(pathstr,MMEXP);

fid1 = fopen (segcodtblss,'r');
if fid1 == -1;  segrecsys = 0; else;  segrecsys = 1; end

fid2 = fopen (segcodtbles, 'r');
if fid2 == -1;   segrecexp = 0; else;  segrecexp = 1; end

fid3 = fopen (malcodtblsm, 'r');
if fid3 == -1;  malrecsys = 0; else;   malrecsys = 1;end

fid4 = fopen (malcodtblem, 'r');
if fid4 == -1;  malrecexp = 0; else;   malrecexp = 1;end

%Extract selected file no.
filno = str2double([imagfilenam(1,11) imagfilenam(1,12)]);

% Read the store of Area of segmented and malignant section for system
fid5 = fopen ('syssumvals.table','r');

% read content of sumarytable
sumparsys = fscanf (fid5,' %4d  %7d  %7d  %5d  %5d  %5d\n',[6,inf]);

if (fid5 == -1);
       sumrecsys = 0;
else
       sumrecsys = 1;
       [mrows,ncols] = size(sumparsys);
% mrows = 6 for one record table,
for trecs = 1 : ncols,
            tblimagfno = sumparsys(1,trecs);

if filno == tblimagfno;
                Asysseg = sumparsys(2,trecs);
                Asysmag = sumparsys(3,trecs);break;
end

end
end

% Read the store of Area of segmented and malignant section for expert
fid6 = fopen ('expsumvals.table','r');

% read content of sumarytable
sumparexp = fscanf (fid6,' %4d  %7d  %7d  %5d  %5d  %5d\n',[6,inf]);

if (fid6 == -1);
```

```
       sumrecexp = 0;
else
       sumrecexp = 1;

       [mrows,ncols] = size(sumparexp);
% mrows = 6 for one record table,
for trecs = 1 : ncols,
            tblimagfno = sumparexp(1,trecs);
if filno == tblimagfno;

                Aexpseg = sumparexp(2,trecs);
                Aexpmag = sumparexp(3,trecs); break;
end
end

end

fclose('all');


end

function analyz_savcompfil(imagfilenam)
% Routine to save the comparisom result in a .mat file

global Accs Specs Accm Specm;

filno = str2double([imagfilenam(1,11) imagfilenam(1,12)]);

compvals = vertcat(filno, Accs, Specs, Accm, Specm);
% comparison values put in an array for entry into table as character
% strings


fid7 = fopen ('syscompreslt.table','r');
if (fid7  == -1);
    fid7 = fopen ('syscompreslt.table','a+');
%append first record from analysis with the format
    fprintf (fid7,' %4.2f  %5.2f  %5.2f  %5.2f  %5.2f\n',compvals);
    fclose(fid7);

else
% read content of comparison table
    comptabl = fscanf (fid7,...
' %4.2f  %5.2f  %5.2f  %5.2f  %5.2f\n',[5,inf]);
    [mrows,ncols] = size(comptabl);

   disp(filno);disp(size(comptabl));

   imagrec = 0;
for trecs = 1 : ncols,
       tblimfno = comptabl(1,trecs);
if filno == int(tblimfno);
           disp(filno);disp(tblimfno);disp(ncols);
           imagrec = 1;%rec is already present overwrite it with new
           disp(comptabl(:,trecs));disp(compvals);
```

```
            fclose(fid7);

            fid7 = fopen ('syscompreslt.table','w');
            comptabl(1,trecs) = filno;    comptabl(2,trecs) = Accs;
            comptabl(3,trecs) = Specs;     comptabl(4,trecs) = Accm;
            comptabl(5,trecs) = Specm;
           fprintf (fid7,' %4.2f  %5.2f  %5.2f  %5.2f  %5.2f\n', comptabl);
break;
end
end
   fclose(fid7);
if imagrec == 0;
       fid7 = fopen ('syscompreslt.table','a');
%append record from analysis to end of file
       fprintf (fid7,' %4.2f  %5.2f  %5.2f  %5.2f  %5.2f\n',compvals);
       fclose (fid7);
end

end

end
```